How Boltzmann Entropy Improves Prediction with LSTM

Grilli, Luca and Santoro, Domenico

22 May 2020

How Boltzmann Entropy Improves Prediction with LSTM

Luca Grilli* Domenico Santoro*

Abstract

In this paper we want to demonstrate how it is possible to improve the forecast by using Boltzmann entropy like the classic financial indicators, throught neural networks. In particular, we show how it is possible to increase the scope of entropy by moving from cryptocurrencies to equities and how this type of architectures highlight the link between the indicators and the information that they are able to contain.

Keyboards: Neural Network; Price Forecasting; LSTM; Entropy

JEL codes: C45; E37; F17; G17

MSC: 62M45; 91G60; 97R40

*Università degli Studi di Foggia, Dipartimento di Economia, Management e Territorio, Via Da Zara, 11, I-71121 Foggia, Italy

Corresponding author: Luca Grilli, email: luca.grilli@unifg.it

1 Introduction

The study of the dynamics of the financial markets and the forecast of stock prices has always attracted researchers, focusing on different methodologies for the study of time series. The study methodology of these series base their assumptions on the *random walk hypothesis*, a concept introduced by Bachelier [2] in 1900. Traditionally, most common practice was to focus on logarithmic returns, bringing the advantage of linking statistical analysis with financial theory. Fama [7] introduced in his EMH theory (*Efficient Market Hypothesis*) the idea that historical prices cannot be used to make predictions since all information is contained in the current price. However, LeRoy [16] showed that the mere concentration on yields was unjustified, defining the stock markets inefficient. The historical approach to the analysis of time series is based on the study of Fama and French [6], Campbell and Shiller [27] and Timmermann [12].

In the last few years, thanks to the development of *artificial neural networks* (ANNs) and their applicability to non-linear modeling [28], there has been a strong interest in the application of these methods to time series prediction, e.g. Refenes et al. [20], Sharda et al. [22] and Dixon [5]. The evolution of the techniques of *Machine Learning* (ML) and *Deep Learning* (DL) has introduced many advantages. As for ML techniques, a great innovation was introduced with the development of the *Support Vector Machine* (SVM) model by Vapnik [25], which solved the problem of pattern classification, with the consequent application to the time series forecasting [1]; Mittermayer et al. [19] and Kara et al. [15]. As for the DL techniques, increasingly complex architectures are being used such as Liu et al. [17] who use a CNN + LSTM, Zhang et al. [29] who use an SFM to predict stock prices, Chen et al. [4] and Mäkinen et al. [18] who propose an LSTM architecture for predicting prices or Sirignano et al. [23] that build a "spatial neural network". However, many other types of more complex networks can be readjusted to time series to make predictions, such as GAN networks (based on the idea of Goodfellow et al. [9]) used for speech synthesis [14] or for the denoising of images [24] and readjusted as in the case of Wiese et al. [26] who build a Quant GANs highlighting the characteristics of the generated data.

In this paper, we want to demonstrate - through the use of an LSTM architecture - that the *Boltz-*

mann's entropy [10] is a reliable indicator for forecasting purposes. In a previous work [11] we have highlighted through a particular GAN network how there is a difference in the "quantity of information" contained in the prices of the financial instruments on the market subject to timetables (stocks) compared to one whose timetables are absent (cryptocurrencies); highlighting that in closing markets there are instruments that despite everything have a great deal of information, e.g. in the case of stocks, some of them behave like cryptocurrencies. In this case we will try to demonstrate how those stocks classified as outliers are more suitable for forecasting.
The paper in section 2 we present the structure of the LSTM and its functioning; in section 3 we explain the construction of the entropy indicator and its use compared with the classic forecast indicators; in section 4 we apply Boltzmann entropy in an LSTM; finally in section 5 some conclusions are drawn.

2 Long-Short Term Memory

2.1 RNN

Recurrent Neural Networks (RNN) are essentially neural networks with feedback connections. In this way, given the considerable flow of information that is generated, training requires considering different time instants (the so-called *unfolding in time*). Thanks to their memory effect, RNN are the most suitable neural networks for managing sequences. In this tipe of neural networks the new state h_t is determined:

$$h_t = f_W(h_{t-1}, x_t) \tag{1}$$

where f_W is the function parameterized by the weights and x_t is the input vector at time step t.

