Comparing classic time series models and the LSTM recurrent neural network: An application to S&P 500 stocks

Javier Oliver Muncharaz

Finance, Markets and Valuation

Comparing classic time series models and the LSTM recurrent neural network: An application to S&P 500 stocks

Comparativa de los models clásicos de series temporales con la red neuronal recurrente LSTM: Una aplicación a las acciones del S&P 500

Javier Oliver Muncharaz [1]

[1] Departamento de Economía y Ciencias Sociales, Universidad Politécnica de Valencia. Valencia, España. Email: jaolmun@ade.upv.es

JEL: G12; G17; C45

Abstract

In the financial literature, there is great interest in the prediction of stock prices. Stock prediction is necessary for the creation of different investment strategies, both speculative and hedging ones. The application of neural networks has involved a change in the creation of predictive models. In this paper, we analyze the capacity of recurrent neural networks, in particular the long short-term recurrent neural network (LSTM) as opposed to classic time series models such as the Exponential Smooth Time Series (ETS) and the Arima model (ARIMA). These models have been estimated for 284 stocks from the S&P 500 stock market index, comparing the MAE obtained from their predictions. The results obtained confirm a significant reduction in prediction errors when LSTM is applied. These results are consistent with other similar studies applied to stocks included in other stock market indices, as well as other financial assets such as exchange rates.

Keywords: Recurrent Neural Network; Long short-term neural network; S&P 500; Arima

Resumen

En la literatura financiera existe un gran interés por la predicción de precios bursátiles que es necesario para la creación de diferentes estrategias de inversion, tanto especulativas como de cobertura. La aplicación de las redes neuronales ha supuesto un cambio en la creación de modelos de predicción. En este trabajo se analiza la capacidad que tienen las redes neuronales recurrentes, en concreto la long short-term recurrent neural network (LSTM) frente a modelos de series temporales clásicos como el Exponential Smooth Time Series (ETS) y el modelo Arima (ARIMA). Para ello se ha estimado dichos modelos para 284 acciones pertenecientes al índice bursátil S&P 500, comparando el MAE obtenido de sus predicciones, con el modelo LSTM. Los resultados obtenidos confirman una reducción importante de los errores de predicción. Estos resultados son coincidentes con otros estudios similares aplicados a acciones de otros índices bursátiles así como a otros activos financieros como los tipos de cambio.

DOI:
10.46503/ZVBS2781

Corresponding author
Javier Oliver Muncharaz

Received: 2 Oct 2020
Revised: 15 Nov 2020
Accepted: 19 Dec 2020

Finance, Markets and
Valuation
ISSN 2530-3163.

Cómo citar este artículo: Oliver Muncharaz, J. (2020) Comparing classic time series models and the LSTM recurrent neural network: An application to S&P 500 stocks. *Finance, Markets and Valuation* 6(2), pp. 137–148.

Keywords: Redes neuronales recurrentes; Long short-term neural network; S&P 500; Arima

1 Introduction

The prediction of stock returns has been widely studied in the financial literature. One of the main objectives is the construction of stock portfolios. In some cases, these portfolios are constructed by applying different optimisation algorithms based on the classic Markowitz model (García, González-Bueno, & Oliver, 2015). Some optimisation algorithms, such as the NSGA-II, allow the construction of these portfolios taking into account more than the two classic return-risk dimensions (W. Chen, Zhang, Mehlawat, & Jia, 2021; García, González-Bueno, Oliver, & Tamošiūnienė, 2019). Other algorithms, such as genetic algorithms together with heuristic techniques allow good solutions to be found in an NP-hard problem (Ahn, Lee, Ryou, & Oh, 2020; García, Guijarro, & Oliver, 2018). Other studies try to find relations between sustainability and stock market portfolios or indexes. Thus, for example, Arribas, Espinós-Vañó, García, and Oliver (2019) analyse the composition and selection of responsible companies for the construction of portfolios and investment funds, while Espinós-Vañó, García, and Oliver (2018) analyse the so-called sustainable stock market indexes to determine whether they can be an investment alternative to traditional stock market indexes.

