



Forecasting financial time series with Boltzmann entropy through neural networks

Luca Grilli¹ · Domenico Santoro²

Received: 15 April 2022 / Accepted: 16 August 2022
© The Author(s) 2022

Abstract

Neural networks have recently been established as state-of-the-art in forecasting financial time series. However, many studies show how one architecture, the *Long-Short Term Memory*, is the most widespread in financial sectors due to its high performance over time series. Considering some stocks traded in financial markets and a crypto ticker, this paper tries to study the effectiveness of the Boltzmann entropy as a financial indicator to improve forecasting, comparing it with financial analysts' most commonly used indicators. The results show how Boltzmann's entropy, born from an Agent-Based Model, is an efficient indicator that can also be applied to stocks and cryptocurrencies alone and in combination with some classic indicators. This critical fact allows obtaining good results in prediction ability using Network architecture that is not excessively complex.

Keywords Neural networks · Price forecasting · LSTM · Boltzmann entropy · Financial markets · Cryptocurrency

JEL Classification C45 · C63 · C88 · G17

1 Introduction

Attention to the dynamics of financial markets and forecast of stock prices has always prompted researchers to develop - and focus - on different-type methodologies. Through the increasingly strong use of neural networks, it has been possible

Luca Grilli and Domenico Santoro contributed equally to this work.

✉ Luca Grilli
luca.grilli@unifg.it

Domenico Santoro
domenico.santoro@uniba.it

¹ Department of Economics, Management and Territory, University of Foggia, Foggia, Italy

² Department of Economics and Finance, University of Bari Aldo Moro, Bari, Italy

to improve the previous regressive models used in the past. These network-based models' characteristics lie in the choice of variables, the so-called features, which can be obtained directly from the markets. Using many features to make predictions is unnecessary, but the main task is to select the most appropriate ones. Among the most used features to forecast prices in the financial markets, we count those relating to the prices recorded at different moments in time and some financial indicators (e.g., MACD, RSI).

In this paper, we want to demonstrate that Boltzmann's entropy is a reliable indicator for forecasting using a Long Short Term Memory (LSTM) architecture. This indicator, developed by Grilli and Santoro (2021), considers an *Agent-Based Model* (ABM) in which, in a specific phase space, the particles are replaced by N economic subjects (agents) and where the movement of these economic agents are proxied by the entropy. Thus, in this way, it is possible to determine the position of the agents—represented by the ability to sell and buy a certain quantity—only through the price—using the Boltzmann formula. The main difference between the previous work and this one concerns the theoretical aspect. While in Grilli and Santoro (2021) the phase space and the possible link between statistical mechanics and Agent-Based Models have been defined (the theoretical background); in this paper, we consider Boltzmann's entropy as a financial indicator (calculated based on what was previously described), whose importance will be studied as a feature to improve price prediction. In particular, we forecast through neural networks and explore the significance of features through factor analysis. Furthermore, this paper considers the case of stocks and cryptocurrencies (Bitcoin), verifying how the Boltzmann entropy indicator can also be applied to the stock market.

1.1 Paper structure

Paper structure is the following: in the following subsection, the most relevant literature is presented; in Sect. 2, we introduce neural networks and the particular structure of the LSTM unit; Sect. 3 introduces ABMs and their applications to the economic/financial world, describing the model from which the Boltzmann entropy was extracted and how this synthetic value can be used as a feature in price prediction; Sect. 4 presents the numerical application of Entropy to some stocks and crypto, determining its importance also through factor analysis. Finally, in Sect. 5 some conclusions are drawn.

1.2 Literature review

The literature on time series bases its assumptions on the *random walk hypothesis*, a concept introduced by Bachelier (1900) in 1900, and its evolution of Cootner (1964) that indicated how the stock price movement could be approximated based on the Brownian motion. Traditionally, a most common practice was to focus on logarithmic returns, bringing the advantage of linking statistical analysis with financial theory. Fama (1970) introduced in his *Efficient Market Hypothesis (EMH)* theory the idea that historical prices are factored into the current prices

of a given market, then deploying these historical data in any analysis would be less valuable (if not completely useless) in making predictions about future prices. However, LeRoy (1989) showed that more concentration on yields was unjustified, defining the stock markets as inefficient. From an econometric perspective, Box and Jenkins (1976) introduced power transformations to statistical models and applied them to a time series. Specifically, they suggested using power transformation to obtain an adequate *Autoregressive Moving Average* (ARMA) model. Several evolutions have followed this pattern, e.g., *Autoregressive Integrated Moving Average* (ARIMA) and *seasonal autoregressive integrated moving average* (SARIMA). In combination with these models, the volatility of time series can be modeled using *AutoRegressive Conditional Heteroskedasticity* (ARCH) and *Generalized ARCH* (GARCH) model, as in the case of Wu (2021), who studied the in-sample coefficient estimation on the crypto market, or Borland (2016), who studied anomalous statistical features of time-series and review models of the price dynamics.