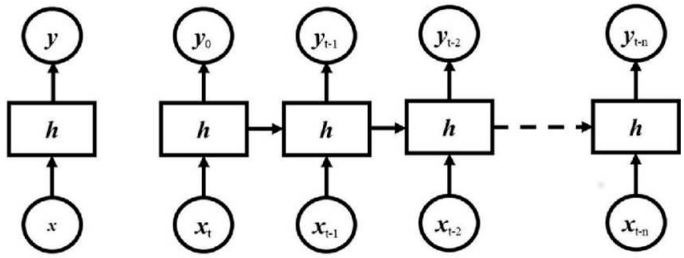


Figure 1: RNN unfolding in time [8]

Backpropagation by Rumelhart et al. [21] is an algorithm for training neural networks through which the gradient of the overall loss function $J(W)$ is computed. In RNN a particular version of this algorithm is used: *Backpropagation Throught Time* (BPTT) the fundamental difference is that in this version of the algorithm gradients are computed for each time step. The main problem is that in this way the network is exposed to the problem of the exploding or (in an opposite way) of the vanishing of the gradient. To prevent this problem we can use specific types of cells to control the transmission of information, such as the *Long-Short Term Memory* (LSTM).

2.2 LSTM

The *Long-Short Term Memory* (LSTM) cell (introduced by Hochreiter et al. [13] in 1997) is one of the most used since it has a memory effect. Thanks to this feature, it is among the most suitable to combat the problem of vanishing gradients [3]. The characteristic of this type of network is that, at each step a level it receives in addition to the input also the output of the previous level, so that it can base its decisions on history. However, since distant memory tends to fade in base cells, the LSTM prevents this from happening through its long-term memory. Each LSTM cell controls the flow of information: forget the irrelevant things, update cell state values and use an output gate to output parts of the cells state,

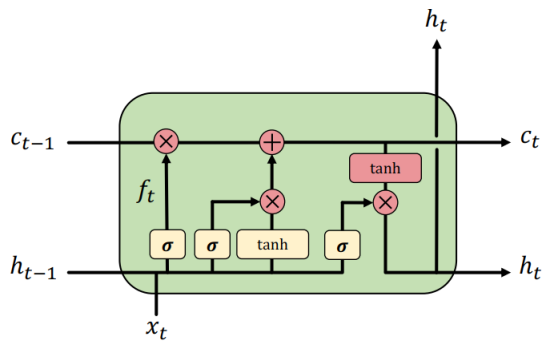


Figure 2: LSTM cell

The operations described above are defined by the following equations:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
\tilde{C}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
C_t &= C_{t-1} \odot f_t + i_t \odot \tilde{C}_t \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + V_o C_t + b_f) \\
S_t &= o_t \odot \tanh(C_t)
\end{aligned}
\tag{2}
$$

where x_t is the input vector at time t, $(W_f, W_i, W_c, W_o, U_f, U_i, U_c, U_o)$ are the weight matrices, (h_f, h_i, h_c, h_o) are the bias vectors, h_t is the value of the cell at time t, i_t and $\tilde{C}_t$ are are values of the input gate and the candidate state of the memory cell, f_t and C_t t are values of the forget gate and the state of the memory cell and o_t is the value of the output gate. Moreover, $\odot$ represents the Hadamard product and σ is the logistic *sigmoid* activation function.

3 Methodology

The aim of this paper is to demonstrate that Boltzmann's entropy determined by us previously [10] can be an effective indicator for making forecast like those traditionally used in technical analysis.

The indicators to which we will refer in this case (which represent the most used for simplicity and effectiveness) are:

- MACD *(Moving Average Convergence/Divergence)*, based on the convergence and divergence of three moving averages. The first at 12 periods, the second at 26 and an exponential average at 9 periods; determined as:

$$\begin{aligned} MACD &= EMA12 - EMA26 \\ Signal &= EMA9(MACD) \end{aligned} \tag{3}$$

- Stochastics oscillator, it is used to study price fluctuations and provides market entry and exit signals. The oscillator $\%D$ is calculated as:

$$\begin{aligned} \%K &= \frac{X - \min 14(X)}{\max 14(X) - \min 14(X)} * 100 \\ \%D &= EMA9(\%K) \end{aligned} \tag{4}$$

  where X is the closing price and $\max 14$ and $\min 14$ refer to a period of 14 days;

- RSI *(Relative Strength Index)*, it is used to identify the oversold and overbought areas, highlighting the ideal timing to enter and exit the market. It is calculated as:

$$RSI = 100 - \frac{100}{1 + \frac{EMA14(X)_{UP}}{EMA14(X)_{DOWN}}} \tag{5}$$

  also in this case the moving averages are calculated over a period of 14 days.

