Very simple design and easy to replicate, data already found and python libraries are indicated. Only one stock (on YF) doesn't have data (can be seen in the notebook).

Prediction of Stock Price Time Series using Transformers

Lorenzo D. Costa[1], **Alexei M. C. Machado** [1,2]

[1] Departamento de Ciência da Computação
Pontifícia Universidade Católica de Minas Gerais
[2] Departamento de Anatomia e Imagem
Universidade Federal de Minas Gerais
Belo Horizonte - Minas Gerais - Brazil

orenzo2846@outlook.com, alexeimcmachado@gmail.com

Abstract. *This work presents an implementation of the Transformer on the problem of predicting stock prices from time series. The model is compared with ARIMA and a neural network with LSTM cells. We hypothesize that, due to the powerful memory capacity and association between series values, the Transformer would be able to achieve better results than other shallow or deep solutions. The data used in the experiments is the average daily prices of 8 shares of the Ibovespa index in the period of 2008. The obtained results corroborated the hypothesis of superiority of the Transformer which predicted the stock prices with higher accuracy in 60% of the times.*

Is this available or open source?
Found it on YF

1. Introduction

The introduction of Transformers in 2017 changed the state of the art for Natural Language Processing problems and, recently, they have proven to be effective in the area of Computer Vision as well. This work aims to evaluate the application of this model in the problem of forecasting time series of stock prices.

Technical Analysis is a very widespread heuristic in the field of Finance, which seeks, through available data, to speculate on asset prices. It is based on the Dow Theory [Brown et al., 1998], that postulates some basic points, one of which is that the market moves in uptrends or downtrends. Furthermore, the Efficient Market Hypothesis [Malkiel and Fama, 1970] states that markets are information efficient, that is, they are a reflection of all available data.

This article is motivated by the importance of market data analysis in the Finance area and the complex nature of the problem, which can be interpreted based on Technical Analysis theories. It is believed that, with the potential shown by the Transformers, it may be possible to make predictions. It is also worth mentioning that this model was chosen taking into account two of its characteristics: its good performance and its ability to memorize series values and to associate them with future prices.

The main objective of this work is to obtain accurate predictions of stock prices using the Transformers. The model will be compared with two others: the ARIMA (Auto-Regressive Integrated Moving Average) and a neural network with LSTM (Long Short-Term Memory) cells. Through this comparison, better results are sought than those more classic models. Figure 1 exemplifies, in general, what is presented in this work.

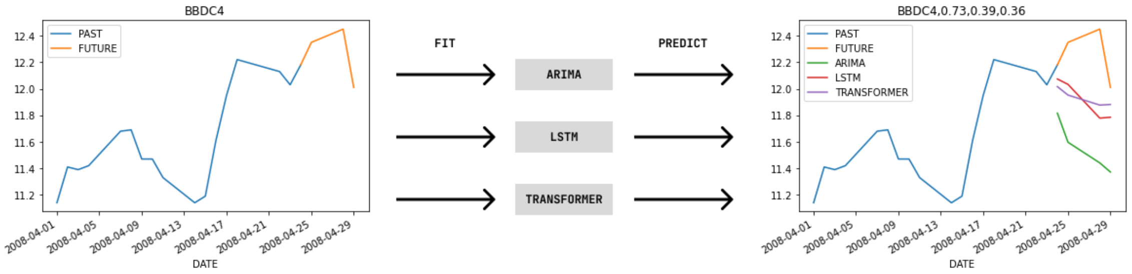

Figure 1. Scheme of prediction process. A partial series is presented to ARIMA, LSTM and the Transformer so that they output the stock price for the next days.

The article is divided into the following sections: Theoretical Framework and Literature Review (2), a study of articles in the area and review of the architecture of the Transformer, ARIMA and the architecture of LSTM cells; Materials (3), the presentation of the used data used; Methods (4) an explanation of the adopted methodology; Results (5), the exposition and analysis of the obtained results' Conclusions and Future Works (6), with the conclusion of the work and evaluation of possible future works.

