International Journal of Business and Management; Vol. 19, No. 4; 2024
ISSN 1833-3850 E-ISSN 1833-8119
Published by Canadian Center of Science and Education

Deep Learning Applied to Stock Prices: *Epoch Adjustment in Training an LSTM Neural Network*

Napoleão Verardi Galegale[1] & Camilo Ilzo Shimabukuro[1]

[1] Centro Estadual de Educação Tecnológica Paula Souza – CEETEPS, Brazil

Correspondence: Napoleão Verardi Galegale, Postgraduate, Extension, and Research Unit,, CEETEPS, Rua dos Bandeirantes, 169, ZIP Code 01124-010, São Paulo, SP, Brazil. E-mail: nvg@galegale.com.br

Received: April 4, 2024 Accepted: May 16, 2024 Online Published: June 8, 2024

doi:10.5539/ijbm.v19n4p80 URL: https://doi.org/10.5539/ijbm.v19n4p80

Abstract

Research on recurrent neural networks applied to financial time series is still underexplored, even more so for series on Brazilian stock prices. The research gap was identified in studies on regularization with early stopping to improve predictive capacity and reduce overfitting for the type of neural network used. This study aims to analyze the effect of the number of epochs on the prediction error dispersion of a recurrent neural network using the Long Short-Term Memory – LSTM approach on the stock prices of a Brazilian company, aiming to minimize prediction error and reduce the risk of overfitting. The method is of an applied nature with a quantitative approach and uses an experimental procedure to analyze the behavior of the prediction error of a recurrent neural network as a function of the number of epochs. As a result, a range of the number of epochs was identified that extracts the best trade-off relation between predictive capacity and overfitting risk for a given network configuration. It was also identified how the dispersion of prediction error initially declines sharply and then stabilizes asymptotically. The study offers a greater understanding of the behavior of the prediction error, seeking greater efficiency in predictive techniques on financial time series in order to add value and reduce uncertainties in the decision-making process for asset managers and investors.

Keywords: deep learning, early stopping, number of epochs, stock prices, recurrent neural networks

1. Introduction

The prediction of asset prices has been of interest both in finance research and the financial industry. Over the past decades, the trading of stocks in stock exchanges has undergone a significant technological transformation as a result of the evolution of computational processing capacity and data transmission methods. The buying and selling of assets have become more efficient through the replacement of traditional open outcry with electronic platforms and the submission of orders through standardized messaging protocols traveling over fiber optic networks. Amid thus evolution, stockbrokers and other institutions participating in the stock market reduced their dependence on floor operators and trading desks, adopting trading based on Algorithmic Trading (AT), which is much more scalable, faster and less prone to errors and biases from human operators. Associated with the use of high-performance computing (HPC) infrastructure, this transformation has also triggered a race for order execution time in milliseconds, commonly referred to as high-frequency trading (HFT). (Nuti et al., 2011; Menkveld, 2013; Menkveld, 2016; Buchanan, 2019; Harvard University – IACS, 2017; MIT Technology Review, 2017).

Since the 2010s, with the diminishing value extraction potential through execution speed (latency), quantitative strategies based on statistical models seeking to improve the predictive process and identify more complex anomalies and patterns have intensified the use of Machine Learning (ML) tools (IEEE Spectrum, 2017; Buchanan, 2019; Arnott, Harvey, & Markowitz, 2019; Martins & Galegale, 2022). During this period, the development of graphics processing units (GPUs) used for algebraic calculations of large matrices at decreasing costs, coupled with the evolution of computing tools and libraries, contributed to the widespread adoption of deep learning (DL), characterized by a higher number of internal layers and neural units. DL has succeeded in areas such as image recognition and natural language processing for its ability to extract nonlinear relationships from data without the intensive work of feature selection typical of ML techniques (Bengio, Courville, Vincent, 2013; Lecun, Bengio, Hinton, 2015).

Until mid-2015, few scientific studies involved DL approaches applied in finance and AT. From the latter half of the last decade, an increasing number of articles began exploring the theme, yet with few publications focused on the Brazilian capital market (Cavalcante, Brasileiro, Souza, Nobrega, & Oliveira, 2016; Henrique, Sobreiro & Kimura, 2018).

Typical DL approaches, deep neural networks overlap representation layers of features with a great capacity to model complex data structures. Generally, DL algorithms involve optimizing an objective function, also called a cost, loss, or error function. The learning process results in a neural unit weight configuration that minimizes the difference between the prediction and training data, i.e., the prediction error. The search for the function's minimum is conducted through gradient descent, which, from a set of multiple inputs, moves in the direction of the steepest negative slope, therefore, the fastest descent rate, controlled by the learning rate and adjusting the weights of each neural unit during the learning process on the training data (Goodfellow, Bengio, & Courville, 2016).

A critical aspect of machine learning is its generalization capacity, i.e., achieving a predictive performance that exceeds a pre-established criterion for new data. However, obtaining satisfactory results and efficient learning depends on reducing data dimensionality, a careful selection of input variables, and proper network parameter adjustment. Nonetheless, the predictive capacity may deteriorate once the learning process overly fits the training data, resulting in increased prediction error for new data, thus a loss of generalization capability known as overfitting (Prechelt, 1998a; Goodfellow et al., 2016; Chong, Han, & Park, 2017; Fischer & Krauss, 2018; Sirignano, 2019).