Thanks to the development of *artificial neural networks* (ANNs) and their applicability to non-linear modeling (Zhang 2003), there has been a strong interest in applying these methods to time series prediction in the last few years. For example, Refenes et al. (1992) proposed using a neural network system for forecasting exchange rates via a feedforward network. Sharda and Patil (1992) compared the prediction made via neural networks and the Box–Jenkins model, which verified that neural networks perform better than the forecast for time series with a long memory. In contrast, the networks outperform the prevision for time series with a short memory. Finally, Dixon (2018) assesses the impact of supervised learning on high-frequency trading strategies. The evolution of *Machine Learning* (ML) and *Deep Learning* (DL) techniques has introduced many advantages. As for ML techniques, a great innovation was introduced with the development of Vapnik (1998)'s *Support Vector Machine* (SVM) model, which solved the problem of pattern classification. Its use was immediately extended to regression, with the consequent application to time series forecasting (Adhikari and Agrawal, 2013). Mittelmayer and Knolmayer (2006) compared different text mining techniques for extracting market response to improve prediction or Kara et al. (2011), which directly uses the SVM for stock price prediction. As for the DL techniques, increasingly complex architectures are being used. For example, Liu et al. (2017) use a CNN-LSTM for strategic analysis in financial markets. Zhang et al. (2017) use an SFM to predict stock prices by extracting different types of patterns. Chen and Ge (2019) use an LSTM-based architecture to predict stock price movement. Mäkinen et al. (2019) propose an LSTM architecture for predicting return jump arrivals one minute in equity markets. Alternatively, Sirignano (2019) builds a “spatial neural network” to use more effective information from the limit order book. 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. (2014)] used for speech synthesis (Kaneko et al. 2017) or the denoising of images (Sun et al. 2018) and readjusted as in the case of Wiese et al. (2020), who build a Quant GANs highlighting the characteristics of the generated data.

2 Neural networks and LSTM units

An artificial neural network (ANN) represents a computational model that takes inspiration from the human brain. Like the human organ, the ANN is composed of neurons [artificial neurons, (McCulloch and Pitts 1943)] that perform computations within them. This fundamental unit performs a combination of functions which, in matrix form, can be defined as:

$$\hat{y} = g(\omega_0 + \mathbf{X}^T \mathbf{W}) \quad (1)$$

where \hat{y} represents the output, g the activation function, ω_0 the bias term, and \mathbf{X} and \mathbf{W} the input vectors and weights, respectively. The most significant advantage of neural networks is the ability to learn: to solve a specific learning problem, which generally represents a problem of adapting the network parameters to data, a set of rules called a learning algorithm is used. There are three learning algorithms: *supervised learning*, in which a domain expert labels the data; *unsupervised learning*, in which the network extracts patterns autonomously from the data and *semisupervised learning*, a combination of the above with a small amount of labeled data.

The first neural network developed is the Feedforward Neural Network (FNN). The connections between the nodes occur in sequence from the previous to the next according to a single direction [for example, this type of network includes the *perceptron*, also called universal approximator (Rosenblatt 1958)]. Against this, a class of neural networks typically used to process data sequences (mainly performing thanks to their memory effect) are the Recurrent Neural Networks (RNNs). These are essentially neural networks with feedback connections in which, given the considerable flow of information generated, training requires considering different time instants (the so-called *unfolding in time*). In contrast to the FNN, in this type of network, the new state h_t is determined as:

$$h_t = f_W(h_{t-1}, x_t) \quad (2)$$

where f_W is the function parameterized by the weights and x_t is the input vector at time step t . Generally, to train a neural network through which the gradient of the overall loss function $J(W)$ is computed, the *Backpropagation* algorithm by Rumelhart et al. (1986) is used. In RNN, this algorithm uses a particular version: *Backpropagation Through Time* (BPTT). The fundamental difference is that gradients are computed for each time step in this algorithm version. The main problem is that the network is exposed to the problem of exploding or (in an opposite way) vanishing the gradient. The latter, in particular better known, occurs because the update of the weights in the neural network is proportional to the partial derivatives of the loss function with respect to the current weight. In this way, the gradient could be so extremely small as to prevent the updating of the weights and block the training of the network (it affects both the FNN and the RNN). To prevent this problem, we can use specific units to control information transmission, such as the *Long-Short Term Memory* (LSTM). These units—introduced by Hochreiter and Schmidhuber (1997)—are most used since they have a long-memory effect, thanks to the ability to receive inputs and outputs from the previous level. Each LSTM unit comprises

an *input gate*, an *output gate*, and a *forget gate* that allows to check the information (forgetting the irrelevant ones) and transmit it to the next unit. The hidden state S_t can be described as:

$$\begin{aligned}
 f_t &= \sigma(X_t U^f + S_{t-1} W^f + b_f), \\
 i_t &= \sigma(X_t U^i + S_{t-1} W^i + b_i), \\
 \tilde{C}_t &= \tanh(X_t U^c + S_{t-1} W^c + b_c), \\
 C_t &= C_{t-1} \odot f_t + i_t \odot \tilde{C}_t, \\
 o_t &= \sigma(X_t U^o + S_{t-1} W^o + b_o), \\
 S_t &= o_t \odot \tanh(C_t).
 \end{aligned} \tag{3}$$

based on the input X_t and the previous hidden state S_{t-1} , where \odot represents the Hadamard product, σ is the *sigmoid* activation function, f the forget gate, i the identify gate, o the output gate, C the cell state, U the input weight matrix, W the recurrent weight matrix, and b the bias. The LSTM is among the most suitable to combat the problem of vanishing gradients (Bao et al. 2017): in fact, the gradient contains the forget gate's vector of activation, which, combined with the additive property of the unit state gradients, allows the network to better determine the best parameters for updating at each time step.

This unit represents one of the most used architectures for time series forecasting. For example, there are several advantages to using LSTM networks compared to ARIMA models. For example, ARIMA models focus on linear relationships in the time series, while LSTM networks capture non-linearity, or using neural networks reduces error rates. Furthermore, as shown by Siami-Namini et al. (2019), the performance of an LSTM network is much more accurate; moreover, this architecture allows to overcome the non-stationarity of prices (Preeti et al. 2019).

