The Research on the Prediction of Cryptocurrency Based on Linear Regression and LSTM

Feng Ding [1*]

[1] Tianjin Xinhua Highschool, Tianjin, China

*100829@yzpc.edu.cn

Abstract. As the growing interest of investment on Cryptocurrencies and the huge volatility of their price, a need for scientific model to predict the future price is growing. In this context, the paper uses linear regression and LSTM model to predict the price of Bitcoin and Ethereum. The result shows that the prediction made by Linear Regression shows less errors but greater lag compared with the prediction made by LSTM method. The lag problem is considered to generate from lack of peripheral information other than previous prices. The prediction implies that the prices of Cryptocurrencies are theoretically predictable, and shows a direction of further research, such as the use of mixed-LSTM model. The methods provided in this paper can be used in development of better models and further investments.

Keywords: Cryptocurrency, Linear Regression, RNN, LSTM

1 Introduction

Cryptocurrency is a form of digital currency that employs cryptographic principles to safeguard trades and govern the production of exchange units. One type of digital currency is so-called Cryptocurrency. The distributed ledger technology used by blockchain technology gives it its decentralized aspect.

Since its birth in 2008, the cryptocurrencies represented by Bitcoin have shown extreme price volatility due to their scarcity and unique issuance mechanism, plus a variety of factors. Take Bitcoin as an example: Since 2013, Bitcoin's price has increased dramatically, from $ 13 to $ 1,200, a nearly 80-fold gain. In 2017, Bitcoin reached a peak of $ 20,000 before dropping to $ 10,000. The price of Bitcoin successfully passed the $ 60,000 mark in 2021, and the market value has reached $ 1.2 trillion, more than the total market value of the two Teslas. The price moved up and down in the blink of an eye. In 2022, Bitcoin fell to $ 40,000 at the beginning of the year, and then it fell to around $20,000 in the middle of the year and continued to fluctuate. Predicting the price of bitcoin is difficult due to the volatility of the Bitcoin market and the price's non-stationarity.

Deep learning stands out among forecasting model tools for its robust learning and performance characteristics, and it has become the favored tool for time series

V. Escudero et al. (Eds.): BADS 2022, AHCS 11, pp. 1472–1481, 2023.
https://doi.org/10.2991/978-94-6463-102-9_154

forecasting in the financial field. Donaldson (1999) used artificial neural networks (ANNs) to predict S&P 500 stock values as early as the end of the twentieth century, and cross-validated the advantages of neural networks over older methods such as weighted least squares [1]. Takeuchi and Lee (2013) employed a stack-restricted Boltzmann machine-based autoencoder to extract stock price feature information, forecasted which stocks would have greater or lower monthly returns than the median, and finally obtain yearly returns with 53% accuracy and 45.93% accuracy [2]. Zhang (2017) forecasted the stock market using a improved BP neural network and attained a remarkable prediction accuracy [3].

Since the LSTM algorithm's outstanding performance in the field of machine translation in 2014, it has been regarded highly and employed extensively in the financial sector. The WMT-14 data set was translated from English to French using a multi-layer Long Short Term Memory network by Sutskever et al. (2014). The total translation score for the full test set was 34.8, which is higher than the phrase-based SMT approach and has more benefits for translating lengthy sentences [4]. In the area of speech recognition, Li et al, (2014) also built a viable Chinese speech recognition system using the LSTM model after the speech recognition for English had been built [5]. Murtaza (2017) deployed a 2-layer LSTM network to forecast the price of the NIFTY50 stock in the processing of financial time series. The test set result was achieved with an RMSE of 0.00859, and the prediction accuracy was shown to be much improved than that of the econometric model [6].

In the research on Bitcoin price prediction, the related models of deep learning and LSTM are also gradually occupying the mainstream. Therefore, this paper compares the linear regression model based on time series with the LSTM model to test the fitting performance of the LSTM model on cryptocurrency prices.

The remaining part of this paper is structured as follows. In the next section, the dataset will be explored, the principles and characteristics of Linear Regression, Recurrent Neural Networks, and Long Short Term Memory networks are introduced; in section three, the construction process and evaluation indicators of the two models in this paper are introduced; in section four and five, the statistical results of prediction and analysis are drawn; and the full study and future prospects are then concluded in section 6.

