Authors: Han Zhang, Qian Zhou

Professor: Dr. Pablo Roldan

Start Date: June 23th, 2021

Change Point Detection methods applied to Financial Time Series

Research Report Document

Introduction

Time series analysis is important in lots of fields including medicine, finance, business, and entertainment. Time series analysis, widely used in technical trading today, focuses on single security over a period of time. Time series data are sequences of data points that occur in successive order and change over some period of time. These behaviors can change over time due to external events and/or internal systematic changes in dynamics/distribution. In the financial field, a time series tracks the movement of the chosen financial data points, such as a security's, stock's, or currency price, over a specified period of time with data points recorded at regular intervals.[1][2] Change-point detection is an important part of time series analysis, as the presence of a change point indicates an abrupt and significant change in the data generating process.

In this project, we apply change point detection methods to financial time series. Since financial crashes occur with very little warning and the noisy and non-stationary characteristics of financial data, our goal is to record observations that undergo a change in their distribution due to changes/disorders in the environment. If the state changes, we detect it as soon as possible, while minimizing false detections. We wish to analyze the distributional changes and detect the change point in real-time of Bitcoin prices and other cryptocurrencies prices, including Ethereum, Litecoin, and Ripple, using changepoint detection methods.

Part one: Change Point Detection of Bitcoin and other Cryptocurrencies

In the usual setting, a sequence of observations x_1, x_2,... is drawn from the random variables X_1, X_2,... and undergoes one or more abrupt changes in distribution at the unknown change points τ_1, τ_2,.... It is usually assumed that the observations are independent and identically distributed between every pair of change points so that the distribution of the sequence can be written as:

$$X_i \sim \begin{cases} F_0 & \text{if } i \leq \tau_1 \\ F_1 & \text{if } \tau_1 < i \leq \tau_2 \\ F_2 & \text{if } \tau_2 < i \leq \tau_3, \\ \cdots & \end{cases}$$

where the F_is represents the distribution in each segment.[3] When there is a point τ_i in the time series, such that the data before and after it has the greatest likelihood of belonging to different distributions F_{i-1} and F_i, then we believe τ_i is a change point. In order to constrain the likelihood of detecting there is a change point where there is none, we let the probability of identifying a change of distributions where there are actually no changes, i.e. Type I errors, to be below a level of significance, defined as $\boldsymbol{\alpha}$.

Since it's hard to analyze and compare different cryptocurrency prices directly, we first convert the Bitcoin prices time series to a Bitcoin daily log-returns time series. Log Return is one of three methods for calculating return and it assumes returns are compounded continuously rather than across sub-periods. It is calculated by taking the natural log of the ending value divided by the beginning value, which is additive. Some properties of additive time series are easier to derive than multiplicative processes. [4] lists some additional desirable and undesirable properties of log returns.