2. Background

2.1. Transformers

The Transformer architecture used is based on the original method proposed by Vaswani *et al.* and can be seen in Figure 2 [Vaswani et al., 2017]. First, Positional Encoding is performed, which incorporates the position of a given quotation to the input. Then, several Encoders and Decoders are stacked, communicating with each other and generating an output. Encoders have two components: a Multi-Head Attention Layer and a Feed Forward Network. Decoders, in turn, have three components: the same as the first and a Masked Multi-Head Attention Layer, which enables model learning. Finally, there is a Linear Layer followed by a Sigmoid Layer, that produces the result. It is worth mentioning that this last layer is no longer a Softmax Layer as proposed in the original architecture, since in the context of time series a numerical result corresponds to the prediction and not to its probability, i.e. it is a regression-like model. Furthermore, the Input/Output Embedding part was replaced by a fully connected layer, as there is no need to map words into numeric vectors given the nature of the problem.

Encoders have two main components. First, input is directed to a Self-Attention Layer. This is responsible for creating associations between the value and the time series. Next, the output elements generated by this layer are sent to a Feed Forward Network, and independently processed. Finally, what was produced by one Encoder is directed to the next one. The Decoders come into play after the completion of the processing done by the Encoders. These have the same components, but with an additional layer responsible for preventing it from accessing terms that will come in the future. At the end of the Transformer architecture, there is a Linear Layer that projects the output generated by the last Decoder into a larger vector called the Logits Vector. The Sigmoid Layer ultimately outputs the forecast value from this.

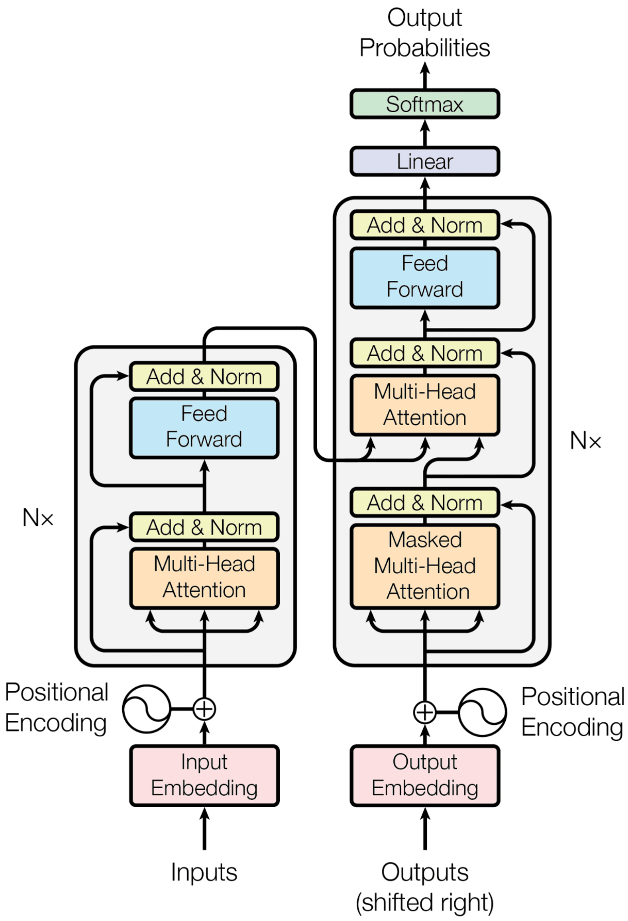

Figure 2. Transformer architecture.

2.2. ARIMA

The ARIMA used is denoted by $ARIMA(p, d, q)$ and is not seasonal. It is the generalization of an auto-regressive model of moving averages, in which parameter p represents the order of the auto-regression, parameter d represents the degree of differentiation and parameter q represents the order of the mean mobile. This model works in such a way that a regression is performed on the analyzed variable from its previous values with the presence of an error. With this, we try to adjust the best possible combination for the time series following the Box-Jenkins Box et al. [2015] approach, which has a series of steps for model determination.

2.3. LSTM

The architecture of the LSTM cells used are the same as originally proposed. They are composed of 3 main mechanisms: an Input Gate, an Output Gate and a Forget Gate, that are responsible for regulating the flow of information that enters and leaves the cell. This, in turn, is capable of remembering values over arbitrary time intervals, what is ideal for processing time series since the duration of important events along the sequence is not known. This memory capacity was essential in overcoming the vanishing gradient problem encountered in Recurrent Neural Networks.

