Understanding and Predicting Trends In Cryptocurrency Prices Using Data Mining Techniques

Adarsh Pal Singh (20161225)
International Institute of Information Technology
Hyderabad, India
adarshpal.singh@research.iiit.ac.in

Saurav Malani (201502047)
International Institute of Information Technology
Hyderabad, India
saurav.malani@research.iiit.ac.in

ABSTRACT

In a span of 10 years, we've witnessed the inception as well as the gain in popularity of the blockchain technology and cryptocurrienc. Various cryptocurrencies have come up and evolved in a short span. Just like the regular stock market, the trends of different cryptocurrency coin prices is of great importance to the cryptocurrency traders and investors. However, unlike stock market analysis which has been going on for decades, cryptocurrency market is still referred to as "highly volatile" thereby giving motivation for researchers to contribute to this theme. In this paper, we attempt to use data mining techniques to understand different factors affecting the price of cryptocurrencies, namely Bitcoin and Etherium, and build models that can predict different future trends of the cryptocurrency market. We first pose the problem as a binary classification one (rise or fall of price in a day) and use supervised learning techniques like Random Forest, SVM and Naive Bayes in order to determine which features of the blockchain technology have a high correlation with the trends of cryptocurrency prices and then move on to more advanced time-series techniques like ARIMA and Bayesian Structural Time Series (BSTS) to predict future values. Later, we show a comparison between the two chosen cryptocurrencies and also the regular stock market.

KEYWORDS

Cryptocurrency, Time Series Analysis, Data Mining

1 INTRODUCTION

A cryptocurrency is a digital currency which is not governed by a central entity but rather exists on a blockchain in a distributed ledger. As the name suggests, its transactions are secured via strong cryptographic techniques. In a nutshell, a cryptocurrency user announces his transaction on this decentralized network and instead of it being recorded by a single entity, the transaction is verified by other anonymous users and stored in a public ledger or the blockchain [3]. The computational power associated with such complex cryptographic zcomputations actually reciprocates in the form of a reward and this process is called mining. Even though the transactions are public, the identities of those involved are completely anonymous; a fact that is often associated with the popularity of cryptocurrencies.

The first and the most popular cryptocurrency to show up was Bitcoin, a name which has nowadays become synonymous with blockchain itself. Another popular cryptocurrency, Ether, exists on the popular Ethereum decentralized platform (In this paper, both Ether and Ethereum are used interchangeably in the context of cryptocurrency). We take these two cryptocurrencies under consideration and try to see what ticks the rise and fall of in their price values. The dataset for these two coins is obtained from Kaggle [7]. Unlike traditional currency prices, cryptocurrencies are known to be volatile. In fact, studies suggest that bitcoin prices are at least 10 times more volatile than US Dollar [2]. Since the cryptocurrency market is relatively new, research is this field is still ongoing.

Since the time cryptocurrencies formed a bubble of their own, a lot of different research ideas have come about trying to predict different trends in the cryptocurrency market with almost all of them focused on the bitcoin market. Autoregression techniques were explored by Garcia et. al. to predict bitcoin price [4, 5]. They showed that the increase in bitcoin prices is usually preceeded by the increase in opinion polarization as well as exchange volume. Devavrat et. al. used Bayesian regression technique to predict bitcoin price and built a model for traders [11]. Amjad et. al. used the historical time series price data to come up with price prediction and trading strategies [1]. Some authors have studied price fluctuations in cryptocurrency prices in general as opposed to time series prediction of price. Kim et. al. performed sentiment analysis on various cryptocurrency web communities to predict price fluctuations [6]. A similar approach of event-driven analysis for bitcoin was done by Pryzmont in [10] in which he compiled a text-based dataset from sources such as journal articles, internet news items and posts on the bitcoin community forums.

Our contributions to this promising field are as follows-

(1) Instead of analyzing and building time series prediction models for just bitcoin, we also do the same for ethereum and present a comparison between the two cryptocurrencies. Note that both these currencies exist on different blockchain technologies making this comparison relevant.
(2) Apart from time series analysis, we also look at the features of blockchain technology itself such as daily hash rate, block size, etc to infer correlation between these and price fluctuations.
(3) Comparison of cryptocurrency trends with that of regular stock market is also documented.

