Machine Learning for Cryptocurrency Pricing

Nijat Khanbabayev & Nicholas Hayeck

Motivation

- Bitcoin and other cryptocurrencies have gotten much coverage in the recent years
- Due to its decentralized nature, market dominance, and adoption by banks, bitcoin is poised to continue its growth and large-scale adoption by investors and every-day consumers
- Complaints of long transaction times and fees have been addressed with the adoption of the Lightning Network

Background

- Bitcoin is a digital asset designed to mimic physical currency that was invented in 2008 by “Satoshi Nakamoto” (likely a pseudonym)

- Using a central ledger, called the blockchain, to verifiably record transactions without the need for intermediaries

- Characterized by many economists as a speculative bubble, its exchange rate with the US Dollar has been wildly volatile since the cryptocurrencies inception

- Research estimates that are approximately four million unique users of Bitcoin globally

Target Task

- We seek to find an efficient machine learning model to predict bitcoin prices given a set of 24 features including the difficulty mining, hash rate, total bitcoins mined, and transactions per block

- We hope for our model to be able to accurately predict bitcoin prices, and possibly apply our model to accurately predict the prices of other cryptocurrencies

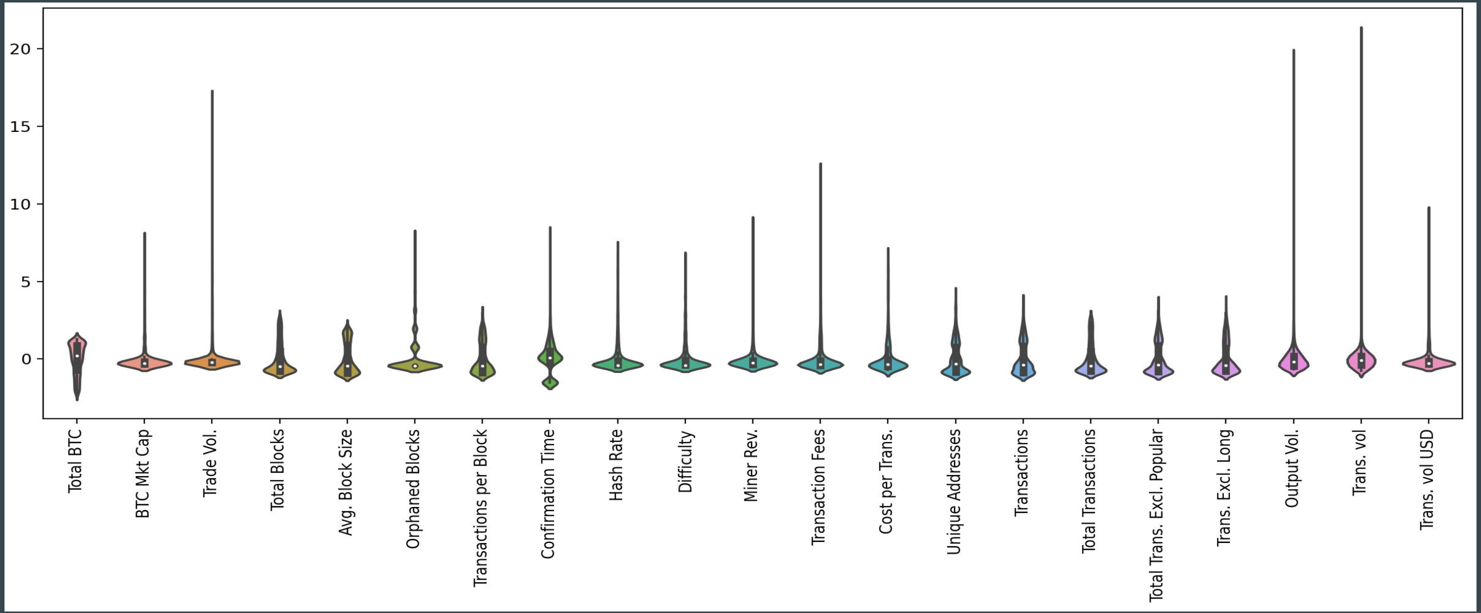
Dataset

- Our dataset includes information on 24 features of Bitcoin documented daily for 8 years. Overall, the dataset contains 2921 total data points

- The diverse set of features include information about the valuation of Bitcoin including the market price, transaction volume in USD, and miner's revenue

Dataset

- However, it also contains technical details regarding Bitcoins day-to-day usage, such as the mining difficulty, hash rate, average confirmation time, and the number of unique addresses