To address the risk of overfitting, neural network regularization strategies can be used, but effective strategies should seek a significant reduction in prediction variance without excessively increasing the training bias, making the development of such strategies the focus of much effort in this área (Goodfellow et al., 2016). Early stopping is a frequently used regularization strategy for its simplicity and good results across various DL approaches. It consists of halting the training process once a performance measure for out-of-sample data (validation data), such as the variance of prediction error, begins to increase after reaching a minimum or remains at a minimum level without improvement over epochs, thereby also minimizing the chance of overfitting occurrence (Prechelt, 1998b; Goodfellow et al., 2016; Chollet, 2021). The number of epochs, a hyperparameter for training neural networks, is the number of times the algorithm completely traverses the training data. In this sense, early stopping also acts as a regularization strategy as it reduces or limits the variation range of a neural network configuration dimension. (Prechelt, 1998a; Prechelt, 1998b; Aggarwal, 2018; Sirignano, 2019).

Among DL approaches, recurrent neural networks (RNNs) are often used for sequential data since they feed the output signal back into the input layer or internal layers of the network (Längkvist; Karlsson; Loutfi, 2014). Long Short-Term Memory (LSTM), a specific case of RNN conceived by Hochreiter and Schmidhuber (1997), has been applied to time series modeling and is considered state-of-the-art in sequential learning with remarkable results in speech recognition and language forecasting, albeit rarely used in financial series prediction (Xiong, Nichols, & Shen, 2015; Fischer & Krauss, 2018).

Considering these points, this study analyzes the impact of the "number of epochs" hyperparameter on the learning process of a recurrent neural network LSTM and the prediction error of a financial time series, starting from the early stopping strategy (Fischer & Krauss, 2018; Sirignano, 2019) applied to a series of stock prices of a Brazilian company. Thus, this research seeks to answer the question: how does the prediction error, given by the difference between the actual price and the price predicted by the model, vary as a function of the number of epochs in the training of a recurrent neural network based on DL on series of Brazilian stock prices? Therefore, the objective of this work is to analyze the effect of adjusting the hyperparameter number of epochs on the prediction error dispersion of a recurrent neural network using the Long Short-Term Memory (LSTM) approach on the stock prices of a Brazilian stock exchange, aiming to minimize prediction error and reduce the risk of overfitting.

The study hopes to contribute to recent research on DL approaches applied in algorithmic stock trading by investigating ways to improve the efficiency of predictive tools on financial time series using a neural network regularization technique such as early stopping. This work aims to practically contribute to creating value and reducing uncertainties for asset managers interested in the strategies discussed and investors seeking a better understanding of potential approaches involving ML and DL in asset management.

This paper is structured into five sections, including this introduction. The following section develops the theoretical construct supporting the empirical research methodology. The methodology's details are described in section three, including sample descriptions, procedures, and techniques used. The results discussion is presented in section four in a descriptive manner and commented on in relation to other studies. Finally, the conclusion wraps up the work and suggests future research avenues.

2. Theoretical Framework

The Efficient Market Hypothesis (EMH) on asset pricing assumes that, in efficient markets, it would not be possible to predict the price of a share based on past prices, and that all known information would already be reflected in its current price. The EMH uses the random walk model to explain that the evolution of investor preferences and the emergence of new information would combine to construct equilibrium situations reflected in the distributions of returns over time. Thus, the main role of the market would be the allocation of ownership of the economy's capital stock, where market prices would signal precisely this allocation of resources. By this reasoning, price patterns could not be used simply because prices would not follow a trend (Fama, 1970; Markov, 1971).

Behavioral finance researchers explore how psychological biases influence investor decisions. They argue that biases such as herd behavior, the anchoring effect, prospect theory, mental accounting, and the illusion of understanding could reflect on market price formation by impacting the rational behavior expected from investors (Tversky & Kahneman, 1974; Shiller, 1995; Thaler, 1999; Kruger & Dunning, 1999; Kahneman, 2011).

Regardless of the price formation approach, seeking higher returns involves extensive data collection and analysis, and the development of dynamic strategies that optimize prescribed performance criteria in uncertain scenarios (Guo et al., 2017). However, uncertainty is among the greatest challenges faced by finance researchers and managers, introducing an inevitable risk factor that complicates decision-making (Gadre-Patwardhan, Katdare, & Joshi, 2016). Though stock prices typically exhibit a non-stationary random walk, under certain conditions and time horizons, they may demonstrate some degree of trend behavior, mean reversion, or even regimes. Such observable behaviors in financial series underscore the importance of predicting trend changes or inflection points. In these situations, ML approaches like Random Forests, Support Vector Machines (SVM), Hidden Markov Model (HMM), among others, can be utilized (Chan, 2021; Längkvist, Karlsson, & Loutfi, 2014).

However, applying ML in the context of finance and Algorithmic Trading (AT) also presents complexities inherent to the market's microstructure, the granularity of data with partial executions, order cancellations, hidden liquidity, where there is at least an intuition of how the liquidity distribution in limited order books (LOB) relates to future price movements. Despite recognizing ML as a powerful, scalable, and fundamentally sound framework for data analysis and prediction, there is no easy path to profitability (Kearns & Nevmyvaka, 2013). Moreover, even with data-adherent approaches, López de Prado (2018) highlights the occurrence of overfitting in backtesting, where models excessively adjust to training data but lose predictive capacity with new data, possibly because, among other reasons, finance research deals with relatively small data sets and a low signal-to-noise ratio.

