LSTM Model to Forecasting Bitcoin: Internal and External Determinants

Sihua Kang

sihua.kang0824@gmail.com

the Virginia Polytechnic Institute and State University, Blacksburg, United States

Abstract. The stock market is difficult for prediction, because of its complexity and randomness. Bitcoin, as the new favorite of stock market, grabs much attention. This article aims to apply the LSTM model for Bitcoin prediction, using multiple financial indices as features of LSTM to find out the relationship between the timestep and the performance. Compare the performance of different models to improve the accuracy of prediction.

Keywords: LSTM, Bitcoin, Forecasting.

1 Introduction

The trend of stocks is nonlinear, dynamic, and extremely difficult to predict. It is a challenging topic to accurately predict stocks. As it has provided that the link between the stock market performance and economic growth both in short run and long run [1]. The prediction of the stock market using various economic and market indicators has been a subject of significant interest and research. There are tons of research that have been subject of using machine learning or deep learning algorithm applying in quantitative finance area and prediction [2]. Recurrent neural networks (RNN), the most popular deep learning model, was the general model of stock price prediction [3]. However, the structure of general RNN limits its performance in prediction as it loses the long-term information. To handle the information lost, applying Long Short-Term Memory (LSTM) is a good choice. After LSTM was published, many research subjects on using LSTM for prediction [4][5].

Using basic data of stock for prediction is not enough. The performance of the model is not good enough to give us a generous return. Fortunately, after the Three Factor Model has published by Fama and French [6], there has been a burst of financial indices. We can use them to improve the performance of the Model.

However, less research has subject on the Bitcoin prediction using financial index. Bitcoin, created in 2009 by Nakamoto, is the first decentralized cryptocurrency. Utilizing blockchain technology for secure and transparent transactions, it operates independently of central banks, challenging traditional financial systems. According to the research [7], "Bitcoin has many similarities to both gold and the dollar. Medium of exchange characteristics are clear, and bitcoin reacts significantly to the federal funds rate, which points to bitcoin acting like a currency." Bitcoin is called digital gold [8] because of its high profits in diversified portfolios, and many people will use Bitcoin as a substitute during a downturn in mainstream financial markets. But high profits are accompanied by extremely high volatility. This also makes Bitcoin more

BDEDM 2024, January 12-14, Ningbo, People's Republic of China

DOI 10.4108/eai.12-1-2024.2347245

difficult to predict than traditional financial markets. According to the research, Bitcoin is significantly affected by investor sentiment. [9] In this research, we will find out the relationship between the performance of the deep learning model and the time step of the dataset using some common market indicators.

2 Data and features

The data under consideration is about Bitcoin, today it is the largest market capitalization cryptocurrency. The data source is from finance.yahoo.com, a widely recognized financial news and data platform providing a comprehensive range of financial information and tools. It offers real-time updates on global financial markets, including stock market data, news, and analysis. In our study, Yahoo Finance was utilized as a primary source for historical closing prices, open prices, high prices, low prices, and volume from February 28, 2018, to March 31, 2023, which was pivotal in calculating features like intraday intensity index, and Exponential Moving Average (EMA), etc. Figure 1 presents the Bitcoin's closing prices from February 28, 2018, to March 31, 2023,