Stock prediction is also applied to speculate with the price evolution of financial assets. Predictive models can be classified into linear and non-linear models. To the first group belong time series models such as autoregressive integrated moving average, exponential smooth models and generalized autoregressive conditional heteroskedasticity, among others. Neural networks are assigned to a second group. Given that the stock returns are not stationary and present long-term dependence (Barkoulas & Baum, 1996), this second group of models have shown to obtain more accuracy and a reduction of errors prediction with respect to linear models.

In the last few decades, neuronal networks have evolved from the initial models of Hebb, which proposed the first rules of learning processes between neurons, Widrow and Hoff with their model Adaline, and Rosenblatt, which proposed the well-known perceptron. There are many works related to the prediction of stock prices and their trends applying models based on artificial neuronal networks. In Qiu and Song (2016), for example, the authors present the analysis of the prediction or the direction of stock price index for the Nikkei 225 stock market index. In this case, they use a backpropagation neural network with two different types of inputs to determine what kind of information improves results. The result suggests that the network is able to select those variables that are suitable for the model. Moghaddam, Moghaddam, and Esfandyari (2016) apply the same type of network for the prediction of the Nasdaq stock index using historical short term data as inputs (between four and nine days). In García, Guijarro, Oliver, and Tamošiūnienė (2018) the authors apply a neural network to predict the trend of the German Dax-30 stock index.

In recent years, recurrent neural networks have been widely used for the analysis of time series with high time dependency, such as stock returns. These types of networks are based on Rumelhart's work on error backpropagation (Rumelhart, Hinton, & Williams, 1986) and Hopfield's networks (Hopfield & Tank, 1985). Some works make a comparative analysis between some already known neural networks against the recurrent neural networks as for example in Saad, Prokhorov, and Wunsch (1998). The authors compare the time delay neural network (TDNN), probabilistic neural network (PNN) and a recurrent neural network to predict short-term stock trends. The conclusion is that the TDNN has the capability to dynamically incorporate past data to internal recurrence, and is the most powerful network among those analysed, although it presents certain complexity regarding its implementation and memory requirement.

On the other hand, Yoshihara, Fujikawa, Seki, and Uehara (2014) compare RNN with another machine learning methodology, the support vector machine (SVM) model. They analyse trend market prediction on Nikkei companies. In this case, in addition to incorporating numerical inputs into the models, information on different economic and financial events is introduced (news reported by newspapers). The advantage of deep learning is its ability to automatically construct different features from data as well as pattern recognition. The results from RNN were compared with SVM, and present the lowest error rate. This study shows that recurrent models are more effective in capturing past events that are significant with respect to long-term effects.

Other studies also confirm these results, like the paper by Rather, Agarwal, and Sastry (2015) who analyse 25 stock returns from Bombay stock exchange indicating that this model is capable of capturing non-linear patterns more efficiently than classical models. In this case it is concluded that the RNN learning process improves as it needs to look for smaller weights. da Silva, Spatti, Flauzino, Liboni, and dos Reis Alves (2016) analyse stocks in the Bovespa index, which is an alternative for decision making in the financial stock markets. RNN has also been applied to other types of financial assets. Ye (2017) focuses on forecasting exchange rate using gradient descent method or hidden layer in the process learning for recurrent neural networks. Recurrence in neurons causes a speed up the weights update as well as convergence. This confirms the reliability and stability of neural networks.

There are different types of recurrent neural networks. The Long Short-Term Recurrent Neural Networks are the most powerful time dynamic neural network (Staudemeyer & Rothstein Morris, 2019). This type of network has been applied in many areas. Among others, in the text translation (Datta, David, Mittal, & Jain, 2020; Nowak, Taspinar, & Scherer, 2017), large vocabulary speech recognition (Li & Wu, 2015), medicine diagnostic (Choi, Schuetz, Stewart, & Sun, 2016; Gao, Zhang, Lu, & Wang, 2019), traffic control in cities or its environmental impact (Awan, Minerva, & Crespi, 2020). Also for forecasting economics and financial time series (Siami-Namini, Tavakoli, & Namin, 2018).

In Section 2, the well-known time series models, Arima and Exponential Smoothing, are described. This is followed by a more in-depth description of Recurrent Neural Networks (RNN) and the particular Long Short-Term Recurrent Neural Network (LSTM). Section 3 describes the main results obtained in the work. Finally, Section 4 summarizes the main conclusions.