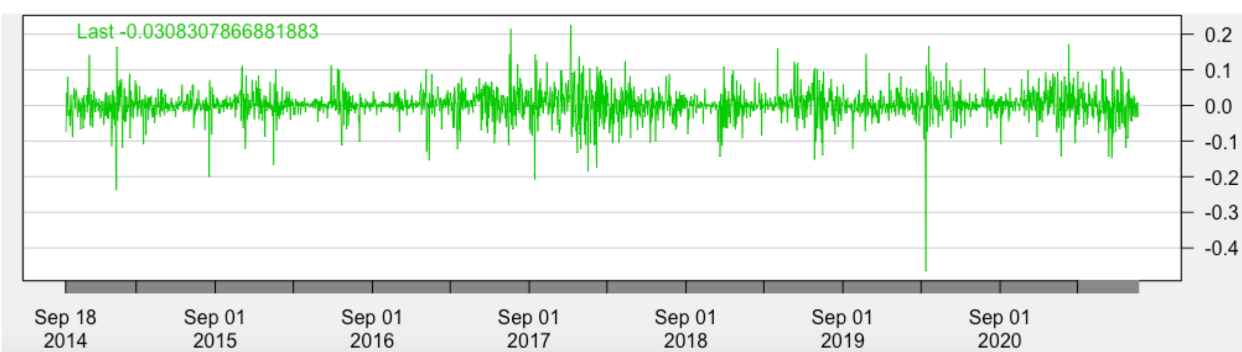


Figure 1. Log-return of bitcoin from 2014-09-17 to 2021-06-29

A. Univariate Time Series of Bitcoin

A changepoint occurs when the time series before and after it are distributed differently. Batch detection and sequential detection are two types of change point detection. For Batch detection, there is a fixed-length sequence consisting of n observations from the random variables $X_1,...,X_n$, and it is required to test whether this sequence contains any change points. When using batch detection, the decision whether a change has occurred at a particular point in the sequence is made using all the available observations, including those which occur later in the sequence. For Sequential detection, the sequence does not have a fixed length. Instead, observations are received and processed sequentially over time. When each observation has been received, a decision is made about whether a change has occurred based only on the observations which have been received so far. This method works better than batch when we are still receiving incoming data, and the time series is not yet complete. If no change is flagged, then the next observation in the sequence is processed. The sequential formulation allows sequences containing multiple change points to be easily handled; whenever a change point is detected, the change detector is simply restarted from the following observation in the sequence. [3] We expect 1/alpha observations before a false alarm for sequential detection. Both of these methods are univariate, and we use the cpm R package to implement them. We use recursion in each half of the changepoint detected for batch to find less significant changepoints, and we recurse after the changepoint found for sequential to find the subsequent changepoints.

The R package cpm contains an implementation of several different CPMs. Detecting changes in sequences of Bitcoin daily log-returns time series requires nonparametric statistics and can be deployed on any stream of continuous random variables without requiring any prior knowledge of their distribution. We use Cramer-von-Mises, Mood, and Mann-Whitney statistics to detect general changes in distributions, and variance and mean changes, respectively.

In order to obtain robust results, we conduct the experiments for batch detection respectively using the level of significance $\boldsymbol{\alpha}$ = 0.01, 0.05, 0.1 from 2014-09-17 to present 2021-06-29. When the alpha is smaller, the change point exists with a high confidence level, but the delay time to detect the change point is longer. On the contrary, when the alpha is larger, we can detect the change point faster, but the probability of false alarm is bigger.

Batch Detection: Change Points and Threshold

Cramer-von-Mises

Mood

Mann-Whitney

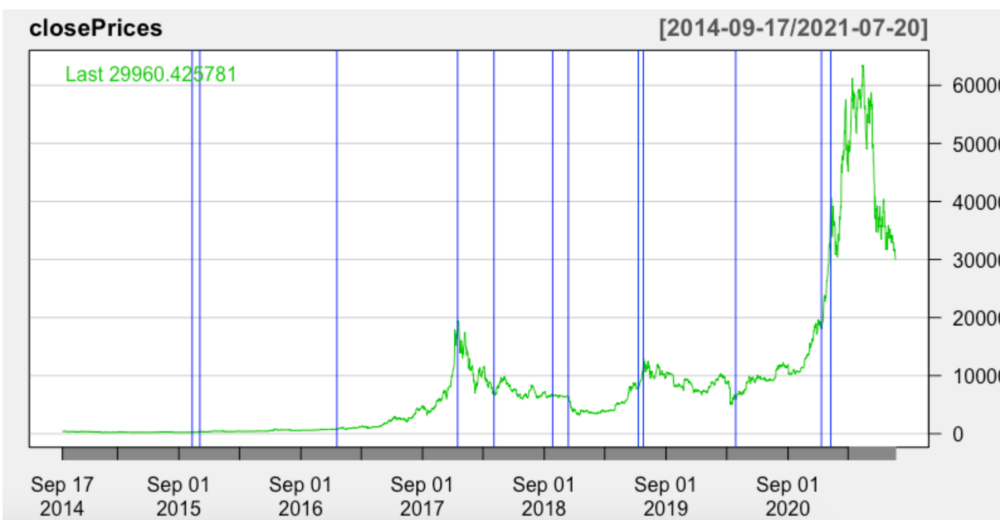
Figure 2. Bitcoin change points were identified by batch detection using cpm R Cramer-von-Mises, Mood, and Mann-Whitney methods from 2014 to the present. The green line is the price of bitcoin in

the US dollar from Yahoo from 2014-09-17 to 2021-08-19. The vertical blue lines indicate all the change points were identified using $\alpha = 0.01$. We did not detect any change point using Mann-Whitney because the threshold is relatively high. The changepoints for $\alpha = 0.05, 0.1$ are graphed in Appendix D. For all detailed change point time, please see Appendix table A.