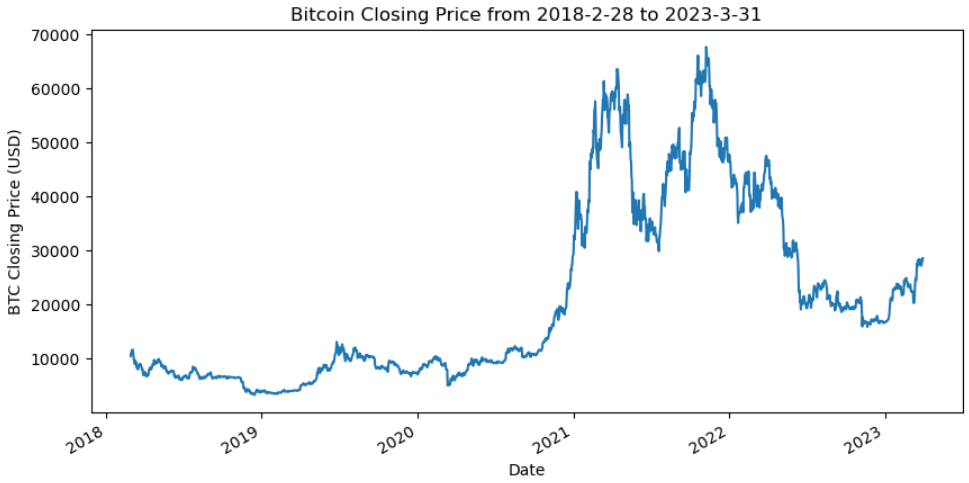

Figure 1: Bitcoin Closing Price.

Figure 1 shows a significant increase between September 2020 and May 2021. The fluctuation is remarkably high after the first peak in May 2021.

Table 1 shows 16 features selected, along with their corresponding formula and explanation. For the row containing missing values, we use k-Nearest Neighbors (KNN) to fix the missing value to avoid algorithm incompatibility error, where k-Nearest Neighbors is a method to impute missing values using the mean or median of the k-nearest neighbors found in the complete cases of the dataset.

Table 1: Feature formula and explanation.

FEATURE NAME	FORMULA	DESCRIPTION
CRYPTO FEAR GREED INDEX	N/A	A daily measure indicating the current sentiment of the cryptocurrency market
EPU	N/A	Economic Policy Uncertainty
GPR	N/A	Geopolitical Risk index
ILLIQUIDITY	$\frac{\lvert \text{Renturn} \rvert}{\text{Volume}}$	The measure of how easily Bitcoin can be bought or sold without affecting the price.
MOMENTUM	$\text{Price}_t - \text{Price}_{t-7}$	A trend-following indicator that shows the strength or weakness of Bitcoin's price over time.
TURNOVER RATIO	$\frac{\text{Volume}}{\text{ClosingPrice}}$	The ratio of traded Bitcoin volume over the total supply.
7DAY MA	$\frac{1}{7}\sum_{i=0}^{6} P_{t-i}$	It calculates the averaging of closing prices P of Bitcoin over past 7 days to identify trends in time series data by smoothing out short term fluctuations.
7DAY EMA	$\text{Value} * \frac{\text{Smothing}}{1+7} + \text{EMA}_{t-1} * \left(1 - \frac{\text{Smothing}}{1+7}\right)$	It gives more weight to the most recent prices, and therefore reacts more quickly to price changes
INTRADAY INTENSITY INDEX	$\frac{\text{Closing} * 2 - \text{High} - \text{Low}}{[(\text{HIgh} - \text{Low}) * \text{Volume}]}$	It provides a continuous volume-focused indicator by using a security's most recent close, high, and low in its calculation while also factoring in volume.
PSY	$\frac{\text{Up}_n}{n} * 100$	Psychological Line, shows the percentage of periods that Bitcoin closed over a given period.
AR	$\frac{\sum_{i=1}^{N} \text{High}_i - \text{Open}_i}{\sum_{i=1}^{N} \text{Open}_i - \text{Low}_i}$	Attach importance to the opening price to reflect the popularity of the current market
BR	$\frac{\sum_{i=1}^{N} \text{High}_i - \text{Close}_{i-1}}{\sum_{i=1}^{N} \text{Close}_{i-1} - \text{Low}_i}$	Attach importance to the closing price to reflect people's willingness to buy and sell in the current market
OBV	$\text{OBV}_{\text{prev}} + \begin{cases} \text{Volume} & \text{if Close} > \text{Close}_{\text{prev}} \\ 0 & \text{if Close} > \text{Close}_{\text{prev}} \\ -\text{Vloume} & \text{if Close} > \text{Close}_{\text{prev}} \end{cases}$	A technical trading momentum indicator that uses volume flow to predict changes in Bitcoin price.

