Sentiment-Driven Price Prediction of the Bitcoin based on Statistical and Deep Learning Approaches

Giulia Serafini*†, Ping Yi*§, Qingquan Zhang¶, Marco Brambilla†, Jiayue Wang‖, Yiwei Hu** and Beibei Li††
* Shanghai Jiao Tong University, Shanghai, China
† Politecnico di Milano, Milan, Italy
¶ University of Illinois, Urbana Champaign, USA
‖ Beijing International Studies University, Beijing, China
** University of California, San Diego, USA
†† Carnegie Mellon University, Pittsburgh, USA
§ Email: yiping@sjtu.edu.cn

Abstract*—Nowadays, Bitcoin has become the most popular cryptocurrency, which gains the attention of investors and speculators alike. Asset pricing is a risky and challenging activity that enchants lots of shareholders. Indeed, the difficulty in making predictions lies in understanding the multiple factors that affect the Bitcoin price trend. Modeling the market behavior and thus, the sentiment in the Bitcoin ecosystem provides an insight into the predictions of the Bitcoin price. While there are significant studies that investigate the token economics based on the Bitcoin network, limited research has been performed to analyze the network sentiment on the overall Bitcoin price. In this paper, we investigate the predictive power of network sentiments and explore statistical and deep-learning methods to predict Bitcoin future price. In particular, we analyze financial and sentiment features extracted from economic and crowd-sourced data respectively, and we show how the sentiment is the most significant factor in predicting Bitcoin market stocks. Next, we compare two models used for Bitcoin time-series predictions: the Auto-Regressive Integrated Moving Average with eXogenous input (ARIMAX) and the Recurrent Neural Network (RNN). We demonstrate that both models achieve optimal results on new predictions, with a mean squared error lower than 0.14%, due to the inclusion of the studied sentiment feature. Besides, since the ARIMAX achieves better predictions than the RNN, we also prove that, with just a linear model, we may obtain outstanding market forecasts in the Bitcoin scenario.

***Index Terms*—Market Stock Prediction, Bitcoin (BTC), Auto-Regressive Integrated Moving Average with eXogenous input (ARIMAX), Recurrent Neural Network (RNN), Sentiment Analysis**

I. Introduction

Market prediction drives investors to identify a reasonable cause-effect relation between all the available current information and the future. However, the prediction gets complex in cryptocurrencies, whose prices depend on multiple factors that are still unknown, and making forecasts becomes more challenging than other commodities [1]. Cryptocurrencies are a new asset class receiving significant attention from the financial and investment community. They are digital currencies traded on special crypto-exchanges not accessible to the majority of traditional investors. In this respect, researchers all over the world have spent an enormous amount of effort to design approaches to predict this inefficient market.

The techniques to predict the market are divided into Statistical, Pattern Recognition, Machine Learning, Sentiment Analysis, and Hybrid method [2]. Statistics is the first and oldest approach adopted for data analysis. Among the statistical models, the Auto-Regressive Moving Average (ARMA) is an effective method for time-series predictions. Its generalization is the Auto-Regressive Integrated Moving Average (ARIMA), useful when the time-series has no constant mean and variance over the time and, therefore, falls into a condition of non-stationarity. Afterwards, Pattern Recognition is a visual approach, widely spread among traders. It consists of recognizing patterns and trends in stock data [3]. Machine Learning is another method used for predicting time-series data, which became more popular since the advent of deep learning tools. Moreover, the spread of computer-based recognition introduced by machine learning made pattern recognition theory acquire even greater importance [4]. Sentiment Analysis applied to crowd-sources is a different approach adopted for stock market analysis [5]. The basic idea relies on the wisdom of crowds principle, which states that the collective opinion of individuals is as reliable as the one of a single expert [6]. Thanks to sentiment analysis, news, current events, public releases, and social media became reliable for market forecasts [7]. Lastly, the Hybrid method is a combination of the approaches mentioned above.

In this paper, we propose a statistical and a machine learning methods to predict the next-day weighted value of BTC market stocks. The first method is a linear model called ARIMAX, the generalization of the ARIMA, which allows to make predictions starting from a multi-features input. On the contrary, the second model is non-linear: the Long-Short Term Memory (LSTM) RNN. For our models, we select a hybrid set of input features. Indeed, we adopt both financial features, like the *BTC weighted price*, and the *BTC volume*, and sentiment features gathered from Twitter, like the *BTC tweets sentiment* and the *BTC tweets volume*. In the end, we find that the best combination of features to predict BTC market stocks is composed of the sole *BTC weighted price* and *BTC tweets sentiment*. Moreover, we prove that both the ARIMAX and the RNN achieve optimal results on new predictions, with a mean

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

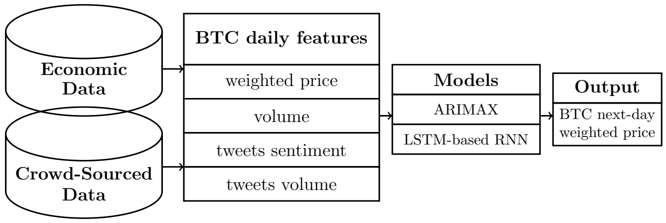

Fig. 1. Research Methodology.