The noise present in financial time series data results from the actions of a large number of market participants buying and selling at different times and for various purposes, presenting characteristics of random sequences. It can also result from price shocks caused by unexpected events like impactful economic news, which can persist over time (Kaufman, 2013).

Thus, neural networks are indicated as data-oriented adaptive methods suitable for situations with few model assumptions because they do not present the large number of parameters of supervised ML models, therefore with a lesser tendency for overfitting, and do not get trapped in local minima in these models' high-dimensional space. Regularization methods, such as dropout and early stopping, drastically reduce the risk of overfitting. Piecewise linear activation functions make training faster and allow the inclusion of additional internal layers in deep networks. In this sense, applying DL approaches through deep neural networks would allow modeling the non-linear relationships of time series, and adding new visualization techniques could provide a better understanding of how neural networks work, mitigating concerns about their black-box nature (Prechelt, 1998b; Goodfellow et al., 2016; Xiong, Nichols, & Shen, 2015).

Among DL modalities, one might choose convolutional neural networks (CNNs) due to the large number of successful studies in image recognition and the capacity to store data from the market's microstructure, like limited offer books (Tsantekidis et al., 2017). However, a recurrent neural network seems more pertinent as its connections form cycles that would allow information persistence over time. Figure 1 illustrates an RNN with the feedback of the output signal at moments: 0, 1, ... t, ... $\tau-1$, and τ. Each input signal x_t is operated by function A^{θ}, where A represents a neural network parameterized by vector θ, and it combines with the previous state signal S_{t-1}, resulting in the output $h_{t+1} = A^{\theta}(S_t, x_{t+1})$. However, the instability resulting from exploding and vanishing gradients and the persistence of information for only a few iterations in memory limit its practical application (Kraus & Feuerriegel, 2017).

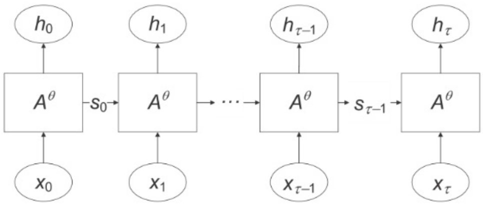

Figure 1. Recurrent Neural Network (RNN)

Source. Kraus & Feuerriegel (2017).

Ultimately, the choice falls on the LSTM modality due to its typical architecture of DL approaches, recognized for its ability to extract characteristics of nonlinear relationships from data, as well as for memorizing long time series, selectively retaining relevant past information, discarding irrelevant data such as noise, and whose nodes close feedback circuits in certain layers of the network, resulting in an approach of extensive memorization suitable for learning from experience even for arbitrary time intervals. Moreover, it supports the learning capacity during training by keeping the gradients of the loss function stable, propagating with a lower risk of vanishing and exploding gradients through the use of sigmoid functions and hyperbolic tangents (Hochreiter & Schmidhuber, 1997; Xiong, Nichols, & Shen, 2015; Bao, Yue, and Rao, 2017; Kraus & Feuerriegel, 2017; Fischer & Krauss, 2018; Kim & Won, 2018).

The central concept of the LSTM unit is to allow data to traverse its entire structure, retaining or eliminating information operated by forget gate, input gate, and output gate, each consisting of a neural unit with its own sigmoid activation function and controlling the content and state of the unit through the state cell that passes the signal forward. The forget gate ft decides which part of the information will be removed from the state cell. The input gate it specifies which information will be added to the state cell. The output gate ot decides which information from the state cell will be used as output. These three gates use the sigmoid function σ, resulting in values between zero and one, quantifying what will be passed forward. The state cell ct traverses the entire network and acts by adding or removing information with the help of gates operated by a hyperbolic tangent function tanh (Bao, Yue, and Rao, 2017; Kraus & Feuerriegel, 2017; Selvin, Vinayakumar, Gopalakrishnam, Menon, & Soman, 2017; Fischer & Krauss, 2018).

The mathematical representation of the vectors of gates and outputs is shown in the equations:

$$\text{Forget gate: } f_t = \sigma(W_f \bullet h_{t-1} + x_t + b_f) \quad (1)$$

$$\text{Input gate: } i_t = \sigma(W_i \bullet h_{t-1} + x_t + b_i) \quad (2)$$

$$\text{Output gate: } o_t = \sigma(W_o \bullet h_{t-1} + x_t + b_o) \quad (3)$$

$$\text{State cell: } c_t = \tanh(Wc \bullet h_{t-1} + x_t + b_c) \quad (4)$$

$$\text{Output vector: } h_t = o_t * \tanh(c_t) \quad (5)$$

where ***f***, ***i***, and *o* are the vectors of forget, input, and output gates; ***c*** is the vector of the state cell; ***x*** and ***h*** are the input and output vectors; ***W*** represents the weight matrices, and ***b*** the bias vectors for the respective gates (subscripts) ***f***, ***i***, and ***o***. The subscripts ***t*** and ***t-1*** refer to the current and one-step lagged times; ***σ()*** is the sigmoid function, and ***tanh()*** is the hyperbolic tangent (Bao, Yue, and Rao, 2017; Kraus & Feuerriegel, 2017; Selvin et al., 2017).