3 Exploratory data analysis

3.1 The correlation of data

Next, we calculate the correlation between all features and close price. Figure 2 is the heatmap to show the correlation of the calculation. We can observe the relationship between each feature and the closing price. The correlation of 7 days MA (Moving Average), 7 days EMA (Exponential Moving Average), and 14 days EMA (Exponential Moving Average) are identical and the same as AR and BR. Therefore, during model training, we contract 7 days MA, 7 days EMA, and 14 days EMA to just 14 days EMA. We also reduce the Emotion Index (AR) and Willingness Index (BR) to just AR, as AR can be used independently.

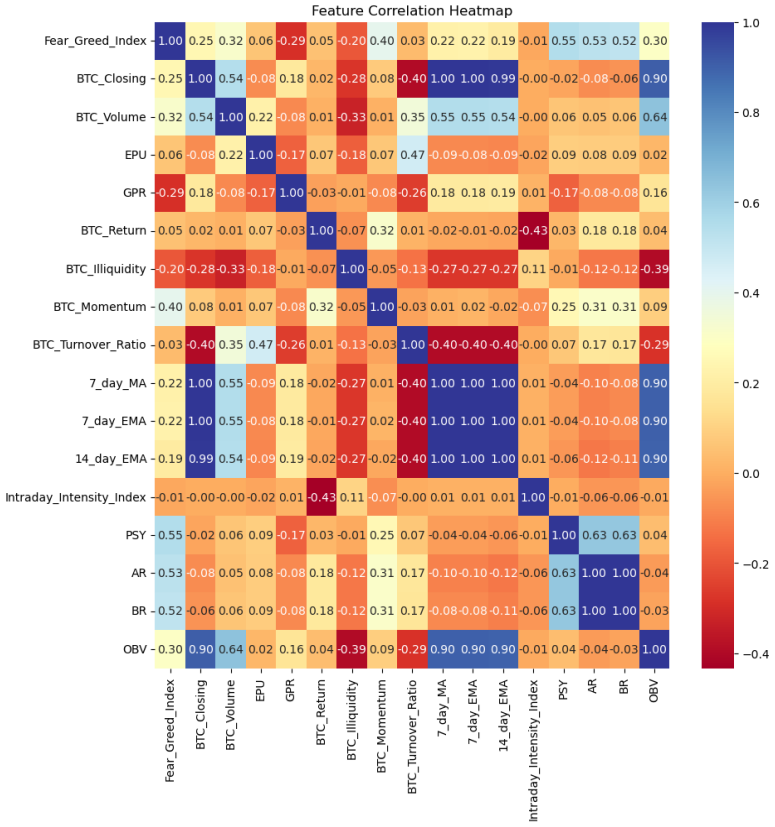

Figure 2: Heat Map.

3.2 Data preprocessing

For data preprocessing, Min-Max normalization was applied to rescale the value in the dataset to the range from 0 to 1. Normalization ensures that each feature contributes equally to the analysis and helps our algorithm converge faster and more accurately during training. The Min-Max method is commonly used in neural networks. It decreases the influence of the outlier and

makes a common scale without distorting differences in the ranges of values, as elucidated in [13]

The equation (1) is the Min-Max normalization function.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

We split a dataset into training and test sets, typically with an 80:20 ratio. Allocating 80% to training ensures the model has enough data to learn effectively. Training on most data allows the model to capture the underlying patterns without being too constrained by a small dataset. The remaining 20% is an independent set to evaluate the model's performance. This split ensures enough data to validate the model's predictions and assess its generalization to unseen data. This balance helps prevent overfitting and underfitting. We use a long short-term memory (LSTM) to predict the closing price of Bitcoin. The following 12 variables will be used as features: trading volume (' BTC Volume'), Economic Policy Uncertainty Index (' EPU'), Global Peace Index (' GPR'), Bitcoin Illiquidity (' BTC Illiquidity'), Momentum (' BTC Momentum'), a 14-day Exponential Moving Average ('14 day EMA'), Turnover Ratio (' BTC Turnover Ratio'), and the Cryptocurrency Fear and Greed Index, Intraday Intensity Index, Psychological Line(PSY), Emotion index (AR), On-Balance Volume (OBV).

