Predicting Crypto Price Trends

Qiuyang Wang
CU Boulder
qiwa8995@Colorado.edu

Xinyu Jiang
CU Boulder
xiji6874@colorado.edu

Yan Zhan
CU Boulder
yan.zhan@colorado.edu

1 Problem Space

Ever since its launching in 2008, Bitcoin has become increasing popular among investors. As the most famous cryptocurrency, Bitcoin as also drawn crucial attention due to its volatile price change. The factors affecting cryptocurrency prices can be divided into internal factors and external factors. The former are driven by the demand and supply of cryptocurrency and the later contains the attractiveness of the cryptomarket, the macro-financial situation and political influences.

Researchers from different fields have been trying to predict the price of Bitcoin with various methods. Jang and Lee (2018) shows that Bayesian Neural Networks can make a good prediction in Bitcoin price time series. Atsalakis et al. (2019) proposes a technique that uses a hybrid Neuro-Fuzzy controller to forecast the direction in the change of the daily price of Bitcoin. Mallqui and Fernandes (2019) compares the behavior of Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Ensemble algorithms for price direction predictions. Wu et al. (2018) adopts Long short-term memory (LSTM) as a new forecasting framework for cryptocurrency price prediction.

In this project, we follow Wu et al. (2018) and try to predict the price of Bitcoin in a short-term future with LSTM. As a time series approach, LSTM outperform others such as Autoregressive (AR), univariate Moving Average (MA) and Autoregressive Integrated Moving Average (ARIMA) in terms of the Bitcoin price prediction. Those methods are more suitable for data with seasonal trends pattern which is not observed in Bitcoin price. LSTM is favored over other time series method due to the temporal nature of the more advanced algorithms.

We summarise the researches that using LSTM as follows. Guo et al. (2018) study the ability to make the short-term prediction of the exchange price fluctuations towards the US dollar for the Bitcoin market and perform experiments to evaluate a variety of statistical and machine learning approach. McNally et al. (2018) tried to ascertain with what accuracy the direction of Bitcoin price in USD can be predicted and compare the results obtained using RNNs, LSTM and ARIMA models. Yamak et al. (2019) compared the ARIMA time series model with the LSTM deep learning algorithm and Gated Recurrent Unit (GRU) for Bitcoin price prediction. Their results shows that ARIMA gives the best results at 2.76% and 302.53 for MAPE and RMSE respectively. The Gated Recurrent Unit (GRU) however performed better than the Long Short-term Memory (LSTM), with 3.97% and 381.34 of MAPE and RMSE respectively.

Overall, the research using LSTM as Bitcoin price prediction is under development and our project plans to contribute to the solution of this problem.

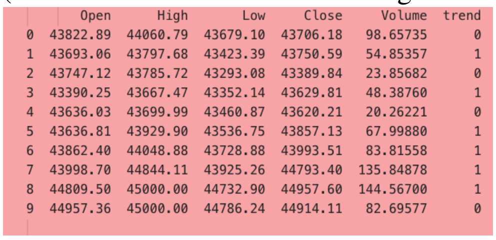
2 Approach

The LSTM networks are composed of an input layer, one or more hidden layers, and an output layer.

Based on the result of the LSTM implementation, without external factors, our model could successfully predict the price of bitcoin. However, because Bitcoin is affected by external factors such as government policies, production costs, supply and demand, etc. For example, according to a report from Chi, Chinese government banned Bitcoin mining in 2020, which caused the price of Bitcoin to drop by 2.9% on that day. Other cryptocurrencies such as Ethereum and Solana were also hugely impacted, Solana dropped by 4.6% and Ethereum by 6.7% in 24 hours after the policy was released. Thus, under the influence of external factors (the prediction is accurate without considering external factors), the accuracy of our model prediction becomes less accurate (the forecast trend of our model is still the same as the trend of Bitcoin price)

Since the predicted price will be affected by many external factors to a large extent, and the predicted trend is indeed correct. We decided to use traditional machine learning methods to directly predict the price trend of Bitcoin.