3 Methodology

The Boltzmann entropy feature arises from considering the stock market as an Agent-Based Model (ABM). The theory of agent-based simulation has been developed since the 1960s, allowing us to study how the application of specific conditions affects a small number of agents (Hamill and Gilbert 2015) (typically heterogeneous). Thanks to the development of processing systems, ABM has evolved into a program that generates an artificial world made up of agents. As a result, studying their interactions through the generated patterns is possible (Squazzoni 2010; Epstein and Axtell 1996). Agents can be any entity, from people to companies to animals: for this reason, ABMs are a fundamental tool in the social sciences for evaluating policy, performance, and perception. When these studies represent economic agents, we refer to *Agent-Based Computational Economics* [ACE, Tesfatsion and Judd (2006)], with which decentralized markets are analyzed under experimental conditions. The main research topics concern (Tesfatsion 2001; Tesfatsion 2002):

- Evolution of behavioral norms, defined as the measure of different behavior than usual seen by other agents (Axelrod 1997). These rules highlight the cooperation between different agents;
- Modeling market processes, to define the self-organization rules typical of different markets;
- Forming networks between agents, through the analysis of strategic interactions between agents to identify their neighbors and the type of relationship between them (from which it is possible to generate graphs completely connected, locally connected, locally disconnected, and so on);
- Design of agents, not only about their heterogeneity but also about the exchanges they can have with other agents, the number of relationships they can have with them, their permanence in a market, and any other condition that can most likely reproduce the system to be analyzed;
- Parallel experimentation, related to the possibility of simulating the behavior of different agents simultaneously, unlike in many current computational systems.

A classic example of ACE is the microeconomic one of supply and demand for a single homogeneous good in the market, in which, through the computation, it is possible to modify some conditions such as non-heterogeneous costs, presence of transaction costs, asymmetric information, and explore the changes to the curves and at their point of intersection (Cliff and Bruten 1998).

ACE theory is used not only for economic models but also to simulate financial markets and analyze patterns within them, despite the difficulties in simulating the complex reality of markets (absence of rational choices and market efficiency). For instance, LeBaron (2000) studied the Santa Fe artificial market, which combines the traditional structure of a financial market with learning using a classifier-based system. Izumi and Ueda (2001) studied the foreign market by proposing an agent-based approach based on behavioral rules. Howitt and Clower (2000) investigated the role of particular agents (*trade specialist*) in a decentralized market model in supporting currency emergencies. Finally, Chen and Yeh (2001) built an ACE framework to analyze the interactions of an artificial stock market, measuring success based on the predictive ability of agents.

3.1 Boltzmann entropy model

In Grilli and Santoro (2021), we defined an ABM in which the particles are replaced by N economic subjects (agents) who intend to trade in cryptocurrencies. In this model, it is possible to determine the movement of economic subjects in a particular "phase space" and whose entropy provides a proxy for this movement. Moreover, we can also fully describe an economic agent in our phase space by two variables, which we can identify as $\{x_i, y_i\}$ where x_i and y_i indicate the ability to buy and sell a certain quantity of cryptocurrencies (both expressed in monetary terms). Finally, let us consider that these two variables are summarized in the cryptocurrency's last prices (*closing price*); in this sense, the latest prices allow us to understand whether the ability to buy or sell prevailed compared to the previous session. In particular,

we have not identified a function such that a change of x_i and y_i leads to a change in price; however, the economic subjects move concerning the quantity they have purchased/sold. In this case, we have a system made up of financial instruments to make some similar assumptions. In particular, we can assume that the reference system includes N agents who intend to trade in stocks. We take a specific time window (5 days, corresponding to a trading week) and group the closing price series based on this window every 5 days. Since each group has a maximum and a minimum price (a gap), we calculate the difference in terms of necessary steps to pass from one to the other, obtaining a particular value of gap G . This assumption is based on the idea that the distance between maximum and minimum is a measure of the dispersion of agents in our phase space. Using combinatorial analysis, we can compute the value used for grouping to determine the "volume" occupied by the disposition of the agents; therefore: $\Gamma = G^5$. The main difference, in this case, is that in calculating the gaps and consequently the entropy value, we still consider 5-days groups. However, these are calculated "dynamically": starting from the last recorded price (indicated with t), we calculated the dynamic gap using the prices of the previous 4 days, creating a range of the type $(t - 4, \dots, t)$. With this method, we obtain several gaps equal to the number of observations in the dataset. 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:

$$S = \kappa_B \ln \Gamma \quad (4)$$

where $\kappa_B \sim 1.3806 \times 10^{-23}$ is the Boltzmann constant, and finally, "rationalizing" multiplying by 10^{23} to make the value more readable from a graphic point of view (e.g., to get 46.6 instead of 0.0496).

Furthermore, we extend the reference market by considering the stocks in this case. The cryptocurrency market is a market open 24 h a day. Therefore, it is possible to carry out transactions at any time. This makes it more idealizable through a physical system, as the particles (agents) do not have the constraint of respecting schedules to move. Instead, the stock market is a market subject to closing times (e.g., the Italian stock market MTA or the Nasdaq), where the previous day's closing and the next day's opening prices often do not match due to events that occurred in the night. However, despite this apparent constraint that "limits" the movement of agents at certain times (corresponding to some volumes of the phase space), we test the Boltzmann entropy indicator on both markets to verify its ability to improve price prediction.