4 Long short-term memory (LSTM) networks

4.1 Introduction

LSTM stands for Long Short-Term Memory. It is a type of artificial recurrent neural network (RNN) architecture used in deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections that make them capable of processing single data points and entire data sequences. This makes them ideal for speech recognition, language modelling, translation, and time-series forecasting tasks.

LSTM is designed by Hochreiter and Schmidhuber [10] to avoid the long-term dependency problem, meaning it can remember information for long periods, which is the main advantage over traditional RNN. This is achieved through its unique gating mechanism, which regulates the flow of information to be remembered or forgotten during the training process. The structure of LSTM includes the cell state and various gates, as shown in Figure 3. The cell state acts as a 'conveyor belt' that runs straight down the entire chain of LSTM cells, allowing information to flow along it unchanged if necessary. It decreases the reliance on short-term memory. The input gate controls the pass rate of new input information into the cell state. The forget gate prevents the retention and discarding of information in the cell state. Finally, the output gate determines the next hidden state, which is used for predictions and passed to the next time step.

As the description for the gates within the LSTM, we can use the sigmoid activation function to achieve the function of three gates and denote them as i_t (2) for input gate, f_t (3) for forget gate, o_t (4) for output gate. Function as:

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + \epsilon_i) \tag{2}$$

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + \epsilon_f) \tag{3}$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + \epsilon_o) \quad (4)$$

The sigmoid function $\sigma \in (0,1)$ acts as the degree of how open the gate is, where w_i, w_f, w_o represent the weights of input, forget, and output. h_{t-1} is the previous hidden state of the LSTM unit at time step t-1. x_t corresponds to the input at time step t. We import the bias for each gate, denote as $\epsilon_i, \epsilon_f, \epsilon_o$.

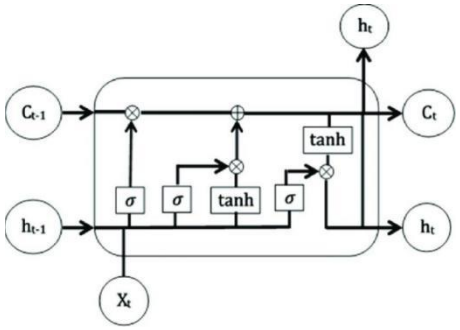

Figure 3: LSTM Structure.

The candidate layer, represented as $\tilde{c}_t$ with equation (5), is a collection of values that could be added to the internal cell state. It is called a "candidate" because whether these values will be added to the cell state is determined by the input gate. It is created by applying a tanh function to a linear combination of the input at the current time step x_t and the previous hidden state h_{t-1} (6), modulated by their respective weights and a bias term. The function ensures that the candidate values are normalized between -1 and 1.

$$\tilde{c}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + \epsilon_c) \quad (5)$$

$$h_t = o_t * \tanh(c_t) \quad (6)$$

where w_c is the weights associated with the candidate layer. Finally, for the cell state update: (7)

$$c_t = f_t * c_{\{t-1\}} + i_t * \tilde{c}_t \quad (7)$$

4.2 Structure of LSTM network

The structure of our model has six layers. It applies to two LSTM layers. One has 128 units and will return the entire sequence of outputs for each sample, and another LSTM layer has 64 units. However, it will produce only the last output in the output sequence, for each LSTM layer follows a Dropout layer with a rate of 0.2. Dropout is a regularization technique that randomly sets input units to 0 at each update during training time, which helps to prevent overfitting. The model has a fully connected Dense layer with 32 neurons using the ReLU activation function. The ReLU (Rectified Linear Unit) activation function is a piecewise linear function that will output the input directly if it is positive. Otherwise, it will output zero. The equation is given as (8)

$$f(x) = \max(0, x) \quad (8)$$

It is used to introduce non-linearity to the model, allowing it to learn more complex patterns. Finally, use a dense layer with a single neuron as the output layer.