2 Data and Methods

In this section, the models used in this work are discussed. First, the well-known time series models, Exponential smoothing and Autoregressive moving average model, are presented in the summary form. Next, the recurrent neural network model is described in more detail, in particular, the long-short term memory recurrent neural network model.

2.1 Exponential smoothing model

The exponential smoothing model was proposed at the end of the ‘50s (Brown, 1959; Holt, 2004; Winters, 1960). In this type of model, past observations are weighted in a way that declines exponentially the further back in time.

In other words, more recent observations are associated with a greater weight while older observations have lower weights.

The model can be expressed as follow:

$$\hat{y}_{t+1} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \ldots \tag{1}$$

Where $\alpha \in [0,1]$ is the smoothing parameter. Thus, the prediction at instant $t+1$ is a weighted average of the observations of y. The degree of the weightings decrease is controlled by the parameter α.

2.2 Autoregressive moving average model

The Autoregressive moving average model (ARMA) was introduced by Box, Jenkins, and Reinsel (1970). It is one of the classic models that analyses time series, and one of the most used in financial literature. These models can become stationary by differencing. Generally, in most economic and financial time series, a single differentiation is enough to make the series stationary and to be able to apply ARIMA models where the "I" represents the level of differentiation (integration) of the series.

Having a time series X_t where t represents the time index, the $ARMA(p,q)$ model is expressed as:

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \ldots + \alpha_p X_{t-p} - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \ldots - \theta_q \epsilon_{t-q} + \epsilon_t \tag{2}$$

Where α and θ are estimated coefficients and ϵ are white noise errors.

The $ARMA(p,q)$ model is built as the combination of two processes. The first is the autoregressive process (AR), which tries to predict the variable using a linear combination of past values of this variable. An autoregressive model of order p, represents the number of lagged variable. On the other hand, the moving average (MA) part gives a prediction of the variable from a moving average model on past prediction errors. The order q of this process represents the number of delays over the prediction errors used in the model.

2.3 Recurrent neural network: LSTM network

Recurrent neural network (RNN) is a network that has backward connections between neurons, which are generally referred to as global recurrent networks. This type of model presents some stability problems in the training process, so it requires complex learning algorithms as well as increasing the training time. Local networks models are global feedforward networks. In this case, a structure of dynamic models of neurons is designed to build a feedback network, in which the connections between these neuron models are strictly feedforward as in the case of Multilayer Perceptron (MLP).

Figure 1 shows an example of the connections between the different layers and neurons in a recurrent neural network. It can be seen how the output obtained in one layer serves as an input for the neurons of layers located in a previous process. Each recurring unit formed by different neurons computes at each time step an output y_t which is time dependent on the current process. In the next time step, the neuron receives a new input vector x_t and additionally incorporates the output obtained previously into this vector (y_{t-1}). The latter is called the recurrent input. In this way, the neuron computes the output vector from the input vector (x_t) and the recurrent input (y_{t-1}) using an activation function θ. This activation function can be of the linear, sigmoidal or tanh type, although the last type of function is often used for time series problems.

$$y_t = \theta\left(W \times x_t, U \times y_{t-1}\right) \tag{3}$$

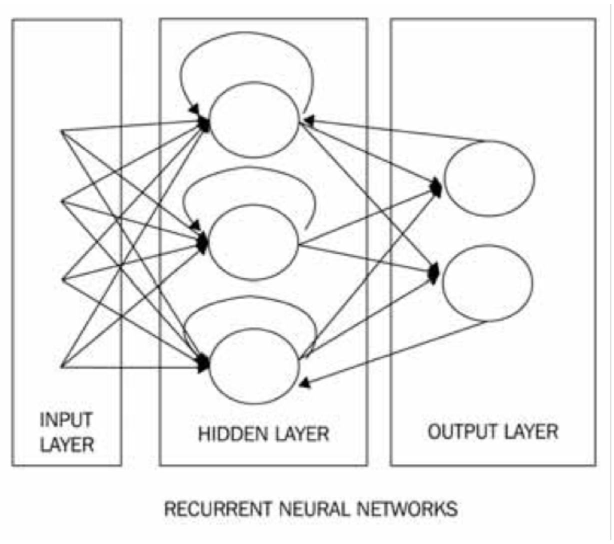


Figure 1. Recurrent Neural Network representation

Source: Ciaburro and Venkateswaran (2017)

Where W and U are weight matrix that multiply the input vector and the recurrent input vector.