Initially, the forget gate combines the previous output h_{t-1} with the current input x_t, returning a vector f_t that corresponds to the intensity with which each element of the state cell c_t should be passed to the next state of the cell. Subsequently, the input gate receives h_{t-1} and x_t and returns a vector i_t, and an additional neural layer c_t calculates a vector of candidate values that are combined through vector multiplication. Finally, the new output h_i is obtained by matrix multiplication $h_t = o_t \bullet c_t$ for each element. The new state cell results from the update rule: $c_t = f_t \bullet c_{t-1} + i_t \bullet c_t$ (Bao, Yue, and Rao, 2017; Kraus & Feuerriegel, 2017; Selvin et al., 2017), as shown in Figure 2.

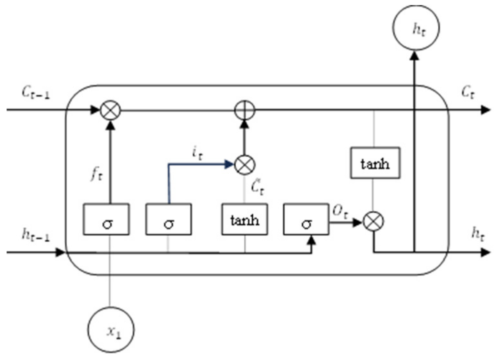

Figure 2. Temporal chaining in an LSTM network

Source. Bao, Yue e Rao (2017).

Despite the convenience of the choice and the suitability of an LSTM network for this study, designing neural network models is not trivial. As pattern recognition models, they depend on a well-crafted data representation, besides involving concerns with performance, statistical methods, generalization capability, and overfitting. Furthermore, financial time series require special attention as they exhibit random characteristics resulting from noise. Thus, their construction should start with a few neural units in the early layers, monitoring the generalization capability as its complexity increases, because adjusting the number of internal layers and the quantity of neural units in each layer is crucial for satisfactory generalization (Zhang, Patuwo, & Hu, 1998; Dunis & Williams, 2003).

Similarly, choosing and optimizing parameters in DL approaches remains a significant challenge for researchers. Although there is a reasonable consensus in studies on machine learning approaches such as decision trees, bagging, boosting, forest trees, and support vector machines, the same cannot be said for deep neural networks. Models such as deep belief networks (DBN), stacked denoising autoencoders, convolutional neural networks, and classifiers based on sophisticated feature extraction techniques may have from ten to fifty hyperparameters, making the minimization of the loss function in the configuration space complex and even hindering the reproducibility of model results, making studies on such methods more of an art than a science. Such challenges, especially in multilayer models, once represented a real obstacle to scientific development in this knowledge area, to the extent that research achieving results by proposing parameter optimization methods contributed more to advances in image classification than to innovative modeling and machine learning strategies (Bergstra, Bardenet, Bengio, & Kégl, 2011).

For example, the learning rate, a hyperparameter that determines the size of adjustments applied to the weights on the neural network units, directly influences its learning capability. Low rates can make learning much slower, while high rates can cause the error function to behave erratically, losing the ability to present continuous improvements and making learning dysfunctional. To refine this process, a momentum parameter is used, which checks how past changes affected the current weight changes to make the next weight adjustment roughly in the same direction as the previous one (Zhang, Patuwo, & Hu, 1998; Dunis & Williams, 2003).

Methods for reducing the risk of overfitting also constitute ways of configuration and parameter optimization. Regularization strategies, for example, seek to reduce the complexity of a neural network by establishing limits in the parameter variation space by reducing the prediction error and, therefore, the risk of overfitting, possibly at the cost of an increase in training error. Significant research efforts in machine learning have concentrated on developing more effective strategies (Goodfellow et al., 2016; Aggarwal, 2018; Sirignano, 2019). Although deep neural networks are recognized as robust systems, overfitting remains a critical problem because large neural networks present slow processing and are difficult to handle due to the combination or co-adaptation of predictions during training and consequent loss of generalization for new data (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

Researchers have many regularization strategies at their disposal; some add restrictions to parameter values, others include penalties in the loss function, which, if carefully chosen, can lead to better performance on test data. They can also encode specific types of prior knowledge, express an unspecific preference for simpler models seeking greater generalization, determine an underdetermined problem, and even combine various hypotheses that explain the training data as ensemble methods (Goodfellow et al., 2016).

A successful strategy for preventing overfitting, applicable to feed-forward neural networks and Boltzmann machines, is dropout, a technique that randomly disconnects some neural units from the network during training, preventing excessive co-adaptation (Srivastava et al., 2014). Another possibility is early stopping, a strategy where the training process is halted when the variance of the prediction error ceases to decrease after reaching a minimum level over epochs, is often used for its ease of implementation and superior results across various approaches. By stopping adjustment at the moment (epoch) when generalization capability deteriorates or stabilizes, excessive adjustment of neural network weights is avoided, reducing the chance of overfitting occurrence (Prechelt, 1998a; Prechelt, 1998b; Goodfellow et al., 2016; Fischer & Krauss, 2018).