squared error lower than 0.14%, due to the inclusion of the *sentiment* feature. However, we demonstrate that the ARIMAX performs better than the RNN, thereby showing that the most straightforward model, i.e., the linear one, is also the most accurate. In the end, we disprove the Jethin et al. statement [8], which asserts that, among the sentiment features, only the *BTC tweets volume*, and not the *sentiment*, is correlated to the BTC market behavior. To confute this statement, we show how both our models achieve better predictions by using the sole *BTC tweets sentiment* rather than the *BTC tweets volume*. More to the point, our contribution is represented in figure 1 and summarized as follows:

- we select proper economic and crowd-sourced data to design our dataset of BTC daily input features;
- we acquire financial features from the economic source, and we extract the sentiment features from the crowd-sourced one, by means of sentiment analysis;
- we build the ARIMAX and the LSTM-based RNN models to predict the BTC next-day weighted price;
- we find the best model and the most straightforward combination of input features for BTC market forecasts.

The rest of the paper is organized as follows. Section II introduces the related work, section III refers to data collection, data pre-processing and models description. Section IV presents the results obtained with our models, and in the end, section V provides the conclusion and future work. We would like to specify that, the experiments, models and results presented in section III and IV have been obtained by using Python.

II. Related Work

This section includes the previous contributions in the field of market forecasts, with a particular focus on the Bitcoin scenario.

In the past, the most well-known models adopted for market forecasting were the Auto-Regressive (AR) and Moving Average (MA), and the combination of both called ARMA and ARIMA. Thanks to [9], these models became popular for time-series predictions since 1970. Later, [10] used the ARIMA model to predict New York and Nigeria stock exchange indexes, proving that this model is not only adequate for short-term prediction but also can compete with the newest techniques for market forecasts. Recently, deep learning models like Convolutional Neural Networks (CNNs), RNNs, and LSTM-based RNNs, achieved significant results in predicting the price of National Stock Exchange (NSE) listed companies [11]. This study shows how, unlike the linear models, the non-linear ones are capable of underlining the dynamics in the data better and identifying patterns through a self-learning process. Moreover, [12] implemented a Feed-Forward Neural Network (FFNN) to predict the Nasdaq index, while [13] opted for a Convolutional Neural Network (CNN) for identifying common patterns of several market stocks. Among all the machine learning techniques mentioned above, the LSTM-based RNN is essential when the time-series depends on large amounts of data and a long-term market history [14]. The spread of social networks pointed out that there exists a correlation between market behavior and public opinion [5]. Based on this finding, [15] and [16] adopted Artificial Neural Network (ANN) and Support Vector Machine (SVM), respectively, to predict the Apple stock index by using the sentiments of the tweets as a valid input feature. Another new approach consists of predicting the market trends by using Generative Adversarial Networks (GANs), a new kind of Neural Networks discovered by [17] in 2014. For the first time, GANs have been implemented to predict market stocks by [18] and [19]. Moreover, there exist several studies that compare various techniques mentioned above. Indeed, [20], by analyzing the RNN and SVM market stock predictions, discovered that the RNN performs better in the presence of high volatility, while the SVM with low volatility. Finally, [21] showed how FFNN, RNN, LSTM, and CNN are more accurate than the ARIMA model to predict the NSE index. Among all the novel deep learning techniques, the LSTM-based RNN is the most popular for its outstanding performance in predicting the market behavior [22], [23]. Over the last few years, several studies carried out to better understand the cryptocurrencies market behavior, with a greater focus on Bitcoin. In this context, [24] exploits the so-called wave theory, while [25], [26] and [27] adopt deep learning techniques, like Bayesian-optimized RNNs, LSTM-based RNNs and FFNN-based auto-regressive models, to predict the Bitcoin price.

From the flow of the previous studies, we notice that, over the years, linear models, like the ARIMA, have been set aside for the rise of novel techniques like machine learning. Indeed, some researches proved that the non-linear models perform better than the linear ones under non-stationarity, seasonality, and high-volatility conditions, which often affect financial time-series [2], [21]. These findings encouraged us to investigate a well-known linear model, the ARIMAX, which no one used for market forecasting before, and compare it with the efficient LSTM-based RNN. Furthermore, since cryptocurrencies belong to a new and not well-explored area, we decided to adopt our models for predicting the Bitcoin price.

III. Methodologies

The core of our research includes dataset and models implementation. In section III-A, we describe the procedures we adopted to select, acquire, and pre-process financial and crowd-sourced data, while section III-B refers to the ARIMAX and the LSTM-based RNN.

A. Data Colletion and Preprocessing

The first step to predict BTC market trends is finding useful resources related to the specific stock trade. Since the excessive volatility of the market does not depend only on economic-financial factors [28], but also on public sentiments [7], we decided to adopt both financial and sentiment features as inputs.
For the financial features, we acquired data by using the library *quandl*. Among all the daily BTC features available with *quandl*, we are interested in the followings:

- *BTC weighted price*: the average price of the BTC cryptocurrency;
- *BTC volume*: the total quantity of shares or contracts traded for the BTC cryptocurrency.