The simplest recurrent neural network, called "vanilla", presents as recurrent inputs only a single output obtained in a previous time step. When the net uses the previous outputs as a new input, the net can remember learned previous data. This process is important for the learning long short-dependencies.

The importance of inputs and recurrent inputs in the net depends on their corresponding weight matrix. During the learning process, the net adjusts the weights to improve the prediction, taking into account the calculated error (bacpropagation process). However, while in a feed forward network the backpropagation process goes back through the hidden layers, for recurrent neural network it is also necessary to adjust the weights of previous time steps (time adjusting). In this type of network, if the sequence is long, there may be a problem in the learning process, since with each prediction the whole way backwards must be covered again. To avoid this, a split of the different sequences is made. In this way, the backpropagation process should only go backwards the length of the subsequence. But in this case, the neural network is only able to determine short dependencies. This is the so-called vanishing gradient problem, in which the further back the sequence is regressed the less important it is in the current prediction, and therefore cannot adequately capture long term dependencies.

The Long Short-Term Neural Network (LSTM) is a more complex kind of recurrent neural network as it is able to capture long-term dependencies. This kind of neural network was proposed by Hochreiter and Schmidhuber (1997) as an evolution of simple RNN. This network can propagate activations over long periods to process different sequences that include long distance dependencies (Kelleher, 2019). This network solves the vanishing gradient problem. In this case, the recurrent unit is modified in blocks, which works like a normal recurrent unit, to which an additional cell and several gates are added. The gates control the flow of information within the recurrent unit (block). In this way it is determined which information is more revealing to improve the prediction and which is not at each of the time steps. These gates (τ) are defined

Type of gate	Role
Forget	Eliminate neuron
Update	Importance of the past
Relevance	Drop previous information
Output	Which information is used

Table 1. Role Gates

Source: Author' elaboration

as follow:

$$\tau = \sigma \left(W \times x_t + U \times y_{t-1} + b \right) \tag{4}$$

Where W, U and b are gate specific coefficients and σ is the common used sigmoid function (while in the input activation function use a tanh function). In the LSTM recurrent neural network, four gates are used, each with a different function. Table 1 summarizes the role function that one of them has.

The forget gate is used to erase a neuron or not, and therefore, forget the information. On the other hand, the update gate indicates what is the past to be taken into account now. The relevance gate defines what information from the past is relevant to incorporate as input to the neuron. Finally, the output gate selects the information that is useful for the neuron in the actual prediction. Each of these four gates uses different weight matrices and are calculated individually during the learning process. In short, these gates control the flow of information in each neuron so that it is useful in predicting at each time step.