The prediction error is the performance variable of the LSTM recurrent neural network under study. Based on the early stopping strategy, the behavior of the prediction error as a function of the number of training epochs of the model can be analyzed. Thus, identifying an interval of the number of epochs that minimizes the prediction error and, as per theory, reduces the risk of overfitting represents a contribution to research in this field of knowledge. It's also worth highlighting that although many studies have analyzed the early stopping strategy for regularization of machine learning models, including deep learning, research on recurrent neural networks applied in finance is less frequent, still being nascent when dealing with time series of Brazilian stock prices.

3. Methodology

This applied nature work with a quantitative approach utilizes an experimental procedure to analyze the behavior of the prediction error of an LSTM network as a function of the number of epochs, a training hyperparameter for neural networks. It seeks to identify an optimal adjustment range that, according to the early stopping strategy, minimizes the prediction error and reduces the risk of overfitting.

The study implements a recurrent neural network LSTM conceived by Hochreiter & Schmidhuber (1997) applied to predicting a financial time series (Fischer & Krauss, 2018) through a computer program developed in Python version 3.7.4 (Karpathy, 2015; Olah, 2015; Palai, 2018; Rohrer, 2020) using software libraries dedicated to data science, numerical processing, data analysis, machine learning, deep learning, and visualization such as: Numpy, Pandas, Scikit-Learn, TensorFlow, MatPlotLib (Géron, 2019; Chollet, 2021) executed on the Anaconda distribution platform v.1.9.7, development interface MS Visual Studio Code v.1.47.2 (Pedregosa et al., 2011), and made available by the authors on the GitHub platform in the file “Modelo LSTM-B3.py” (GitHub, 2021).

The network is composed of two layers with 200 neuronal units each and uses the ADAM (Adaptive Moment Estimation) gradient optimizer for its ease of implementation, computational efficiency, and suitability to non-stationary data, sparse gradients, and low signal/noise ratio (Kingma, 2014; Ruder, 2016). The learning rate was set at 0.001, and the batch size at 50. These hyperparameters are kept fixed to observe the behavior of the prediction error as a function of the number of epochs. The number of epochs was adjusted to vary in the range of ten to one hundred with increments of ten epochs.

The time series data are the closing prices of Petrobras' preferred shares, PETR4, traded on “B3 Brazilian Stock Exchange” for the period from 2008 to 2018 and made available on the site (B3 Historical Prices, 2020). The data are contained in a ".csv" file and made available on the GitHub platform (GitHub, 2021).

Prices were standardized to range between 0 (zero) and 1 through the MinMaxScaler function, grouped into sequences of 20 data points, and segmented into 80% for training, 10% for validation, and 10% for testing, in line with the works of Fischer and Krauss (2018), Xiong, Nichols, and Shen (2015), and Kraus and Feuerriegel (2017). Thus, out of 2719 lines of data, 2160 are allocated for training, another 271 for validation, and the remaining 271 for testing.

4. Results Presentation and Discussion

The data and charts for Prediction (Target vs. Prediction) and Error (Target - Prediction) generated by the program were recorded for epochs starting at ten and ending at one hundred in increments of ten epochs, keeping the other hyperparameters constant. Graphs of Prediction and Error (Accuracy) for Petrobras' preferred shares (PETR4) for ten, fifty, and one hundred epochs are presented in Figures 3, 4, and 5, respectively. The other charts with intermediate epochs are available on the GitHub platform (2021).

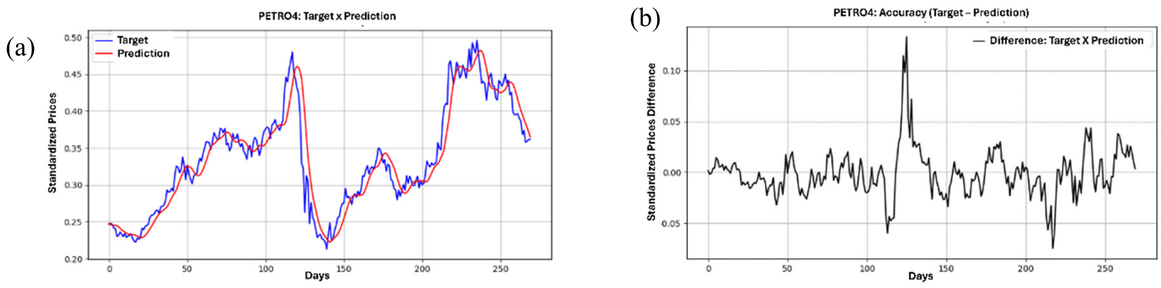

Figure 3. PETR4 Graphs for 10 Epochs: (a) Prediction Graph; (b) Error Graph

Source. Prepared by the authors.