2 Data and Method

2.1 Data

The paper uses the Close price of Bitcoin and Ethereum from July 30, 2019 to July 15, 2022. The data set takes every half hour as a sampling moment, with a total of 51828 pieces of data. The dataset is divided by ratio of 80:20. 80% of the data serves for training, and the 20% of the data is used as validation set. Both models take Close price as the prediction target. The data is download freely from Kaggle, a well-known website for statistical learning, the dataset for the price of Bitcoin and Ethereum are respectively called Bitcoin Cryptocurrency Historical Data and Ethereum Cryptocurrency Historical Data.

2.2 Method

Linear Regression.

Provided a dataset with N statistical units $\{y_n, x_{n1}, \ldots, x_{np}\}_{n=1}^{N}$, a linear relationship between the dependent variable y and the vector of regressors x is presumed by a typical Linear Regression model. This link is represented by a term that indicates disturbance or error variable ε, which is an unobserved random variable that put "noise" into the linear relationship between the dependent variable and the regressors. As a result, the model assumes the form

$$y_n = \beta_0 + \beta_1 x_{n1} + \cdots + \beta_p x_{np} + \varepsilon_n \tag{1}$$

These n equations are typically packed altogether and represented as matrix form: $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where,

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$
$$X = \begin{pmatrix} \mathbf{x}_1^{\top} \\ \mathbf{x}_2^{\top} \\ \vdots \\ \mathbf{x}_n^{\top} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}, \tag{2}$$
$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Recurrent Neural Network (RNN).

The difference between plain neural networks and RNN is that RNN obtains contextual input by building connections between neurons in the same hidden layer in order to exploit its short-term memory advantage. Figure 1 depicts an expanded view of the Recurrent Neural Network. As it is depicted in Figure 1, in addition to the current input information, the hidden layer's input information also includes the hidden layer's output information from the previous moment, forming a time dependency. Using this connection structure of hidden nodes at various moments, RNN may realize the memory of historical moment value and use it to the computation of current output.

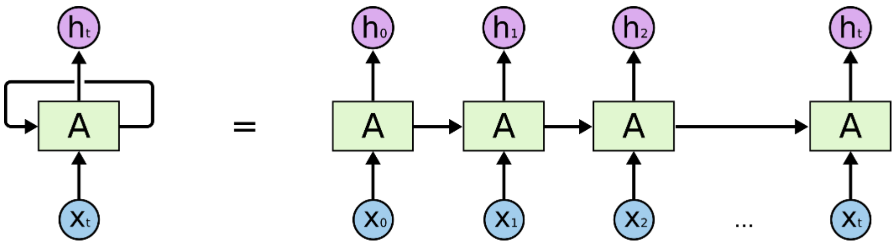


Fig. 1. The brief principle of an RNN unit and RNN network [11]

However, when the parameters of RNN are optimized and applied to long-term sequence processing, due to the fact that each layer shares the same weight parameters and the activation function derivative multipliers are continuously accumulated, etc., gradient disappearance and gradient explosion often occur, resulting in too small memory value. Therefore, RNNs are often ineffective in dealing with long-term timing problems.

Long Short Term Memory network (LSTM).

Under this problem, Long Short Term Memory network (LSTM) was introduced to tackle the aforementioned RNN limitations. As a specialized recurrent neural network, LSTM introduces a gate system based on RNN, which includes an input gate, a forget gate, and an output gate. These three types of gates govern the update and output of current input data and historical data to the memory cell state value, respectively. The many gates selectively control information passing, allowing the network learning to properly forget the old knowledge and update the cells in accordance with the new information. Figure 2 illustrates its basic structure.

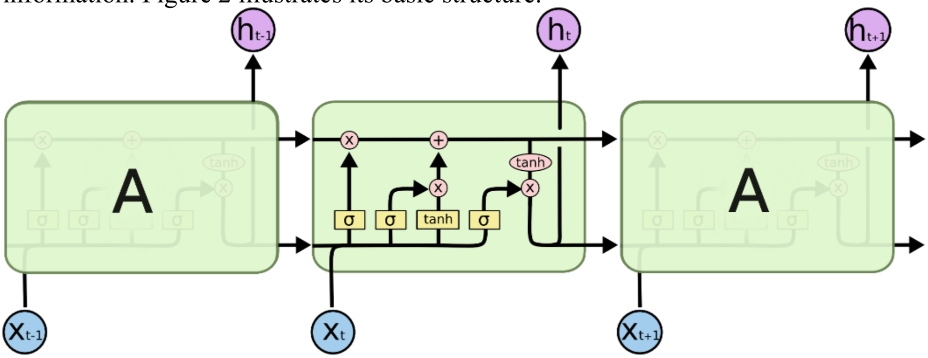
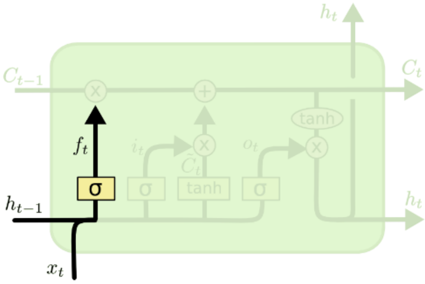