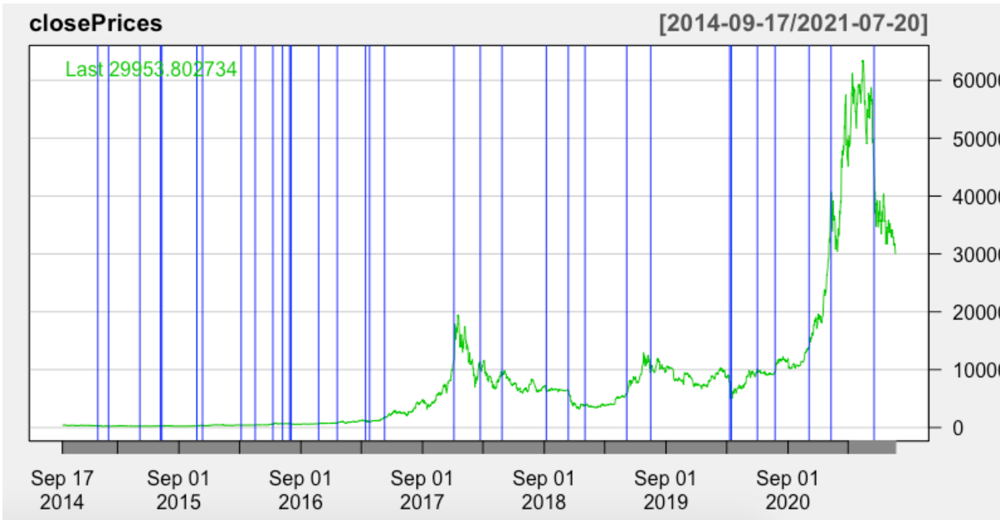
For sequential, The false alarm rate is constant over time: assuming that there is no change point, at each time instant t, the probability to raise a false alarm, i.e. wrongly detect there is a change point, is kept constant= alpha. This implies that the expected number of observations received until a false alarm is raised is $1/\alpha$. This quantity is referred to as the average run length or ARL_0. Long ARL means there is a low possibility of false alarm rate α, but it's a less sensitive detection method. Short ARL means it's a more sensitive detection that has fast detection times, but there is a high possibility of false alarm rate α. We use $ARL=1/\alpha = 1000, 900, 800$ respectively. Results for the latter two are in the appendix.