4.3 The optimization algorithm

The model is compiled with the Adam optimizer, and the loss function is set to mean squared error. Adaptive Moment Estimation (Adam) is an algorithm for gradient-based optimization of stochastic objective functions. It was developed by Kingma and Ba [11]. It computes adaptive learning rates for each parameter. It is computationally efficient, only requires first-order gradients with little memory requirement and is well-suited for problems with large datasets or parameters. Adam is an advanced algorithm combining the advantages of the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp), the other two extensions of stochastic gradient descent. It stores an exponentially decaying average of past squared gradients like RMSProp and an exponentially decaying average of past gradients like momentum. The requirement of Adam includes stepsize α, exponential decay rates for the moment estimates $\beta_1, \beta_2 \in [0,1]$, stochastic objective function $f(\theta)$, where θ is the parameter vector. The good default settings for these variables are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{\{-8\}}$. Equation (9) refers to the gradients of the stochastic objective at time step t written as g_t. The biased first-moment estimate at time step t, denoted as m_t, has the equation (10) Also, the biased second moment estimate at time step t, denoted as v_t, has the equation (11). After we get m_t, v_t, we can use them to calculate a bias-corrected estimate of the first moment and the second moment, denoted as $\widehat{m}_t, \widehat{v}_t$, with equation (12) and (13). Equation (14) is the update equation of theta at time step t. also we initial θ_0 as $m_0 = 0, v_0 = 0, t = 0$

$$g_t = \nabla_{\{\theta\}} f_t\left(\theta_{\{t-1\}}\right) \tag{9}$$

$$m_t = \beta_1 \cdot m_{\{t-1\}} + (1 - \beta_1) \cdot g_t \tag{10}$$

$$v_t = \beta_2 \cdot v_{\{t-1\}} + (1 - \beta_2) \cdot g_t \tag{11}$$

$$\widehat{m}_t = \frac{m_t}{1-(\beta_1)^t} \tag{12}$$

$$\hat{v}_t = \frac{v_t}{1-(\beta_2)^t} \tag{13}$$

$$\theta_{\{t+1\}} = \theta_t - \frac{\hat{\alpha} m_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{14}$$

The algorithm will stop when θ_t is converged. Adam is suitable for handling non-stationary objective functions, as in the case of RMSPorp, and it does not require a stationary objective. It is also invariant to the diagonal rescale of the implement.

5 Empirical results

5.1 Goal of Experiment

Our goal for this result is to compare the accuracy of predicted value in models with different t time step inputs, which means use t days data to predict the next closing price of Bitcoin. We choose four different time lengths: 60 days, 30 days, 14 days, and 7 days as our experimental objects. We will use RMSE, MAE, MAPE to determine the performance of models.

5.2 RMSE

The Root Mean Square Error (RMSE) is a widely used metric that quantifies the difference between the values predicted by a model and the values observed. It is calculated as the square root of the average of the squared differences between prediction and actual observation. It has the formula like equation (15)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{15}$$

where y_i represents the observed values, $\hat{y}_i$ represents the predicted values, and n is the number of observations.

RMSE is expressed in the same units as the predicted and observed values, which facilitates an intuitive understanding of the magnitude of the error. A lower RMSE value means there is less of the 'average' error in the predictions, with RMSE giving a higher weight to larger errors.