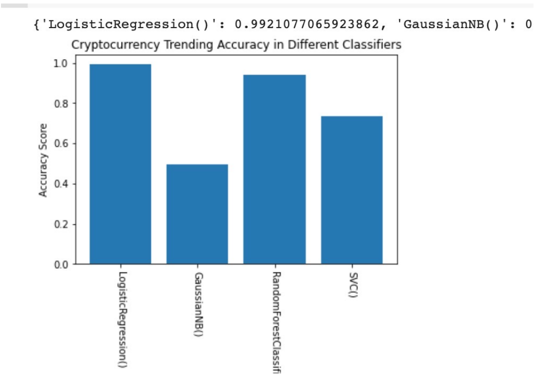
We first add another column of data called "trend"that compares the current Bitcoin price with previous price, if the price goes up we mark it as "1" and if it has fallen, we will mark it as "0" (Below shows the data after adding trending).



```
       Open      High       Low     Close     Volume  trend
0  43822.89  44060.79  43679.10  43706.18   98.65735      0
1  43693.06  43797.68  43423.39  43750.59   54.85357      1
2  43747.12  43785.72  43293.08  43389.84   23.85682      0
3  43390.25  43667.47  43352.14  43629.81   48.38760      1
4  43636.03  43699.99  43460.87  43620.21   20.26221      0
5  43636.81  43929.90  43536.75  43857.13   67.99880      1
6  43862.40  44048.88  43728.88  43993.51   83.81558      1
7  43998.70  44844.11  43925.26  44793.40  135.84878      1
8  44809.50  45000.00  44732.90  44957.60  144.56700      1
9  44957.36  45000.00  44786.24  44914.11   82.69577      0
```

After that, we use the train_test_split method provided by the sklearn library to split the data into 80% training data and 20% testing data. Then we use logistic regression, svm, random forest and naive bayes algorithms to train our data, and compare it with the test data after getting the prediction results. The result is surprisingly good.

	Accuracy
Logistic Regression	0.9921
Naive Bayes	0.4957
Random Forest	0.9391
SVM	0.7370



```
{'LogisticRegression()': 0.9921077065923862, 'GaussianNB()': 0
```

Based on the graph above , logistic regression and random forest classifier all yield pretty good results with over 93% accurate prediction. SVM gets about 74% accuracy and naive bayes 50% accuracy.

However, the output of the above model is not practical, since we are training the model and testing the model with the data in the same month. When we test the regression model with the data in the next month, the accuracy of the model is just over 0.5, and since we were predicting trend(a classification problem), the performance of the regression model was obviously really bad and it could not be used as trend/price prediction. With the failure of predicting trend using regression model, we turned to look at neural network and hope that such model can help us to deliver the task.

After weeks of discussing and seeking, we finally decided to use LSTM as our model and we also decided to change the scope of the project, from predicting trend to predicting crypto price in the next hour. To reach the goal, we also need to reshape and edit the raw data to make sure we can feed the correct data to our model.

In terms of predicting the crypto price in the next hour, we found a way to create the label which acts like the crypto price in the next hour. Then we fit the LSTM model with the preprocessed data and feed test data to the model as well. The model generated the predicted crypto prices in the next hour just after the last close time of the last row of the data. And we validated the predicted labels with our test y, and generated mse, loss, residual plots to evaluate the performance of our model.

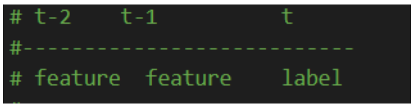
3 Data

We used the data from Binance Data Collection which is a website provides free crypto data sets in daily and monthly with complete time resolutions(minutes, hours, days, months). The we used klines data sets in 1-hour resolution, the time range of the data sets are from December 2020 to November 2021. For the original data sets, each data set contains fixed numbers of features, but we only used close feature(the price of a crypto when the market is closed) as our main feature. Only one column of feature is definitely not enough to train our LSTM model. Therefore, we used 4 crypto close prices in total and we did data preprocessing on these 4 crypto close prices. The data from the website are zipped in month scale, we concatenated the data from December 2021 to

October 2020 as four dataframes for each crypto. Since we want to be able to predict the Bitcoin price in next few hours, we need to find a way to ”create” features and labels so we can train our model and generate the predicted labels we want. Therefore, we took the data whose time ranges are from December 2020 to October 2021 as our features, and we did data preprocessing on these data in terms of get rid of nan and 0 values, select Close(close prices of crypto) column of each dataframe and concat them as a new dataframe for the future usage.

