Journal of Mathematical Finance, 2020, 10, 132-139
https://www.scirp.org/journal/jmf
ISSN Online: 2162-2442
ISSN Print: 2162-2434

Bitcoin Price Prediction Based on Deep Learning Methods

Xiangxi Jiang

Barstow School of Ningbo, Ningbo, China
Email: xavier194152@163.com

How to cite this paper: Jiang, X.X. (2020) Bitcoin Price Prediction Based on Deep Learning Methods. *Journal of Mathematical Finance*, **10**, 132-139.
https://doi.org/10.4236/jmf.2020.101009

Received: August 30, 2019
Accepted: February 8, 2020
Published: February 11, 2020

Abstract

Bitcoin is a current popular cryptocurrency with a promising future. It's like a stock market with time series, the series of indexed data points. We looked at different deep learning networks and methods of improving the accuracy, including min-max normalization, Adam optimizer and windows min-max normalization. We gathered data on the Bitcoin price per minute, and we rearranged them to reflect Bitcoin price in hours, a total of 56,832 points. We took 24 hours of data as input and output the Bitcoin price of the next hour. We compared the different models and found that the lack of memory means that Multi-Layer Perceptron (MLP) is ill-suited for the case of predicting price based on current trend. Long Short-Term Memory (LSTM) provides relatively the best prediction when past memory and Gated Recurrent Network (GRU) is included in the model.

Keywords

Deep Learning Model, Multi-Layer Perceptron, Gated Recurrent Network, Long Short-Term Memory, Cross-Validation, Normalization

1. Introduction

Bitcoin is a cryptocurrency and a form of electronic cash. It is a digital currency that can be sent from user to user on the peer-to-peer Bitcoin network without intermediaries. It keeps a record of trading among peers and every record is encrypted. Each new record created contains the cryptographic hash of a previous block. Each record contains a timestamp and the data of the sender, the receiver, and the amount. Given Bitcoin is an emerged technology, few predictions is made on Bitcoin future value. Greaves and Au used linear regression, logistic regression and support vector machine to predict Bitcoin future price with low performance [1]. Indira *et al.* proposed a Multi-layer Perceptron based non-linear autoregressive with External Inputs (NARX) model to predict Bitcoin price of the next day [2]. Jakob Aungiers proposed a long-short term memory deep neural networks to predict S & P 500 stock price [3]. His research sheds light on Bitcoin prediction which is similar to stock price. Madan *et al.* used more machine learning approaches like generalized linear models and random forest to address Bitcoin prediction problem [4].

Researches mentioned above focuses on predicting the Bitcoin price of the next day. However, Bitcoin is traded frequently in a much smaller interval. In this research, we try to use historical data to predict next hour's price instead of next day's price which may have better application in real world. First we implemented data normalization like min-max normalization and normalization with window [5] where the data is normalized based on the window's initial value and the percentage of change. Multiple Layer Perceptron (MLP), Long-Short-Term-Memory (LSTM) and Gated recurrent units (GRU) models are compared on the test dataset with cross-validation.

2. Dataset Exploration

Data used in this research is collected from Kaggle [6]. Bitcoin data from Jan 2012 to July 2018 is collected. It has a timestamp, the value at Open, High, Low, Close, the volume traded in Bitcoin and USD, the weighted price and the date. This research focuses on predicting Bitcoin price in the future hour by using the price of past 24 hours, so only the timestamp and the weighted price are used in the model.

3. Pre-Processing

As shown in **Figure 1**, the dataset is by minute, and contains around 3,409,920 points. Since we predicted the price by hours, we have had 1,409,920/60 which is 56,832 datapoints. The dataset is further split into training, validating and testing sets. As shown in **Figure 2**, training data takes up to 80% of the entire dataset, and validating and testing 10% respectively. As the time series data, samples are not randomized. We used the first 24 hours' Bitcoin price as input to predict the next hours' Bitcoin price. Several other pre-processing methods are implemented to improve data processing and model convergency efficiency. Mini-batch is used to split large data into small batches, which improves memory efficiency. Minimum-Maximum normalization and window-based normalization is used to set the whole training dataset to (−1, 1) scale. Window normalization is based on the reference of stock market. The normalization methods will take each sized window and normalize each one to reflect percentage changes from the start hour of the window [3].