The structure of the rest of the paper is as follows: Section 2 describes the dataset and initial inferences from different features. We also describe the experiments here. In section 3, we describe the results of the binary classification experiment and discuss the importance of different features. Feature comparison between Bitcoin and Ethereum is also presented in this section. Time series analysis of Bitcoin and Ethereum prices is done in section 4.

DWDM, Monsoon 2018, IIIT Hyderabad
2018. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

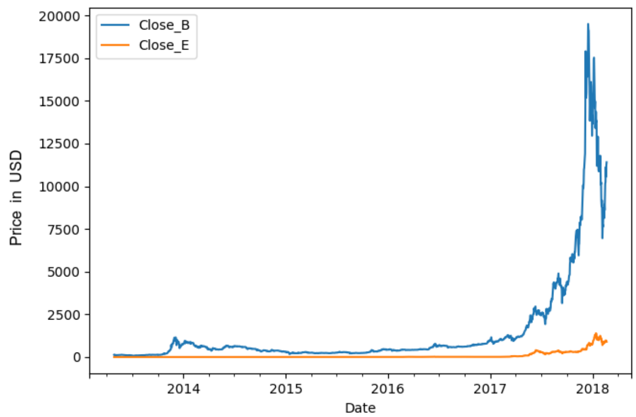


Figure 1: Date vs. Bitcoin/Ethereum price plot for a period of five years.

2 DATASET DESCRIPTION

We are using the "Cryptocurrency Historical Prices" dataset [7]. This dataset includes 15 different cryptocurrencies with features such as date, opening price, closing price, highest, lowest, volume and market cap documented of each day from April 2013 to March 2018 (almost 5 years). Table 1 gives a simple description of the bitcoin and ethereum price dataset. Notice the humongous jump between the min and max values of all the features. In fact, the transition from min to 25 percentile till 75 percentile seems like a smooth increase suggesting that there is a huge jump towards the end. The mean being around the 1500 mark for BTC also suggests this. Notice how significantly different mean is from the 50% mark. The plot of closing price for the entire duration of 5 years for BTC and around 3 years for Ethereum is shown in Figure 1. This clearly validates our hypothesis of a sudden jump which occurs at the end of 2017 for BTC. The prices fall down in the beginning of 2018 but the trend seems to be increasing again towards the end. For Ethereum, the trend is increasing but not nearly as sharply as Bitcoin. It is still below the 1000 dollar mark whereas Bitcoin is hovering around the 10,000 dollar mark. However, the juxtaposition of both the plots do reveal a similar trend even though they are on a completely different scale.

Table 1: A simple description of a subset of the dataset features

Parameter	Open BTC	Open Ether	High BTC	High Ether	Close BTC	Close Ether
Mean	1479.57	146.89	1536.99	154.36	1485.74	147.78
Std	2950.45	263.21	3092.95	277.62	2959.26	264.13
Min	68.50	0.431	74.56	0.48	68.43	0.43
25%	274.73	7.89	279.85	8.27	274.87	7.91
50%	480.71	12.05	495.19	12.43	482.81	12.02
75%	870.08	245.2	900.22	257.00	871.37	246.00
Max	19475.8	1397.48	20089	1432.88	19497.4	1396.42

For Bitcoin and Ethereum, another dataset (referred to as dataset-2 throughout this paper) exists in the repository which includes daily blockchain features for the same time-frame as the previous dataset of opening/closing prices. For Bitcoin, there are 21 features such as btc-market-price, btc-avg-block-size, btc-hash-rate, btc-miners-revenue, etc. For complete feature list with description, refer to [7]. For ethereum, there are 14 features many of which are similar to those of Bitcoin.

There are two experiments that we perform on the datasets that we have. First, we use the features from dataset-2 and the closing price feature from dataset-1 and pose a binary classification problem. Since dataset-2 has vectors pointing to days before April 2013 (start point in dataset-1), we prune these vectors to bring the date-stamps equal in both the datasets. In the second experiment, we deal with dataset-1 (opening/closing prices) and use various time series algorithms in an attempt to predict future values.