The fourth indicator used is Boltzmann entropy. In our previous work [10] we have shown how it is possible to assimilate a cryptocurrency system to a thermodynamic system whose entropy can subsequently be calculated. In this case we have a system made up of financial instruments in which we can make some similar assumptions. In particular, we can assume that the reference system is made up of N economic subjects (agents) who intend to trade in stocks. These agents, also in this case, are completely identified by 2 variables $\{x_i, y_i\}$ where, however, x_i and y_i indicate the ability

to buy and to sell a quantity of a certain stocks (expressed in monetary terms). We suppose that the market to which we refer is influenced only by the supply and demand leverage and the key assunption is that we can consider as a conserved quantity the total number of stocks in circulation which by their definition is constant over a suitable time interval (fixed by definition) throught the function $M(x_i, y_i)$. Moreover, we know that every subjects in our new system is described by its ability to buy and sell and that these two variables are summarized in the *closing price* of the stocks: the closing price allow us to understand whether the ability to buy or sell prevailed.

In this case, the main difference is that in calculating the gaps and consequently the entropy value we still consider 5-day clusters, but these clusters are calculated "dynamically" e.g. for the first closing price the maximum and the minimum are calculated from the period t_1 to t_5; for the second price the maximun and the minimum are calculated from the period t_2 to t_6 and so on. With this method we obtain a number of gaps equal to the number of observations in the dataset (the gap $\mathbf{G}$ is however calculated as the difference in terms of necessary steps to pass from max to min). Having such a large number of gaps we can calculate as many "volumes" Γ occupied by the disposition of the agents and consequently as many Boltzmann entropies through the classic formula

$$S = \kappa_B \ln \Gamma \tag{6}$$

where $\kappa_B \sim 1.3806 * 10^{-23}$ is the Boltzmann constant, and finally "rationalizing" multiplying by 10^{23}.

Through an LSTM architecture we want to demonstrate that the entropy indicator calculated in this way has a predictive capacity at least equal to that of the indicators most used in technical analysis and in addition to these, how the predictive ability of the features varies overall. However, given the simplicity of the data, the structure of the network is composed of only 1 input layer with a number of neurons from 7 to 9 (according to the widespread theory that the number of neurons in the input layer is equal to the number of features plus a bias), 1 output layer with 1 neuron only and no hidden layer. To highlight the results obtained we will consider the $RMSE$ (Root-Mean Square Error) and the R^2 of the forecast obtained on the test set.

3.1 Dataset

Empirical analysis were carried out on the closing prices of two widespread stocks[1] (therefore having a very high number of stocks in circulation which allows to fall within the assumptions of the model for entropy):

- Apple Inc. (AAPL listed on NASDAQ) whose price with 2 decimal places provides for a step equal to 0.01;
- Tesla Inc. (TSLA listed on NASDAQ) whose price with 2 decimal places provides for a step equal to 0.01.

Prices of stocks are considered with a daily time frame from 10/01/2011 to 31/12/2019. In particular, the dataset on which the calculations of the various indicators were made goes from 20/12/2010 to 8/1/2020, but the one used for the train and the network test is the one described above. The dataset consists of several columns (features), each of which (especially in the graphical analysis) will be indicated with the first letter of the column (2 in some cases); these features are: **Open** (O), **High** (H), **Low** (L), **Close** (C), **Adj Close** (A_D), **Volume** (V), **MACD** (M), **Stochastic** (S_O), **RSI** (R), **Entropy** (E).
Furthermore, the choice of titles is not entirely random but conveyed by what is defined in [11], so as to demonstrate that the forecast on the stocks classified as "outliers" is better than the competitors on a closing market.