The ABMs allow, especially in recent times thanks to the high computational capacity of the machines, to carry out simulative and forecasting analyses ever higher, helping in the definition of strategies/policies. However, there are several problems affecting these models. First, as introduced by Axtell and Farmer (2022), there is the issue of parallel execution: generally, simulations using ABM occur in a single thread, whereby each agent acts once per machine cycle. However, in reality, the agents carry out actions asynchronously and, above all, simultaneously. ABM algorithms are being developed to solve this problem, especially in recent times,

allowing multi-threaded execution¹. Another problem concerns the level of representation of the economic system through the ABM. In fact, due to the high complexity, it is impossible to fully represent the variables that influence an agent nor all the relationships that could be generated. For this reason, an ABM represents only a restricted portion of the economic system (so-called *nanoeconomics*). Again, the *curse of dimensionality* (Bellman 1957) is another problem linked to the impossibility of fully representing the economic system in which agents can move. This occurs when the size of the system parameter space increases, and the data representation creates sparsity, resulting in worse analyses. Finally, the problem of the burn-in phase is the need for an agent-based model to carry out a series (often elevated) of simulations before entering total capacity and representing existing relationships. This phase adds to the computational capacity and increases the time required to obtain results.

The main advantage of using the Boltzmann entropy model is the possibility of summarizing agents' behavior in a single variable, similar to a financial indicator. We are not interested in understanding how agents can move within the phase space but only in observing, after having performed a movement (a transaction), how their position has changed, summarized by a single indicator. Thanks to the formalization in the phase space, we avoid some of the previous problems typical of ABMs, such as the curse of dimensionality or the burn-in phase. Similar approaches are present in Fraunholz et al. (2021), where the authors use an ANN to identify the endogenous relationships between some variables of their ABM model for price prediction in the energy market. Furthermore, Ghosh and Raju Chinthalapati (2014) developed an ABM model by linking the functioning of the economic system to a physical system through a minority game, considering the stock market and the Foreign Exchange Market (FOREX). In this way, based on whether or not the agents have completed a transaction and based on the construction of these, they can make price predictions in the various markets using Genetic Algorithms (GA). Zhang (2013) uses an ABM to study the interactions between agents in the markets, particularly highlighting some mechanisms of the stock market and exploiting them to predict aggregate behaviors (specifically return signs underlying the prediction of strategies). Arthur et al. (1997) propose a theory of asset pricing based on heterogeneous agents, considering the ABM market of Santa Fe (LeBaron 2000), highlighting how these agents modify their expectations according to the transactions carried out. Shi et al. (2019) an ABM representative of a market with two types of agents (investors and speculators). The price is predicted based on the expectations of these agents considering external information (the so-called jump process). Finally, Rekik et al. (2014) model the financial market as a complex system characterized by the interaction of agents, developing an artificial market to verify the dynamics that lead to the price prediction based on the exchanges of 3 types of agents.

¹ For example, in the case of the Zero Intelligence agents, the multi-threaded version of the Bristol Stock Exchange [TBSE, (Rollins and Cliff 2020)]

To show the behavior of the Boltzmann entropy-based indicator in the prediction phase, we can compare its performance with that of some of the leading financial indicators used by analysts, which are:

- *MACD (Moving Average Convergence/Divergence)* is based on two moving averages' convergence and divergence. The first at 12 periods and the second at 26. In particular, EMA_{12} represents the 12-days *Exponential Moving Average* of closing prices while EMA_{26} represents the 26-days *Exponential Moving Average*. So the MACD indicator is determined as follows:

$$MACD = EMA_{12} - EMA_{26} \quad (5)$$

- *SI (Stochastics Index)* studies price fluctuations and provides market entry and exit signals. For example, considering X as the last closing price, H_{14} as the highest price of the 14 previous days, and L_{14} as the lowest price of the 14 previous days, the oscillator *SI* is calculated as:

$$SI = \frac{X - L_{14}}{H_{14} - L_{14}} \times 100 \quad (6)$$

- *RSI (Relative Strength Index)* is used to identify the oversold and overbought areas, highlighting the ideal timing to enter and exit the market. Considering U as the average of the upward closing differences over a certain period (e.g., 14 days) and D is the average of the absolute value of the downward closing differences over a certain period, the RSI is calculated as:

$$RSI = 100 - \frac{100}{1 + \frac{EMA_{14}(U)}{EMA_{14}(D)}} \quad (7)$$

3.2 Setting up the machine

Through this type of architecture, we want to demonstrate that the entropy indicator calculated in this way has a predictive capacity at least equal to the indicators most used in technical analysis and, in addition to these, how the predictive ability of the features varies overall. With *Google Colab* and given the simplicity of the data, we have set the structure of the network with only 1 input layer with several neurons from 7 to 9 (according to the general 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, based on the work of Ketsetsis et al. (2021). The remaining hyperparameters, which control the learning process, have been tuned using the state-of-the-art values in the literature and are shown in Table 1. We will consider the Root-Mean Square Error (RMSE) to highlight the results obtained. The dataset was divided into a training set (80%) and a test set (20%).

Table 1 Values of hyperparameters for LSTM network

Parameters	Values
Layers	2
N. of neurons	{7, 8, 9}
Activ. func.	<i>sigmoid</i>
Learning rate (η)	0.0005
Optimizer	Adam(η , 0.9, 0.999)
Loss	Mean square error
Batch size	32
Epochs	300
Time step	3

3.3 Dataset

The empirical analysis was carried out on the closing prices of three widespread stocks² (therefore having a very high number of stocks in circulation, which allows falling within the assumptions of the entropy model) and the last price of a classical cryptocurrency, the Bitcoin price (referred to USD):

- Apple Inc. (AAPL listed on NASDAQ) with a tick size of 0.01;
- Tesla Inc. (TSLA listed on NASDAQ) with a tick size of 0.01;
- Amazon Inc. (AMZN listed on NASDAQ) with a tick size of 0.01;
- Bitcoin (BTC-USD CoinMarketCap Exchange) with a tick size of 0.001.

