Machine Learning for Cryptocurrency Pricing

Nicholas G. Hayeck
University of Virginia
Charlottesville, Virginia, USA
hayeck@virginia.edu

Nijat U. Khanbabayev
University of Virginia
Charlottesville, Virginia, USA
nijatk@virginia.edu

ABSTRACT

In this paper we tackle the task of bitcoin pricing utilizing machine learning models. We compared the performance of a multilayer perceptron (MLP) and a random forest classifier with that of a Long short-term memory (LSTM) model on our detailed dataset and noted the differences in the speed and accuracy of the models. Our MLP and random forest classifiers predict the sign of the change in bitcoin pricing from day-to-day, serving as a binary classifier, while our LSTM model predicts the price of Bitcoin on a given day. The dataset we used catalogued a set of 24 features communicating Bitcoin price and market presence, measured daily over a span of 8 years. We found that the MLP and random forest algorithms significantly underperformed relative to the LSTM, with the latter model still falling somewhat short of expectations. Our main contributions are that we highlighted the importance of analysing time series data in the correct manner and the the erratic nature of the cryptocurrency market, and financial markets in general.

KEYWORDS

datasets, neural networks, cryptocurrency pricing, bitcoin

ACM Reference Format:
Nicholas G. Hayeck and Nijat U. Khanbabayev. 2020. Machine Learning for Cryptocurrency Pricing. In *Proceedings of ACM Conference (Conference'20).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 The Problem

In finance, there is a great deal of competitive advantage to be found in the prediction of currency exchange rates. If one could be given a somewhat accurate estimate of the exchange rate between, say, the Euro and the dollar for the following trading day based solely upon current data, one could conceivably take short-term positions in the currency expected to appreciate and make sizeable returns. This is the problem we attempt to showcase except, rather than the euro and dollar (which have a whole host of geopolitical reasons for their relative values), we examine the case of cryptocurrency and the U.S. Dollar. While cryptocurrency is clearly not the perfectly uncorrelated asset some may make it to be, we make an attempt to price it based upon daily trading data, with some promising results.

1.2 Bitcoin

Founded in 2009 by the pseudonymous Satoshi Nakamoto, Bitcoin quickly gained popularity amongst some communities due to its unregulated nature, and has become the predominant cryptocurrency based on its market share. All Bitcoin transactions are stored in a decentralized ledger distributed across participating computers around the world. Instead of relying on a central authority to verify transactions, Bitcoin centers on a proof-of-work concept that utilizes the difficulty of certain computational tasks to verify the legitimacy of its transactions. The individual who is the first to complete one of these problems is awarded newly created Bitcoin for their efforts, increasing the total supply of Bitcoin. The amount awarded to each miner steadily lowers over time, bounding the total number of Bitcoin in circulation to approximately 21 million. The popularity of Bitcoin has greatly increased in recent years, and while some criticize the excitement around Bitcoin as a financial bubble, others believe Bitcoin will only become more integrated into everyday life. Many investors and everyday people are realizing the utility of Bitcoin and using it in place of fiat currencies, or as a way to diversify one's portfolio. Governments, however, are becoming wary of its usage and increasingly look to tamp down its prospects for use as a global currency via regulation. Either way, Bitcoin has established itself as the premier digital currency across the globe and will be a force in the financial world for years to come.

1.3 Dataset

Our dataset consist of 24 features that were updated daily from 2010 to 2018. This resulted in 2921 total real-world datapoints. We selected this specific dataset due to its combination of features reflecting the current-time valuation of Bitcoin, including market price, transaction volume in USD, and miner's revenue, as well as information regarding the technical details of Bitcoin such as mining difficulty, hash rate, average confirmation time, and the number of unique addresses. We were hoping these two spheres of information could be utilized by our model to both predict the usage of Bitcoin as well as the current public perception. The problem of alternative asset pricing, specifically applied to Bitcoin, presents a unique challenge in that alternative asset prices are often highly correlated with the equity capital markets and thus our feature set will often come up short in presenting "learnable" data for pricing. Regardless of this, there is likely to be some learnable features within the data that will bring some insight to the way in which cryptocurrency, and alternative assets as a whole, are priced.