2.4. Related Works

Transformers were initially presented by Vaswani et al. [2017], and changed the state of the art for the Natural Language Processing problem. The introduction of a model that had only attention mechanisms and discarded recurrence, which was considered the best approach to this challenge, changed the panorama of the solutions that would come to be made in the future. It became possible to train models much faster, what was previously not feasible due to the inefficiency of Recurrent Neural Networks (RNNs), so that significantly better results were obtained. In the original article, the authors test the proposed architecture in two language translation tasks: WMT 2014 English-to-French and WMT 2014 English-to-German. In the first task, Transformer surpassed the best state-of-the-art results, spending 1/4 less than the training time previously reported by the models, evidencing the improvement in performance. For the second Dataset, the model achieved a BLEU (Bilingual Evaluation Understudy) score of 2 points above the best scores ever observed.

For the problem of predicting asset prices from texts, Batra and Daudpota [2018] presents a sentiment analysis of Tweets about Apple products. The result, combined with indicators of the company in question, are supplied to a Support Vector Machine (SVM) with the aim of predicting the stock's movement on the following day. The results obtained from the tests showed that it is possible to associate people's opinion about such a company with the course of their quotations. The accuracy was approximately 77%. The articles of Liu et al. [2019] and Othan et al. [2019] continue working on stock price forecasting, but now using Transformers. The first presents a variation of the architecture, called Capsule Network Based on Transformer Encoder (CAPTE), capable of capturing relevant semantic information from investment texts about S&P500 shares. In the second, the Bidirectional Encoder Representations from Transformers (BERT) is used to classify Tweets about investments related to BIST100 assets. In both cases, the models used by the authors stood out in the tests, obtaining the best accuracy among the compared models. CAPTE scored with an accuracy of approximately 64% and BERT with 96%. In Daiya and Lin [2021], the authors present a high-performance model called TRANSDICE to predict the movement of stocks based on financial indicators and investment news. For this architecture, both Transformers and Dilated Causal Convolutions are used to extract characteristics from the data, considering its context. The evaluated dataset involved S&P500 assets and the accuracy obtained in the tests proved to be 3% greater than the state of the art.

Based on numerical indicators for forecasting stock prices, Caron and Müller [2020] present an evaluation of state-of-the-art Transformers models for predicting asset volatility. Furthermore, the authors expose a new type of architecture based on features and context incorporation. The dataset used was the FIN10K and the results showed that an ensemble of the authors' models reached an accuracy of 34% greater than the competitors. Ding et al. [2020] explore the estimation of the direction of stock prices using Deep Learning techniques. Predictions were made on numerical data from China A-Shares and NASDAQ using a model proposed by the authors: the Hierarchical Multi-Scale Gaussian Transformer. The results obtained were satisfactory, for the first dataset an accuracy of 58% was obtained and for the second of 57%. A few months later, Ramos-Pérez et al. [2021], present models to make predictions about the volatility of assets, which are hybrids of Transformers and Multi-Transformers Layers, with GARCH algorithms and

LSTM units. Using data from the last 650 days of the S&P, the authors obtained good results in predicting the next day in the tests, showing that the model provided good estimates and adequate risk management. In Lim et al. [2021], a new model called Temporal Fusion Transformer (TFT) is proposed, with the aim of making predictions for Multi-Horizon Time Series, which are more complex series. The predictions are made based on the OMI Realized Library Dataset, which contains the daily realized volatility of asset indices. The results obtained from the tests showed that TFT has a high predictive capacity for complex data scenarios. Finally, Li et al. [2022], propose an architecture called Transformer Encoder Attention (TEA), with the aim of predicting whether the price of an asset will rise or fall on a given day. This model combines the evaluation of historical values of quotes with the sentiment analysis of Headlines and Tweets about investments. The results obtained were very satisfactory, with the model obtaining an accuracy of approximately 64% in two of the three datasets used.