Stocks' prices are considered with a daily time frame from 02/01/2011³ to 12/31/2019. In contrast, Bitcoin's price is considered from 09/01/2015 to 12/31/2019 (the difference is that for cryptocurrencies, 365 days are recorded while for stocks, 250, to balance the number of observations in the dataset). The dataset consists of several columns (features), each of which will be indicated with the first letter of the column; these features are: Open (O), High (H), Low (L), Adj Close/Last (C), Volume (V), MACD (M), Stochastic (SI), RSI (R), Entropy (E). In Table 2 there is a representation of the dataset used in the analysis with all features.

The most important feature to predict is the closing price. Furthermore, it is to be specified that the different instruments record different price levels. So, to highlight the closing price differences between the different datasets, the central statistics are shown in Table 3, such as number ("No." column), mean, standard deviation, skewness, kurtosis, minimum and maximum.

In particular, Bitcoin recorded the most substantial price variation after the extreme speculative bubbles created in 2015 and 2017. These differences, often very pronounced, are essential because they can lead to different levels of RMSE.

² Source: finance.yahoo.com.

³ All dates are in US format.

Table 2 Sample of the Apple dataset (values from beginning and ending dates)

Date	Open	High	Low	Adj close	Volume	MACD	SI	RSI	E
2011-02-01	12.18929	12.34464	12.17786	10.5803	426633200	0.006359	84.14532	52.71597	6.953876
2011-02-02	12.30179	12.33036	12.26964	10.55853	258955200	0.011935	80.88183	49.91702	6.953876
2011-02-03	12.27857	12.29429	12.09107	10.53154	393797600	0.0136	76.83798	48.13209	6.953876
2011-02-04	12.27286	12.38214	12.26821	10.62538	321840400	0.019879	100	48.35597	5.448233
2011-02-07	12.42464	12.61607	12.41571	10.79035	485021600	0.03361	100	59.71958	6.564281
2011-02-08	12.63143	12.69714	12.57679	10.89216	381040800	0.048188	100	63.79891	8.854192
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2019-12-26	71.205	72.495	71.175	71.5209	93121200	0.25615	100	85.35301	22.44524
2019-12-27	72.78	73.4925	72.03	71.49376	146266000	0.29672	99.52138	82.4768	22.44524
2019-12-30	72.365	73.1725	71.305	71.91808	144114400	0.330453	100	95.02158	20.16067

Table 3 Main statistics of the time series used

Title	No.	Mean	St. Dev	Skewness	Kurtosis	Min.	Max.
AAPL	2243	27.66	13.49	0.79	-0.15	9.62	71.61
TSLA	2243	38.74	22.04	-0.34	-1.10	4.36	86.18
AMZN	2243	37.02	29.02	0.89	-0.68	8.04	101.97
BTC	1583	4530.80	3983.18	0.74	-0.08	227.08	19497.40

4 Numerical results

To test the effectiveness of the different indicators, we will analyze the different features first individually and then combine the different features in other datasets to see how the values of the *RMSE* change (the features being forecast will always remain "Adj Close"). We try to show that entropy, in some cases, can be an indicator that, due to its construction, significantly improves the forecast.

As shown in Table 4, obtained by training the previously defined network architecture with the different combinations of datasets, the *RMSE* values differ according

Table 4 RMSE (lower the better) for the different datasets

	AAPL	TSLA	AMZN	BTC
OHLV	11.554	6.901	203.127	1299.767
OHLM	6.971	4.744	169.931	1082.248
OHL SI	11.032	4.731	1096.145	1664.770
OHLR	5.678	2.689	184.241	1195.360
OHLE	3.057	4.162	348.731	532.271
OHLVM	4.831	3.176	234.345	644.492
OHLV SI	3.346	13.266	103.814	1256.008
OHLVR	3.758	2.884	104.190	1049.686
OHLVE	13.462	3.241	218.341	993.780
OHLVMR	2.659	3.190	111.695	846.491
OHLVRE	2.801	2.893	137.989	796.520
OHLVME	3.291	3.306	183.588	565.063
OHLV SE	6.215	2.826	128.667	1772.583
OHLVMRE	3.400	4.376	47.113	1117.730
μ_{RMSE}	5.86	4.45	233.70	1058.34
σ_{RMSE}	3.619	2.781	258.81	372.84

to the type of instruments since the prices inside move on different levels (as shown by the different μ and σ values of the RMSE). Therefore, this indicator measures the goodness of the forecast made on a test set of over 300 values (20% of the initial dataset) for each dataset type. The results show that in the first combination of features (the classic OHLC without Volume) with the addition of entropy, this is a good indicator for prediction, especially for Bitcoins that demonstrate their high predictive capacity and Apple. Figure 1 shows the predictions on the part of the test set of the different datasets.

By adding more features, the predictive accuracy of the model increases. Neural Networks can perceive the relationships between the features, particularly from the forecast improvement with the combined use of the Volume and RSI, or Volume and MACD. The combination with entropy gives an excellent result (OHLVRE and OHLVME case), while these features combined worsen the *RMSE*. This effect can be due to the redundancy of information created by combining features. For example, in the case of RSI - shown in Fig. 2-, the entropy determined 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 this characteristic, which we hypothesized as a tool for making a prediction, makes the neural network able to improve the forecast. We can also assume that the reduction of *RMSE* 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 that could somehow condition the network itself.