3 BINARY CLASSIFICATION EXPERIMENT

We use the features from dataset-2 and the closing price feature from dataset-1 for this. We converted the closing price feature into '1' and '0' where '1' denotes a rise in price as compared to the previous day and a '0' denotes a fall in price of bitcoin as opposed to the previous day. This is done by simply taking the difference of one feature vector with its previous vector. A positive quantity so achieved for the price difference constitutes a rise and is put as '1' whereas for the fall case, a '0' is registered. Since we are taking the difference of the two adjacent vectors, we get rising/falling values of all the features as well which are kept as it is (in this difference form). This new dataset so obtained is referred to as the binary-dataset from here onwards. Note that there was not a single case where the price stayed the same and hence we went with a binary classification problem instead of a 3-class classification problem. Finally, this binary-dataset has 1759 data points for Bitcoin and 925 points for Ethereum. As far as the skewness of the dataset is concerned, for the Bitcoin dataset, we have around 968 ones and 791 zeros and for the Ethereum dataset, 462 ones and 463 zeros. The datasets are almost balanced.

We can now use supervised learning techniques on these binary-datasets (one for Bitcoin and one for Ethereum). We make use of three popular techniques- Support Vector Machine or SVM (both with linear and radial basis function (RBF) kernel to allow for non-linear decision boundaries), Random Forest (RF) and Naive Bayes (NB). In SVM, each n-dimensional feature vector is basically a point in an n-dimensional feature space, where n is the number of features present in the dataset. Each vector belongs to one of the k classes. SVM does not make any assumptions about the data. The algorithm attempts to fit an optimal hyperplane between the two classes with the help of support vectors. A non-linear separation boundary can be obtained by using a kernel. SVM has a tunable penalty hyperparameter for indicating the tolerable error or misclassifications in fitting the hyperplane. The RF classifier, on the other hand, is an ensemble of decision trees where each tree is grown on some part of the dataset with replacement and has a vote. The final outcome is the average of the votes of all the trees. A decision tree, as the name suggests, is a tree-shaped structure in which each internal node represents a logical test on some feature and each leaf node

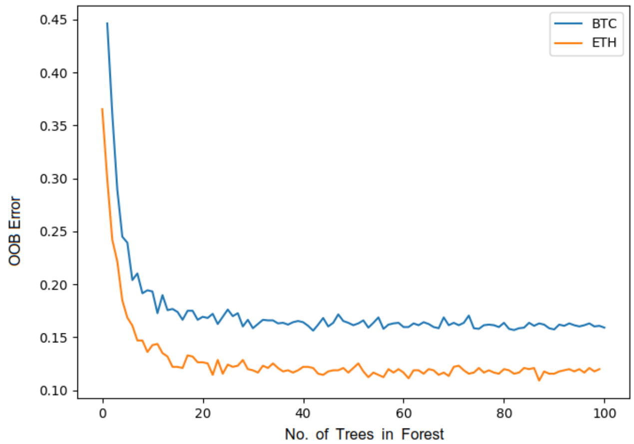
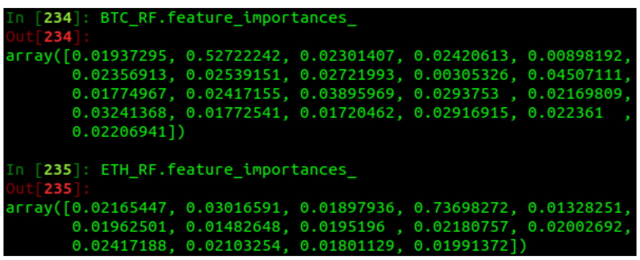


Figure 2: OOB error rate for Bitcoin and Ethereum coin dataset.



```
In [234]: BTC_RF.feature_importances_
Out[234]:
array([0.01937295, 0.52722242, 0.02301407, 0.02420613, 0.00898192,
       0.02356913, 0.02539151, 0.02721993, 0.00305326, 0.04507111,
       0.01774967, 0.02417155, 0.03895969, 0.0293753 , 0.02169809,
       0.03241368, 0.01772541, 0.01720462, 0.02916915, 0.022361  ,
       0.02206941])

In [235]: ETH_RF.feature_importances_
Out[235]:
array([0.02165447, 0.03016591, 0.01897936, 0.73698272, 0.01328251,
       0.01962501, 0.01482648, 0.0195196 , 0.02180757, 0.02002692,
       0.02417188, 0.02103254, 0.01801129, 0.01991372])
```

Figure 3: Feature importances as reported by Random Forest Classifier.

represents one of the outcome classes. The tree forming algorithm used in this experiment is the classification and regression tree (CART) algorithm which utilizes the gini index for determining the quality of a split. The number of trees is a tunable hyperparameter along with a few others for pruning and preventing overfitting. We tuned the minimum samples split for the latter case. Naive Bayes is a simple algorithm which exploits the Bayes theorem with the assumption that there is conditional independence between every pair of features given the value of class variable. We make use of the popular Gaussian Naive Bayes algorithm in which the likelihood of features is assumed to be Gaussian. 5-fold cross validation is done and accuracy as well as F1 score is reported for each classifier. All the classification-based experiments have been done using Scikit-learn library in Python [9].

