Bitcoin Value Prediction

Longtian Ye, Jianghao Wu, Shiqi Zheng, Zenan Wang

December 10, 2022

1 Introduction

The use of cryptocurrencies, such as Bitcoin, has increased significantly in recent years. As a result, there is a growing interest in understanding and predicting their prices. The price of Bitcoin is known to be highly volatile, making it a challenging task to predict. Traditional methods, such as linear regression, have been applied to Bitcoin price prediction, but have shown limited success. On the other hand, neural networks such as RNN and LSTM models have been shown to be effective at modeling time series data with long-term dependencies. In this paper, we leverage the use of supervised machine learning models for predicting the weighted price of Bitcoin.

2 Dataset & Data Cleaning

To predict the weighted price of Bitcoin, we trained several models on a dataset containing the daily weighted prices of Bitcoin from January 1, 2012 to March 31, 2021. The dataset was obtained from Bitcoin Historical Data on Kaggle[1]. The dataset involved 8 distinct features related to each transaction:Timestamp: Start time of time window (60s window), in Unix time, Open: Open price at start time window, High: High price within time window, Low: Low price within time window, Close: Close price at end of time window, Volume_(BTC): Volume of BTC transacted in this window, Volume_(Currency): Volume of corresponding currency transacted in this window, Weighted_Price: VWAP- Volume Weighted Average Price. Preprocessing a dataset for input into a machine learning model involves cleaning the data, transforming it to a consistent scale, and selecting relevant features. In the case of a dataset containing the daily weighted prices of Bitcoin, the following preprocessing steps were applied:

1. Remove any rows with missing or incorrect values.
2. Relabel the timestamp and combine the data from the same date. Also, exclude the data before 2017/01/01 since the Bitcoin price before 2017/01/01 is low and stable, which is not helpful for predicting the future price because the Bitcoin price had a sharp increase over the recent two years.
3. Transform the data to a consistent scale by MinMaxScaler from the sklearn.
4. Select relevant features e.g, weighted price.

3 Methodology

3.1 Training and Testing

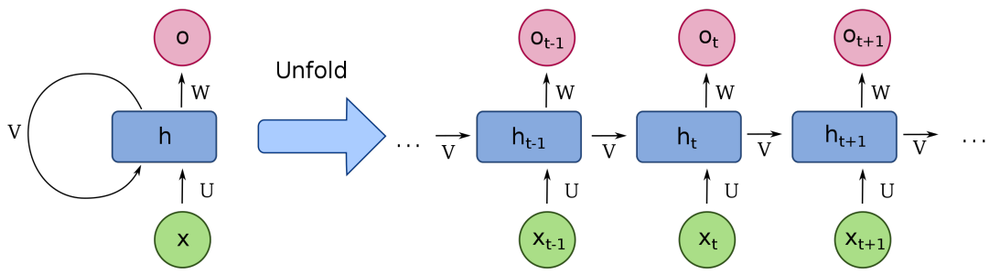
In order to train our chosen models on historical data for Bitcoin, it is vital to split the data into a training set and a test set. The training set is used to train the model, while the test set is used to evaluate the performance of the trained model. To split the data, it is first necessary to compile a dataset of historical prices for Bitcoin. The training set has been set to the data from 2017/01/01 to 2020/12/30 since the Bitcoin price had a sharp increase in the fourth season in 2020 and we want the trend to be learned by the machine. The data after 2020/12/31 is used as the testing set to test how well the model predicts the Bitcoin price.

3.2 Linear Regression

We start to predict the weighted price of Bitcoin by using a linear regression model. To train the linear regression model, we used the same dataset containing the daily weighted prices of Bitcoin from 2017/01/01 to 2021/12/31. We split the dataset into a training set, which was used to fit the model, and a test set, which was used to evaluate the model's performance. The linear regression model was implemented using the scikit-learn library in Python. The model was trained using the ordinary least squares method, which minimizes the sum of the squared residuals between the predicted and actual values.

3.3 Recurrent Neural Network (RNN)

In addition to the linear regression model, we also investigated the use of an RNN model for predicting the weighted price of Bitcoin. We constructed an RNN model consisting of 5 RNN layers and a dense layer. The RNN layers are connected in a sequential manner, with the output of each layer serving as the input for the next layer. The first RNN layer has 32 dimensions, the second has 64 dimensions, and the third hits a 128 dimensions. The output layer is a dense layer with a single unit, which outputs the predicted weighted price of Bitcoin. The model is trained using the Adam optimizer with a learning rate of 0.001 and a mean squared