Fig. 2. The LSTM units [11]

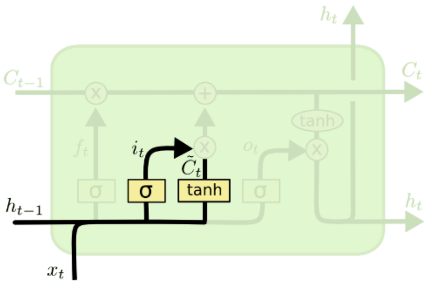
The calculation process of the LSTM unit is divided into the following steps. As shown below (figure 3 shows the calculation process of the forget gate; figure 4 shows the calculation process of input gate and candidate cell state at time t; figure 5 shows the update of the cell state and figure 6 shows the forward of output).

(1) At time t, respectively for the value of candidate memory unit $\tilde{C}t$, the value of the input gate it and the value of the forgetting gate ft are calculated, the calculation process and formulas are shown as below.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Fig. 3. The process of forget gate [11]



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Fig. 4. The process of input gate and candidate cell state [11]

where W_C, W_i, W_f are the corresponding weight matrices; b_C, b_i and b_f are the corresponding biases; h_{t-1} is the forward of output value of the former LSTM unit; While xt is the memory cell value at time t; $\sigma$is the sigmoid function.
sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3}$$

(2) Multiply the old state with the forget gate information and discard some information, add the input gate and the candidate memory unit value, and obtain the value Ct of the memory unit at the current moment, the formula is:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

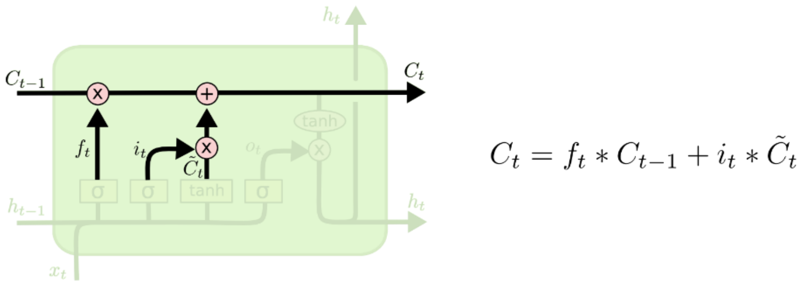


Fig. 5. The process of memory cell [11]

Then, determine o_t by passing through the output gate, and determine the output part h_t. The formulae are shown below.

$$o_t = \sigma W o h_{t-1}, x_t + b_o \tag{5}$$

$$ht = ot * \tanh(ct) \tag{6}$$

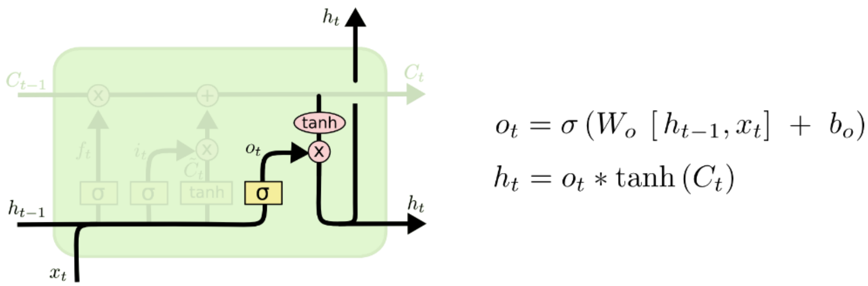


Fig. 6. The forward of cell state [11]

3 Model building

3.1 Linear regression

Firstly, after preprocessing to the data set, add five dimensions to the data set, which represent the Close price at the previous five moments. After analysis, it is concluded that the new-added five dimensions are highly linearly correlated to the Close price of the day, as shown in the left of Figure 7.

However, according to the analysis, the transaction volume at the previous moment is not turn out to be linearly correlated to the current Close price, so it is not considered. All datasets are not normalized.

For the data set of Ethereum, the same process was conducted as the process of Bitcoin.