5.3 MBE

The Mean Bias Error (MBE) calculates the average bias in a set of predictions, and it can be calculated by subtracting the observed values from the predicted values and then finding the average of these differences, as equation (16).

$$\mathrm{MBE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i) \tag{16}$$

where y_i represents the observed values, $\hat{y}_i$represents the predicted values, and n is the number of observations. MBE near zero indicates that, on average, the model's predictions are not systematically biased, meaning it is providing accurate and unbiased estimates of the target values. The sign of MBE means the model is either overestimated or underestimated. The smaller absolute value, whether positive or negative, suggests a better performance of bias.

5.4 MAPE

The Mean Absolute Percentage Error (MAPE) (17) is a statistical measure expressing the average absolute error as a percentage of the observed values, providing a relative measure of prediction accuracy. It allows for a straightforward interpretation of the model's predictive performance in terms of average percentage error. MAPE avoids the issue of error compensation seen with measures that consider positive and negative deviations together, which can sometimes cancel each other out. It also shows the error relative to the magnitude of the quantities being predicted. Thus, a lower MAPE will have better performance.

$$\mathrm{MAPE} = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) \times 100\% \tag{17}$$

5.6 Performance

In Table 2, the models trained with longer time steps, 60 and 30 days, outperformed their shorter time step counterparts on all metrics except for R-squared. The 60-day model reported the lowest RMSE and MBE at 2.53% and 0.004, respectively, while the 30-day model followed closely. Conversely, models with a 14-day and 7-day time step exhibited higher errors, with the 7-day model marking the highest RMSE at 3.4515% and MBE at 0.016.

Table 2: Performance in Different Time Step.

TIME STEP	RMSE	|MBE|	MAPE
60	0.0253	0.004	0.058
30	0.0232	0.007	0.056
14	0.0329	0.015	0.065
7	0.0345	0.016	0.077

5.7 Comparison

These findings suggest that, within the scope of our dataset, LSTM models with longer time steps capture the underlying trends more effectively, leading to enhanced prediction accuracy and reduced bias. This may reflect the influence of longer-term dependencies in the data, better encapsulated by models operating over extended time steps. Also, comparing research performance (using the average value of RMSE, MBE, MAPE) [12], as shown in Figure 4, the LSTM model always has lower RMSE, lower absolute MBE, but higher MAPE than ANN and RF models.