Sequential Detection

Cramer-von-Mises



Mood



Mann-Whitney

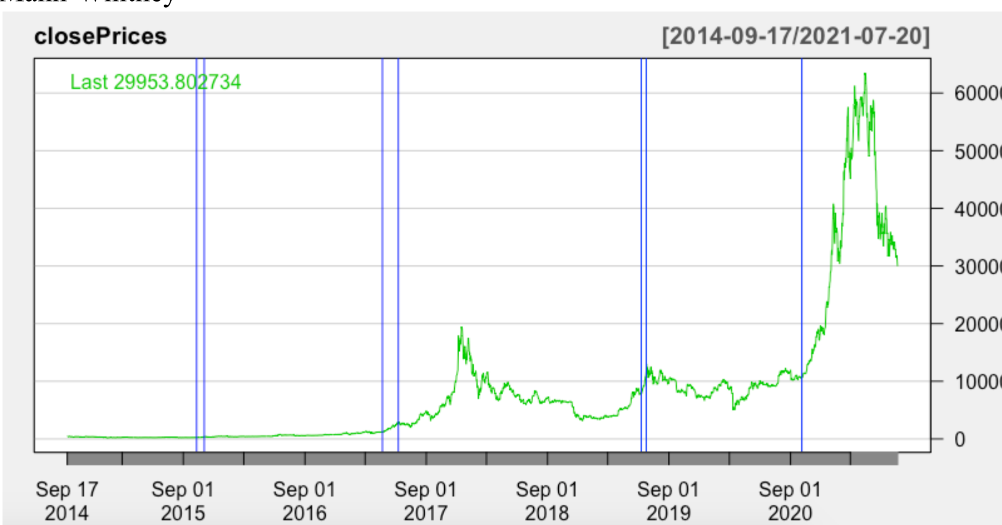


Figure 3. Bitcoin change points were identified by sequential detection using cpm R Cramer-von-Mises, Mood, and Mann-Whitney methods from 2014 to the present. The green line is the log-return calculated by the price of bitcoin in the US dollar from Yahoo from 2014-09-17 to 2021-06-29. The vertical blue lines indicate all the change points were identified using ARL = 1000. For figures of ARL = 900, 800, and all detailed change point time, please see Appendix table B.

In order to more intuitively see Changepoint's detection of mean and variance changes, we calculate the log-return of bitcoin's mean and variance of each interval between the two change points detected by Mann-Whitney and Cramer-von-Mises Sequential methods with ARL0=1000.

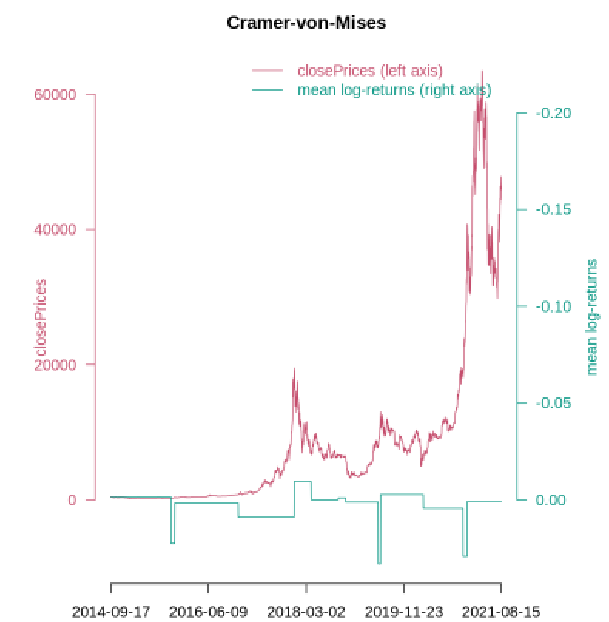
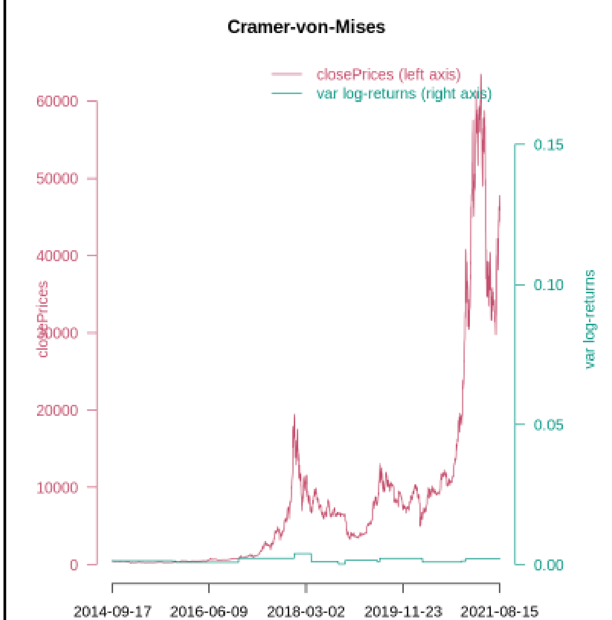
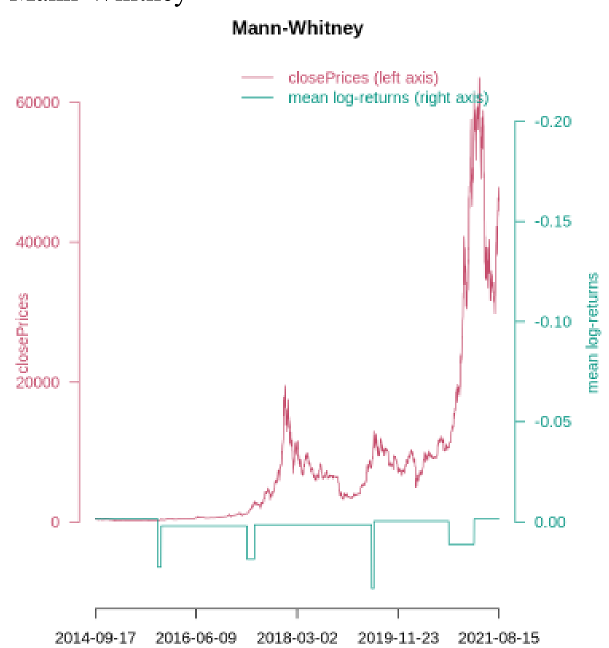
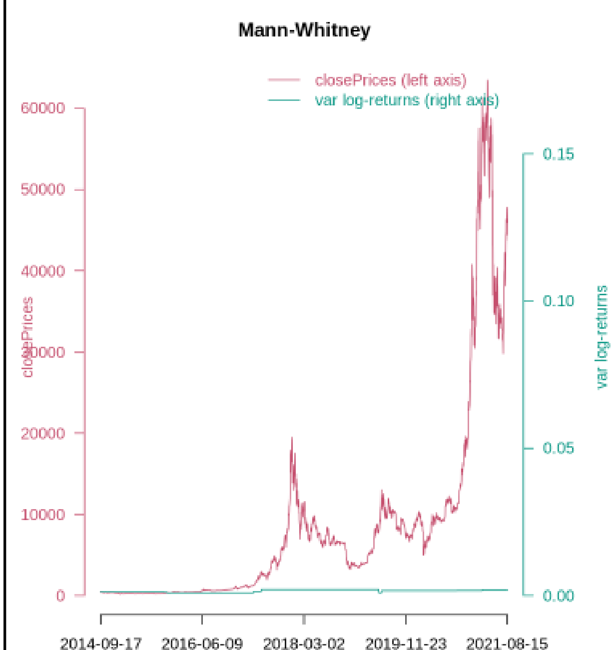
Mean	Variance
	