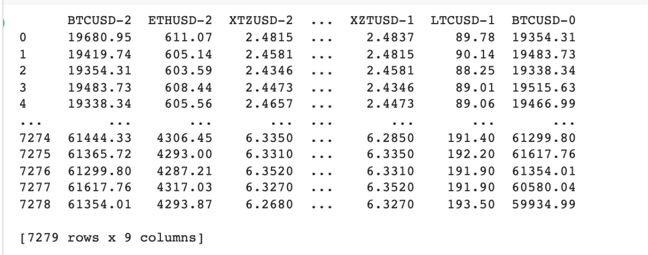
For our new dataframe, we call it ”close prices” it contains 5 features: close time(when the crypto is priced, but we dropped it after we finished feature correlations), BTCUSD price(Bitcoin price in USD), LTCUSD(litecoin price in USD), XTZUSD(tezos price in USD) and ETHUSD(ethereum price in USD). And then, to allow our model to predict the Bitcoin price in next 3 hours, we delete the first row of the features(for LTCUSD, BTCUSD, XTZUSD, ETHUSD) and append one row of corresponding features(from November 2021) then put them into a new dataframe to create the data for [t-1](means the crypto prices in an hour ago) hours. And the the data [t-2](crypto prices in two hours ago) which is already there(the original data), we put them into a new dataframe for later concatenation. As for the crypto prices now[t], we delete the first two rows of data, and append two rows of new data from November 2021 as our label dataframe. Therefore, we have 8 features and 1 label ready for the future training.

This is the timeline of our features and label. Considering t as the time where we are at right now.



```
# t-2     t-1           t
#--------------------------
# feature  feature     label
```

This is what our data looks like after preprocessed. As you can see, xxxUSD-2 are the features of crypto prices in 2 hours ago, and xxxUSD-1 are the crypto prices in an hour ago. And our label BTCUSD-0 are the Bitcoin crypto prices currently.

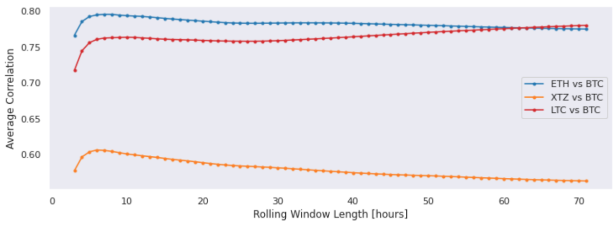


```
      BTCUSD-2  ETHUSD-2  XTZUSD-2  ...  XZTUSD-1  LTCUSD-1  BTCUSD-0
0     19680.95    611.07    2.4815  ...    2.4837     89.78  19354.31
1     19419.74    605.14    2.4581  ...    2.4815     90.14  19483.73
2     19354.31    603.59    2.4346  ...    2.4581     88.25  19338.34
3     19483.73    608.44    2.4473  ...    2.4346     89.01  19515.63
4     19338.34    605.56    2.4657  ...    2.4473     89.06  19466.99
...        ...       ...       ...  ...       ...       ...       ...
7274  61444.33   4306.45    6.3350  ...    6.2850    191.40  61299.80
7275  61365.72   4293.00    6.3310  ...    6.3350    192.20  61617.76
7276  61299.80   4287.21    6.3520  ...    6.3310    191.90  61354.01
7277  61617.76   4317.03    6.3270  ...    6.3520    191.90  60580.04
7278  61354.01   4293.87    6.2680  ...    6.3270    193.50  59934.99

[7279 rows x 9 columns]
```

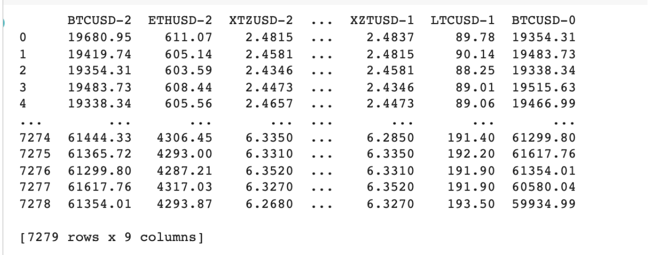
4 Results

4.1 Data Visualization



Based on the closing price sequence of Ethereum (ETH/USD), Tezos (XTZ/USD), and Litecoin (LTC/USD), we hope to predict the closing price of BTC/USD. To correctly find the correlation between each cryptocurrencies, we created plots showing the average correlation for all the crypto currencies with bitcoin. Based on the graph above, we find that Ethereum and Litecoin have strong correlation with the price of Bitcoin and there is no need to consider the sequence of time.