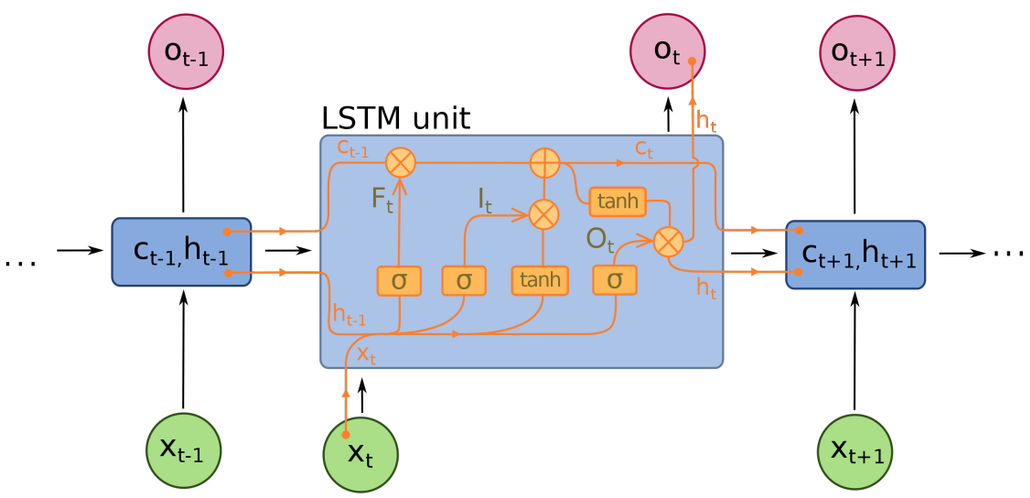
[2]

Figure 1: A typical RNN architecture

error loss function. The model is trained for 50 epochs with a batch size of 32. This means that the model weights are updated 32 times per epoch, and the training process is repeated for a total of 50 epochs.

3.4 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) model is a type of recurrent neural network (RNN) that is designed to capture long-term dependencies in time series data. We constructed the LSTM architecture following the



[3]

Figure 2: A typical LSTM architecture

Keras library instructions with Tensorflow which is composed of a series of LSTM layers and an output layer. The LSTM layers consist of multiple LSTM cells, which are responsible for learning and remembering the long-term dependencies in the data. The output layer is a dense layer, which is a fully connected layer that outputs a single value. The output value is the predicted weighted price of Bitcoin, based on the input data and the learned dependencies in the LSTM layers. All these layers are connected in a sequential manner. The model was trained using the Adam optimizer with a learning rate of 0.001 and a mean squared error loss function. The model was trained for 50 epochs with a batch size of 32.

4 Results

To evaluate the model, the predicted prices for Bitcoin can be compared to the actual market prices for the same time period. This comparison can be used to calculate a variety of metrics, such as the mean squared error (MSE). MSE is a measure of the difference between the predicted and actual Bitcoin weighted prices. A lower MSE indicates a better performance, with a value of 0 indicating a perfect prediction. The results showed that the linear regression model achieved an MSE of 31878.33 on the test set. On the other hand, the RNN achieved a MSE of 17961.22 which outperformed our first attempt with linear regression. Nonetheless, we could still capture the big gap between the actual and predicted price, which could still incur problems for investors who try to use the model as a reference.

Fortunately, this problem was effectively tackled through the use of LSTM model. Indeed, the model was able to effectively predict the weighted price of Bitcoin. On the test set, it achieved a MSE of 3455.97. That is, LSTM's error in accurately predicting the weighted price was very small and far exceeding our expectations, suggesting that the model was able to capture the underlying trends in the data. Consequently, we were able to arrive at a safe conclusion that the LSTM model is the most effective at predicting the weighted price of Bitcoin than the former two. Again, the underlying reason for this is that the linear regression and the RNN model do not take into account the time dependence of the data, whereas the LSTM model is specifically designed to model time series data with long-term dependencies. Additionally, the linear regression model only considers a linear relationship between the dependent and independent variables, whereas the LSTM model can capture more complex relationships.

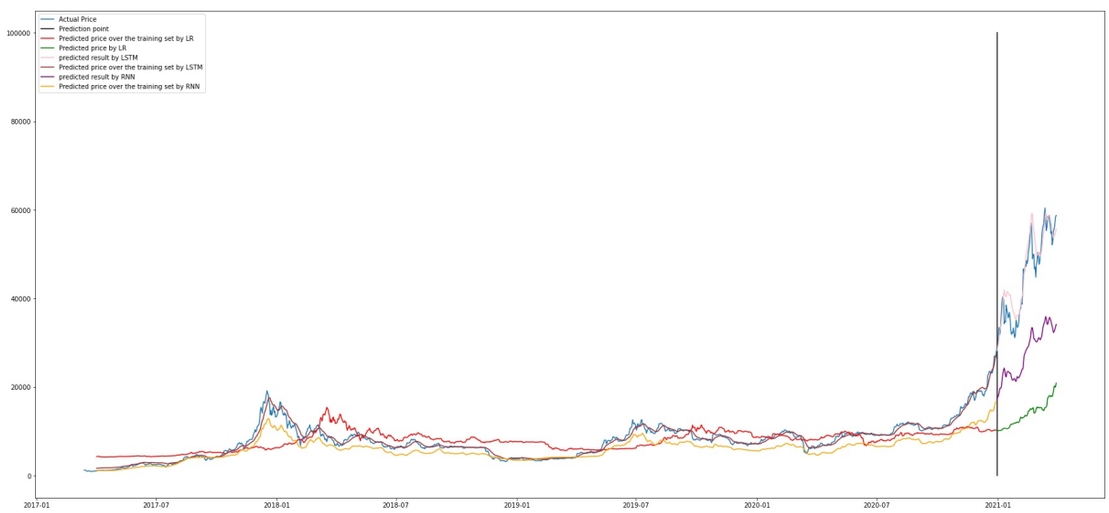


Figure 3: Visualization of the training and testing progress of the selected models, including the original dataset