4. Models

Deep learning network is a type of computer modeling that finds the pattern within the given datasets and categorize the input accordingly. There are many

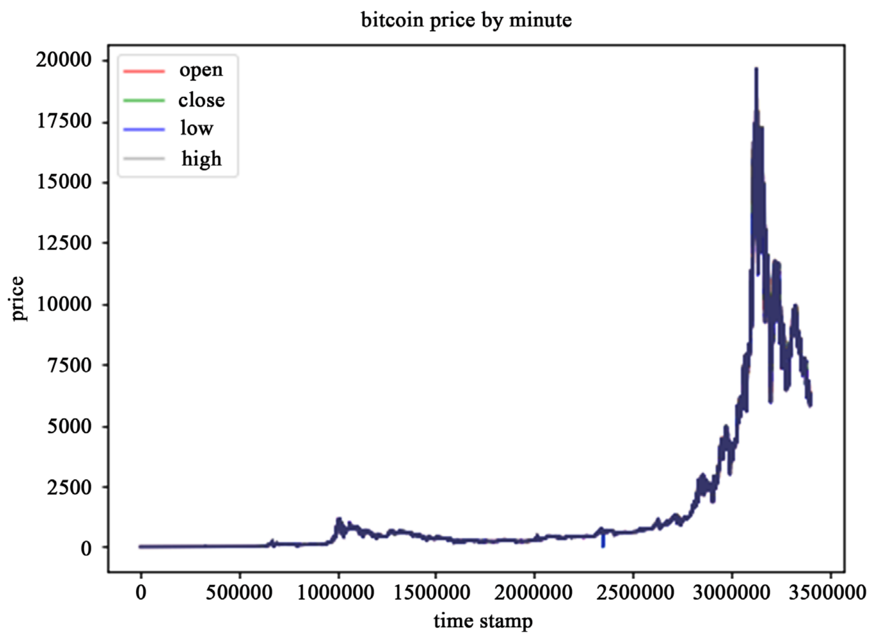

Figure 1. The overview of data listed by minutes.

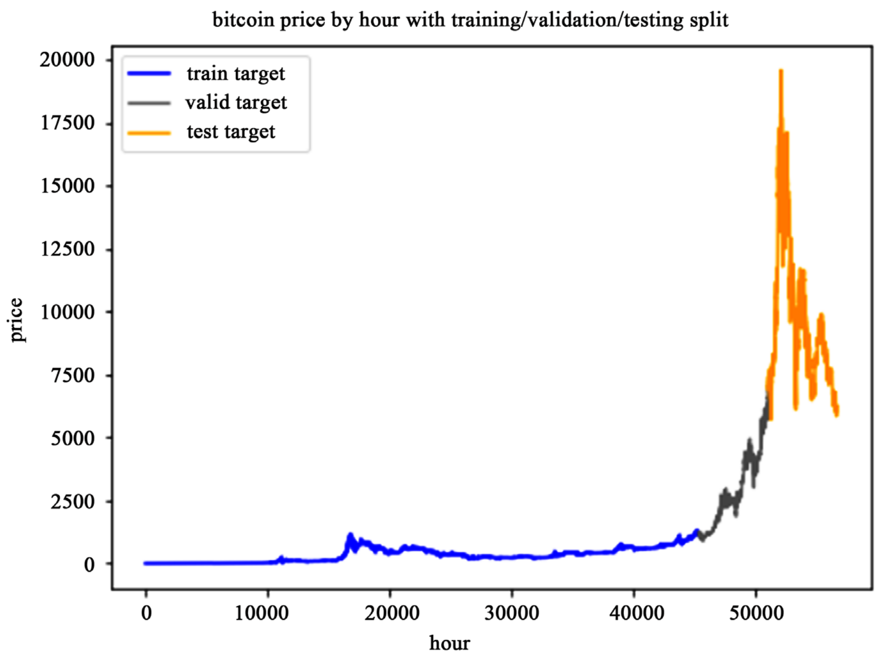

Figure 2. Training, validating and testing dataset.

different structures for deep learning network, including Multiple Layer Perceptron (MLP) that has a linear activation function, Recurrent Neural Network (RNN) that records a separate hidden unit to influence the next calculation. Extensions of RNN include Long Short-Term Memory (LSTM) and Gated Recurrent Model (GRU).

MLP is a basic method in prediction. It reads all input with no ordering and then determine the relationship between the independent variables and the dependent variables. Hidden layers can be added between the input layer and the output layer together with the activation function, to better describe the non-linear relationship.