4.2 Feature Selection and Division



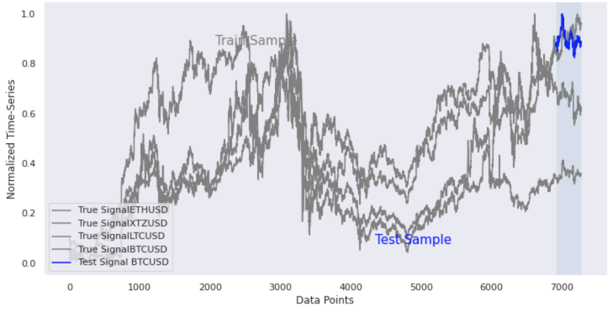
```
      BTCUSD-2  ETHUSD-2  XTZUSD-2  ...  XZTUSD-1  LTCUSD-1  BTCUSD-0
0     19680.95    611.07    2.4815  ...    2.4837     89.78  19354.31
1     19419.74    605.14    2.4581  ...    2.4815     90.14  19483.73
2     19354.31    603.59    2.4346  ...    2.4581     88.25  19338.34
3     19483.73    608.44    2.4473  ...    2.4346     89.01  19515.63
4     19338.34    605.56    2.4657  ...    2.4473     89.06  19466.99
...        ...       ...       ...  ...       ...       ...       ...
7274  61444.33   4306.45    6.3350  ...    6.2850    191.40  61299.80
7275  61365.72   4293.00    6.3310  ...    6.3350    192.20  61617.76
7276  61299.80   4287.21    6.3520  ...    6.3310    191.90  61354.01
7277  61617.76   4317.03    6.3270  ...    6.3520    191.90  60580.04
7278  61354.01   4293.87    6.2680  ...    6.3270    193.50  59934.99

[7279 rows x 9 columns]
```

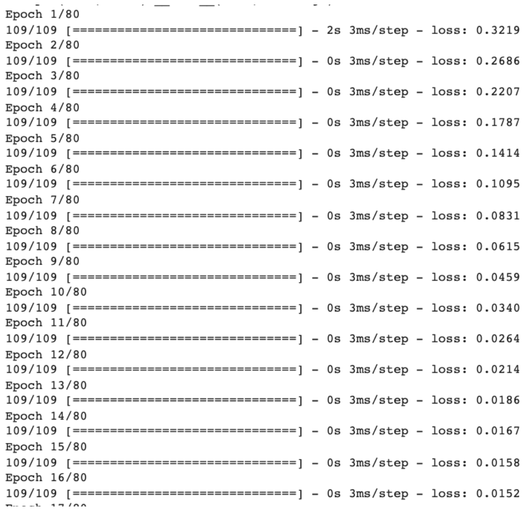
1. To correctly predict the price of BTC, we first divide the features and labels. Through the label, the model will generate the value we want to predict. In the training sample, the label is used for training. To this end, we provide a series of features and display relevant tags. The graph above shows the data after pre-processing.

2. To divide the training data and testing data, we use *traintestsplit* provided by SKlearn library and set the random state to 42. In addition to that, we decided to train our LSTM model on 95 percent of our data, so we set the

testing size to 0.05.Below is the graph of our train-testing splitting results. The grey lines represents data for training, and the blue lines are testing data.

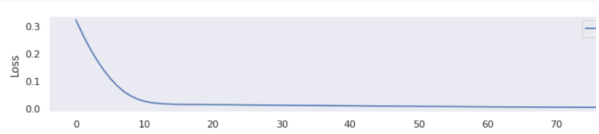


4.3 LSTM Model Design



```
Epoch 1/80
109/109 [==============================] - 2s 3ms/step - loss: 0.3219
Epoch 2/80
109/109 [==============================] - 0s 3ms/step - loss: 0.2686
Epoch 3/80
109/109 [==============================] - 0s 3ms/step - loss: 0.2207
Epoch 4/80
109/109 [==============================] - 0s 3ms/step - loss: 0.1787
Epoch 5/80
109/109 [==============================] - 0s 3ms/step - loss: 0.1414
Epoch 6/80
109/109 [==============================] - 0s 3ms/step - loss: 0.1095
Epoch 7/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0831
Epoch 8/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0615
Epoch 9/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0459
Epoch 10/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0340
Epoch 11/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0264
Epoch 12/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0214
Epoch 13/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0186
Epoch 14/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0167
Epoch 15/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0158
Epoch 16/80
109/109 [==============================] - 0s 3ms/step - loss: 0.0152
```