The reason for this result is traceable in the construction of the entropy indicator, which being constructed "dynamically" takes into account a certain

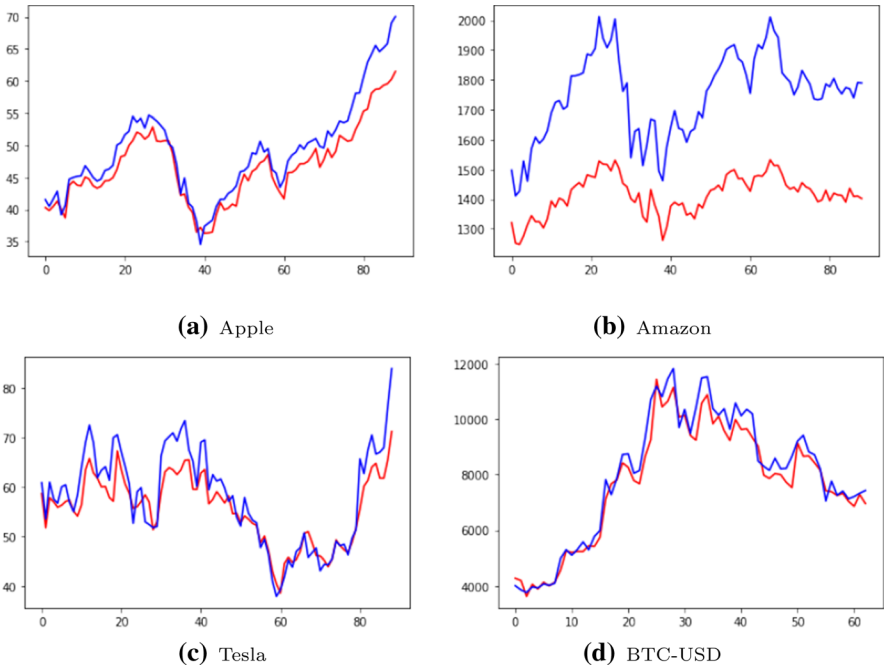


Fig. 1 Real price (blue) and predicted price (red) on part of the OHLE test set (colour figure online)

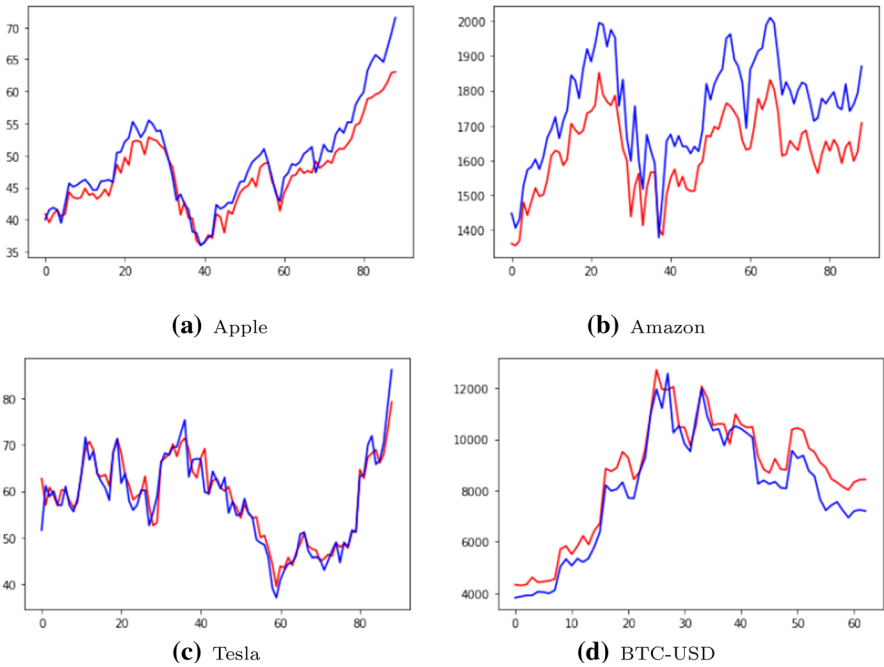


Fig. 2 Real price (blue) and predicted price (red) on part of the OHLVRE test set (colour figure online)

Table 5 Communalities of features from factor analysis (all values are %)

Title	Open	High	Low	Volume	MACD	SI	RSI	Entropy
AAPL	99.8	99.8	99.8	60.9	55.1	72.4	94.8	63.6
TSLA	99.7	99.8	99.8	86.6	73.6	99.5	80.6	77.3
AMZN	99.8	99.8	99.8	68.7	50.6	71.6	84.8	73.1
BTC	99.6	99.7	99.7	54.9	32.1	95.1	91.3	70.2

amount of information (which represents the position of economic agents concerning buying or selling), based on which it is possible to understand when there will be a movement of agents. However, when entropy is used together with the other indicators, this significant presence of captured information generates information redundancy. In this sense, multiple points could be where all three/four indicators have captured the same type of information. However, the neural network does not capture this (in particular since the indicators, despite the same type of information, could have opposite movements), producing a higher *RMSE* than the single indicator.

4.1 Factor analysis

Through the LSTM architecture, we could highlight how using the Boltzmann entropy feature can improve price prediction. However, to quantify the importance of this feature compared to the others used, we use factor analysis. Through Google Colab and the Factor package, we perform a 4-factor analysis. This dimensionality reduction technique used to reduce the number of features has the advantage of reporting the variability explained by each variable. In particular, reporting the *communalities*, we determine the portion of each variable's variance explained by the factors. In this way, the variables with a higher value are the most represented by the factors and, therefore, the most useful. Using an orthogonal varimax rotation, the communalities are shown in Table 5.