4 Numerical examples

To test the effectiveness of the different indicators, we will analyze the different features first individually and then combining the different features in other datasets to see how the values of the R^2 changes (the features being forecast will always remain "Adj Close"). What we will demonstrate

[1] Source: finance.yahoo.com

is that entropy in some cases can be an indicator that, probably due to its construction, makes a significant improvement to the forecast.

	AAPL		TSLA	
DATASET	RMSE	R^2	RMSE	R^2
OHLCA_D	14.357	0.777	15.722	0.893
OHLA_DV	11.576	0.866	15.964	0.890
OHLA_DM	11.573	0.855	16.128	0.887
OHL$A_D S_O$	12.674	0.826	14.878	0.904
OHLA_DR	12.527	0.830	15.731	0.893
OHLA_DE	9.412	0.904	15.876	0.891
OHLA_DVM	9.806	0.896	16.112	0.888
OHLA_DVS_O	11.273	0.863	15.717	0.893
OHLA_DVR	8.650	0.919	15.904	0.891
OHLA_DVE	10.071	0.890	15.920	0.890
OHLA_DVMR	8.482	0.922	15.877	0.891
OHLA_DVRE	10.190	0.888	15.691	0.893
OHLA_DVME	9.358	0.905	15.769	0.892
OHLA_DVMRE	11.638	0.854	15.963	0.890

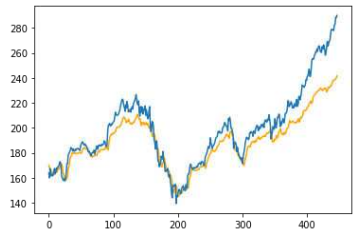
Table 1: RMSE (lower the better) and R^2 (higher the better) for the different datasets

As you can see in table 1, the RMSE values differ according to the type of stock since the prices inside move on different levels. These two indicators measure the goodness of the forecast made on a test set of over 400 values for each type of dataset and stock; however we can see that in the cases of Apple stocks, entropy without volume leads to an excellent result as well as in the dataset with the MACD.

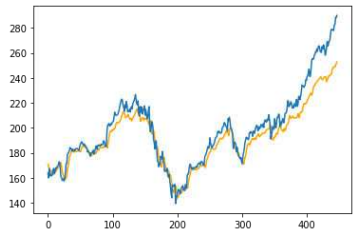
We can also notice how the neural network is able to perceive the relationships between the features,

in particular from the improvement of the forecast with the combined use of the Volume and RSI features. The entropy determined in these cases respects the main property, according to which when it reaches a local maximum and is followed by a drastic descent then at the time point following the descent it will necessarily have to rise to "re-balance" the amount of information. We can assume that it is precisely this characteristic, which we hypothesized as a tool for making prediction, that makes the neural network improve the forecast. We can also assume that the reduction of R^2 with the use of all the features is linked to the fact that entropy not only moves on different ranges from the other indicators, but that in some cases (especially with the RSI) it has peaks tendentially opposites that could somehow condition the network itself.

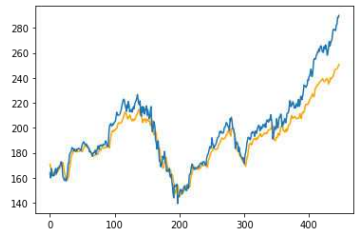
Graphically, we can compare the forecasts made for each type of dataset with respect to the actual closing prices as shown in figure 3 for AAPL stocks and figure 4 for TSLA stocks.