Figure 2 shows the Out-Of-Bag or OOB error score vs. the number of trees in the forest for the binary-datasets for Bitcoin as well as Ethereum. After around 30 trees, we see not much change in the OOB error. Table 2 shows the accuracy and F1 score pertaining to RF, SVM (Linear), SVM (RBF) and NB models. As discussed above, RF classifier consists of 30 different decision trees trained on differnet parts of the datset and the minimum-samples-split parameter was kept at 30 to prevent overfitting. In case of SVM, we tuned the penalty parameter and found C=1.0 to be optimal. For NB, there is no hyperparameter to tune. Clearly, we get the best accuracy with the RF classifier in both the cases. We get around 80% accuracy and F1 score for predicting the rise/fall of Bitcoin price and around 87% accuracy for predicting the rise/fall in Ethereum price. This shows how useful ensemble models are. Another inference from Table 2 is that linear SVM outperforms non-linear SVM suggesting that the data spread is more towards the linearly separable side and that a kernel trick does not help to make it any more separable but rather decreases the performance. SVM linear also has a lower compulational rate is less prone to overfitting making it a good choice. Naive Bayes' performance is poorer than non-linear SVM. The Gaussian assumption for the features simply do not hold in this case.

A useful property of RF classifier is that it gives us a vector pertaining to the importance of all features that we have in the input. Note that all the features mentioned below are relative difference of current day and previous day since this is how we formed the dataset in the first place. Feature importances for both the datasets are shown in Figure 3. Feature 2 for Bitcoin and feature 4 for Ethereum clearly outshine all other features. Feature 2 of Bitcoin is the btc-market-cap and feature 4 of Ethereum is the exact same measure eth-marketcap. Market capitalization specifies the total USD value of bitcoin supply in circulation. While the price changes frequently based on the number of shares bought and sold, the market cap is a broader scenario of a company and represents how much a company is worth in publicly traded markets. It shows the value at which investors place the company at a given time [12]. Also, the next best feature for Bitcoin is the btc-miners-revenue feature. This denotes the total value of coinbase block rewards and transaction fees paid to bitcoin miners. Also, it can be seen that feature 5 and feature 9 for Bitcoin give the least correlation with the bitcoin price fluctuations. Feature 5 is the btc-n-orphaned-blocks which denotes the total number of blocks mined but ultimately not attached to the main bitcoin blockchain and feature 9 is the btc-difficulty which is a relative measure of how difficult it is to find a new block. On the other hand, the next best feature for the Ethereum is feature 2 or the eth-address feature which is a cumulative measure of the growth of Ethereum address generation. Feature 5 or the eth-hashrate (hash rate in GH/s) and feature 7 or the eth-blocks (number of blocks processed per day) give the least correlation with the rise/fall of price for Ethereum.

Even though we have almost half the data points for Ethereum than for Bitcoin, almost all models work better for predicting its

Table 2: Accuracy and F1 Score of different classifiers

Currency	Classifier	Accuracy	F1 Score
Bitcoin	RF	0.795	0.798
Ethereum	RF	0.874	0.839
Bitcoin	SVM Linear	0.626	0.556
Ethereum	SVM Linear	0.720	0.713
Bitcoin	SVM RBF	0.545	0.508
Ethereum	SVM RBF	0.550	0.509
Bitcoin	NB	0.570	0.468
Ethereum	NB	0.496	0.417

price rise/fall. This may imply that the ethereum price is less volatile and more predictable than Bitcoin price. Figure 1 also shows us a glimpse of that. Which cryptocurrency to invest in is a whole new ball-game though. Though Ethereum seems to be steadier and more easily predictable, it is nowhere near the price of Bitcoin which even soared to $20,000 at one point. This concludes our experiments with the binary-datasets. Note that we cannot actually use these results in real life since the differential features (current day - previous day) of marketcap, miners-revenue or even address can only be obtained after the current day has passed. This experiment was done to gain an insight into what ticks these cryptocurrencies. The next experiment with time series datasets does the job of predicting future trends.