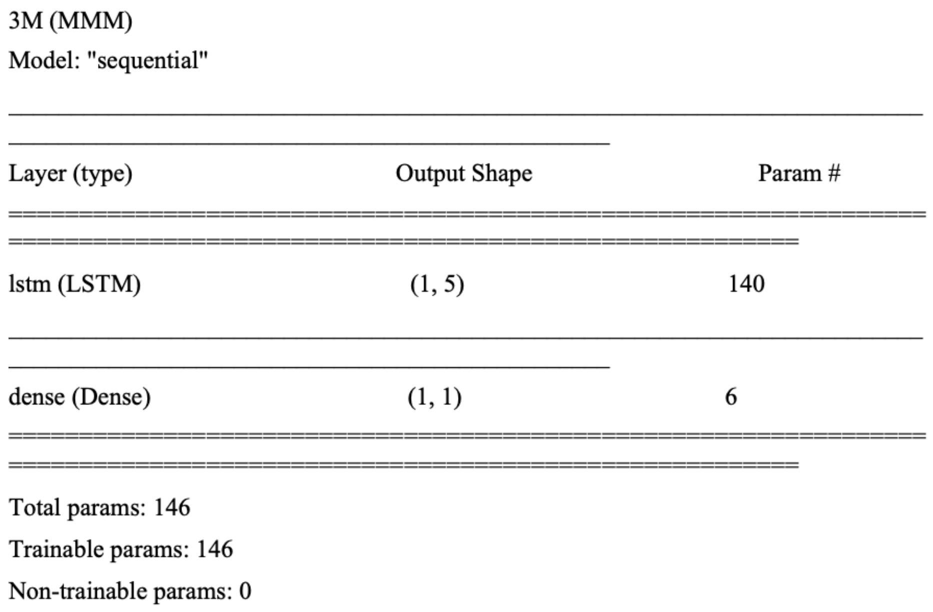
3 Results

In this work, it is intended to confirm the efficiency of a LSTM Neural Network, as opposed to some classic models applied to time series. In this case, it is going to be compared with an Exponential Smooth Time Series model and an ARIMA model. For this purpose, each of them has been applied to a sample of 284 stocks from the S&P 500 index with daily data from the last 20 years. The sample has been divided into 70% for the estimation and training processes and 30% for its validation.

For the Exponential Smooth Time Series and ARIMA model, the number of differentiations needed to obtain stationary time series, which is required in this type of models, has been taken into account. As with many economic time series (McCabe & Tremayne, 1995), only one integration was necessary to achieve stationarity.

For the Exponential Smooth Time Series model, the AIC (Akaike) crystals have been used to select the appropriate delays in each case. In the case of the ARIMA models, Phillips Perron's criteria has been used to determine the delays of both the autoregressive part and the moving averages.

In the case of the LSTM Neural Network model, the series has been standardised, both for training and for testing. There are many works that verify that standardisation improves the learning process in neural networks such as Lachtermacher and Fuller (1995); Shen, Zhang, Lu, Xu, and Xiao (2020); Zhang, Patuwo, and Hu (1998).



```
3M (MMM)
Model: "sequential"
_________________________________________________________________
_____________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
=======================================================
lstm (LSTM)                   (1, 5)                  140
_________________________________________________________________
_____________________________________________
dense (Dense)                 (1, 1)                  6
=================================================================
=======================================================
Total params: 146
Trainable params: 146
Non-trainable params: 0
```

Figure 2. Example LSTM parameters for 3M stock

Source: Author' elaboration

Figure 2 shows the type of network applied to each of the actions. As already indicated, it is a recurrent long short-term neural network. In this case, the processes the sequence of vectors using a LSTM layer of input data. This model presents the different layers in a sequential way. As the model needs to know in the first layer the type of input it should expect, (since in the rest of layers it is inferred automatically), the number of samples per gradient updates is one. That is, the number of batch size of the inputs for the layer is one. The dimension of the output space of this first layer is five. This parameter is subject to tuning. The output layer is full-fully connected (dense layer), it has been configured with a batch size of one and one unit.

The three models have been estimated for each stock with the corresponding appropriate delays according to the criteria already indicated. For the evaluation of the efficiency of each model, the Mean Absolute Error (MAE) has been calculated on the predicted observations. Figure 3 shows a boxplot with the errors of each model for the 3M action. In the 284 stocks analysed, two important issues have been observed. Firstly, the Exponential Smooth Time Series model and the ARIMA model present similar MAE. For example, in the case of 3M the MAE obtained in the first model was 0.8847, while for the ARIMA model it was 0.8857. It is possible that by applying other exponential models such as Holt-Winters' that could relatively improve the ARIMA model (Maria & Dezsi, 2011). Secondly, even in the case of using other exponential models that improve the ARIMA model, they are far removed from the results obtained from MAE for the LSTM model. For example, for 3M the MAE obtained was 0.1823, that is, 79% less error than the other models.

Table 2 describes the main statistics on the distribution of the MAE obtained in the total number of actions analysed and for each of the models. On the one hand, it can be seen that the classic time series models (ETS and ARIMA) present a higher MAE for all the quantiles of the sample in comparison with the LSTM model, as it was already advanced in the previous