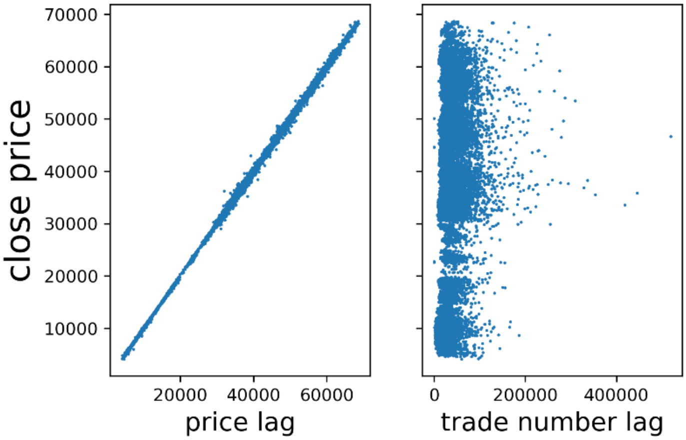


Fig. 7. The correlations between different attributes [original]

3.2 Long Short Term Memory (LSTM)

The model uses 6 features as parameters, which are Open, High, Low, Close, Volume and Number of Trades. The initial learning rate of the model is set to be 0.01, and the optimizer uses "ADAM" (adaptive momentum) optimizer. In order to be consistent with the linear regression model, the algorithm forecasts the next day's close price using data from the previous five days.

Model with two layers of hidden layers. The first layer has 150 computational units and the second layer has 50 computational units. 30 is the maximum number of iterations allowed (30 epochs).

4 Evaluation of the models

In purpose of assessing the performance of the models comprehensively, different methods of evaluation should be applied on the model.

4.1 Assessment method

Common assessment methods include Mean Squared Error, Root Mean Squared Error, Mean Absolute Error and R-squared value.

The dimension of Mean Squared Error is at a higher rank than the original data, so it is not that intuitive to use this measure to evaluate the models. The RMSE value is more sensitive to outliers, as stated by Chai (2014) [7]. The prediction should not contain too much outliers, hence the model accepts RMSE as its metric. R-squared value can be a good evaluation of the fitting, according to Cameron (1997) [8]. In this part, the model mainly uses RMSE and R-squared value as evaluation standards, the following is the calculating formula:

$$\mathrm{SSR} = \sum_{n=1}^{N} \left(\mathrm{y_n} - \overline{\mathrm{y}}_\mathrm{n}\right)^2 \tag{7}$$

$$\mathrm{SSE} = \sum_{\mathrm{n}=1}^{\mathrm{N}} (\mathrm{y_n} - \hat{\mathrm{y}}_\mathrm{n})^2 \tag{8}$$

$$\mathrm{SST} = \mathrm{SSR} + \mathrm{SSE} = \sum_{\mathrm{n}=1}^{\mathrm{N}} (\mathrm{y_n} - \overline{\mathrm{y}})^2 \tag{9}$$

$$\mathrm{R}^2 = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}} \tag{10}$$

$$\mathrm{RMSE}\ (y_n, \hat{y}_n) = \sqrt{\frac{\sum_{n=1}^{N} (y_n - \hat{y}_n)^2}{N}} \tag{11}$$

4.2 Model evaluation and prediction visualization

As shown in table 1 and Figure 8, the result can be seen. Figure 8 compares the outcomes of the linear regression model with the actual value in the upper half of the graph, and the LSTM model's prediction outcomes with the actual value in the lower part. It can be seen that both models perform excellently in fitting the real data, but the linear regression model predicts obvious lag, and this problem is significantly improved in the LSTM model. As can be seen from table 1, the linear regression model exhibits lower RMSE and a slightly higher R-square value, but the lag of its predictions is much higher than that of the LSTM model. This problem can have a huge negative impact on the meaning of model predictions.

Table 1. Evaluation using two methods [original]

	R^2	RMSE
Linear Regression – BTC	0.9998	195
Linear Regression – ETH	0.9998	14.5
LSTM – BTC	0.9995	207
LSTM – ETH	0.9995	14.6