The great variation in the bitcoin feature values was a cause for concern, but after normalization the issue was resolved and we

Conference'20, December 2020, Charlottesville, VA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

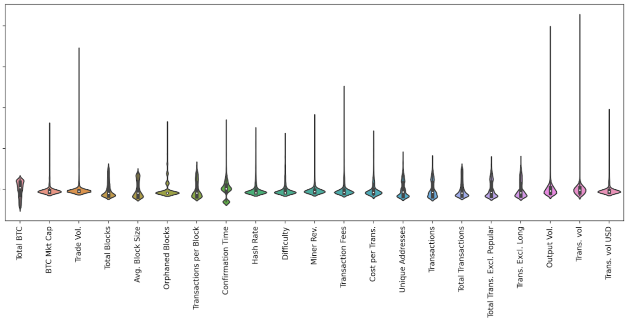
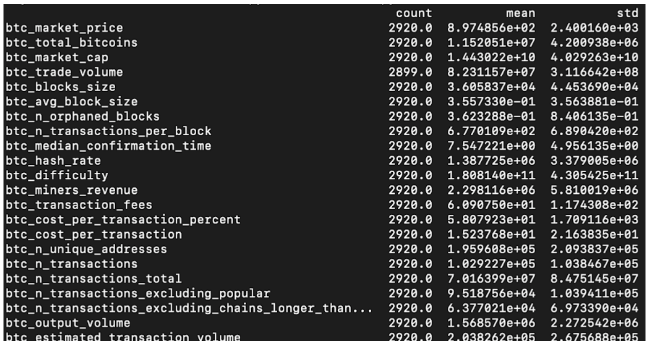


Figure 1: A violin plot showcasing the distribution of the normalized data



```
                                                 count          mean           std
btc_market_price                                2920.0  8.974856e+02  2.400160e+03
btc_total_bitcoins                              2920.0  1.152051e+07  4.200938e+06
btc_market_cap                                  2920.0  1.443022e+10  4.029263e+10
btc_trade_volume                                2899.0  8.231157e+07  3.116642e+08
btc_blocks_size                                 2920.0  3.605837e+04  4.453690e+04
btc_avg_block_size                              2920.0  3.557330e-01  3.563881e-01
btc_n_orphaned_blocks                           2920.0  3.623288e-01  8.406135e-01
btc_n_transactions_per_block                    2920.0  6.770109e+02  6.890420e+02
btc_median_confirmation_time                    2920.0  7.547221e+00  4.956135e+00
btc_hash_rate                                   2920.0  1.387725e+06  3.379005e+06
btc_difficulty                                  2920.0  1.808140e+11  4.305425e+11
btc_miners_revenue                              2920.0  2.298116e+06  5.810019e+06
btc_transaction_fees                            2920.0  6.090750e+01  1.174308e+02
btc_cost_per_transaction_percent                2920.0  5.807923e+01  1.709116e+03
btc_cost_per_transaction                        2920.0  1.523768e+01  2.163835e+01
btc_n_unique_addresses                          2920.0  1.959608e+05  2.093837e+05
btc_n_transactions                              2920.0  1.029227e+05  1.038467e+05
btc_n_transactions_total                        2920.0  7.016399e+07  8.475145e+07
btc_n_transactions_excluding_popular            2920.0  9.518756e+04  1.039411e+05
btc_n_transactions_excluding_chains_longer_than...  2920.0  6.377021e+04  6.973390e+04
btc_output_volume                               2920.0  1.568570e+06  2.272542e+06
btc_estimated_transaction_volume                2920.0  2.038262e+05  2.675688e+05
```

Figure 2: A summary of each feature including the average value, and the standard deviation

could not identify specific features that were heavily skewing our results.