We selected financial features that are relevant for short-term analysis, considering a small number of timestamps. We did not include other financial features like the well-known *High*, *Low*, *Open* and *Close*, that might be relevant for the long-term analysis, which takes into account, for example, the previous market breakouts.
Instead, for the sentiment features, we collected data from Twitter through the *twitterscraper* library. We consider Twitter a solid sentiment resource for several reasons. First, Twitter reflects the wisdom of crowds principle [6]: an average of 500 million tweets are posted every day [8], which makes Twitter be a diversified and robust resource, big enough to state that the public opinion about the market forecasts has the same value of the one of an expert. Secondly, the tweets standard format not only differs from others like the one of newspapers articles, but also suggests that each tweet sample refers to a unique topic-sentiment pair. The daily sentiment features we extracted from Twitter are the followings:

- *BTC sentiment*: the average sentiment of tweets related to BTC;
- *BTC tweets volume*: the total quantity of tweets related to BTC.

Each sample of our dataset is, indeed, composed of the daily *BTC Volume*, *Weighted Price*, *Sentiment*, and *Tweets Volume* features (figure 2). We collected data from April 2017 to October 2019, with a total number of 944 days[1].

To obtain the daily *BTC sentiment* and *BTC tweets volume*, we first pre-processed the tweets and then calculated each tweet sentiment. If we look at the tweets content, we can see that it is not always suitable for sentiment analysis. For this reason, before estimating the daily sentiments, we first decided to filter and normalize the tweets. First of all, we considered only tweets written in English and related to the $bitcoin and $btc tickers. Afterwards, we modified the tweets content by adopting lexicon normalization techniques, like stop-words removal, stemming, lemmatization, and spelling correction. After the initial pre-processing step, we estimated

[1]Our dataset is publicly available at https://github.com/giuliaserafini/btc/.

Timestamp	BTCVolume	WeightedPrice	Sentiment	TweetsVolume
2019-09-26	10825.2	8096.93	0.173406	1084
2019-09-27	7405.80	8047.29	0.206158	1851
2019-09-28	3824.93	8171.37	0.236713	1421
2019-09-29	4937.63	8040.93	0.197057	1298
2019-09-30	7532.45	8051.43	0.189400	1823

Fig. 2. Dataset Features Samples.

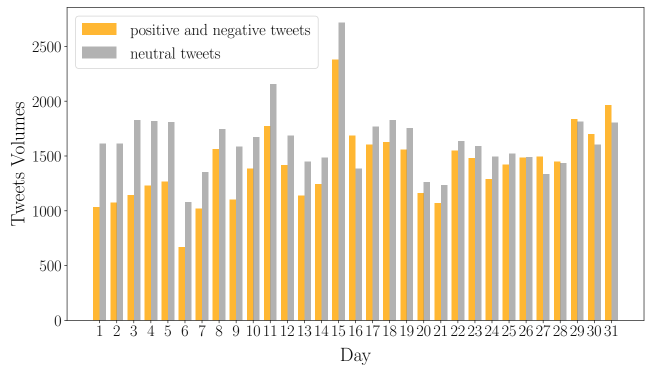

Fig. 3. BTC Tweets Volumes (October 2018).

the sentiment of each tweet. Tweets can be classified as positive, negative, or neutral according to their sentiment. We determined the sentiments by using the library *Vader*, which is appropriate for sentiment expressions in social media contexts [29]. With *Vader*, the polarity score of a sentence is calculated as the sum of all the lexicon ratings of the sentence normalized between -1 (extremely negative) and $+1$ (extremely positive). After this step, we selected the October 2018's data samples to analyze the distribution of the tweets sentiment volumes (figure 3). From the chart, we can see that every day, a considerable amount of tweets is neutral. This is due to the fact that often, tweets related to stock trades contain not only opinions, but also objective events that lack sentiments. Moreover, Twitter has also bots and advertising users, whose sentiments fall into the neutral class. Since tweets without sentiment are not meaningful, we decided to discard them from our dataset. After that, we grouped the tweets based on their date, and we calculated the average daily sentiments and the volumes.

B. Models

In this paragraph, we describe the models adopted for predicting BTC market stocks. In particular, we introduce the ARIMAX and the LSTM-based RNN in section III-B1 and III-B2.

1) ARIMAX: The Auto-Regressive Moving Average, also called ARMA(p,q), is a linear model with the following form:

$$y_t = \varepsilon_t + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} \quad (1)$$

where α_i and β_i are respectively the parameters of the auto-regressive and the moving average parts, y_{t-i} are the time-series terms, and ε_{t-i} are the error terms. The error terms ε_{t-i} are assumed to be independent and identically distributed (i.i.d), with a Gaussian curve with zero mean distribution (more details can be found in [9]). If the time-series is non-stationary, it is appropriate to opt for the generalization of the ARMA model: the ARIMA(p,i,q). The p and q parameters represent, as for the ARMA model, the order of the auto-regressive and moving average parts, while the integrated term i refers to the number of times the time-series has been differentiated to be stationary. The time-series can be differentiated by applying the following formula:

$$y_t^{'} = y_t - y_{t-1} \tag{2}$$

where $y_t^{'}$ is the value of y at time t after one order of differentiation (*i*=1). If the output y_t also depends on an exogenous input, it is necessary to use the ARIMAX model, which adds an additional term X to equation 1, described as follows:

$$X = \sum_{v=1}^{V} \sum_{i=1}^{r} \gamma_{v,i} x_{v,t-i} \tag{3}$$

where $|V|$ is the number of inputs, $x_{v,t-i}$ is the input v at time t, and $\gamma_{v,i}$ and r are respectively the parameters and the order of the exogenous part. In conclusion, the ARIMAX model aims to predict y_t by exploiting the auto-regressive, moving average, and exogenous components.