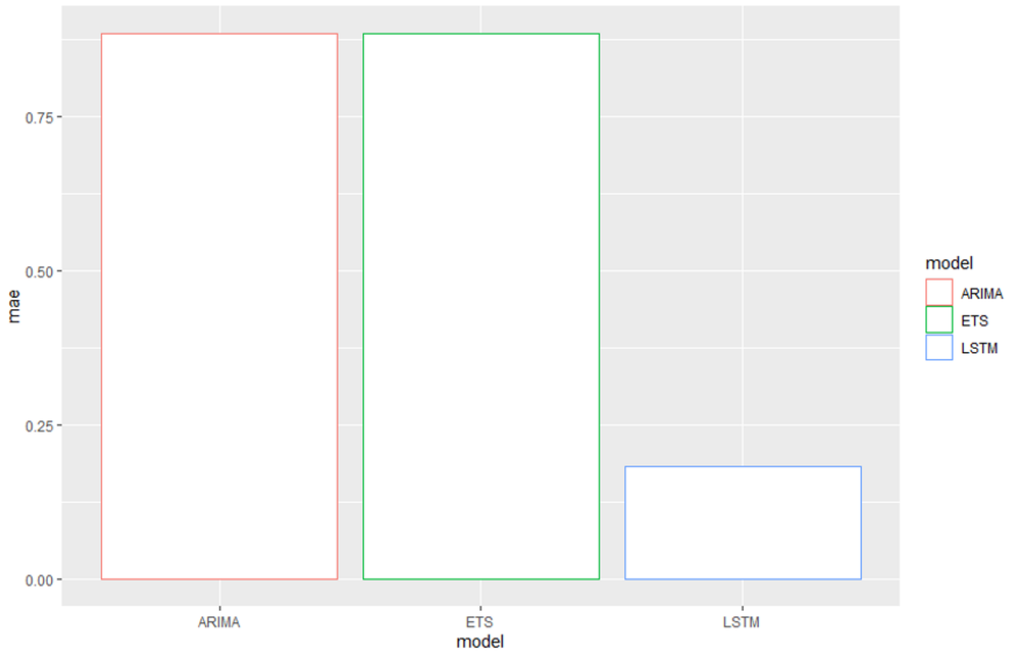


Figure 3. MAE comparison model for 3M stock

Source: Author' elaboration

example. These results are consistent with other studies such as Siami-Namini et al. (2018) that compare these three models for various stock market indexes such as Nasdaq, Nikkei, Hang Seng with monthly data. The results suggest that the LSTM model obtains, on average, a reduction in prediction error of between 84 and 87 percent. Baughman, Haas, Wolski, Foster, and Chard (2018) compare the ARIMA model with the LSTM for Amazon stock, obtaining an error reduction of 95%.

However, four stocks have been detected in which the MAE obtained by the LSTM model is superior to any of the other two models (Table 3). Each of these stocks has been analyzed in detail to detect if this result is due to some kind of error in the sample. In all four cases, there are a sufficient number of observations (several thousand). Neither have any missing or anomalous data been detected. Likewise, the quoted prices have been visually contrasted without apparently detecting any errors. The first two shares, LDOS and IRM, are listed on the NYSE, while CTSH and CHTR are listed on the NASDAQ. It can therefore be concluded that the results obtained for these four shares are plausible. However, the LSTM model has outperformed the classical time series models in 98.59% of the sample analysed, so recurrent neural networks are a good alternative for time series prediction in general, and for stocks and stock indices in particular. Abdoli (2020) analyses the Tehran Stock Exchange confirming the results obtained in the present work, where the LSTM outperforms ARIMA model, in terms of error of accuracy.

4 Conclusions

In this work, the efficiency of the Long short-termn recurrent neural network has been analysed in comparison with other time series models. The main conclusion that can be drawn is that there is a large reduction in the prediction error of more than 85%, which is in line with previous

	ETS	ARIMA	LSTM
Min.	0.0626	0.1441	0.0009
1st Qu.	0.3842	0.3866	0.0565
Median	0.5678	0.5766	0.1289
Mean	0.8177	0.8320	0.2199
3rd Qu.	0.8374	0.8397	0.2708
Max.	9.6813	9.7285	3.1888
Sd.	1.0555	1.0693	0.3000

Table 2. Descriptive Statistics

Source: Author' elaboration

TICKER	ETS	ARIMA	LSTM
LDOS	0.6080	0.6118	0.9766
IRM	0.2709	0.2701	0.5742
CTSH	0.3843	0.3853	0.9812
CHTR	2.6014	2.6120	3.1888

Table 3. Descriptive Statistics

Source: Author' elaboration

results from other studies on other financial assets. Recurrent neural networks in general, and the LSTM in particular, may be an alternative to consider in the creation of stock price prediction models. However, to confirm these results, this analysis should be extended to other aspects such as the application of a larger number of fully connected intermediate layers or the application of tuning of other network parameters.