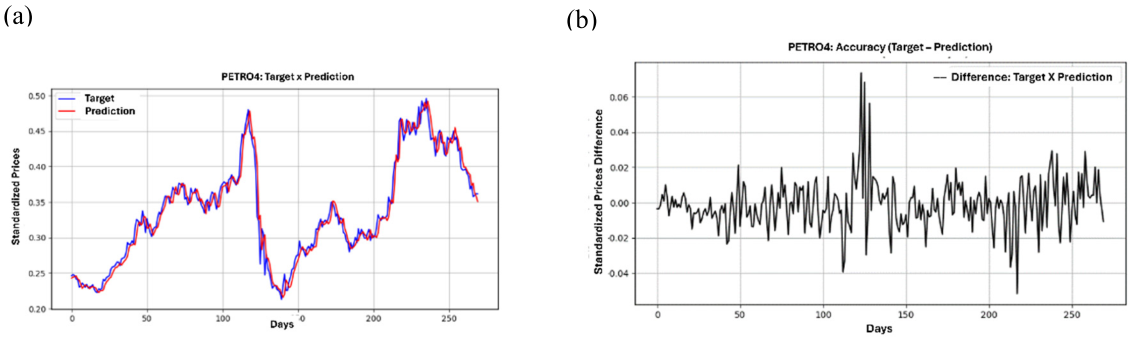

Figure 4. PETR4 Graphs for 50 Epochs: (a) Prediction Graph; (b) Error Graph

Source: Prepared by the authors

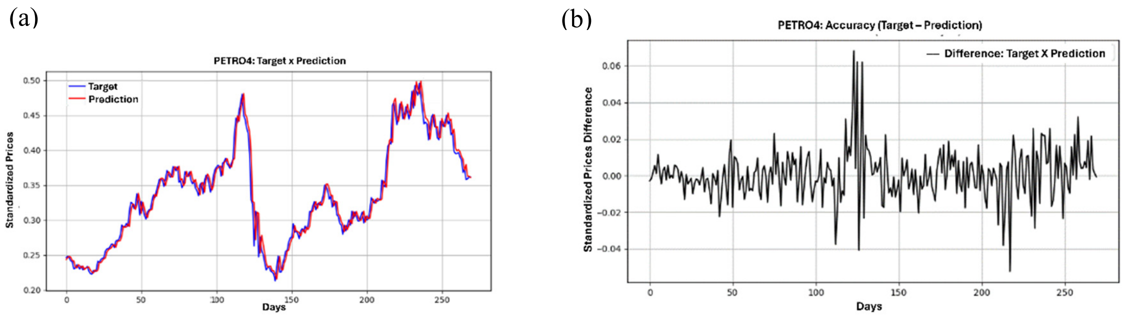

Figure 5. PETR4 Graphs for 100 Epochs: (a) Prediction Graph; (b) Error Graph

Source: Prepared by the authors

The mean, measures of dispersion (variance and standard deviation), and the coefficient of determination R2 as a function of epochs are presented in Table 1.

Table 1. Mean, dispersion measures and coefficient of determination

Epochs	Mean	Variance	Standard Deviation	R2
10	-0.001260	0.000542	0.023290	0.895881
20	- 0.006160	0.000357	0.018892	0.929903
30	0.001015	0.000257	0.016028	0.950584
40	- 0.001234	0.000210	0.014484	0.959468
50	- 0.001052	0.000197	0.014042	0.962305
60	- 0.001320	0.000189	0.013736	0.963244
70	- 0.000841	0.000186	0.013620	0.963935
80	0.001911	0.000194	0.013913	0.962786
90	- 0.002884	0.000189	0.013735	0.963005
100	0.000733	0.000190	0.013768	0.963688

Source. Prepared by the authors.

In the graphs of Figures 6 and 7, it's observed that the measures of dispersion, variance, and standard deviation, significantly decrease in the range between 10 and 50 epochs but show little variability from 50 epochs onwards.

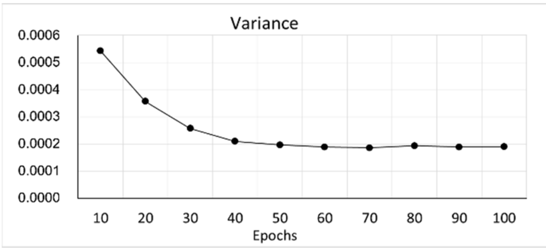

Figure 6. Variance of the prediction error as a function of the number of Epochs

Source. Prepared by the authors.

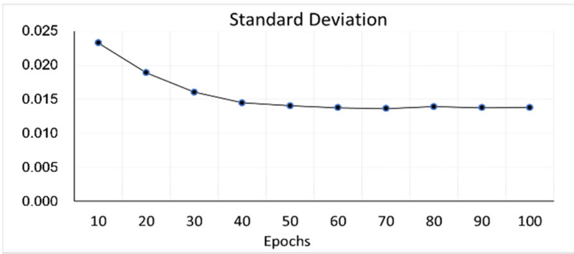

Figure 7. Standard Deviation of the prediction error as a function of the number of Epochs

Source. Prepared by the authors.

In the graph of Figure 8, the coefficient of determination increases in the range of 10 to 50 epochs, also showing almost constant values from 50 epochs onwards.

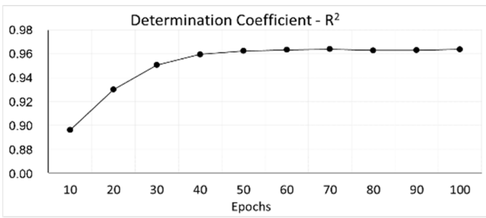

Figure 8. Determination Coefficient of prediction error as a function of the number of Epochs

Source. Prepared by the authors.