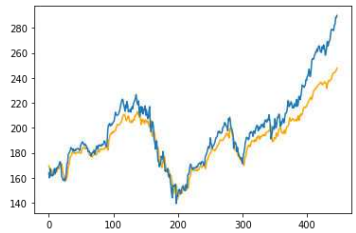
(a) OHLCA_D



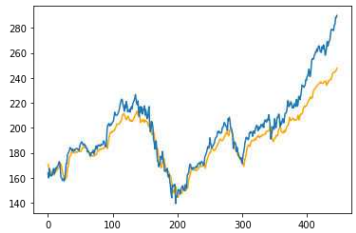
(b) OHLA_DV



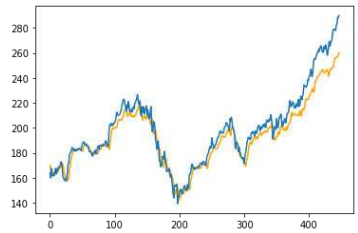
(c) OHLA_DM



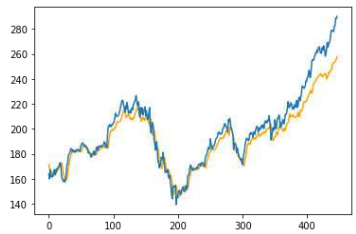
(d) OHL$A_D S_O$



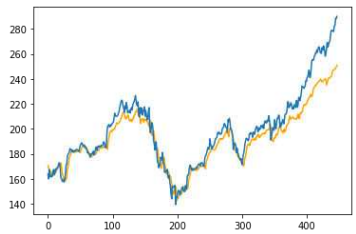
(e) OHLA_DR



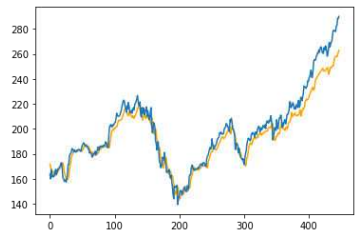
(f) OHLA_DE



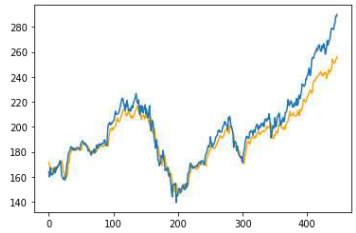
(g) OHLA_DVM



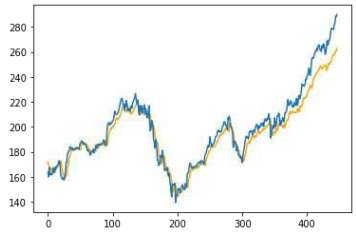
(h) OHLA_DVS_O



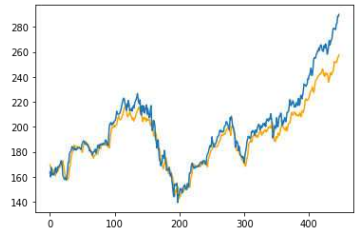
(i) OHLA_DVR



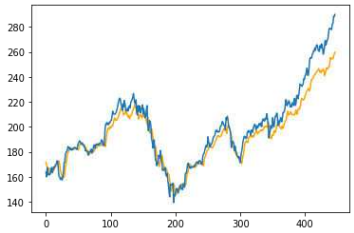
(j) OHLA_DVE



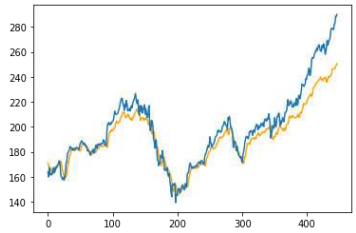
(k) OHLA_DVMR



(l) OHLA_DVRE

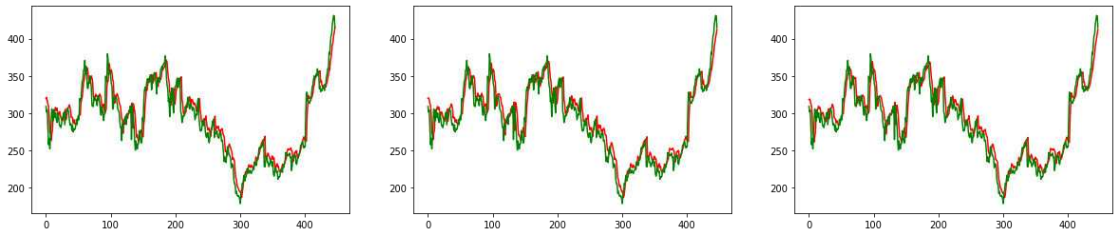


(m) OHLA_DVME



(n) OHLA_DVMRE

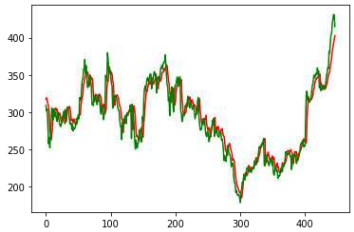
Figure 3: Original test set (blue) and predicted values (orange) for Apple stocks