2 IMPLEMENTATION

2.1 General Approach

To explore the full gamut of To explore the full gamut of machine learning techniques as they apply to this problem, we decided on using three forms of ML which are among the most commonly used in practice today: a multi-layer perceptron, a random forest classifier, and a long short-term memory model. The first two would use similar data, taking in variables such as trading volume, total bitcoins in circulation, etc., and predicting whether the USD/BTC exchange rate would increase or decrease the following trading day. The LSTM would predict the next trading day's price, using a "look-back" period of three trading days (i.e. the model would have access to the previous three days' data when predicting the next day's exchange rate).

2.2 Multi-layer Perceptron

With our dataset containing 22 features, the architecture of the model was defined as follows:

```
# Define model architecture
model = keras.Sequential([  keras.layers.Input(shape=(22)),
                            keras.layers.Dense(128),
                            keras.layers.Dense(128),
                            keras.layers.Dense(1),
                            keras.layers.Activation(keras.activations.sigmoid)
                            ])
```

It consists of two dense layers, each containing 128 nodes, with a single output node, activated by a logistic function. This is a pretty standard MLP. For this task, we used Adam's optimizer with a learning rate of approximately 0.001, trained over 100 epochs with a 90/10 validation split.

2.3 Random Forest Classifier

Our random forest classifier attempted the same task as the Multi-layer Perceptron, namely attempting to predict the sign of the daily price-change for Bitcoin. We utilized a randomized split of 85% in the training set and 15% in the testing set repeated 10 times to analyze the efficacy of our random forest classifier. Notably, we did not initially utilize the time-series data to predict the price change, but when this feature was implemented, the accuracy did not rise substantially. We optimally tuned the hyperparameters and found that a max depth of 10, a value of 200 for the estimator count, and a value of 7 for the number of jobs optimized our results.

```
classifier = RandomForestClassifier(n_estimators=100)
accuracy = 0
v=10
for i in range(v):
    x_train, x_test, y_train, y_test = train_test_split(
        x, y,test_size=0.15
    )
    model = classifier.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print("Random Forest accuracy on trial",i+1,":", accuracy_score(y_test, y_pred))
    accuracy += accuracy_score(y_test, y_pred)
print("Average Random Forest accuracy:", accuracy/v)
```

2.4 Long Short-Term Memory Model

For the LSTM, an incredibly simple model architecture was used:

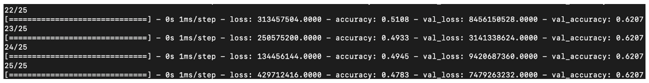
```
# Define model architecture
model = keras.Sequential([  keras.layers.LSTM(50),
                            keras.layers.Dense(1),
                            keras.layers.Activation(keras.activations.relu)
                            ])
```

It consists of a single LSTM layer, with fifty nodes, and a singular output node, activated by the relu function defined by $f(x) = \max(0, x)$ since we assume the exchange rate to be non-negative. Again, we have used Adam's optimizer, with a learning rate of 0.5, but have used mean absolute error to reflect the quantitative nature of the output.

3 RESULTS

3.1 MLP and Random Forest Models

Unfortunately, both of these models achieved suboptimal results. Without any way for the algorithm to orient itself in time, the algorithms were unable to exceed chance. Even with hyperparameter tuning, all sorts of different layer combinations, and a dozen different optimization methods, the perceptron was unable to get any sort of a grasp on the data. We found that the loss slowly converged over a large number of epochs, with the convergence slowing after around 100 epochs. Figures for the accuracy and loss of the two models are shown below:



```
22/25
[==============================] - 0s 1ms/step - loss: 313457504.0000 - accuracy: 0.5108 - val_loss: 8456150528.0000 - val_accuracy: 0.6207
23/25
[==============================] - 0s 1ms/step - loss: 250575200.0000 - accuracy: 0.4933 - val_loss: 3141338624.0000 - val_accuracy: 0.6207
24/25
[==============================] - 0s 1ms/step - loss: 134456144.0000 - accuracy: 0.4945 - val_loss: 9420687360.0000 - val_accuracy: 0.6207
25/25
[==============================] - 0s 1ms/step - loss: 429712416.0000 - accuracy: 0.4783 - val_loss: 7479263232.0000 - val_accuracy: 0.6207
```