These results indicate a nonlinear behavior of the measures of dispersion of the prediction error, variance, and standard deviation. The continuous and significant reduction of the dispersion of the prediction error in the range of 10 to 50 epochs suggests that the neural network was 'learning' from the training data. From 50 epochs onwards, the dispersion of the prediction error practically remains unchanged, suggesting that the neural network was no longer benefiting from successive adjustments in weights, and therefore the learning was no longer resulting in a decreased variance of the prediction error, indicating a potential risk of overfitting. It can be inferred that there is an optimal range around 50 epochs for the LSTM network in this study, beyond which not only does the quality of prediction stabilize but the predictive capacity for new out-of-sample data could also be impaired.

5. Conclusions

The choice of the LSTM network is justified not only by its suitability for financial series and its growing use in recent research but also by its characteristics of recurrent feedback, extraction of characteristics from nonlinear relationships in data, memorization of long time series, selective retention of relevant information, all while maintaining learning capacity with a low risk of vanishing gradients.

Based on the concepts of the early stopping regularization strategy, this work investigated how adjusting the number of training epochs of a recurrent neural network LSTM affects the dispersion of the prediction error, its indicator of prediction performance for the specific case of a time series of stock prices of a Brazilian company.

The analysis of the measures of dispersion of the prediction error, variance, and standard deviation, allowed identifying a range between 10 and 50 training epochs in which the error dispersion reduces continuously and sharply, indicating that in this range the LSTM recurrent neural network was productively adjusting the weights of the neural units, capturing the relationship between input and output signals, therefore learning from the training data. However, in the range of 50 to 100 epochs, the dispersion of the prediction error remained almost constant, suggesting that adjustments in weights did not improve learning, and, as per the literature, could be incurring overfitting. It can be concluded that for the researched LSTM recurrent neural network, choosing a range around 50 epochs extracts the best trade-off relationship between predictive capacity and overfitting risk. It is noteworthy that the dispersion of the prediction error did not show an increase after reaching its minimum, a behavior characteristic of convolutional networks, showing initially a sharp decline and then stabilizing asymptotically.

Future works could utilize a multivariate analysis for determining ranges for other hyperparameters and establish a hyperparameter space from which configurations that minimize prediction error in association with other regularization techniques such as dropout can be extracted. It is also recommended to conduct a study applied to larger neural networks on broader data sets that involve intraday data on all transactions carried out (tick-by-tick), better leveraging the capacity of deep learning by using a wider diversity of assets, not only stocks but also currencies, commodities, futures, and other derivatives, aiming for the generalization of results. In addition, a cross-sectional analysis with competing predictive methods for a comparative evaluation that considers the effect of parameter optimization on predictive capacity is suggested.

Informed consent

Obtained.

Ethics approval

The Publication Ethics Committee of the Canadian Center of Science and Education.

The journal and publisher adhere to the Core Practices established by the Committee on Publication Ethics (COPE).

Provenance and peer review

Not commissioned; externally double-blind peer reviewed.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Data sharing statement

No additional data are available.

References

Aggarwal, C. C. (2018). Neural networks and deep learning. *Springer, 10,* 978-3. https://doi.org/10.1007/978-3-031-29642-0

Arnott, R., Harvey, C. R., & Markowitz, H. (2019). A backtesting protocol in the era of machine learning. *The Journal of Financial Data Science, 1*(1), 64-74. http://dx.doi.org/10.2139/ssrn.3275654

B3 Preços históricos (2020). Retrieved from http://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/historico/mercado-a-vista/series-historicas/

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one, 12*(7), e0180944. https://doi.org/10.1371/journal.pone.0180944

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798-1828. http://dx.doi.org/10.1109/TPAMI.2013.50

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems, 24.*

Buchanan, B. (2019). Artificial intelligence in finance. *The Alan Turing Institute.* https://doi.org/10.5281/zenodo.2612537

Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P., & Oliveira, A. L. (2016). Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications, 55,* 194-211. https://doi.org/10.1016/j.eswa.2016.02.006

Chan, E. P. (2021). Quantitative trading: how to build your own algorithmic trading business. John Wiley & Sons.

Chollet, F. (2021). Deep learning with Python. Simon and Schuster.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications, 83*, 187-205. https://doi.org/10.1016/j.eswa.2017.04.030

Dunis, C. L., & Williams, M. (2003). Applications of advanced regression analysis for trading and investment. *Applied quantitative methods for trading and investment,* 1-40. https://doi.org/10.1002/0470013265.ch1

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance, 25*(2), 383-417. https://doi.org/10.2307/2325486

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054

Gadre-Patwardhan, S., Katdare, V., & Joshi, M. (2016). A Review of Artificially Intelligent Applications in the Financial Domain. Cap. 1 In Dunis, C. L., Middleton, P. W., Karathanasopoulos, A., Theofilatos, K. (Eds.), *Artificial Intelligence in Financial Markets.* Palgrave MacMillan. https://doi.org/10.1057/978-1-137-48880-0

Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc.