RNN is a group of method to calculate products from previous result of the model and new input data. In fact, it is better to MLP that it has "experience" from last calculations that will influence its calculations. The "experience" is gained from the model, is kept privately but is allowed to pass onto the next model. This private variable is called the hidden state and is passed on from the current calculation to the future calculation. It determines independently the output of the model, apart from the algorithm itself. However, RNN model depends on the continuous flow, which is sequential like the time series, in order to input data for the training. If the pattern repeats only over the long term, the previous repetition may be not influential enough to affect it at the next repetition. It also requires the data to be in order of time. Therefore determines, unlike MLP, RNN cannot be given random samples.

Long Short-Term Memory solves the issue that the diminished influence of distant events on the RNN network. It has a switch that can choose certain events to remember. It also is not long-term dependent and doesn't require as much training. It has four layers to determine the output, then passes the hidden state with the result to the next cycle. "Forgetting gates" exists in addition to four layers to determine if the experience should not be counted. Four layers and forgetting gates can be given different information to focus on either short-term or long-term memory.

GRU or Gated Recurrent Model is considered as one of the simpler model compared to the LSTM model, combination of the "forget" step with the "input" step into one, and as a result, requires only one hidden unit.

Among the three methods, MLP is mostly credited with its simplicity and the need for less computational power. They have the same amount of information as input. However, the number of hidden layers and the hidden units are more magic numbers. Some number turns out to work well especially, while some may turn out to be just the opposite. RNN accounts on the previous model through the hidden unit. The value uses in the calculation but does not need intervention. It can be very accurate, given the fact that the model has a large training set. However, long term patterns cannot be memorized and this may result in inaccuracy, especially when rapid changes take place in recent years. LSTM can choose whether it should "forget" previous states. Therefore, it is better capable of dealing with data that has repetitive trend over a long time. GRU model is also able to choose whether it should recall previous experience, but it is capable of learning more rapidly and need a bit less resource.

Six models are compared in this research. The model setups are listed in the following Table 1 and training results will be discussed in the next part.

5. Results

As shown in Figure 3, in MLP and RNN frameworks, we find the similar

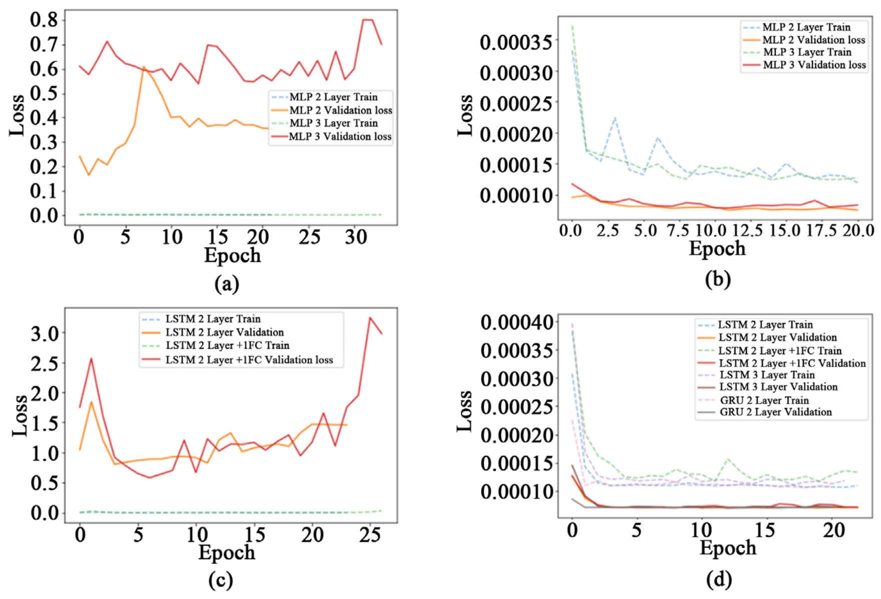

Figure 3. Training performance of models. (a) MLP with whole-dataset-normalization; (b) MLP with window-normalization; (c) RNN with whole-dataset-based normalization; (d) RNN with window-normalization.

Table 1. Font sizes of headings. Table captions should always be positioned above the tables.

Model	Layer Information
2 Layer MLP	[256, 256]
3 Layer MLP	[256, 128, 64]
2 Layer LSTM	[256, 256]
2 Layer LSTM + 1Fully Connected layer	[256, 256] + [128]
3 Layer LSTM	[256, 256, 128]
2 Layer GRU	[256, 256]

conclusion that window-based normalization is much better than whole-dataset-based normalization. Because of time-series data feature, the RNN frameworks converge faster than MLP methods. Model performance in this research is evaluated by Root Mean Square Error (RMSE) of the predicted price and the true price of the dataset. The results are listed in the following table. As shown in Table 2, normalization by window method performs much better.