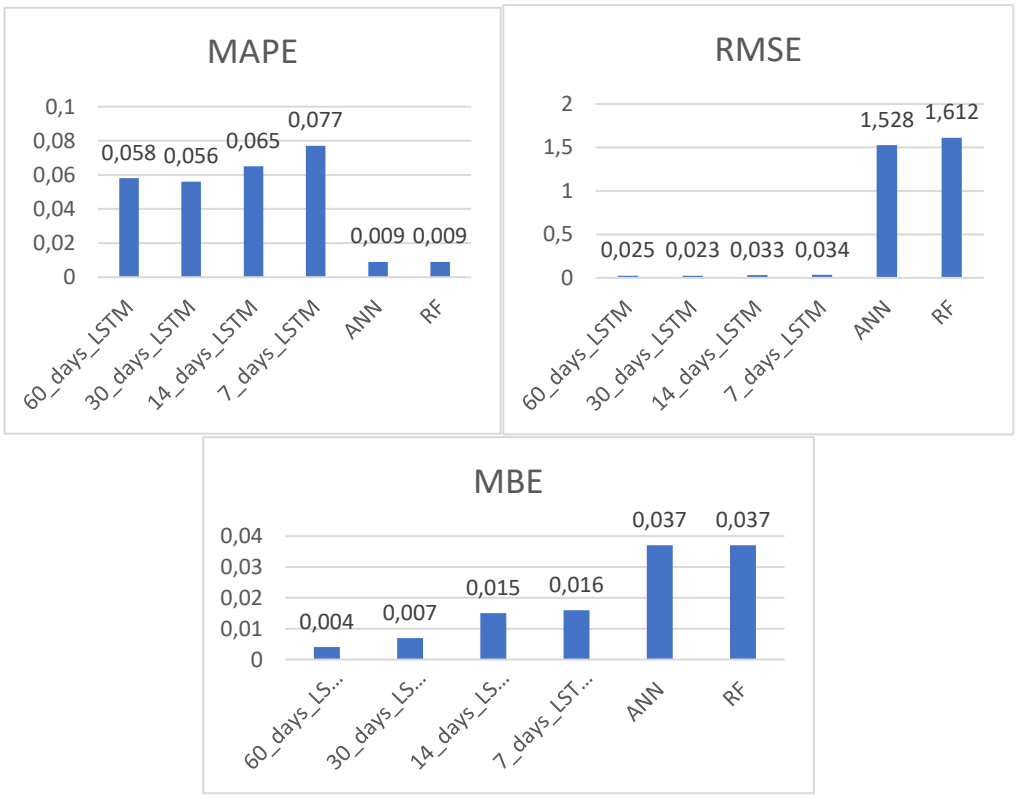

Figure 4: Performance.

As the compared method we provide, a model with better performance should have lower RMSE, absolute MBE, and MAPE. From RMSE, we can see that LSTMs have less average error. From MBE, the predictions of LSTMs have lower systematically biased. But a higher MAPE of

LSTMs means each error relative to the magnitude of the quantities being predicted is higher. This might be because of the features we selected that make the models slightly overfitting. Comparing 4 LSTM models, we can see a clear decrease in three methods, which means that the model has negative relation between the timestep and those methods.

6 Conclusion

We have the subject of the prediction of Bitcoin using various financial index. As the competition of different time step using same LSTM structure. The performance has given us some clues. In conclusion, selecting an appropriate time step is crucial for the performance of the LSTM model. It should be tailored to the data's specific characteristics and the application domain's forecasting requirements. The timestep has a positive relation with accuracy or performance. Comparing the decreasing of RMSE, MBE, and MAPE, we can get an average decrease of 0.17%, 0.0002, and 0.03% in RMSE, MBE and MAPE for each additional day. However, we should consider the overfit case. Also, comparing the performance of research [12], we can get that the average error of prediction LSTM always have better performance than ANN and RF, but for the magnitude of the error, LSTM does not perform well. These could be because of feature selection or overfit case. For our future work, we will try to find the relationship between the time step of the model and the epochs of different time step. Try to find a good strategy to balance the time step and the epochs avoiding the overfit.

References

[1] Guru, B. K., & Yadav, I. S. (2019). Financial development and economic growth: panel evidence from BRICS. Journal of Economics, Finance and Administrative Science, 24(47), 113-126.

[2] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, 2000, pp. 348-353 vol.6, doi: 10.1109/IJCNN.2000.859420.

[3] Jahan, I., & Sajal, S. (2018). Stock price prediction using Recurrent Neural Network (RNN) algorithm on time-series data. In 2018 Midwest instruction and computing symposium. Duluth, Minnesota, USA: MSRP.

[4] Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. Procedia Computer Science, 170, 1168-1173.

[5] Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science, 167, 2091-2100.

[6] Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. the Journal of Finance, 47(2), 427-465.

[7] Dyhrberg, H. A. (2016) Bitcoin, gold, and the dollar – A GARCH volatility analysis, Finance Research Letters, Volume 16, Pages 85-92, ISSN 1544-6123

[8] Taskinsoy, J. (2021). Bitcoin: A New Digital Gold Standard in the 21st Century? Available at SSRN 3941857.

[9] Eom, C., Kaizoji, T., Kang, S. H., & Pichl, L. (2019). Bitcoin and investor sentiment: statistical characteristics and predictability. Physica A: Statistical Mechanics and its Applications, 514, 511-521.

[10] S. Hochreiter & J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[11] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[12] Vijh, M.& Chandola, D.& Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. Procedia computer science, 167, 599-606.
[13] Coronel, C., & Morris, S. (2019). Database systems: design, implementation, and management. Cengage learning.