3. Materials

In this work, data from 8 preferred shares of the Ibovespa Index were used: BBDC4, BRAP4, CMIG4, GGBR4, GOAU4, GOLL4, ITSA4 and ITUB4. Each one with its own dataset, which contained the average daily prices in 2008, totaling 80 values per share. It is worth mentioning that the choice of stocks was made in alphabetical order, in order to guarantee the integrity of the model and avoid any choice bias. Figure 3 and Table 1 show data for the analyzed period.

They are available on YF, but they have used AVERAGE daily prices

Table 1. Stock prices in each time period.

Date	BBDC4	BRAP4	CMIG4	GGBR4
01/02/2008	R$ 12.30	R$ 31.45	R$ 4.58	R$ 21.70
01/03/2008	R$ 11.67	R$ 31.17	R$ 4.58	R$ 22.06
...	...	...	...	...
04/28/2008	R$ 12.45	R$ 31.55	R$ 5.07	R$ 27.07
04/29/2008	R$ 12.01	R$ 31.04	R$ 4.96	R$ 26.72
Date	GOAU4	GOLL4	ITSA4	ITUB4
01/02/2008	R$ 29.57	R$ 40.90	R$ 3.61	R$ 15.42
01/03/2008	R$ 29.91	R$ 38.87	R$ 3.50	R$ 14.87
...	...	...	...	...
04/28/2008	R$ 35.57	R$ 24.26	R$ 3.61	R$ 15.44
04/29/2008	R$ 35.03	R$ 24.73	R$ 3.54	R$ 15.13

4. Methods

The proposed Transformer model was compared with a more classic approach in the area of time series prediction, ARIMA, and with a more consolidated Artificial Intelligence model in the area of temporal sequence prediction, a neural network of LSTM cells. The entire implementation of the work was done in the Python [Van Rossum and Drake Jr, 1995] programming language.

The period analyzed for each action was divided into series of 20 days, in order to ensure more assertive predictions and better training performance. The division of data

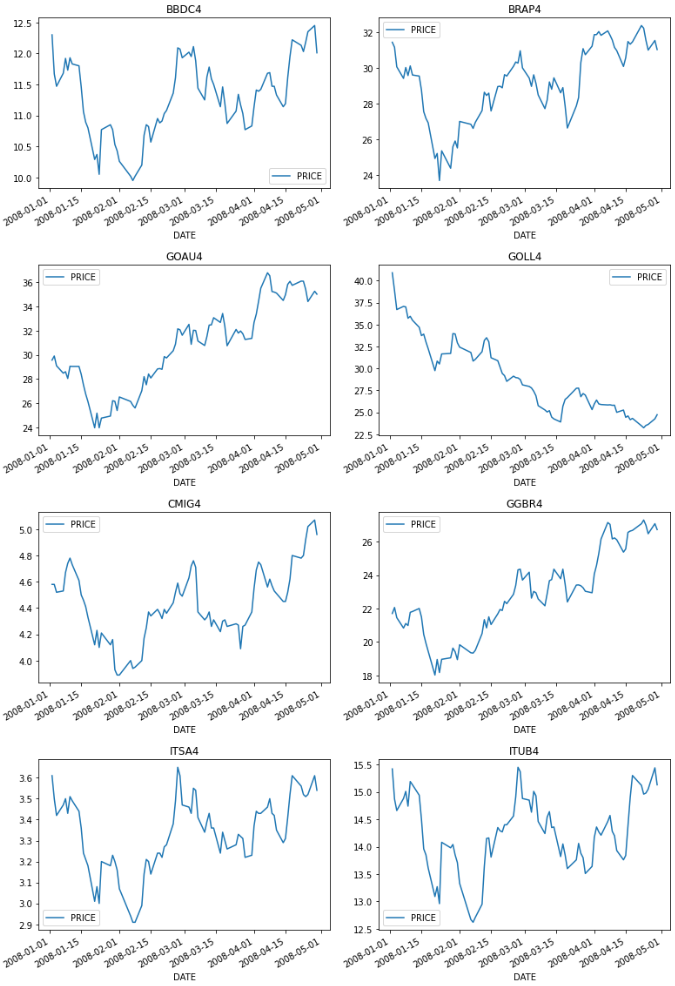

Figure 3. Stock prices in each time period.

into training and testing was done with a sequential separation of 80% for the first and 20% for the second. In this work, the terms "PAST" and "FUTURE" were adopted as training and test, respectively. Figure 4 presents the data with this division made in one of the analyzed periods.