We visualize the predicted price in the test dataset against true values in Figure 4 and zoom in to have a closer look at the predicted price in Figure 5. We can find that LSTM with normalization by window is the best combination.

A ten-fold cross-validation is conducted on all the models. As shown in Figure 6, we can see that the error goes down after the training set is enlarged. At the end of the time-series data, when the fluctuation goes high, the error goes up a little again. Based on the cross-validation results, as summarized in Table 3, 2 layers of GRU is the best, and 2 layers of LSTM are very close to the performance.

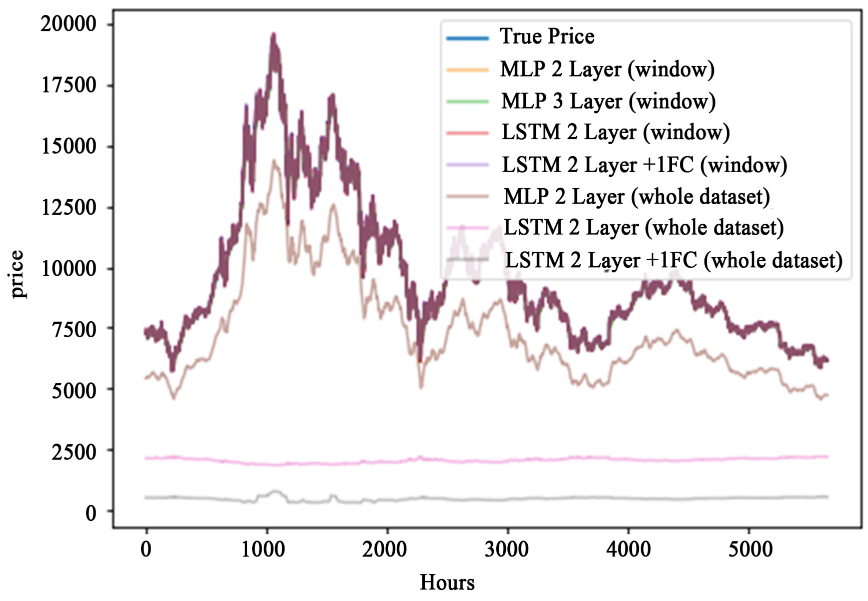

Figure 4. Predicted price on the test dataset.

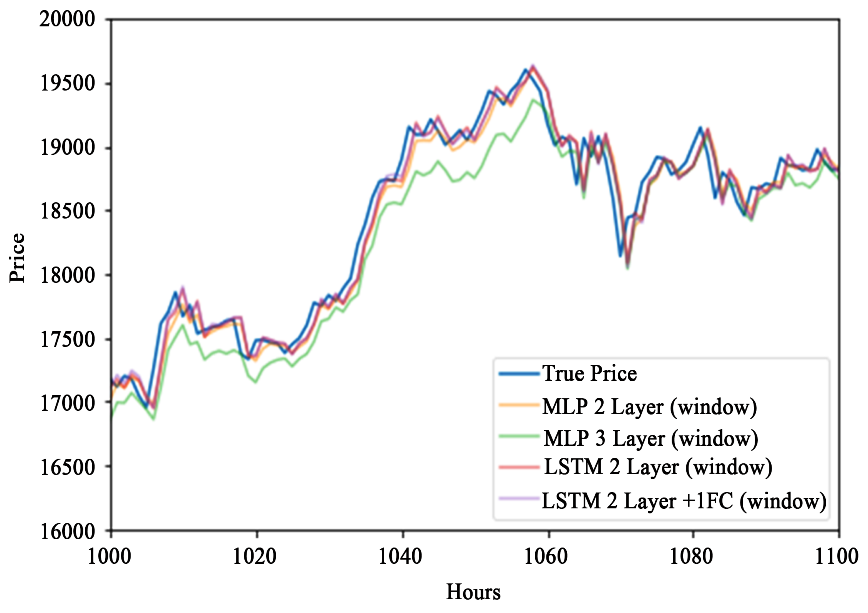

Figure 5. Zooming in.

Table 2. RMES of six models by different normalization methods.