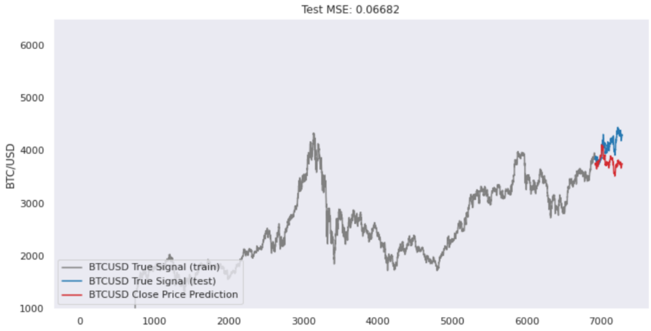
To build the LSTM model, we first design a simple network architecture using TensorFlow 2.1.x (TF) that uses a wrapper based on keras. In the process of building the model, we added LSTM, Dropout and Time Distributed as three layers and trained with 80 epochs. Based on the result of 80 epochs, the loss during training is decreasing, which meets our goal. In addition to that, To visualize the evolution of the loss function, we also design a plot that shows the process of training. Based on the graph below, with times of transformation, the loss is decreasing and eventually becomes flat/consistent.

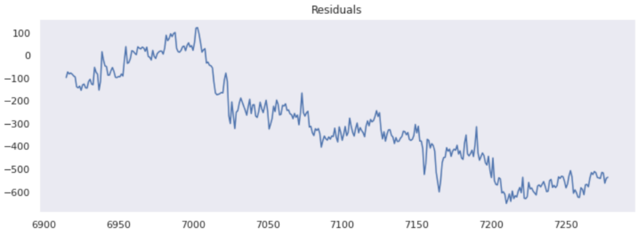


4.4 Bitcoin Price Prediction

1. Prediction Result
   In order to test the prediction of bitcoin price and actual value, we use the mean square error regression loss (MSE) to measure. Based on the graph below, the test MSE result we got is 0.06682 which represents the lowest error measurement. The pot shows without the influence of particularly large external factors, there is not much difference between the predicted price and the actual price. However, after the interference of external factors, there are some differences between our predicted results(the blue line) and actual results(the red line), however, our model can still successfully predict the tendency of bitcoin price(based on the graph, the red line shows the same tendency with the blue line).

2. Residuals
   The second graph represents the residuals of difference between the predicted value and the actual Bitcoin closing price. Based on the graph we can conclude that the RNN LSTM network cannot predict a sudden price drop that is caused by external factors(like covid, or government policies). Prior to this, it can successfully handle forecasts that deviate from actual prices, ranging from 100 to -250 US dollars. In addition to that, the result shows that our false predictions generated by our model are more likely to be slightly below the actual bitcoin price.





5 Discussion

1. External Factors
   As you can see in the graphs and images we

listed here, our model is able to predict the Bitcoin price in short term(3 hour) when there is no external factor or the external factor is not able to cause rapid increase and decrease of Bitcoin prices(and other crypto prices as well). However, when there are external factors that they are large enough to result the rapid validation of crypto prices, our model is not be able to predict the accurate prices(with in certain range). Instead, it will predict the similar trend of the Bitcoin price with large gap between real Bitcoin price and predicted Bitcoin price. And according to the residual graph we have generated, the observed price difference between real price and predicted price is ranged from 100 to -600, considering the Bitcoin price is nearly 50k usd, our model is ”accurate” enough for predicting the trend in the next hour but not the exact price. However, we used a year long data set to predict the next hour Bitcoin price with probability of high error, with this efficiency, I don’t think we can say that we have reached our previous goal which is our model has the ability to predict Bitcoin price in the next hour.