4 TIME SERIES EXPERIMENT

4.1 Making Bitcoin Dataset Stationary

The objective of a predictive model is to estimate the value of an unknown variable. A time series has time t as an independent variable and a target dependent variable. The output of the model is the predicted value for y at time t. Components of any time-series data is as follows-

(1) Trend: A trend exists when a series increases, decreases, or remains at a constant level with respect to time. Therefore, the time is taken as a feature.
(2) Seasonality: This refers to the property of a time series that displays periodical patterns that repeats at a constant frequency m.
(3) Cycles: Cycles are seasons that do not occur at a fixed rate.

A time-series (TS) is said to be stationary if its statistical properties such as mean, variance remain constant over time. Most of the TS models work on the assumption that the TS is stationary. So, from Figure 4, we can get a better insight into the data by looking at the rolling mean and variance. Intuitively, we can state that if a TS has a particular behaviour over time, there is a very high probability that it will follow the same in the future. Also, the theories related to stationary series are more mature and easier to implement as compared to non-stationary series.
We can test the stationarity of a dataset using augmented Dicky Fuller test. The augmented Dicky Fuller (ADF) test is a type of statistical test called a unit root test. The intuition behind a unit root test is that it determines how strongly a time-series is defined by a trend. There are number of unit root tests and ADF is one of the most widely used. Null hypothesis of the test is that the time series can be represented by a unit root that is not stationary. Alternative hypothesis of the test is that the time series is stationary. In Dicky Fuller test, if $P > 0.05$ then the data has a unit root and is non-stationary and if $P <= 0.05$, then reject the Null Hypothesis H_0, the data is stationary. For the original dataset, we obtained the value of P to be 0.99, which is greater that 0.05. Hence, our Bitcoin data is not stationary. There are three often used approaches to make time-series stable based on three difference scenarios, i.e., difference for linear trend, log for non-linear trend and log seasonal difference for seasonality. Here, we are using log seasonal difference. On applying log seasonal difference, we get the a pretty stable data with P value of 0.000014 as evident from Figure 5.

4.2 Holt Winters Forecasting Method on Bitcoin Dataset

Forecasting techniques can be largely classified as judgmental, univariate or multivariate. Univariate forecasts involve just one explanatory variable, whilst multivariate forecasts involve more than one explanatory variable. Holt-Winters is a univariate method. There is also a distinction between automatic and non-automatic

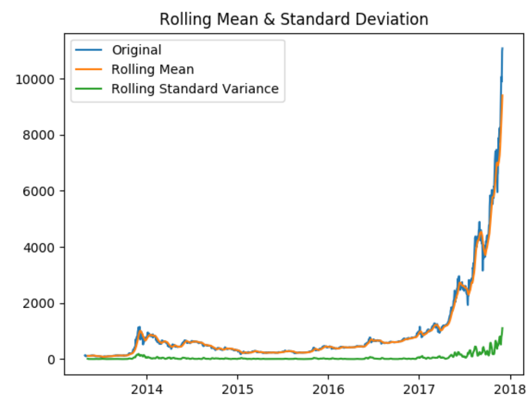
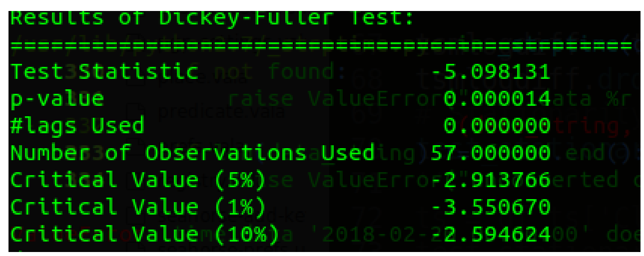


Figure 4: 12-month moving rolling mean and variance.



```
Results of Dickey-Fuller Test:
Test Statistic                  -5.098131
p-value                          0.000014
#lags Used                       0.000000
Number of Observations Used     57.000000
Critical Value (5%)             -2.913766
Critical Value (1%)             -3.550670
Critical Value (10%)            -2.594624
```

Figure 5: Dickey-Fuller test result for bitcoin dataset.