5 Discussion

Linear regression is a simple algorithm that models the relationship between a dependent variable and one or more independent variables using a linear function. RNNs are a type of neural network that are designed

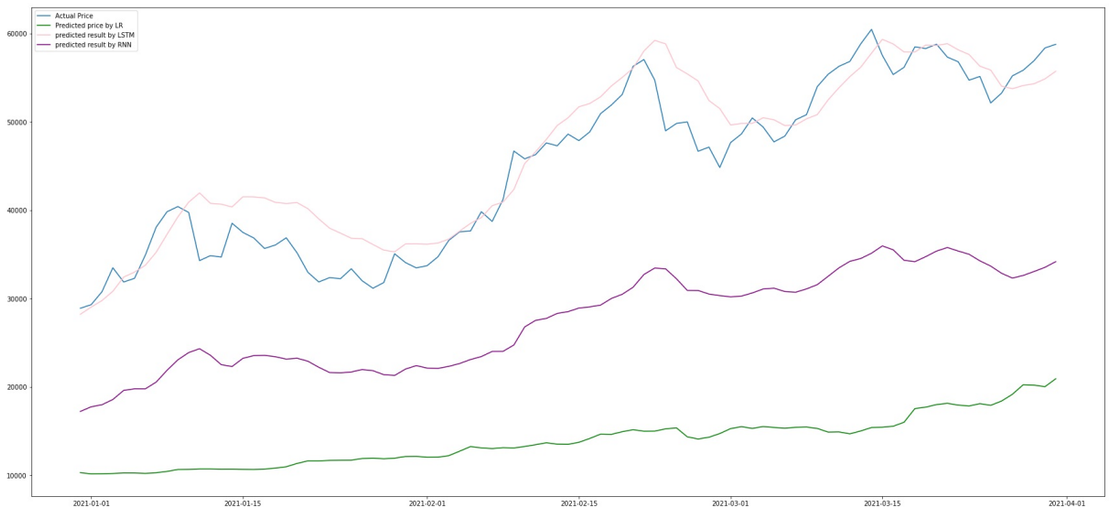


Figure 4: A closer view of results of different models: Linear model, RNN, and LSTM, compared to the ground truth

to process sequential data, such as time series data. LSTMs are a type of RNN that are specifically designed to capture long-term dependencies in sequential data.

When analyzing historical data for Bitcoin, linear regression can be used to model the relationship between the price of Bitcoin and other factors, such as market conditions or the adoption of Bitcoin by merchants. RNNs and LSTMs, on the other hand, are ideal to analyze the price of Bitcoin over time and make predictions about future prices. One advantage of linear regression is that it is relatively easy to understand and interpret, but it is limited in its ability to capture complex non-linear relationships. RNNs and LSTMs, on the other hand, are able to capture complex patterns in sequential data, but they can be difficult to train and require a lot of data to achieve good performance. Overall, linear regression, RNNs, and LSTMs are all viable options for analyzing historical data for Bitcoin, and the choice of which algorithm to use will depend on the specific goals and requirements of the analysis.

In this paper, we showed empirically that the LSTM model is the state-of-the-art model among others for prediction of Bitcoin weighted price. This could serve as a potential reference for speculators in the financial sectors and facilitate their decision-making upon Bitcoin trading. However, we admit that this study has several limitations. Firstly, the dataset only covered a limited time period from 2017 to 2021. In the future, it will be interesting to train the model on a longer time period to see if it can still accurately predict the weighted price of Bitcoin. Secondly, the model was only trained on the daily weighted prices of Bitcoin. Incorporating additional features, such as trading volume and news sentiment, may improve the performance of the model.

6 Conclusion

In conclusion, this study aimed to evaluate the performance of various models employed to predict the weighted price of Bitcoin. Particularly, the LSTM model far exceeded the linear regression and RNN model in terms of accuracy and the capability of capturing the underlying trends in the time series data. Further research is needed to improve the model and evaluate its performance on longer time periods and with additional features.

7 Acknowledgments

For this report, Longtian Ye and Jianghao Wu constructed the linear regression and LSTM model, Zenan Wang and Shiqi Zheng constructed the RNN model. All of us participated in the final report writing. The dataset we use is from Bitcoin Historical Data on Kaggle[1]. We would also like to thank Professor Jorge Silva for amazing lectures in machine learning this semester.

References

[1] Kaggle: Bitcoin Historical Data
https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data?resource=download

[2] Wikipedia: Compressed (left) and unfolded (right) basic recurrent neural network
https://en.wikipedia.org/wiki/Recurrent$_neural_network/media/File:Recurrent_neural_network_unfold.svg$

[3] Wikipedia: Long short-term memory unit
https://en.wikipedia.org/wiki/Recurrent$_neural_network/media/File:Long_short-Term_Memory.svg$