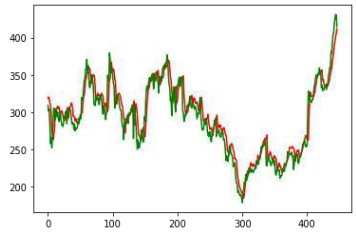
(a) OHLCA_D

(b) OHLA_DV

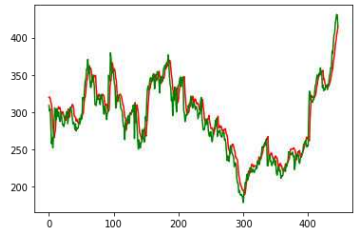
(c) OHLA_DM



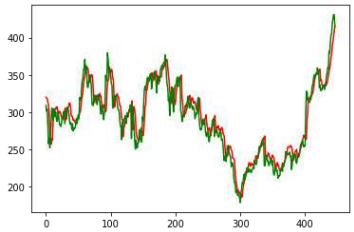
(d) OHL$A_D S_O$



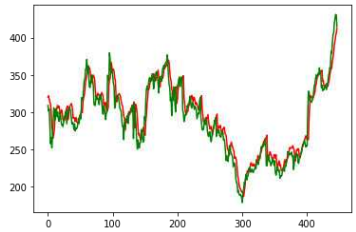
(e) OHLA_DR



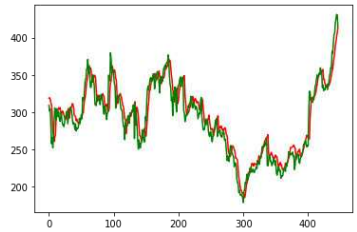
(f) OHLA_DE



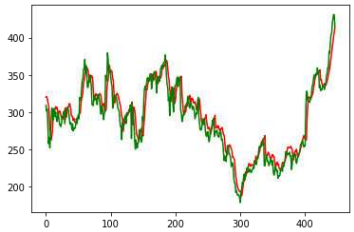
(g) OHLA_DVM



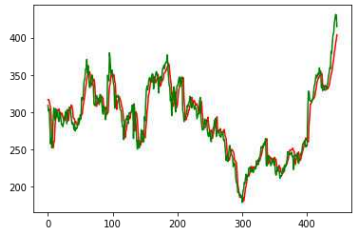
(h) OHLA_DVS_O



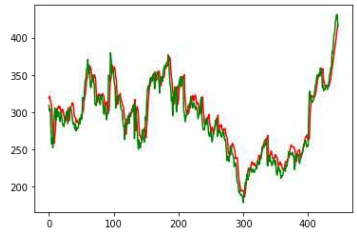
(i) OHLA_DVR



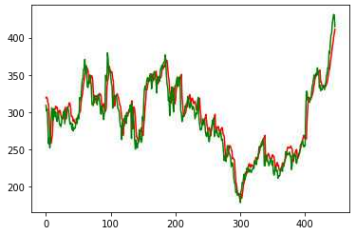
(j) OHLA_DVE



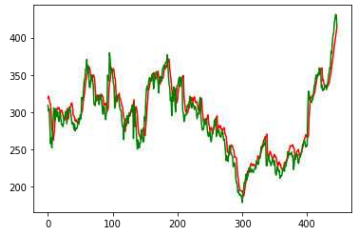
(k) OHLA_DVMR



(l) OHLA_DVRE



(m) OHLA_DVME



(n) OHLA_DVMRE

Figure 4: Original test set (green) and predicted values (red) for Tesla stocks

From this analysis we can put forward a hypothesis even if very strong. In all cases we note that entropy alone (in addition simply to the classic OHLA_DV dataset) is a very effective indicator that on average allows to reach a R^2 of 90%. The reason for this optimal result is traceable in the construction of this indicator, which being constructed "dynamically" takes into account a certain amount of information (which represents the position of economic agents with respect to buying or selling) on the basis of which it is possible to understand when there will be a movement of agents; however, this great presence of captured information, when entropy is used together with the other indicators, generates **information redundancy**. In this sense there could be multiple points where all three/four indicators have captured the same type of information, but this is not captured by the neural network (in particular since the indicators despite the same type of information could have completely opposite movements) producing a lower R^2 than using the single indicator.