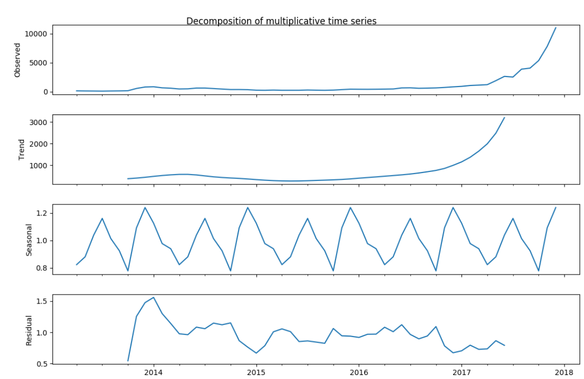


Figure 6: Multiplicative time-series decomposition for bitcoin dataset.

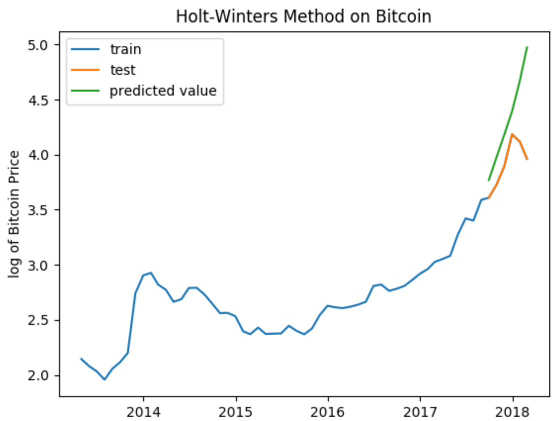


Figure 7: Holt Winters performance on Bitcoin Dataset

forecasting procedures. Non-automatic procedures require user intervention whilst automatic procedures do not. Holt-Winters is generally viewed as an automatic procedure but the user can intervene if required. The Holt method uses simple exponential smoothing in order to forecast. The forecast is obtained as a weighted average of past observed values where the weights decline exponentially so that the values of recent observations contribute to the forecast more than the values of earlier observations. Most time-series have three components: trend, seasonal and irregular. The irregular component is the residual after trend and seasonality have been removed. The Holt method accounts for only the trend and irregular components. The Holt-Winters method builds on this by allowing for seasonality. Forecasted values are dependent on the level, slope and seasonal components of the series being forecast. Hence, Holt-Winters method is suitable for data with trends and seasonalities which include a seasonality smoothing parameter . To analyze and compare the performance measure of different algorithm we used root mean square error (RMSE). In case of Holt Winters on Bitcoin dataset, we got an $rmse$ of 0.5073. Recall that we took the log of the whole dataset to make it stationary. Hence, $rmse$ is found for the log (Bitcoin close value). Figure 7 shows the Holt-Winter's performance on bitcoin dataset. So, the accuracy is approximately $(1 - rmse) * 100$.

4.3 ARIMA Forecasting Method on Bitcoin Dataset

ARIMA stands for auto-regressive integrated moving averages. The ARIMA forecasting for a stationary time-series is nothing but a linear equation. ARIMA models have three parts, although not all parts are always necessary. The three parts are the autoregression part (AR), the integration part (I) and the moving average part (MA). The predictors depend on the parameters (p, d, q) of the ARIMA model. The parameter p indicates the number of AR terms, q indicates the number of MA terms and d indicates number of differences. The main assumption surrounding the AR part of a time-series dataset is that the observed value depends on some linear combination

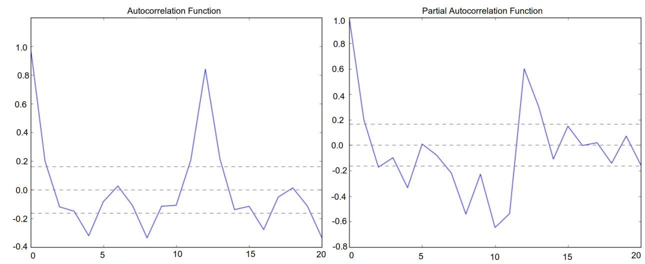


Figure 8: Autocorrelation and partial autocorrelation function.