Figure 3: Accuracy ratings of the MLP, including validation accuracy, validation loss, and loss for the final 3 epochs

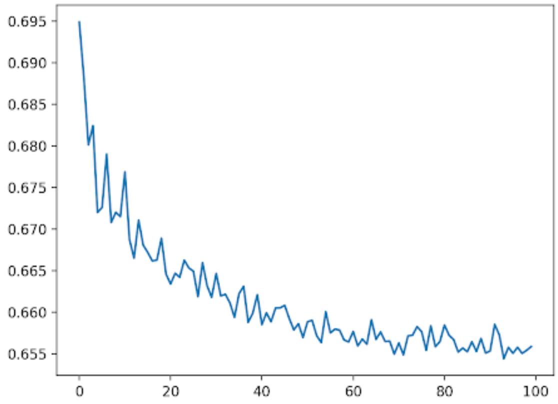
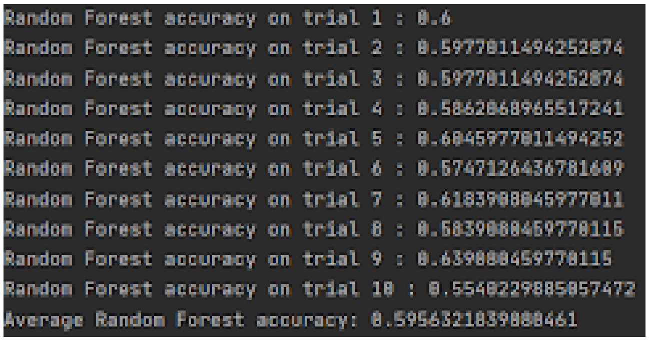


Figure 4: MLP loss plotted over the 100 epochs the model was ran for



```
Random Forest accuracy on trial 1 : 0.6
Random Forest accuracy on trial 2 : 0.5977011494252874
Random Forest accuracy on trial 3 : 0.5977011494252874
Random Forest accuracy on trial 4 : 0.5862068965517241
Random Forest accuracy on trial 5 : 0.6045977011494252
Random Forest accuracy on trial 6 : 0.5747126436781609
Random Forest accuracy on trial 7 : 0.6183908045977011
Random Forest accuracy on trial 8 : 0.5839080459770115
Random Forest accuracy on trial 9 : 0.639080459770115
Random Forest accuracy on trial 10 : 0.5540229885057472
Average Random Forest accuracy: 0.5956321839080461
```

Figure 5: Accuracy ratings of the random forest classifier over 10 trials with a randomized 85%-15% training/testing split

3.2 LSTM Model

The LSTM showed more promising results, and the figure below clearly show that the LSTM was able to follow the trend line of the exchange rate quite well. While this is the case, it certainly wouldn't be a candidate for practical deployment. It would be unwise to set it free in the market and have it place trades or anything of that sort; it's clearly not ready for that sort of task. That aside, it is an interesting view into how markets that have traditionally been entirely uncorrelated to the equity markets can still have some internal structure that can be found and (admittedly, somewhat poorly) learned by the algorithm.

In previous work (e.g. see Madan, et. al), it is mentioned that an issue with their model is that it only looks at price changes at single points in time, which is non-optimal for checking complex and overarching patterns in the data. While we believe we have solved this challenge (an LSTM solves this issue by utilizing time series data to predict future prices, rather than focusing on single points), our results clearly show that, while encoding time into the data set shows some improvements, it is not a sufficient criteria for a practically deployable model.