On the other hand, other authors have proposed other types of neural networks that seem to offer very efficient alternatives, as well. In M, E.A., Menon, and K.P. (2018) the authors compare several linear time series models (ARIMA) with non-linear models such as ARCH, GARCH and Neural Networks. In this case, they apply two types of recurrent neural networks, one LSTM model and the other Convolutional Neural Network. This network is applied to five stocks of the National Stock Exchange (NSE) of India. The results suggest that the Convolutional Neural Network outperforming the other models, even against the LSTM model. In the same line, the works of Y. Chen, Wei, and Huang (2018)applying a Convolutional model to the prediction of the stock market in Mainland China incorporating related corporations' information to create more accuracy in predictions are presented. In addition, other works propose the use of a hybrid model between the Convolutional Neural Network and the recurrent neuronal network LSTM (Kim & Kim, 2019).

References

Abdoli, G. (2020). Comparing the prediction accuracy of LSTM and ARIMA models for time-series with permanent fluctuation. *SSRN Electronic Journal*. doi:

https://doi.org/10.2139/ssrn.3612487

Ahn, W., Lee, H. S., Ryou, H., & Oh, K. J. (2020). Asset allocation model for a robo-advisor using the financial market instability index and genetic algorithms. *Sustainability*, *12*(3), 849. doi: https://doi.org/10.3390/su12030849

Arribas, I., Espinós-Vañó, M. D., García, F., & Oliver, J. (2019). Defining socially responsible companies according to retail investors' preferences. *Entrepreneurship and Sustainability Issues*, *7*(2), 1641–1653. doi: https://doi.org/10.9770/jesi.2019.7.2(59)

Awan, F. M., Minerva, R., & Crespi, N. (2020). Improving road traffic forecasting using air pollution and atmospheric data: Experiments based on LSTM recurrent neural networks. *Sensors*, *20*(13), 3749. doi: https://doi.org/10.3390/s20133749

Barkoulas, J. T., & Baum, C. F. (1996). Long-term dependence in stock returns. *Economics Letters*, *53*(3), 253–259. doi: https://doi.org/10.1016/s0165-1765(96)00935-4

Baughman, M., Haas, C., Wolski, R., Foster, I., & Chard, K. (2018). Predicting amazon spot prices with LSTM networks. In *Proceedings of the 9th workshop on scientific cloud computing.* ACM. doi: https://doi.org/10.1145/3217880.3217881

Box, G. E., Jenkins, G. M., & Reinsel, G. (1970). *Time series analysis: forecasting and control holden-day san francisco*. Holden Day.

Brown, R. G. (1959). *Statistical forecasting for inventory control*. McGraw/Hill.

Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean–variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, *100*, 106943. doi: https://doi.org/10.1016/j.asoc.2020.106943

Chen, Y., Wei, Z., & Huang, X. (2018). Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management.* ACM. doi: https://doi.org/10.1145/3269206.3269269

Choi, E., Schuetz, A., Stewart, W. F., & Sun, J. (2016). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, *24*(2), 361–370. doi: https://doi.org/10.1093/jamia/ocw112

Ciaburro, G., & Venkateswaran, B. (2017). *Neural networks with r: Smart models using cnn, rnn, deep learning, and artificial intelligence principles*. Packt Publishing Ltd.

da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., & dos Reis Alves, S. F. (2016). Forecast of stock market trends using recurrent networks. In *Artificial neural networks* (pp. 221–227). Springer International Publishing. doi: https://doi.org/10.1007/978-3-319-43162-8_13

Datta, D., David, P. E., Mittal, D., & Jain, A. (2020). Neural machine translation using recurrent neural network. *International Journal of Engineering and Advanced Technology*, *9*(4), 1395–1400.