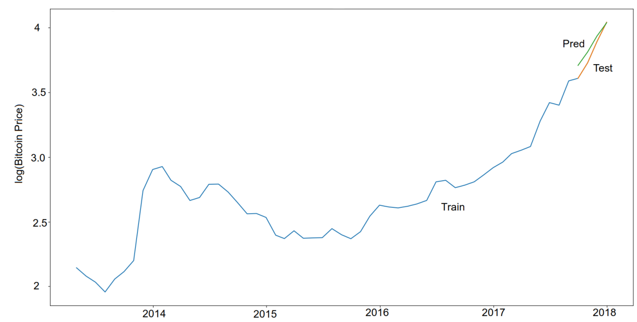


Figure 9: ARIMA model performance on the bitcoin dataset.

of the previously observed values up to a defined maximum lag (denoted p), plus a random error term ϵ_t . The main assumption surrounding the MA part of a time-series is that the observed value is a random error term plus some linear combination of the previously random error terms up to a defined maximum lag (denoted q). An importance concern here is how to determine the value of p and q. We use two plots to determine these numbers. Autocorrelation function (ACF) is the measure of the correlation between the the TS with a lagged version of itself. Partial autocorrelation function (PACF) is the measure of the correlation between the TS with a lagged version of itself but after eliminating the variations already explained by the intervening comparisons. In the plot, the two dotted lines on either sides of 0 are the confidence intervals. p is the lag value where the PACF chart crosses the upper confidence interval for the first time and q is the lag value where the ACF chart crosses the upper confidence interval for the first time. Based on the optimal values of p and q, obtained from the Figure 8, the ARIMA model is trained with $p = 2$ and $q = 2$. Figure 9 shows the performance of our ARIMA model on the bitcoin dataset. The $rmse$ for ARIMA with $p = 2$ and $q = 2$ is 0.4659.

The above analysis is repeated for the ethereum dataset. The $rmse$ value for Ethereum dataset in case of Holts Winter and ARIMA is 0.2634 and 0.1942 respectively.

5 COMPARISON WITH REGULAR STOCK MARKET

We use the stock price dataset of Google to highlight the differences between the cryptocurrency and the regular stock market. The

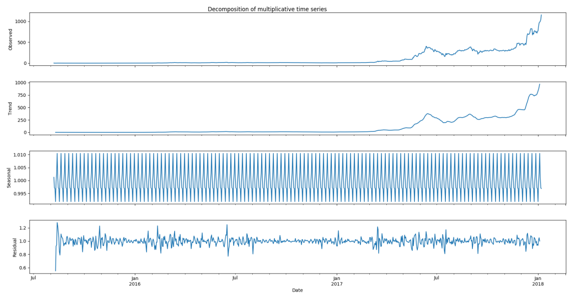


Figure 10: Multiplicative time-series decomposition for the bitcoin dataset.

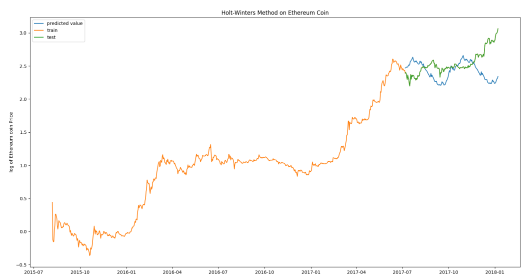


Figure 11: Holt's Winter's performance on the Ethereum dataset.

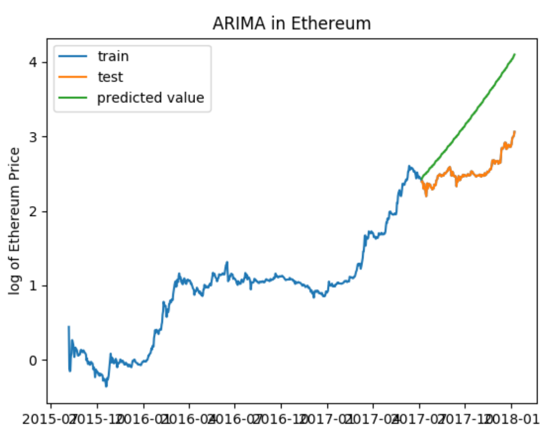


Figure 12: ARIMA model performance on the ethereum dataset.

dataset [8] description is tabulated in Table 3 as we did for the cryptocurrency dataset in Table 1. The first distinguishing point here is that the mean coincides with the 50% mark, a feat that was far from true for cryptocurrencies. There is a steady trend which

Table 3: A simple description of a subset of the dataset features.