In our model, y_t refers to the next-day *BTC weighted price*, y_{t-i} to the *BTC weighted price* at day $t-i$, and $x_{v,t-i}$ to the *BTC volume*, *BTC sentiment* and *BTC tweets volume* values at day $t-i$. More specifically, each sample of our model is defined as follows:

$$\begin{aligned} btc\_weighted\_price_t = & \sum_{i=1}^{p} \alpha_i \cdot btc\_weighted\_price_{t-i} + \\ & + \sum_{i=1}^{q} \beta_i \cdot \varepsilon_{t-i} \ + \ \varepsilon_t + \\ & + \sum_{i=1}^{r} \gamma_{1,i} \cdot btc\_volume_{t-i} + \\ & + \sum_{i=1}^{r} \gamma_{2,i} \cdot btc\_sentiment_{t-i} + \\ & + \sum_{i=1}^{r} \gamma_{3,i} \cdot btc\_tweets\_volume_{t-i} \end{aligned} \tag{4}$$

Before building the linear model, we first need to examine the time-series stationarity and define the parameter i. Indeed, we adopted the Augmented Dickey-Fuller (ADF) test, which is a promising method mostly used to test the stationarity of financial time-series [30], and we evaluated the result by analyzing one of the ADF outputs: the *p-value*. The *p-value* indicates the probability that our hypothesis, i.e., the stationarity of the

Input: $training\_set, test\_set, p\_threshold, q\_threshold$
Output: $best\_p, best\_q, lowest\_mse$

$best\_p \leftarrow -1$
$best\_q \leftarrow -1$
$i \leftarrow 1$
for p in $p\_threshold$ **and** q in $q\_threshold$ **do**
    $model \leftarrow ARIMAX(p, i, q)$
    $model.train(training\_set)$
    $mse = model.predict(test\_set)$
    if $mse < lowest\_mse$ **then**
        $best\_p \leftarrow p$
        $best\_q \leftarrow q$
        $lowest\_mse \leftarrow mse$
    end if
end for

Algorithm 1. ARIMAX Parameters Tuning.

time-series, is correct. We set the *p-value* confidence threshold to 1%. Since initially, the *p-value* confidence was not below the threshold, we differentiated the dataset once (i.e., *i*=1) to finally obtain a stationary time-series. After that, we built an algorithm that iterates on a finite number of *p* and *q*, and finds the best parameters of the ARIMAX, in terms of the lowest mean squared error (MSE) on new predictions (algorithm 1). At each iteration, we built a different ARIMAX(p,i,q) model by using the *pyflux* library[2]. The model's parameters have been estimated by using the Maximum Likelihood Estimation (MLE).

2) LSTM-based Recurrent Neural Network: RNN is a class of neural networks useful for processing sequential data [31]. Its basic structure consists of a directional graph that connects a sequence of nodes. This kind of network is suitable for predicting time-series data, and, besides, it is one of the most popular deep neural networks used for market forecasts [2]. The issue introduced by recurrent networks is that, during the backpropagation, the gradient may vanish or explode. A variation of the traditional RNN solved the backflow problems by adding a memory unit, called LSTM, in the RNN structure, composed by input gate, output gate, and forget gate [32], [33].

Before building the RNN model, we determined the shape of each sample, which is made of a combination of features described in section III-A, repeated for a sliding-window (i.e., a sequence of consecutive units) of three days. Each sample is associated with a label that refers to the next-day *BTC weighted price*. Our RNN model is composed of an input layer, an output layer type Dense, and three LSTM hidden layers (figure 4). The input is a tensor, whose dimensions are the number of samples, the sliding-window size, and the number of features related to each sample[3]. Instead, the output is a single value that represents the estimated next-day BTC weighted price. The LSTM layers are composed of 30 units

[2]We set the *p* and *q* thresholds to 3, while the value of *r* is 1 by default.
[3]In figure 4, the number of input features is 4, but it can vary depending on the input features combination selected (more details in section IV).