The definition of the ARIMA parameters p, q and d was done automatically, using the PMDARIMA [Smith et al., 2017] library. The method works in such a way that differentiation tests are performed on the training data in order to determine the its differentiation order. After obtaining this value, several adaptations of ARIMA models with different parameters are made, with the objective of minimizing an information criterion. The criterion used was the library default: AIC (Akaike Information Criterion) and the only necessary adjustment was the definition of the existence of seasonality of the time series as false.

For LSTM and Transformer, a random search was performed to determine the best combination of hyperparameters for a validation set. This was defined as the last 2 days of the training period. The variation of the values of each hyperparameter for the LSTM can be observed in Table 2 and for the Transformer in Table 3. A model was considered the best if it obtained a Root Mean Squared Error (RMSE) smaller than the error of the

RMSE is used to judge models

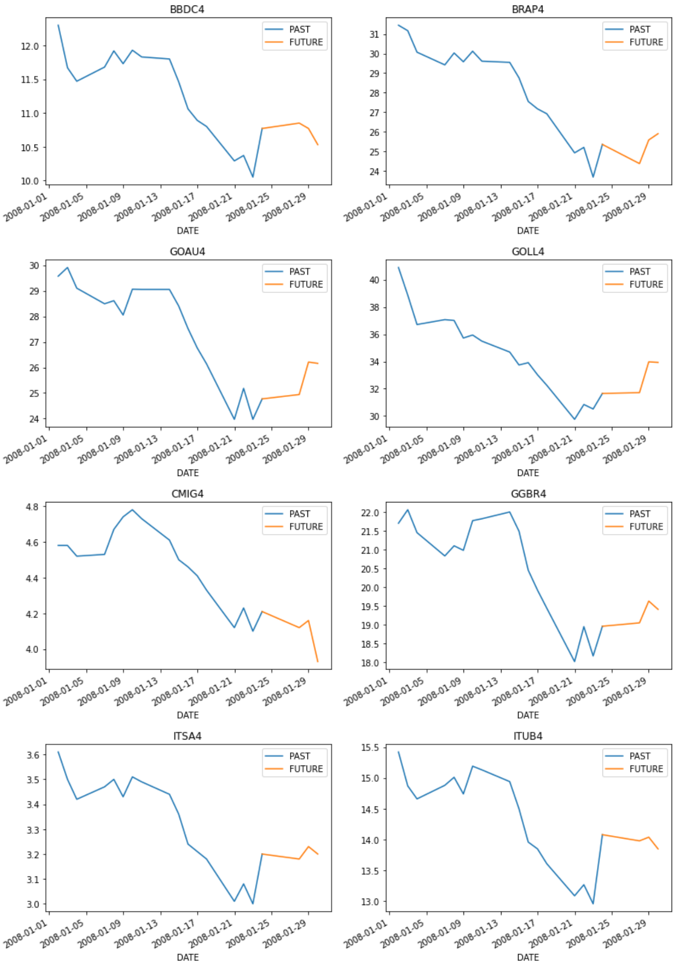

Figure 4. Stock prices in each time period, divided into training and test sets.

current best combination in the validation set minus 0.1. This value was subtracted, as a small improvement in the error doe not mean an improvement in the model, given the random nature of the problem.

Table 2. Hyperparameters used in the LSTM.

Learning Rate	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}
Generator Length	1, 3, 5, 7, 9
Hidden Layers	0, 1, 2
Neurons	16, 32, 64
Epochs	128, 256, 512

Python is always used, packages are here specified

The implementation of LSTM used the Keras Chollet et al. [2015] library and the Transformer was implemented through the Python Package Index (PyPI) "time-series-transformer" package. In both cases, before being fed to the models, the training data was passed through a Standard Scaler from the SKLearn library [Pedregosa et al., 2011], so that each value is standardized. Both were compiled using the Adam optimizer from the PyTorch library [Paszke et al., 2019] and the Mean Squared Error (MSE) loss function. After training each model, the RMSE obtained in the test sets and the price prediction of

Table 3. Hyperparameters used in the Transformer.