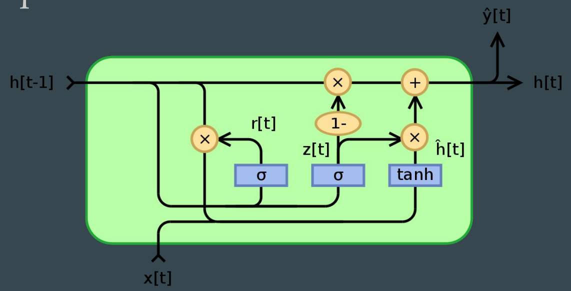
	count	mean	std
btc_market_price	2920.0	8.974856e+02	2.400160e+03
btc_total_bitcoins	2920.0	1.152051e+07	4.200938e+06
btc_market_cap	2920.0	1.443022e+10	4.029263e+10
btc_trade_volume	2899.0	8.231157e+07	3.116642e+08
btc_blocks_size	2920.0	3.605837e+04	4.453690e+04
btc_avg_block_size	2920.0	3.557330e-01	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
btc_estimated_transaction_volume_usd	2920.0	2.118495e+08	5.903811e+08

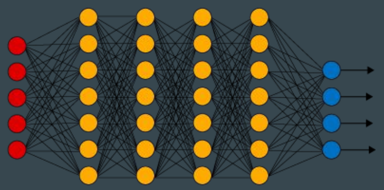
A View of the Data

General Approach

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:

(1) Deep Neural Network
  (a) This model would take in variables such as trading volume, total bitcoins in circulation, etc.
  (b) Predicts whether the price will go up or down the next trading day
(2) Recurrent Neural Network:
  (a) Uses same data as DNN, in addition to previous time series data
  (b) Predicts the USD/BTC exchange rate
(3) Random Forest:
  (a) Uses the DNN data
  (b) Predicts price movement direction





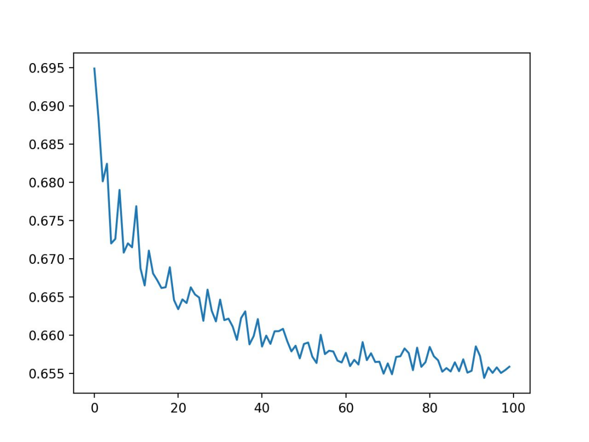
Attempt 1: Deep MLP

Deep Neural Network

Failure. Without any way for the algorithm to orient itself in time, the algorithm was 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.

```
loss: 313457504.0000 - accuracy: 0.5108 - val_loss: 8456150528.0000 - val_accuracy: 0.6207
loss: 250575200.0000 - accuracy: 0.4933 - val_loss: 3141338624.0000 - val_accuracy: 0.6207
loss: 134456144.0000 - accuracy: 0.4945 - val_loss: 9420687360.0000 - val_accuracy: 0.6207
loss: 429712416.0000 - accuracy: 0.4783 - val_loss: 7479263232.0000 - val_accuracy: 0.6207
```



Attempt 2: Random Forest Classification

Random Forest

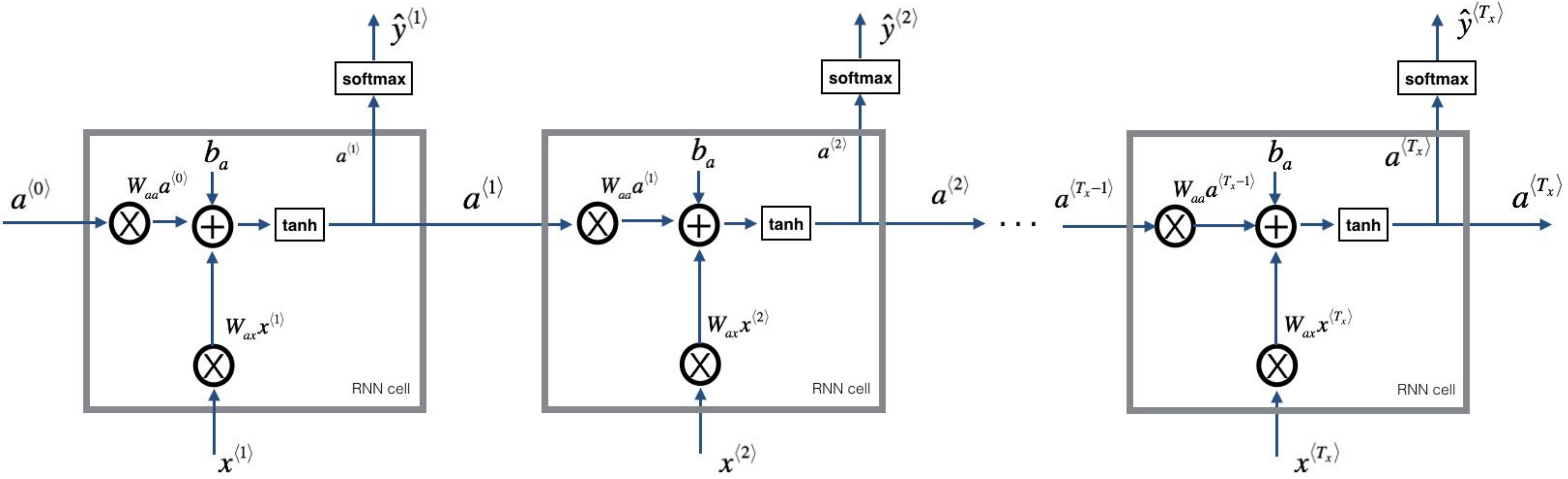
- We then tried utilizing Random Forest alongside standard scaling to predict the direction of price changes