5 Conclusions

In this paper, we have shown how the dynamically determined Boltzmann entropy for stocks can be an indicator on a par with those most commonly used in technical analysis. In particular, we have shown how the single use of this indicator performs better than the joint use. In addition, stocks such as Tesla (outliers in markets subject to timetables [11]) prove to be optimal for forecasting, as the predictive ability of the different features allows even with only a few of these an excellent forecast. At this point, the objective of the next works will be to exploit the Entropy indicator as a tool to verify the possible presence of *cyclicity* in the movement of economic agents.

References

[1] R. Adhikari and R. K. Agrawal. An introductory study in time series modeling and forecasting. *arXiv*, 2013.

[2] L. Bachelier. Theorie de la speculatione. *Ph.D. Thesis*, 1900.

[3] W. Bao, J. Yue, and Y. Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE 12(7)*, 2017.

[4] S. Chen and L. Ge. Exploring the attention mechanism in lstm-based honk kong stock price movement prediction. *Quantitative Finance*, 2019.

[5] M. Dixon. A high-frequency trade execution model for supervised learning. *High Frequency*, 2017.

[6] Fama E. F. and French K. R. Dividend yields and expected stock returns. *Journal of Financial Economics*, 1988.

[7] E. Fama. Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 1970.

[8] B. Freeman, G. Taylor, B. Gharabaghi, and J. Thé. Forecasting air quality time series using deep learning. *Journal of the Air & Waste Management Association*, 1995.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing System*, 2014.

[10] L. Grilli and D. Santoro. Boltzmann entropy in cryptocurrencies: A statistical ensemble based approach. *MPRA Munich Personal RePEc Archive*, 2020.

[11] L. Grilli and D. Santoro. Generative adversarial network for market hourly discrimination. *MPRA Munich Personal RePEc Archive*, 2020.

[12] Perasan M. H. and Timmermann A. Predictability of stock returns: Robustness and economic significance. *The Journal of Finance*, 1995.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[14] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino. Generative adversarial network-based postfilter for statistical parametric speech synthesis. *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2017.

[15] Y. Kara, M. Kara, and O. Kaan. Expert systems with applications predicting direction of stock prce index movement using artificial neural networks and support vector machines. *Expert Syst. Appl.*, 2011.

[16] S. F. LeRot. Efficient capital markets and martingales. *Journal of Economic Literature*, 1984.

[17] S. Liu, C. Zhang, and J. Ma. Cnn-lstm neural network model for quantitative strategy analysis in stock markets. *Neural Information Processing*, 2017.

[18] Y. Mäkinen, J. Kanniainen, M. Gabbouj, and A. Iosifidis. Forecasting jummp arrivals in stock prices: new attention-based network architecture using limit order book data. *Quantitative Finance*, 2019.

[19] M. Mittelmayer and G. F. Knolmayer. Text mining systems for market response to news: A survey. 2006.

[20] A. N. Refenes, M. Azema-Barac, and S. A. Karoussos. Currency exchange rate forecasting by error backpropagation. *Proceedings of the Twenty-Fifth Annual Hawaii International Conference on System Sciences*, 1992.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representation by backpropagation errors. *Nature*, 1986.

[22] R. Sharda and R. B. Patil. Connectionist approach to time series prediction: An empirical test. *Journal of Intelligent Manufacturing*, 1992.

[23] J. Sirignano. Deep learning for limit order books. *Quantitative Finance*, 2018.

[24] Y. Sun, L. Ximing, P. Cong, L. Li, and Z. Zhao. Digital radiography image denoising using a generative adversarial network. *Journal of X-Ray Science and Technology*, 2018.

[25] V. Vapnik. Statistical learning theory. *Wiley*, 1998.

[26] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer. Quant gans: deep generation of financial time series. *Quantitative Finance*, 2020.

[27] Campbell J. Y. and Shiller R. J. Stock prices, earnings, and expected dividends. *The Journal of Finance*, 1988.

[28] G. P. Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing 50*, 2003.

[29] L. Zhang, C. Aggarwal, and G. J. Qi. Stock price prediction via discovering multifrequency trading patterns. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.