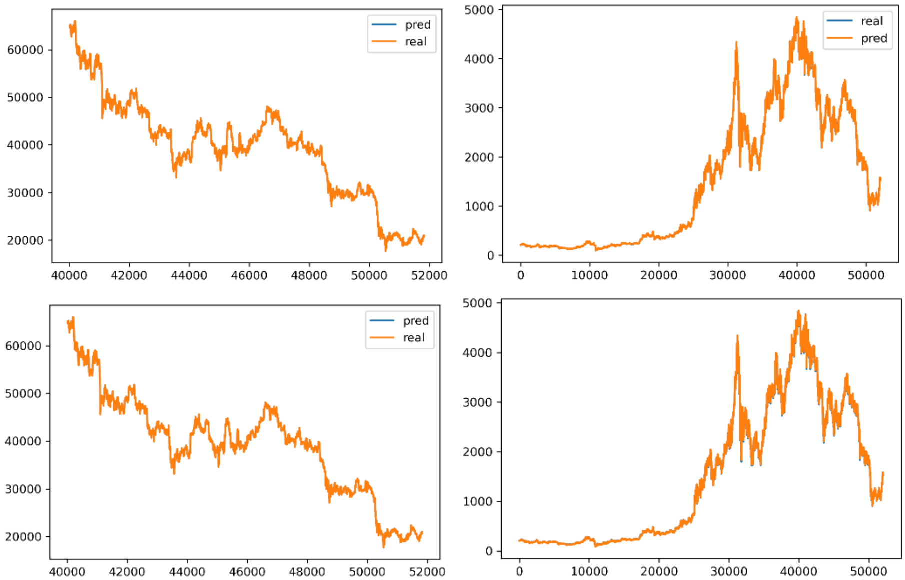


Fig. 8. The visualization of fitting results [original]

5 Discussion

In term of the R-squared value and RMSE value, the prediction made by Linear Regression turns out to be more accurate. For the R-squared value closer to 1 and lower RMSE values in predictions of both kinds of Cryptocurrency.

Through the visualization, a more obvious lag-behind can be observed in the fitting of Linear Regression, which is a crucial disadvantage in practical use.

The prediction is based on only previous prices, so the results turned out the problem of lag. In further research, more peripheral parameters could be taken into consideration.

6 Conclusions

Two models established in this paper provide some prediction ideas for the price of Bitcoin. Using this idea, the improved model in the future should be able to obtain more accurate prediction results. The insights may aid in the decision-making of governmental regulatory bodies in multiple countries. They may also assist Bitcoin investors in lowering their investment risks and obtaining steady profits. Particularly, it may be used to various financial time series forecasts as evidenced by the verification of the impact of the aforementioned models on Bitcoin price prediction. It has a wide range of applications, including commodity spot futures pricing, real estate buying and selling rental prices, and asset securitization risk management.

Due to the highly nonlinear and non-stationary characteristics of Bitcoin price, the traditional linear regression model is difficult to deal with and shows great hysteresis. While LSTM has advantages compared with traditional prediction models, but only uses a single LSTM model for prediction. However, the accuracy of single LSTM model is limited, and lag problem is still significant, hence, an improved model is expected. Because of the difficulty in capturing multi-scale periodic features, there is still much room for improvement in the final accuracy. Ma (2021) proposed a mixed LSTM model to improve the prediction of charge stations occupancy [9]; Cheng (2020) uses mixed LSTM model to analysis the sentiment of language [10]. Both of them gained more remarkable results than single LSTM model by using mixed models.

References

1. Donaldson R G, Kamstra M. Neural network forecast combining with interaction effects [J]. Journal of the Franklin Institute, 1999, 336(2):227-236.
2. Takeuchi L, Lee Y. Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks. 2013.
3. S Zhang, T S Xu. Stock Market Prediction with Improved BP Neural Network [M]. 2017.
4. Sutskever I, Vinyals O, Q V Le. Sequence to Sequence Learning with Neural Networks[C]// NIPS.MIT Press, 2014.
5. J Li, H Zhang, X Y Cai, et al. Towards end-to-end speech recognition for Chinese Mandarin using long short-term memory recurrent neural networks [C]// Interspeech 2015. 2015.
6. Roondiwala M, Patel H, Varma S. Predicting Stock Prices Using LSTM [J]. International Journal of Science and Research (IJSR), 2017, 6(4).
7. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE). Geosci. Model Dev. Discuss.2014,7, 1525–1534.
8. Cameron A C, Windmeijer F. An R-squared measure of goodness of fit for some common nonlinear regression models [J]. Journal of Econometrics, 1997, 77(2):329-342.
9. T Y Ma, Faye S. Multistep Electric Vehicle Charging Station Occupancy Prediction using Mixed LSTM Neural Networks [J]. 2021.
10. K Cheng, Zhili, et al. Chinese Aspect-Level Sentiment Analysis CNN-LSTM Based on Mixed Vector [M], 2020.
11. Christopher O. Understanding LSTM Networks, 2015.