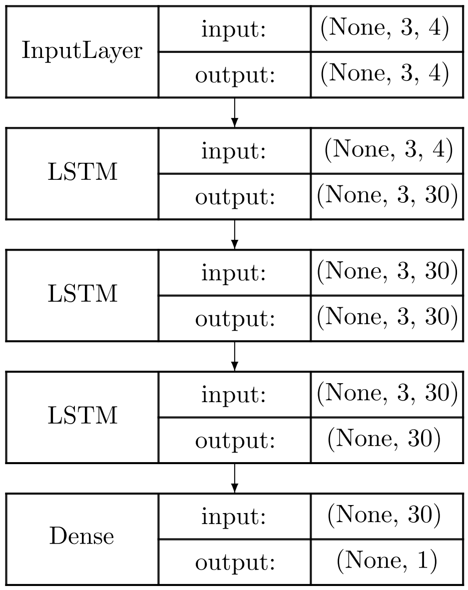

Fig. 4. RNN with LSTM hidden layers.

each, and the last one also includes a dropout of 50% to avoid overfitting. Since the activation function of the LSTM units is the hyperbolic tangent (*tanh*), whose range is between -1 and 1, we decided to scale the input features between those values before fitting the model. Moreover, the optimizer and the loss function adopted are respectively the Adam Optimizer and the MSE. The model has been trained for 400 epochs, with a batch size of 5 samples.

To build the RNN model, we adopted the *keras* library.

Combination	ARIMAX MSE	(p,i,q)	RNN MSE
1	0.00034745	1,1,2	0.00210033
2	0.00032384	2,1,2	0.00229036
3	0.00036445	1,1,2	0.00219284
4	0.00035345	1,1,2	0.00179336
5	0.00033647	1,1,2	0.00274001
6	0.00030187	3,1,1	0.00134103
7	0.00036392	3,1,0	0.00177107

TABLE I
ARIMAX AND RNN MSEs FOR EACH FEATURES COMBINATION (ENUMERATED IN SECTION IV-A), WITH RELATIVE ARIMAX(P,I,Q) ORDERS.

IV. Results

Our contribution is divided into three parts. In section IV-A, we first test the ARIMAX and RNN models with different combinations of input features. Then, in section IV-B, we compare the overall performance of our models and we find the most performing one. Finally, in section IV-C, we check the robustness of the models.

A. Features Combinations

The performance of a model not only depends on its intrinsic structure but also on its input. For this reason, we first decided to test our models with several combinations of input features. Each features combination is made of *BTC weighted price*, plus one of the following sequences of features:

1) *BTC volume*, *sentiment*, *tweets volume*
2) *BTC volume*, *sentiment*
3) *BTC volume*, *tweets volume*
4) *BTC sentiment*, *tweets volume*
5) *BTC volume*
6) *BTC sentiment*
7) *BTC tweets volume*

Before building our models, we first need to analyze the correlation between the *BTC weighted price* and the exogenous input features. Indeed, we adopted the Granger's causality test to determine the relationship between *BTC weighted price* and *BTC volume*, *tweets sentiment* and *tweets volume*. We discovered that the *BTC weighted price* is strongly correlated to all the exogenous input features, except for the *BTC volume*. However, we decided not to discard the *BTC volume* and build the models by considering all the possible combinations of input features.

Afterwards, we split the dataset into training (80%) and test (20%) set. Then, we trained the models for all the different combinations of features mentioned above, and we calculated the MSE on the predictions based on the test set. In particular, unlike the RNN which has a fixed structure, for the ARIMAX case, we first found the best p, q, and i parameters for each features combination by applying algorithm 1. The resulting MSEs are shown in table I[4] and represented in figure 5 and 6. From figure 5 and 6, we can see that for both the models, the lowest MSE (which is numerically smaller than 0.14%) is obtained by using the only *BTC weighted price* and *sentiment* as features (6^{th} combination, table I). From this observation, we understand that not always a higher number of input features leads to better results, because both the models achieved the best outcome with only two input features. Moreover, if we compare the 5^{th} and 6^{th} features combinations of table I, for both the ARIMAX and the LSTM-based RNN models, we can state that the predictions are more accurate with the *BTC sentiment* and not the *BTC tweets volume*. This result disproves the Jethin et al. statement [8], which affirms that there is no correlation between the *BTC sentiment* and market trends.

B. ARIMAX and RNN Performances

At this point, we want to compare the overall performance of our models based on the MSEs described in table I and represented in figure 7. From this figure, we can see that, for all the different combinations of input features, the ARIMAX MSEs are lower than the RNN ones. We can also verify the same results by looking at the ARIMAX and RNN predictions on the test set. Indeed, we plotted the predictions of the

[4]The MSEs are approximated to the 8^{th} decimal place.

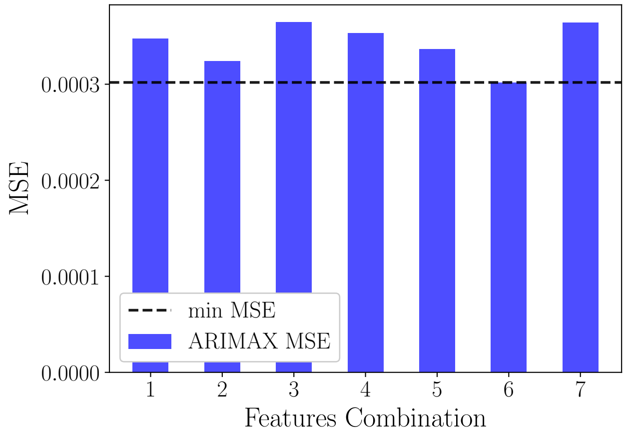

Fig. 5. ARIMAX MSEs with different features combinations.