The reason why our model can’t perform its job largely is due to unpredictable external factor and we can’t really forecast how these external factors are going to affect the market. In terms of external factors, they could be political orders, like banning crypto as valid currencies, banning mining crypto, etc. These actions are complete break-down the investors’ confidences and largely affect crypto prices negatively. However, there are many other external factors like Elon Musk’s tweet, he can tweet whatever he wants about crypto and these tweets will result in market shock. Although the shock could be negative or positive, not like political actions which most of them are negative. And there is no build-in logic to analyze how these tweets/news will affect the market, considering currently, crypto are built upon nothing- there is no actual asset to guarantee the value of crypto.

One of the other interesting part of our data could be the training data we used is already tainted since the data is ranged from December 2020 to October 2021, as I remembered, the crypto market had been affected by many external factors. So we probably used dirty data to train our model and our prediction is not accurate as well. One of the solution could be using the data from few years ago, when the crypto market was not largely affected by politics, social media, etc. The prediction could be much more accurate if our model is well trained.

2. Internal Factors.
Besides the effect of external factors have applied to our data, there could be some problems with our model. We did some major changes to our data to allow the model to fit it, like get rid of nan and zero values(I think it will definitely affect the prediction, since we deleted rows of values. The other problem with our data was we reshaped our data before we fit them, in terms of for LSTM model, we need to reshape the data from 2 dimensions to 3 dimensions, and meanwhile, we used minmaxscaler to rescale the data in range(0,1). I don’t think the process is the problem but probably if we change the scale range, our model could perform better. And changing epochs, add more layers, recombing different layers, using different batch size, etc. could also help the model to increase the accuracy of its prediction.
Apart from model problem, we had a problem while we fit and trained our model. In the start, we have found that our loss was nan when we fit the model using our data, which could be caused by: gradient explosion, low learning rate, nan values in our data, data label is not correct. And after weeks of testing, we finally figured out the reason why the loss value is nan: the data we used is not correct. The method I mentioned above: using minmaxscaler to rescale the data and reshape it to correct dimension and split the data in correct proportion, is the method we used after we found out that our previous approach was not correct. What we did previously was using train test split from sklearn library to split the data so we can have train and test data. But after we printed out the data the function returned, we had found out that the values of the returned data are too small that their magnitude was power of -10

of the original data. Therefore we rewrote the train test split function on our own, and reduce the learning rate of the model. Then we finally had loss value while we trained the model. Although we had our model successfully predict some crypto prices, but I think there is better much better way to preprocess the data before we fit the model with the data. It is sad that due to time limit we can't find a better way to deal with the data, but it is definitely going to be our future work.

3. Follow up
   Like we discussed above, in the future, we are going to find a proper way to preprocessing the data and keep adjusting the model until we have found out the proper values of the variables which allow the model to have the lowest loss and MSE. Further more, we also want to use the old data(in years ago) to re-train our model to see if it can actually predict accurate price of Bitcoin. And we are going to train our model using the data ranged from 2019 to 2020, we hope to observe some rapid drop/increase in residual plot. Since COVID is one of the biggest external factor for crypto during that period of time. COVID is not only the black swan for the whole world, but also for crypto, and it would be very interesting if we can observe the rapid changes in residual plot before 2020 and after 2020.

References

China already banned Bitcoin mining—now it's cracking down on holdouts — Fortune. https://fortune.com/2021/11/17/china-bitcoin-mining-ban-crypto-holdouts-ether-solana-price/.

George S. Atsalakis, Ioanna G. Atsalaki, Fotios Pasiouras, and Constantin Zopounidis. 2019. Bitcoin price forecasting with neuro-fuzzy techniques. *European Journal of Operational Research*, 276(2):770–780.

Tian Guo, Albert Bifet, and Nino Antulov-Fantulin. 2018. Bitcoin Volatility Forecasting with a Glimpse into Buy and Sell Orders. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 989–994.

Huisu Jang and Jaewook Lee. 2018. An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information. *IEEE Access*, 6:5427–5437.

Dennys C.A. Mallqui and Ricardo A.S. Fernandes. 2019. Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Applied Soft Computing*, 75:596–606.

Sean McNally, Jason Roche, and Simon Caton. 2018. Predicting the Price of Bitcoin Using Machine Learning. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 339–343.

Chih-Hung Wu, Chih-Chiang Lu, Yu-Feng Ma, and Ruei-Shan Lu. 2018. A New Forecasting Framework for Bitcoin Price with LSTM. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 168–175.

Peter T. Yamak, Li Yujian, and Pius K. Gadosey. 2019. A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pages 49–55, Sanya China. ACM.