Parameter	Open	High	Close
Mean	682.35	687.36	682.23
Std	187.40	188.53	187.57
Min	384.96	390.16	383.34
25%	543.66	547.58	543.02
50%	651.57	658.25	652.47
75%	805.96	810.73	806.40
Max	1188.00	1198.00	1187.56

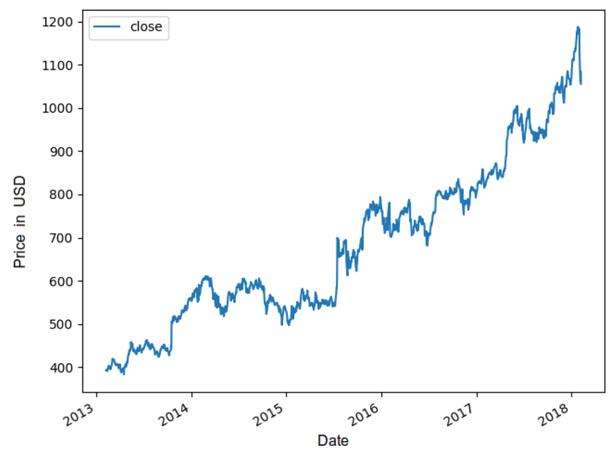


Figure 13: Date vs. Google stock price plot for a period of five years.

can be seen from the min to 25% to 50% to 75% to max row. The price plot figure of Google shows a clear trend as we predicted from the table. From just one look we can say with confidence that even a simple linear regression model can do a good job on this dataset. The *rmse* value for ARIMA and Holt Winters is 0.6957 and 0.6475 respectively.

6 CONCLUSION

In this paper, we performed two different experiments with the cryptocurrency datasets on Bitcoin and Ethereum. In the first experiment, we framed a binary classification problem to see what features were actually responsible for the rise and fall of the price of cryptocurrencies. In the next section, we attempted to use various time-series models to predict these prices. Lastly, a juxtaposition of the regular stock prices and cryptocurrency prices revealed just how volatile the latter is as compared to the former.

7 ACKNOWLEDGEMENT

The authors would like to thank Prof. Kamalakar Karlapalem for his guidance and support for not only this project, but throughout the DWDM course.

REFERENCES

[1] M. Amjad and D. Shah. 2016. Trading Bitcoin and Online Time Series Prediction. *NIPS Time Series Workshop* (2016), 1–15.

[2] Jonathan T. Barker. 2017. Why Is Bitcoin's Value So Volatile? *Online Article, Investopedia* (Dec. 2017). https://www.investopedia.com/articles/investing/052014/why-bitcoins-value-so-volatile.asp

[3] A. Bouveret and V. Haksar. [n. d.]. What Are Cryptocurrencies? *Finance and Development, IMF* 55, 2 ([n. d.]).

[4] D. Garcia and F. Schweitzer. 2015. Social Signals and Algorithmic Trading of Bitcoin. *Royal Society Open Science* 2, 9 (2015).

[5] D. Garcia, C. J. Tessone, P. Mavrodiev, and N. Perony. 2014. The Digital Traces of Bubbles: Feedback Cycles Between Socio-economic Signals in the Bitcoin Economy. *Journal of The Royal Society Interface* 11, 99 (2014).

[6] Young B. Kim, Jun G. Kim, W. Kim, Jae H. Im, Tae H. Kim, Shin J. Kang, and Chang H. Kim. 2016. Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies. *PLOS ONE* 11, 8 (2016).

[7] S. Kumar. 2018. Cryptocurrency Historical Prices Dataset. *Kaggle* (Feb. 2018). https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory/version/13

[8] C. Nugent. 2018. SP 500 Stock Data. *Kaggle* (Feb. 2018). https://www.kaggle.com/camnugent/sandp500

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[10] P. Pryzmont. 2016. An Empirical Study of how Bitcoin Related Incidents Impact its Price Volatility. *MS Thesis, National College of Ireland* (Aug. 2016).

[11] D. Shah and K. Zhang. 2014. Bayesian Regression and Bitcoin. *ArXiv* (2014). https://arxiv.org/abs/1410.1231

[12] G. Terzo. 2018. Market Value vs. Market Capitalization. *Investing for Beginners, Zacks* (March 2018). https://www.imf.org/external/pubs/ft/fandd/2018/06/what-are-cryptocurrencies-like-bitcoin/basics.htm