	Normalized By Window	Normalized Whole Dataset
2 Layer MLP	131.023	2541.128
3 Layer MLP	156.922	3692.356
2 Layer LSTM	125.387	8312.999
2 Layer LSTM + 1FC	126.016	9187.938
3 Layer LSTM	125.414	Not tried
2 Layer GRU	126.512	Not tried

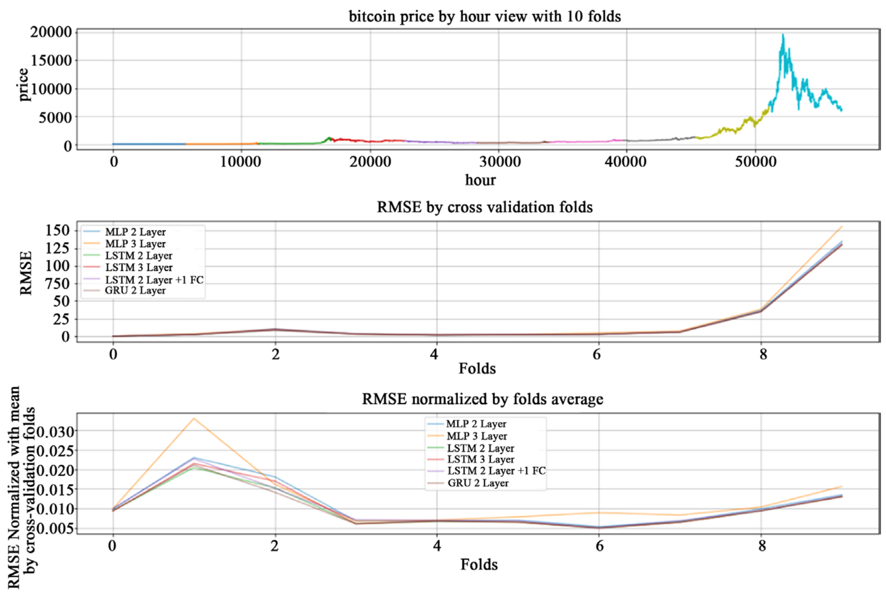

Figure 6. Cross validation results. The top one is the 10-fold split of original data, the middle one is the average RMSE for each fold, the bottom one is the RMSE/average price in that fold.

Table 3. Summarize of cross-validation results.

	Mean RMSE	Std RMSE
2 Layer MLP	20.093	39.630
3 Layer MLP	22.736	45.711
2 Layer LSTM	19.121	38.214
2 Layer LSTM + 1FC	19.486	38.808
3 Layer LSTM	19.250	38.177
2 Layer GRU	19.020	38.146

6. Conclusion and Discussion

According to cross-validation results, 2 layers of LSTM has the best performance on the original test dataset and 2 layers of GRU is the best. All six models have close performance, so different models may be preferred in different scenarios. MLP model requires less computing power while it has slightly lower performance than RNN model. Our study combines several unique features, including the hour-based prediction, the usage of data from the past 24 hours, normalization by window and the comparison of different types of model with different amounts of layers. Based on this research, future work can be done on predicting a sequence of estimation so that it can be applied in more common Bitcoin trading scenarios.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Alex, G. and Au. B. (2015) Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin.

[2] Indera, N.I., Yassin, I.M., Zabidi, A. and Rizman, Z.I. (2017) Non-Linear Autoregressive with Exogeneous Input (NARX) Bitcoin Price Prediction Model Using PSO-Optimized Parameters and Moving Average Technical Indicators. *Journal of Fundamental and Applied Sciences*, **9**, 791-808. https://doi.org/10.4314/jfas.v9i3s.61

[3] Aungiers, J. (2018) Time Series Prediction Using LSTM Deep Neural Networks. https://www.altumintelligence.com/articles/a/Time-Series-Prediction-Using-LSTM-Deep-Neural-Networks

[4] Isaac, M., Saluja, S. and Zhao. A. (2015) Automated Bitcoin Trading via Machine Learning Algorithms. http://cs229.stanford.edu/proj2014/Isaac%20Madan,%20Shaurya%20Saluja,%20Aojia%20Zhao,Automated%20Bitcoin%20Trading%20via%20Machine%20Learning%20Algorithms.pdf

[5] Pedregosa, F., *et al.* (2011) Scikit-Learn: Machine Learning in Python. *Journal of machine learning research*, **12**, 2825-2830.

[6] Zielak. (2019) Bitcoin Historical Data, Bitcoin Data at 1-Min Intervals from Select Exchanges, Jan 2012 to July 2018, Version 14. https://www.kaggle.com/mczielinski/Bitcoin-historical-data