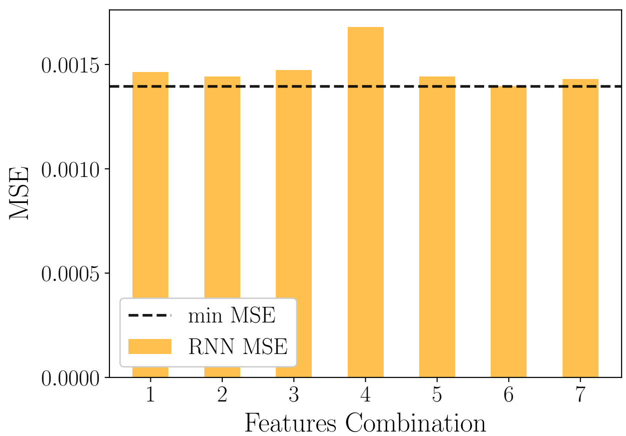

Fig. 6. RNN MSEs with different features combinations.

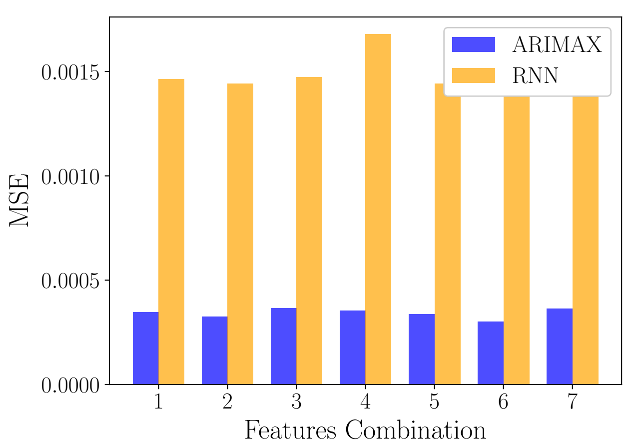

Fig. 7. ARIMAX and RNN MSEs.

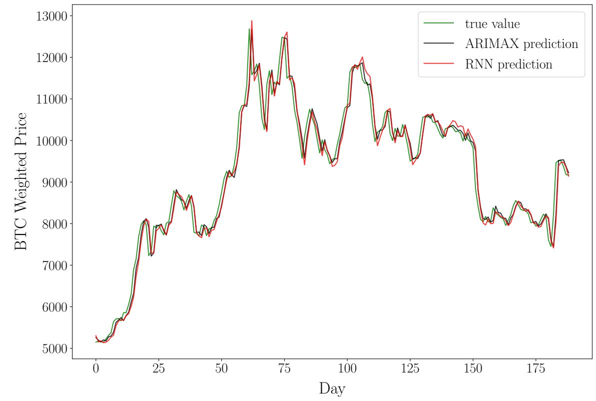

Fig. 8. ARIMAX(3,1,1) and RNN predictions by using the *BTC weighted* price and *sentiment* as features.

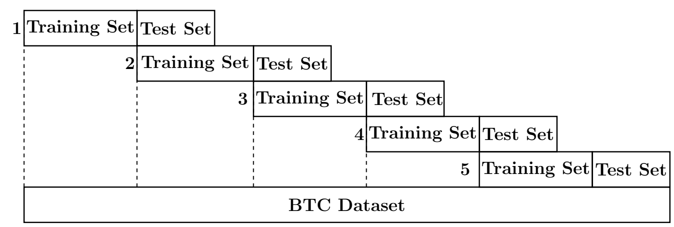

Fig. 9. Cross-Validation: dataset blocks division.

ARIMAX and RNN models which obtained the lowest MSEs (6^{th} combination, table I), compared with the real value of the *BTC weighted price* (figure 8), and we observed that, especially in proximity of the local minima and maxima, the ARIMAX predictions are closer to the true values than the RNN ones. From these results, we also state that not always a more complex model, like the LSTM-based RNN, is better than a linear one. In fact, for BTC market predictions, the linear ARIMAX(3,1,1) with the *BTC weighted price* and *sentiment* as features, achieves better performance than the non-linear RNN.

C. Robustness Validation

A financial model is robust if its predictions are accurate in any condition, even if the market is affected by dramatic changes. In that respect, cross-validation is an effective technique used to generalize the results of a model, estimate its overall performance, and check its robustness. In our case, cross-validation is important for verifying the results presented in section IV-A and IV-B. We opted for the so-called walk-forward cross-validation, suitable for time-series models [34]. Indeed, we equally divided the dataset into a sequence of five blocks, each made up of training and test set (figure 9), and we determined the overall performance of the models by averaging the MSEs on the test sets.

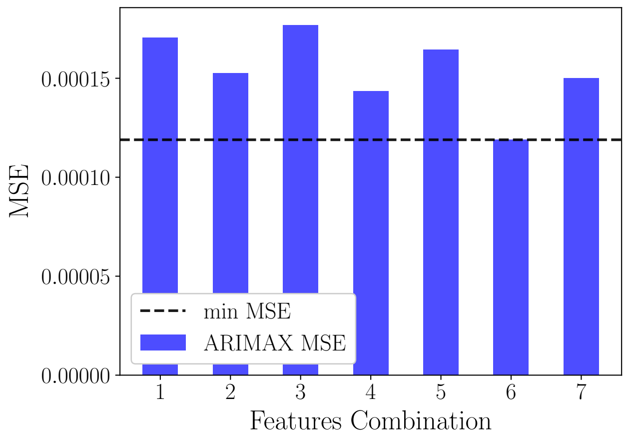

Fig. 10. Cross-Validation: ARIMAX MSEs with different features combinations.