Learning Rate	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}
Attention Length	1, 3, 5, 7, 9
Encoders, Decoders	1, 3, 5, 7, 9
Q, V, H	1, 3, 5, 7, 9
Epochs	128, 256, 512

each model were collected.

5. Results

After carrying out all the tests, the performance of the models and the results were interpreted in 2 different ways: by determining which model obtained the lowest RMSE in each sub-period and by determining which model obtained the lowest total RMSE. Some graphical examples of predictions can be seen in Figure 5. In Table 4, some predictions of each model are compared with real prices. With respect to the first evaluation metric, which can be seen in Table 5. Transformer was the best model, obtaining the lowest RMSE in 60% of the tests, followed by LSTM with 22 % and, finally, ARIMA with 18%. With repect to the second metric, presented in Table 6, the same pattern was observed, with Transformer being the best model with the lowest total RMSE in all experiments, followed by LSTM and ARIMA. These results may come from Transformer's strong ability to remember data over time and to perform associations between values. In contrast, the ARIMA model had the worst performance, given its strong dependence on non-stationary and seasonal data, which is not the case for stock price series. The neural network with LSTM cells had an median performance, showing an evolution upon ARIMA, but without reaching the association and memory potential of the Transformer.

Table 4. Comparison between actual prices and predictions for each model.

Date	Price	ARIMA	LSTM	Transformer
01/24/2008	R$ 24.77	R$ 23.60	R$ 27.47	R$ 24.27
01/28/2008	R$ 24.94	R$ 23.22	R$ 27.66	R$ 24.26
01/29/2008	R$ 26.21	R$ 22.65	R$ 27.67	R$ 24.27
01/30/2008	R$ 26.16	R$ 22.48	R$ 27.67	R$ 24.29

Table 5. Percentage of the tests in which each method presented RMSE.

ARIMA	LSTM	Transformer
18%	22%	60%

5.1. Other Experiments

It is noteworthy that other experiments were carried out throughout the development of the work, with the objective of evaluating the performance of the models for different combinations of input size. Initially, instead of using daily average prices, monthly average prices were used. This choice provides a longer period for evaluation, while the variance between prices is much greater, given the large time span from one month to another. Due

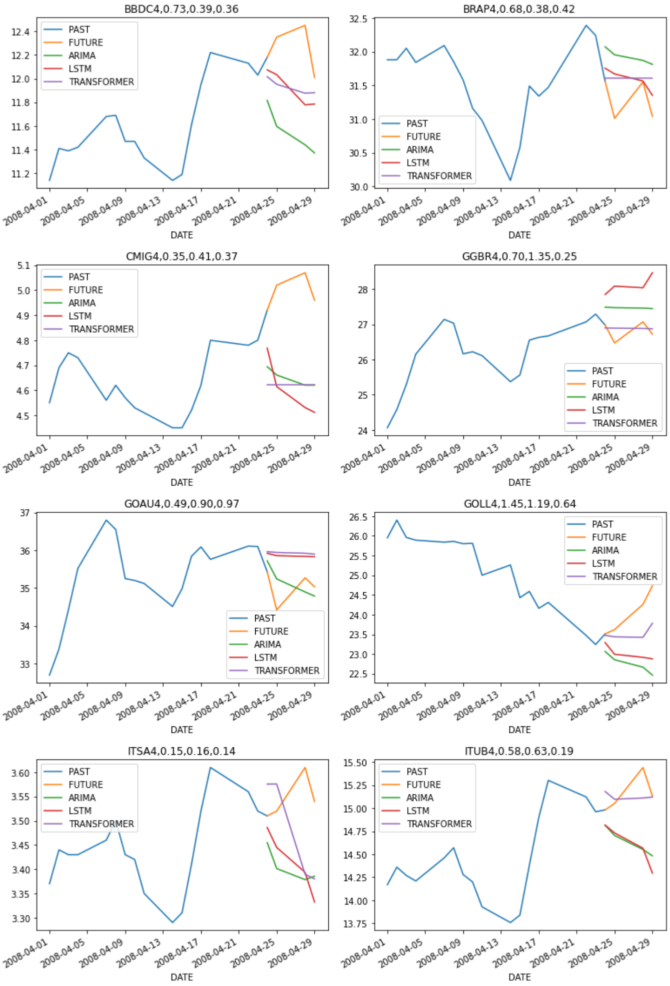

Figure 5. Predicted prices.