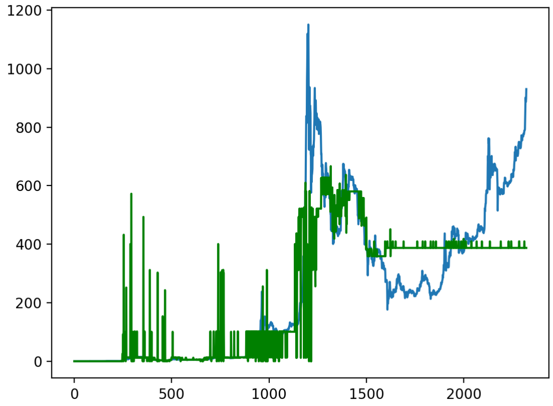


Figure 6: Comparison between actual bitcoin price (blue) and the price predicted by the LSTM (green) over the entire dataset

4 CONCLUSION

Despite the numerous models attempted, we were not able to create a model that accurately predicted bitcoin prices. There were a few technical challenges that arose when interpreting the dataset. Most importantly, in the latter half of 2017, Bitcoin had a surge in popularity that took its price from around 1000 USD in April of 2017 to a high of 17, 000 USD in January 2018. A similarly drastic spike had not occured at any time prior in bitcoin's history, as can be see in the figure below: This volatility, possibly due to increased publicity around Bitcoin not fully reflected in our dataset, heavily skewed our models resulting in inaccurate predictions. We found that omitting around the last 200 days avoided this spike, and resulted in more accurate models for the rest of the dataset. Also, our dataset was

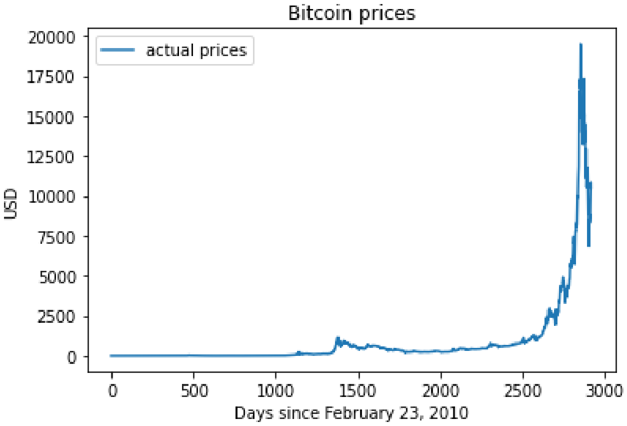


Figure 7: Bitcoin prices in USD starting from 2010

only 2921 points, which when coupled with validation splits, meant that optimizing out models could only be done on a relatively small number of datapoints.

Prior attempts at Bitcoin pricing using machine learning models have not been very successful, probably due to the complex nature of the currency's pricing. However, our use of time-series data did offer promising results. Future explorations could attempt to utilize an XGBoost model on the dataset formatted to represent the time-series data. This approach would combine the utility of the time-series approach the LSTM used, while also combining many less accurate models in a decision tree-style format to optimize the accuracy of the overall conglomeration. Our results are promising are further development could result in a viable model.

REFERENCES

[1] Anonymous Author. 2020. Bitcoin Machine Learning. https://ryxcommar.com/2020/12/08/bitcoin-machine-learning/

[2] Zheshi Chen, Chunhong Li, and Wenjun Sun. 2020. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *J. Comput. Appl. Math.* 365 (2020), 112395.

[3] Anne Haubo Dyhrberg. 2016. Bitcoin, gold and the dollar–A GARCH volatility analysis. *Finance Research Letters* 16 (2016), 85–92.

[4] Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, Vol. 1. IEEE, 278–282.

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1996. LSTM can solve hard long time lag problems. *Advances in neural information processing systems* 9 (1996), 473–479.

[6] Isaac Madan, Shaurya Saluja, and Aojia Zhao. 2015. Automated bitcoin trading via machine learning algorithms. *URL: http://cs229. stanford. edu/proj2014/Isaac%20Madan* 20 (2015).

[7] Satoshi Nakamoto. 2019. *Bitcoin: A peer-to-peer electronic cash system.* Technical Report. Manubot.

[8] Devavrat Shah and Kang Zhang. 2014. Bayesian regression and Bitcoin. In *2014 52nd annual Allerton conference on communication, control, and computing (Allerton)*. IEEE, 409–414.