Github (2021) Dados de Mercado, Códigos Fonte e Resultados da Pesquisa. Retrieved from https://github.com/CamiloIlzo/Dados-de-Mercado-B3

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Guo, X., Lai, T. L., Shek, H., & Wong, S. P. S. (2017). Quantitative trading: algorithms, analytics, data, models, optimization. Chapman and Hall/CRC. https://doi.org/10.1201/9781315371580

Harvard University – IACS (2017). An Economy of algorithms. Publ. 19-jan-2017. Retrieved from https://www.seas.harvard.edu/news/2017/01/economy-algorithms

Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry, 33*(1), 3-12. https://doi.org/10.1002/asmb.2209

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of Finance and Data Science, 4*(3), 183-201. https://doi.org/10.1016/j.jfds.2018.04.003

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

IEEE Spectrum (2017). Hedge Funds Look to Machine Learning, Crowdsourcing for Competitive Advantage. Retrieved from https://spectrum.ieee.org/tech-talk/computing/software/fintech-trends-hedge-funds-look-to-machine-learning-crowdsourcing-for-competitive-advantage

Kahneman, D. (2011). Thinking, fast and slow. Macmillan. https://doi.org/10.1007/s00362-013-0533-y

Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog, 21*, 23. Retrieved from http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Kaufman, P. J. (2013). Trading Systems and Methods. John Wiley & Sons.

Kearns, M., & Nevmyvaka, Y. (2013). Machine learning for market microstructure and high frequency trading. *High Frequency Trading: New Realities for Traders, Markets, and Regulators.*

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications, 103*, 25-37. https://doi.org/10.1016/j.eswa.2018.03.002

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems, 104*, 38-48. https://doi.org/10.1016/j.dss.2017.10.001

Kruger, J., & Dunning, D. (1999). Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology, 77*(6), 1121. https://doi.org/10.1037//0022-3514.77.6.1121

Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters, 42*, 11-24. https://doi.org/10.1016/j.patrec.2014.01.008

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444. https://doi.org/10.1038/nature14539

López de Prado, M. L. (2018). Advances in financial machine learning. John Wiley & Sons.

Markov, A. A. (1971). Extension of the limit theorems of probability theory to a sum of variables connected in a chain. *Dynamic Probabilistic Systems, 1*, 552-577.

Martins, E. & Galegale, N. V. (2022). Retail sales forecasting information systems: comparison between traditional methods and machine learning algorithms. Proceedings of the 15th IADIS International Conference Information Systems 2022, IS 2022 (pp. 30-38). Retrieved from https://www.iadisportal.org/digital-library/retail-sales-forecasting-information-systems-comparison-between-traditional-methods-and-machine-learning-algorithms

Menkveld, A. J. (2013). High frequency trading and the new market makers. *Journal of Financial Markets, 16*(4), 712-740. https://doi.org/10.1016/j.finmar.2013.06.006

Menkveld, A. J. (2016). The Economics of High-Frequency Trading. *Annual Review of Financial Economics, 8*, 1-24. https://doi.org/10.1146/annurev-financial-121415-033010

MIT Technology Review. (2017). As Goldman Embraces Automation, Even the Masters of the Universe Are Threatened. Nanette Byrnes. 7 fev. 2017. Retrieved from http://www.technologyreview.com/s/603431/as-goldman-embraces-automation-even-the-masters-of-the-uni verse-are-threatened/

Nuti, G., Mirghaemi, M., Treleaven, P., & Yingsaeree, C. (2011). Algorithmic trading. *Computer, 44*(11), 61-69. https://doi.org/10.1109/MC.2011.31

Olah, C. (2015) Understanding LSTM Networks. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Palai, U. (2018) Algorithmic Trading. Retrieved from https://github.com/umeshpalai/AlgorithmicTrading-MachineLearning

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research, 12*, 2825-2830.

Prechelt, L. (1998a). Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks, 11*(4), 761-767. https://doi.org/10.1016/s0893-6080(98)00010-0

Prechelt, L. (1998b). *Early stopping – but when? In Neural Networks: Tricks of the trade* (pp. 55-69). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-49430-8_3

Rohrer, B. (2020). Retrieved from https://brohrer.github.io/blog.html

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1643-1647). IEEE. https://doi.org/10.1109/ICACCI.2017.8126078

Shiller, R. J. (1995). Conversation, information, and herd behavior. *The American Economic Review, 85*(2), 181-185.

Sirignano, J. A. (2019). Deep learning for limit order books. *Quantitative Finance, 19*(4), 549-570. https://doi.org/10.1080/14697688.2018.1546053

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15*(1), 1929-1958.

Thaler, R. H. (1999). Mental accounting matters. *Journal of Behavioral Decision Making, 12*(3), 183-206. https://doi.org/10.1002/(SICI)1099-0771(199909)12:3<183::AID-BDM318>3.0.CO;2-F

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2017, July). Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th Conference on Business Informatics* (CBI) (Vol. 1, pp. 7-12). IEEE. https://doi.org/10.1109/CBI.2017.23

Tversky, A., & Kahneman, D. (1974). Judgment under Uncertainty: Heuristics and Biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *Science, 185*(4157), 1124-1131. https://doi.org/10.1126/science.185.4157.1124

Xiong, R., Nichols, E. P., & Shen, Y. (2015). Deep Learning Stock Volatility with Google domestic trends. arXiv preprint arXiv:1512.04916. https://doi.org/10.48550/arXiv.1512.04916

Focus to learn moreZhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*(1), 35-62. https://doi.org/10.1016/S0169-2070(97)00044-7