Table 6. RMSE per method, considering all time periods.

ARIMA	LSTM	Transformer
30.63	27.77	23.76

to the complex nature of the problem, the results were not assertive enough for a conclusion, since the models were equally correct and obtained a considerably high RMSE. In addition to this choice, the use of a much larger volume of data was also taken into account, in which, instead of 20 values per subperiod, there were approximately 60,000 prices. Despite the greater availability of information, this option was discarded given the very high computational cost for training the models and the low impact of periods very distant from the present on the prediction.

6. Conclusions and Future Work

6.1. Conclusions

From the obtained results, it is clear that the Transformer is a promising alternative in the field of predicting complex time series, such as stock prices over time, as it obtained better performance in all experiments. Its memory capacity proved to be essential for the production of predictions. Stock price forecasting is a complex problem, since there is not a single variable capable of stipulating the value of a quote on the next day. There are many factors that influence the price of a share, such as public opinion about it, the global context in which it is inserted and the results of the company responsible for the share. Therefore only Technical Analysis is not enough to make accurate predictions.

6.2. Future Work

With the use of more computational power and dedicated GPUs, the evaluation of several other sub-periods is desirable, since this can provide even more assertive results for the problem. In addition, an even more in-depth study will be important to develop more combinations of hyperparameters, ensuring that new possibilities are tested and better models are sought. Another point to be evaluated is how the presence of a greater oscillation in prices or a daily or monthly seasonal behavior can impact the results of the predictions. Given this, it is also proposed as future work the insertion of new variables in the problem, such as the use of Sentiment Analysis of social networks to determine the population's position towards a stock and how this can impact its price.

References

R. Batra and S. M. Daudpota. Integrating stocktwits with sentiment analysis for better prediction of stock price movement. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–5. IEEE, 2018.

G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.

S. J. Brown, W. N. Goetzmann, and A. Kumar. The dow theory: William peter hamilton's track record reconsidered. *The Journal of finance*, 53(4):1311–1333, 1998.

M. Caron and O. Müller. Hardening soft information: A transformer-based approach to forecasting stock return volatility. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4383–4391. IEEE, 2020.

F. Chollet et al. Keras, 2015. URL `https://github.com/fchollet/keras`.

D. Daiya and C. Lin. Stock movement prediction and portfolio management via multimodal learning with transformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3305–3309. IEEE, 2021.

Q. Ding, S. Wu, H. Sun, J. Guo, and J. Guo. Hierarchical multi-scale gaussian transformer for stock movement prediction. In *IJCAI*, pages 4640–4646, 2020.

Y. Li, S. Lv, X. Liu, and Q. Zhang. Incorporating transformers and attention networks for stock movement prediction. *Complexity*, 2022, 2022.

B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

J. Liu, H. Lin, X. Liu, B. Xu, Y. Ren, Y. Diao, and L. Yang. Transformer-based capsule network for stock movement prediction. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 66–73, 2019.

B. G. Malkiel and E. F. Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.

D. Othan, Z. H. Kilimci, and M. Uysal. Financial sentiment analysis for predicting direction of stocks using bidirectional encoder representations from transformers (bert) and deep learning models. In *Proc. Int. Conf. Innov. Intell. Technol.*, volume 2019, pages 30–35, 2019.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

E. Ramos-Pérez, P. J. Alonso-González, and J. J. Núñez-Velázquez. Multi-transformer: A new neural network-based architecture for forecasting s&p volatility. *Mathematics*, 9(15):1794, 2021.

T. G. Smith et al. pmdarima: Arima estimators for Python, 2017. URL `http://www.alkaline-ml.com/pmdarima`. [Online; Acessado em 12/11/2022].

G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.