After removing the “*Adj Close*” feature for each dataset, we consider all the remaining ones so that we can compare them with the importance of the entropy feature. The first three features (Open, High and Low) are closely linked since the prices recorded in these variables are very similar (which is why they are so important). On the other hand, as often highlighted by analysts, Volume is not a fundamental feature, so much so that, in this case, it has lower communalities than the indicators. Finally, among the constructed indicators (MACD, SI, RSI, and Entropy), globally, the most important are SI and RSI. In some cases, our Entropy indicator obtained a higher value (e.g., in the case of AMZN, Entropy > SI). While in comparison with MACD, also Entropy got higher values for all the instruments considered, highlighting the importance that this feature derived from an ABM can have in the predictive process.

5 Conclusions

This paper shows how the dynamically determined Boltzmann entropy for stocks and cryptocurrency can be an indicator on a par with those most commonly used in financial data analysis. We tested this indicator alone and in combination with other features, both in the case of stocks and cryptocurrency, using a neural network architecture with LSTM units to make the price prediction and evaluate the importance of this feature through factor analysis. The results show that entropy is a good indicator already at the level of relatively simple datasets (think of the possibility of using a dataset with the classic OHL features). In this sense, we can believe that the representation through an Agent-Based Model is functional in determining the entropy indicator and effective for improving the predictive accuracy. Future works' objective will be to exploit the Entropy indicator as a tool to verify the possible presence of *cyclicality* in the movement of economic agents.

Funding Open access funding provided by Università di Foggia within the CRUI-CARE Agreement. The authors declare that received no financial support for the research, authorship, and/or publication of this article.

Declarations

Conflict of interest The authors certify that there is no actual or potential conflict of interest in relation to this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adhikari R, Agrawal RK (2013) An introductory study in time series modeling and forecasting. LAP LAMBERT Academic Publishing, Sunnyvale, p 76
- Arthur WB, Holland JH, LeBaron B, et al (1997) Asset pricing under endogenous expectations in an artificial stock market. In: The economy as an evolving complex system II, ISBN: 9780429496639
- Axelrod R (1997) The complexity of cooperation: agent-based models of conflict and cooperation. Princeton University Press, Princeton
- Axtell RL, Farmer JD (2022) Agent-based modeling in economics and finance: past, present, and future. INET Oxford Working Paper (2022-10)
- Bachelier L (1900) Theorie de la speculation. PhD Thesis
- 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>
- Bellman R (1957) Dynamic programming. Princeton University Press, Princeton

- Borland L (2016) Exploring the dynamics of financial markets: from stock prices to strategy returns. *Chaos Solitons Fractals* 88:59–74. <https://doi.org/10.1016/j.chaos.2016.03.014>
- Box GEP, Jenkins GM (1976) *Time series analysis forecasting and control*. Holden-Day, San Francisco
- Chen S, Ge L (2019) Exploring the attention mechanism in lstm-based Hong Kong stock price movement prediction. *Quantit Financ* 19(9):1507–1515. <https://doi.org/10.1080/14697688.2019.1622287>
- Chen SH, Yeh CH (2001) Evolving traders and the business school with genetic programming: a new architecture of the agent-based artificial stock market. *J Econ Dyn Control* 25(3):363–393. [https://doi.org/10.1016/S0165-1889\(00\)00030-0](https://doi.org/10.1016/S0165-1889(00)00030-0)
- Cliff D, Bruten J (1998) Less than human: simple adaptive trading agents for CDA markets. *IFAC Proceedings Volumes* 31(16):117–122. [https://doi.org/10.1016/S1474-6670\(17\)40468-X](https://doi.org/10.1016/S1474-6670(17)40468-X), iFAC Symposium on Computation in Economics, Finance and Engineering: Economic Systems, Cambridge, UK, 29 June - 1 July
- Cootner PH (1964) *The random character of stock market prices*. MIT Press, Cambridge
- Dixon M (2018) A high-frequency trade execution model for supervised learning. *High Frequency*. <https://doi.org/10.1002/hf2.10016>
- Epstein J, Axtell R (1996) *Growing artificial societies: social science from the bottom up*. MIT Press, Cambridge
- Fama E (1970) *The Journal of Finance* 25(2):383–417. <https://doi.org/10.2307/2325486>. papers and Proceedings of the Twenty-Eighth Annual Meeting of the American Finance Association New York, N.Y. December, 28–30, 1969
- Fraunholz C, Kraft E, Keles D et al (2021) Advanced price forecasting in agent-based electricity market simulation. *Appl Energy* 290(116):688. <https://doi.org/10.1016/j.apenergy.2021.116688>
- Ghosh P, Raju Chinthalapati VL (2014) Financial time series forecasting using agent based models in equity and FX markets. In: 2014 6th computer science and electronic engineering conference (CEEC), pp 97–102. <https://doi.org/10.1109/CEEC.2014.6958562>
- Goodfellow IJ, Pouget-Abadie J, Mirza M, et al (2014) Generative adversarial nets. In: Proceedings of the 27th international conference on neural information processing systems-Volume 2. MIT Press, Cambridge, MA, USA, NIPS'14, p 2672–2680
- Grilli L, Santoro D (2021) Cryptocurrencies markets and entropy: a statistical ensemble based approach. *Appl Math Sci* 15(7):297–320
- Hamill L, Gilbert N (2015) *Agent based modelling in economics*. Wiley, Hoboken
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Howitt P, Clower R (2000) The emergence of economic organization. *J Econ Behav Org* 41(1):55–84. [https://doi.org/10.1016/S0167-2681\(99\)00087-6](https://doi.org/10.1016/S0167-2681(99)00087-6)
- Izumi K, Ueda K (2001) Phase transition in a foreign exchange market-analysis based on an artificial market approach. *IEEE Trans Evol Comput* 5(5):456–470. <https://doi.org/10.1109/4235.956710>
- Kaneko T, Kameoka H, Hojo N, et al (2017) Generative adversarial network-based postfilter for statistical parametric speech synthesis. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP) pp 4910–4914. <https://doi.org/10.1109/ICASSP.2017.7953090>
- Kara Y, Boyacioglu MA, Baykan OK (2011) Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert Syst Appl* 38(5):5311–5319. <https://doi.org/10.1016/j.eswa.2010.10.027>
- Ketssetsis AP, Giannoutakis KM, Spanos G et al (2021) A comparative study of deep learning techniques for financial indices prediction. In: Maglogiannis I, Macintyre J, Iliadis L (eds) *Artificial intelligence applications and innovations*. Springer, Cham, pp 297–308
- LeBaron B (2000) Agent-based computational finance: suggested readings and early research. *J Econ Dyn Control* 24:679–702
- LeRoy SF (1989) Efficient capital markets and martingales. *J Econ Lit* 27(4):1583–1621
- Liu S, Zhang C, Ma J (2017) Cnn-lstm neural network model for quantitative strategy analysis in stock markets. In: Liu D, Xie S, Li Y et al (eds) *Neural information processing*. Springer, Cham, pp 198–206
- Mäkinen Y, Kannianen J, Gabbouj M et al (2019) Forecasting jump arrivals in stock prices: new attention-based network architecture using limit order book data. *Quant Financ* 19(12):2033–2050. <https://doi.org/10.1080/14697688.2019.1634277>
- McCullock WS, Pitts WH (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115