Espinós-Vañó, M. D., García, F., & Oliver, J. (2018). The ethical index FTSE4good IBEX as an alternative for passive portfolio strategies in spain. *Finance, Markets and Valuation*, *4*(1), 85–93. doi: https://doi.org/10.46503/mukb2397

Gao, J., Zhang, H., Lu, P., & Wang, Z. (2019). An effective LSTM recurrent network to detect arrhythmia on imbalanced ECG dataset. *Journal of Healthcare Engineering*, *2019*, 1–10. doi: https://doi.org/10.1155/2019/6320651

García, F., González-Bueno, J., Oliver, J., & Tamošiūnienė, R. (2019). A credibilistic mean-semivariance-per portfolio selection model for latin america. *Journal of Business Economics and Management*, *20*(2), 225–243. doi: https://doi.org/10.3846/jbem.2019.8317

García, F., González-Bueno, J. A., & Oliver, J. (2015). Mean-variance investment strategy applied in emerging financial markets: Evidence from the colombian stock market. *Intellectual Economics*, *9*(1), 22–29. doi: https://doi.org/10.1016/j.intele.2015.09.003

García, F., Guijarro, F., & Oliver, J. (2018). Index tracking optimization with cardinality constraint: a performance comparison of genetic algorithms and tabu search heuristics. *Neural Computing and Applications*, *30*(8), 2625–2641. doi: https://doi.org/10.1007/s00521-017-2882-2

García, F., Guijarro, F., Oliver, J., & Tamošiūnienė, R. (2018). Hybrid fuzzy neural network to predict price direction in the german dax-30 index. *Technological and Economic Development of Economy*, *24*(6), 2161–2178. doi: https://doi.org/10.3846/tede.2018.6394

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

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, *20*(1), 5–10. doi: https://doi.org/10.1016/j.ijforecast.2003.09.015

Hopfield, J. J., & Tank, D. W. (1985). “neural” computation of decisions in optimization problems. *Biological cybernetics*, *52*(3), 141–152.

Kelleher, J. D. (2019). *Deep learning*. Mit Press.

Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLOS ONE*, *14*(2), e0212320. doi: https://doi.org/10.1371/journal.pone.0212320

Lachtermacher, G., & Fuller, J. D. (1995). Back propagation in time-series forecasting. *Journal of Forecasting*, *14*(4), 381–393. doi: https://doi.org/10.1002/for.3980140405

Li, X., & Wu, X. (2015). Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In *2015 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 4520–4524).

M, H., E.A., G., Menon, V. K., & K.P., S. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science*, *132*, 1351–1362. doi: https://doi.org/10.1016/j.procs.2018.05.050

Maria, F. C., & Dezsi, E. (2011). Exchange-rates forecasting: Exponential smoothing techniques and arima models. *Annals of Faculty of Economics*, *1*(1), 499–508.

McCabe, B. P., & Tremayne, A. R. (1995). Testing a time series for difference stationarity. *The Annals of Statistics*, 1015–1028.

Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, *21*(41), 89–93. doi: https://doi.org/10.1016/j.jefas.2016.07.002

Nowak, J., Taspinar, A., & Scherer, R. (2017). LSTM recurrent neural networks for short text and sentiment classification. In *Artificial intelligence and soft computing* (pp. 553–562). Springer International Publishing. doi: https://doi.org/10.1007/978-3-319-59060-8_50

Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLOS ONE*, *11*(5), e0155133. doi: https://doi.org/10.1371/journal.pone.0155133

Rather, A. M., Agarwal, A., & Sastry, V. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, *42*(6), 3234–3241. doi: https://doi.org/10.1016/j.eswa.2014.12.003

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Saad, E., Prokhorov, D., & Wunsch, D. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, *9*(6), 1456–1470. doi: https://doi.org/10.1109/72.728395

Shen, Z., Zhang, Y., Lu, J., Xu, J., & Xiao, G. (2020). A novel time series forecasting model with deep learning. *Neurocomputing*, *396*, 302–313. doi: https://doi.org/10.1016/j.neucom.2018.12.084

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th ieee international conference on machine learning and applications (icmla)* (pp. 1394–1401).

Staudemeyer, R. C., & Rothstein Morris, E. (2019). Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv*, arXiv–1909.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, *6*(3), 324–342. doi: https://doi.org/10.1287/mnsc.6.3.324

Ye, Y. (2017). Study on exchange rate forecasting using recurrent neural networks. *International Journal of Economics, Finance and Management Sciences*, *5*(6), 300. doi: https://doi.org/10.11648/j.ijefm.20170506.14

Yoshihara, A., Fujikawa, K., Seki, K., & Uehara, K. (2014). Predicting stock market trends by recurrent deep neural networks. In *Pacific rim international conference on artificial intelligence* (pp. 759–769).

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:. *International Journal of Forecasting*, *14*(1), 35–62. doi: https://doi.org/10.1016/s0169-2070(97)00044-7