- While more successful than an MLP, this approach only hovered around 59% accuracy

```
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
```

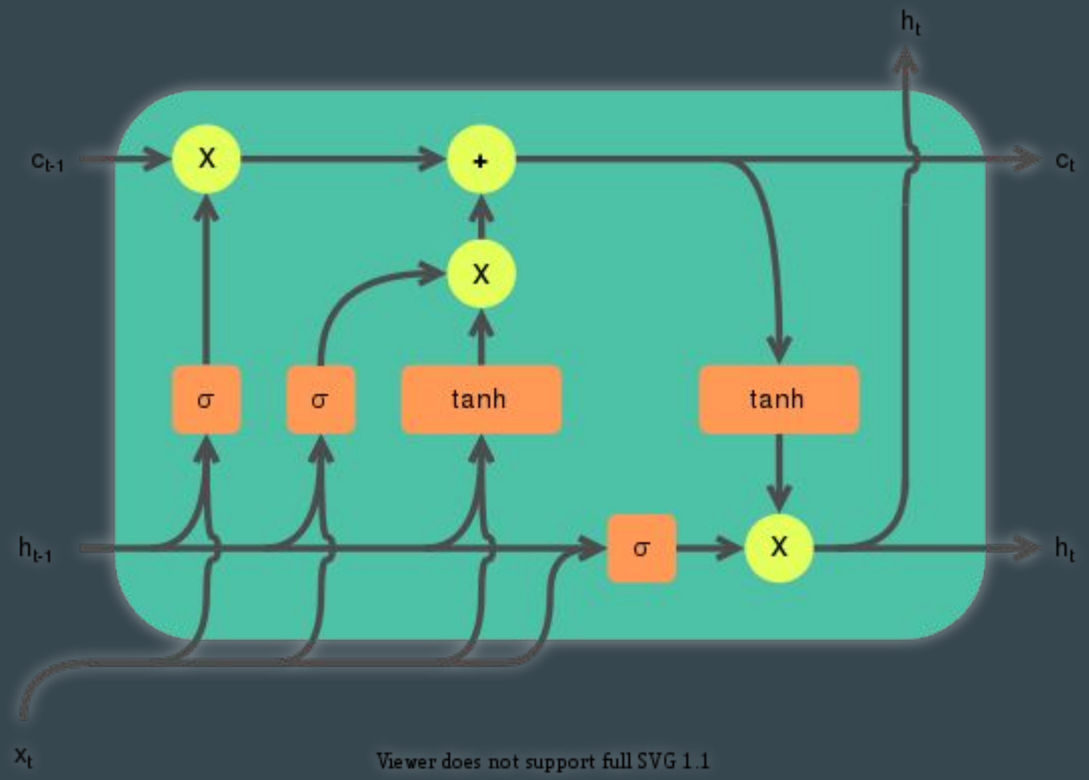
Attempt 3: Recurrent Neural Network

RNN Primer

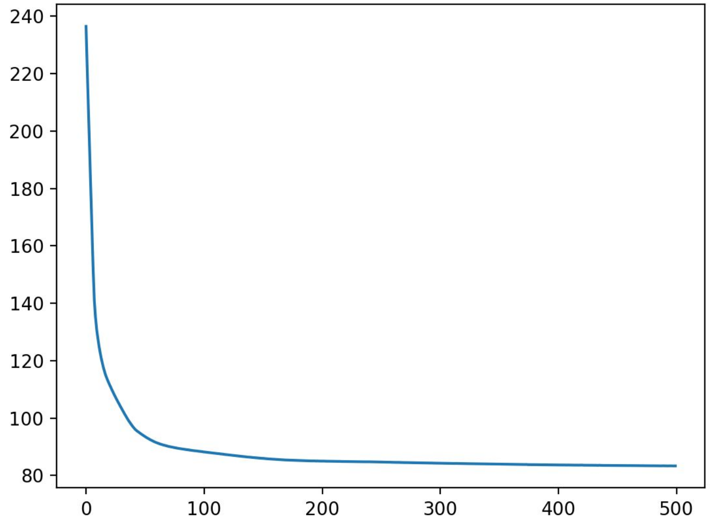


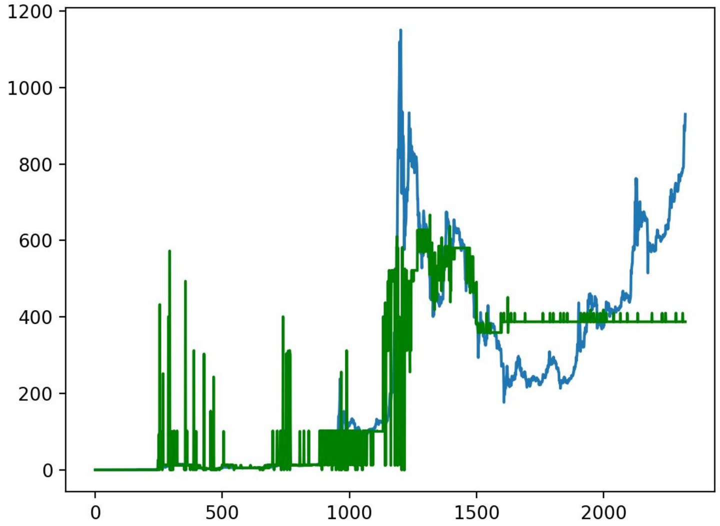
$\hat{y}^{\langle 1 \rangle}$
softmax
$a^{\langle 0 \rangle}$
$W_{aa}a^{\langle 0 \rangle}$
b_a
$a^{\langle 1 \rangle}$
tanh
$W_{ax}x^{\langle 1 \rangle}$
RNN cell
$x^{\langle 1 \rangle}$
$a^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
softmax
$W_{aa}a^{\langle 1 \rangle}$
b_a
$a^{\langle 2 \rangle}$
tanh
$W_{ax}x^{\langle 2 \rangle}$
RNN cell
$x^{\langle 2 \rangle}$
$a^{\langle 2 \rangle}$
. . .
$a^{\langle T_x-1 \rangle}$
$\hat{y}^{\langle T_x \rangle}$
softmax
$W_{aa}a^{\langle T_x-1 \rangle}$
b_a
$a^{\langle T_x \rangle}$
tanh
$W_{ax}x^{\langle T_x \rangle}$
RNN cell
$x^{\langle T_x \rangle}$
$a^{\langle T_x \rangle}$

Long short-term memory RNN



LSTM Results






```
Epoch 500/500
73/73 [==============================] - 0s 2ms/step - loss: 83.3430
Test RMSE: 331.758
```

Technical Challenges

- 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; this volatility, possibly due to increased publicity around Bitcoin not fully reflected in our dataset, heavily skewed our models resulting in inaccurate predictions

- The size of the dataset was only 2921 data points which might not be sufficient to create a robust and accurate model

- Some of the features given in the dataset may have been entirely uncorrelated to trading activity and thus the exchange rate

Related Work

- Bitcoin price prediction using machine learning: An approach to sample dimension engineering
  - https://www.sciencedirect.com/science/article/abs/pii/S037704271930398X
- 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

Contributions & Conclusion

- Although optimal results were not achieved with our models, the vast difference in ability of the three models presents some clear lessons, highlighting:
    - the importance of analysing time series in the correct manner
    - the erratic nature of the cryptocurrency market, and financial markets in general
- For financial markets in general, this experiment has shown the importance of external factors, not found in the trading data or internal BTC metrics
- 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

References

[1] Ho, T. K. (1995, August). Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278-282). IEEE.

[2] Hochreiter, S., & Schmidhuber, J. (1996). LSTM can solve hard long time lag problems. Advances in neural information processing systems, 9, 473-479.

[3] Nakamoto, S. (2019). Bitcoin: A peer-to-peer electronic cash system. Manubot.

[4] Dyhrberg, A. H. (2016). Bitcoin, gold and the dollar–A GARCH volatility analysis. Finance Research Letters, 16, 85-92.

Thank You