- Mittelmayr M, Knolmayer GF (2006) Text mining systems for market response to news: a survey. In: IADIS European Conference Data Mining 2007 (part of MCCSIS 2007) pp 164–169. ISBN: 978-972-8924-40-9
- Preeti, Bala R, Singh RP (2019) Financial and non-stationary time series forecasting using lstm recurrent neural network for short and long horizon. In: 2019 10th international conference on computing, communication and networking technologies (ICCCNT) pp 1–7. <https://doi.org/10.1109/ICCCNT45670.2019.8944624>
- Refenes A, Azema-Barac M, Karoussos S (1992) Currency exchange rate forecasting by error back-propagation. In: Proceedings of the twenty-fifth Hawaii international conference on system sciences iv:504–515 vol.4. <https://doi.org/10.1109/HICSS.1992.183441>
- Rekik YM, Hachicha W, Boujelbene Y (2014) Agent-based modeling and investors' behavior explanation of asset price dynamics on artificial financial markets. *Procedia Econ Financ* 13:30–46. [https://doi.org/10.1016/S2212-5671\(14\)00428-6](https://doi.org/10.1016/S2212-5671(14)00428-6)
- Rollins M, Cliff D (2020) Which trading agent is best? Using a threaded parallel simulation of a financial market changes the pecking-order. [arXiv:2009.06905](https://arxiv.org/abs/2009.06905)
- Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65:386
- Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representation by back-propagation errors. *Nature* 323:533–536. <https://doi.org/10.1038/323533a0>
- Sharda R, Patil RB (1992) Connectionist approach to time series prediction: An empirical test. *J Intell Manuf* 3:317–323. <https://doi.org/10.1007/BF01577272>
- Shi Y, Luo Q, Li H (2019) An agent-based model of a pricing process with power law, volatility clustering, and jumps. *Complexity*. <https://doi.org/10.1155/2019/3429412>
- Siami-Namini S, Tavakoli N, Namin AS (2019) A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. [arXiv:1911.09512v1](https://arxiv.org/abs/1911.09512v1)
- Sirignano JA (2019) Deep learning for limit order books. *Quant Financ* 19(4):549–570. <https://doi.org/10.1080/14697688.2018.1546053>
- Squazzoni F (2010) The impact of agent-based models in the social sciences after 15 years of incursions. *History Econ Ideas* 18(2):197–233
- Sun Y, Ximing L, Cong P et al (2018) Digital radiography image denoising using a generative adversarial network. *J Xray Sci Technol* 26(4):523–534. <https://doi.org/10.3233/XST-17356>
- Tesfatsion L (2001) Special issue on the agent-based modeling of evolutionary economic systems. *IEEE Trans Evolut Comput* 5(5):437
- Tesfatsion L (2002) Agent-based computational economics: growing economies from the bottom up. *Artif Life* 8(2):55–82. <https://doi.org/10.1162/106454602753694765>
- Tesfatsion L, Judd K (2006) Handbook of computational economics: agent-based computational economics. North Holland, iISBN: 978-0444512536
- Vapnik V (1998) Statistical learning theory. Wiley, Hoboken, p 768
- Wiese M, Knobloch R, Korn R et al (2020) Quant gans: deep generation of financial time series. *Quant Financ* 20(9):1419–1440. <https://doi.org/10.1080/14697688.2020.1730426>
- Wu C (2021) Window effect with markov-switching garch model in cryptocurrency market. *Chaos Solitons Fractals* 146(110):902. <https://doi.org/10.1016/j.chaos.2021.110902>
- Zhang G (2003) Time series forecasting using a hybrid arima and neural network model. *Neurocomputing* 50:159–175. [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0)
- Zhang L, Aggarwal C, Qi GJ (2017) Stock price prediction via discovering multi-frequency trading patterns. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA, KDD '17, p 2141–2149. <https://doi.org/10.1145/3097983.3098117>
- Zhang Q (2013) Disentangling Financial Markets and Social Networks: Models and Empirical Tests. PhD dissertation, ETH