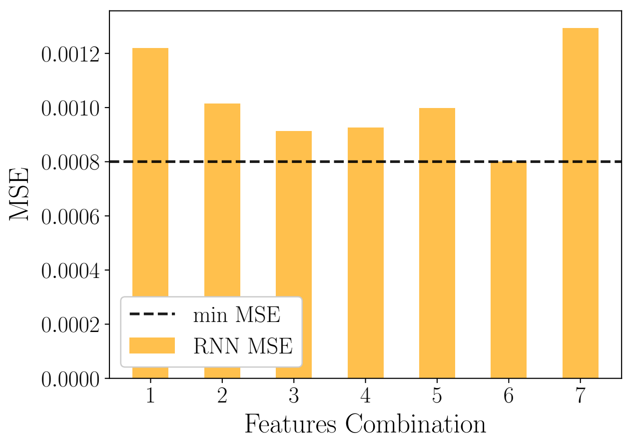

Fig. 11. Cross-Validation: RNN MSEs with different features combinations.

Combination	ARIMAX MSE	RNN MSE
1	0.00017052	0.00288201
2	0.00015253	0.00181126
3	0.00017685	0.00199402
4	0.00014348	0.00264277
5	0.00016447	0.00128573
6	0.00011893	0.00100892
7	0.00015008	0.00203627

TABLE II
CROSS-VALIDATION: ARIMAX AND RNN MSES FOR EACH FEATURES COMBINATION (ENUMERATED IN SECTION IV-A).

Table II[5], figure 10, figure 11, and figure 12 show the averaged MSEs of the ARIMAX and the RNN, obtained through the walk-forward cross-validation. From these findings, we can verify the results achieved by the ARIMAX and RNN models, described in the previous sections:

- the combination of features that leads to the most accurate *BTC weighted price* predictions is composed of the sole *BTC weighted price* and *tweets sentiment*, for both the ARIMAX and the RNN models (table II, figure 10 and figure 11, 6^{th} combination);
- between the RNN and the ARIMAX, the model with the lowest MSE on new predictions is the ARIMAX (figure 12).

[5]The MSEs are approximated to the 8th decimal place.

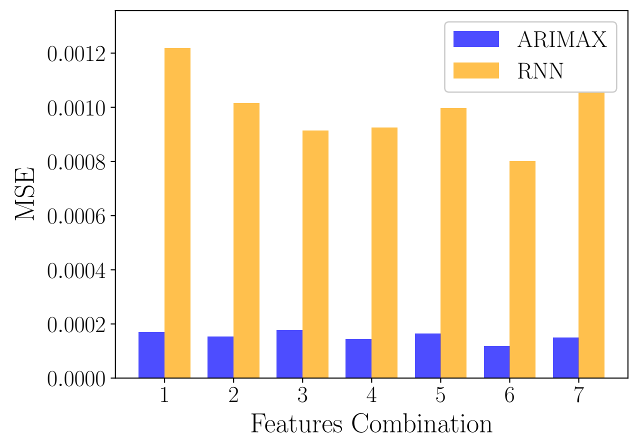

Fig. 12. Cross-Validation: ARIMAX and RNN MSEs.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a statistical and a deep-learning based models to predict the daily weighted price of Bitcoin, the most prominent stock trade among all the cryptocurrencies. In particular, we trained the ARIMAX and LSTM-based RNN with several combinations of financial and sentiment input features, and we found that the best combination is made up of the sole *BTC weighted price* and *tweets sentiment*. From this result, we observed that not always a higher number of input features leads to better outcomes. Moreover, we proved that, between our models, the linear ARIMAX is not only the most straightforward but also the most performing model, with an MSE of 0.00030187 on new predictions. Thanks to this achievement, we also proved that the ARIMAX outperforms the LSTM-based RNN, the most popular machine learning technique that nowadays is used for predicting market stocks.

In our future work, we would like to verify if our models can be generalized for predicting other different stock trades. Moreover, we wish to compare our ARIMAX model with other novel machine learning techniques adopted for market predictions, like CNNs and GANs.

ACKNOWLEDGEMENT

The work is supported by the National Key Research and Development Program of China (No.2019YFB1405000).

References

[1] V. Rajput and S. Bobde, "Stock market prediction using hybrid approach," presented at the International Conference on Computing, Communication and Automation (ICCCA), 2016, pp. 82–86.

[2] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: a review and taxonomy of prediction techiques," *International Journal of Financial Studies*, vol. 7, no. 26, pp. 1–22, 2019.

[3] K. Fu and A. Rosenfeld, "Pattern Recognition and Image Processing," *IEEE Transactions on Computers*, vol. 25, pp. 1336–1346, 1976.

[4] B. Zhou and J. Hu, "A dynamic pattern recognition approach based on Neural Network for stock time-series," presented at the World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009, pp. 1552–1555.

[5] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, pp. 1–8, 2011.

[6] D. Goldberg, D. A. Nichols, B. M. Oki, and D. B. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, pp. 61–70, 1992.

[7] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," presented at the International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), 2016, pp. 1345–1350.

[8] A. Jethin, D. Higdon, J. Nelson, and J. Ibarra, "Cryptocurrency price prediction using tweet volumes and sentiment analysis," *SMU Data Science Review*, vol. 1, no. 3, 2018.

[9] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc. San Francisco, CA, USA, 1990.

[10] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," presented at the 16^{th} International Conference on Computer Modelling and Simulation, 2014, pp. 106–112.

[11] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," presented at the International Conference on Advances in Computing Communications and Informatics (ICACCI), 2017, pp. 1643–1647.

[12] A. H. Moghaddam, M. H. Moghaddam, and M. Esfandyari, "Stock market index prediction using Artificial Neural Network," *Journal of Economics, Finance and Administrative Science*, vol. 21, pp. 89–93, 2016.

[13] M. Wen, P. Li, L. Zhang, and Y. Chen, "Stock market trend prediction using high-order information of time series," *IEEE Access*, vol. 7, pp. 28 299–28 308, 2019.

[14] I. Parmar, N. Agarwal, S. Saxena, R. Arora, S. Gupta, H. Dhiman, and L. Chouhan, "Stock market prediction using Machine Learning," presented at the 1^{st} International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018, pp. 574–576.

[15] M. A. A. Roslan and M. H. F. Rahiman, "Stock prediction using sentiment analysis in Twitter for day trader," presented at the 9^{th} IEEE Control and System Graduate Research Colloquium (ICSGRC), 2018, pp. 177–182.

[16] R. Batra and S. M. Daudpota, "Integrating StockTwits with sentiment analysis for better prediction of stock price movement," presented at the International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative Adversarial Networks," *ArXiv*, vol. abs/1406.2661, 2014.

[18] X. Zhou, Z. Pan, G. Hu, S. Tang, and C. Zhao, "Stock market prediction on high-frequency data using Generative Adversarial Nets," *Mathematical Problems in Engineering*, 2018.

[19] K. Zhang, G. Zhong, J. Dong, S. Wang, and Y. Wang, "Stock market prediction based on Generative Adversarial Network," *Procedia Computer Science*, vol. 147, pp. 400–406, 2019.

[20] Z. Li and V. Tam, "A comparative study of a Recurrent Neural Network and Support Vector Machine for predicting price movements of stocks of different volatilites," presented at the IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–8.

[21] M. Hiransha, E. A. Gopalakrishnan, K. M. Vijay, and K. P. Soman, "NSE stock market prediction using Deep-Learning models," *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018.

[22] B. Lin, W. Chu, and C. Wang, "Application of stock analysis using Deep Learning," presented at the 7^{th} International Congress on Advanced Applied Informatics (IIAI-AAI), 2018, pp. 612–617.

[23] S. Liu, G. Liao, and Y. Ding, "Stock transaction prediction modeling and analysis based on LSTM," presented at the 13^{th} IEEE Conference on Industrial Electronics and Applications (ICIEA), 2018, pp. 2787–2790.

[24] A. A. Tadvi, "Bitcoin price prediction," *International Journal of Academic Research and Development*, vol. 3, pp. 1112–1114, 2018.

[25] S. McNally, J. Roche, and S. Caton, "Predicting the price of Bitcoin using Machine Learning," presented at the 26^{th} Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018, pp. 339–343.

[26] H. Jang and J. Lee, "An empirical study on modeling and prediction of Bitcoin prices with Bayesian Neural Networks based on Blockchain information," *IEEE Access*, vol. 6, pp. 5427–5437, 2017.

[27] N. I. Indera, I. M. Yassin, A. Zabidi, and Z. I. Rizman, "Non-Linear Autoregressive with Exogenous input (NARX) Bitcoin price prediction using PSO-optimized parameters and moving average technical indicators," *Journal of Fundamental and Applied Sciences*, vol. 9, pp. 791–808, 2017.

[28] D. Peng, "Analysis of investor sentiment and stock market volatility trend based on Big Data strategy," presented at the International Conference on Robots & Intelligent System (ICRIS), 2019, pp. 269–272.

[29] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," presented at the 8^{th} International AAAI Conference on Weblogs and Social Media (ICWSM), 2014.

[30] B. Li, J. Zhang, Y. He, and Y. Wang, "Short-Term load-forecasting method based on wavelet decomposition with second-order gray neural network model combined with ADF test," *IEEE Access*, vol. 5, pp. 16 324–16 331, 2017.

[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[32] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, pp. 1735–1780, 1997.

[33] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," presented at the 9^{th} International Conference on Artificial Neural Networks (ICANN), 1999, p. 850855.

[34] M. Y. Hu, G. P. Zhang, C. X. Jiang, and B. E. Patuwo, "A CrossValidation analysis of Neural Network OutofSample performance in exchange rate forecasting," *Decision Sciences*, vol. 30, pp. 197–216, 2007.