Mann-Whitney 	

Figure 3b. The Green lines are means and variances of the Bitcoin log-returns in between changepoints detected by Mann-Whitney and Cramer-von-Mises Sequential CPD methods with ARL0=1000. The red lines are prices, plotted for reference. Note that the means are positive when they are below the zero line.

Looking at the changepoints for Mann-Whitney ARL0=1000, and the means in between, it seems that the mean log-returns are quite large in the short periods in 2015, 2017, and 2019, suggesting that they are short periods of growth, followed by periods of slower growth or decay in the case of the first and last periods, and the one from 2019 to 2020. The period from 2017 to 2019 seem to correspond with the 2017 boom and the lasting effects of the crash, and it seems the positive effects of the boom have averaged out the negative ones from the crash to produce a small positive average return.

We also see that the changepoints for sequential Mood, ARL0=1000, seem to be a superset of the changepoints for Batch mood, alpha=0.01.

Elisheva's Method

Elisheva (private communication) proposed a method similar to that of Islambekov et al. [10] discussed in Part Three, in which we focus our change point detection method on the time series generated by finding the variance of discrete windows with the size of *w* sliding over the time series. She argues that this method is more sensitive to changes in variance in a time series. Applying this method with a window size of 5, we were able to find identical changepoints using Cramer-von-Mises and Mann-Whitney using the Batch method (and none using Mood) for alpha = 0.01. For Sequential (ARL0=50000), the results are almost identical for Cramer-von-Mises and Mann-Whitney. These results suggest sudden increases in variance during the 2017-2018 crash, as well as periods of lower but still large variance before and after the crashes. The first two periods and the period after the 2018 crash are periods of greater stability.

(data below are calculated from log-returns directly, not from preprocessed)

mean -0.00321706757584759, variance 0.00189656428943762, from 2014-09-19

mean 0.00238447514393266, variance 0.000852651122272751, from 2015-04-01

mean 0.00510775880763427, variance 0.00308186995312668, from 2017-05-05 mean 0.000759686795768944, variance 0.0013513588779323, from 2018-04-30

mean 0.0035743370308594, variance 0.00198729521452346, from 2020-11-25

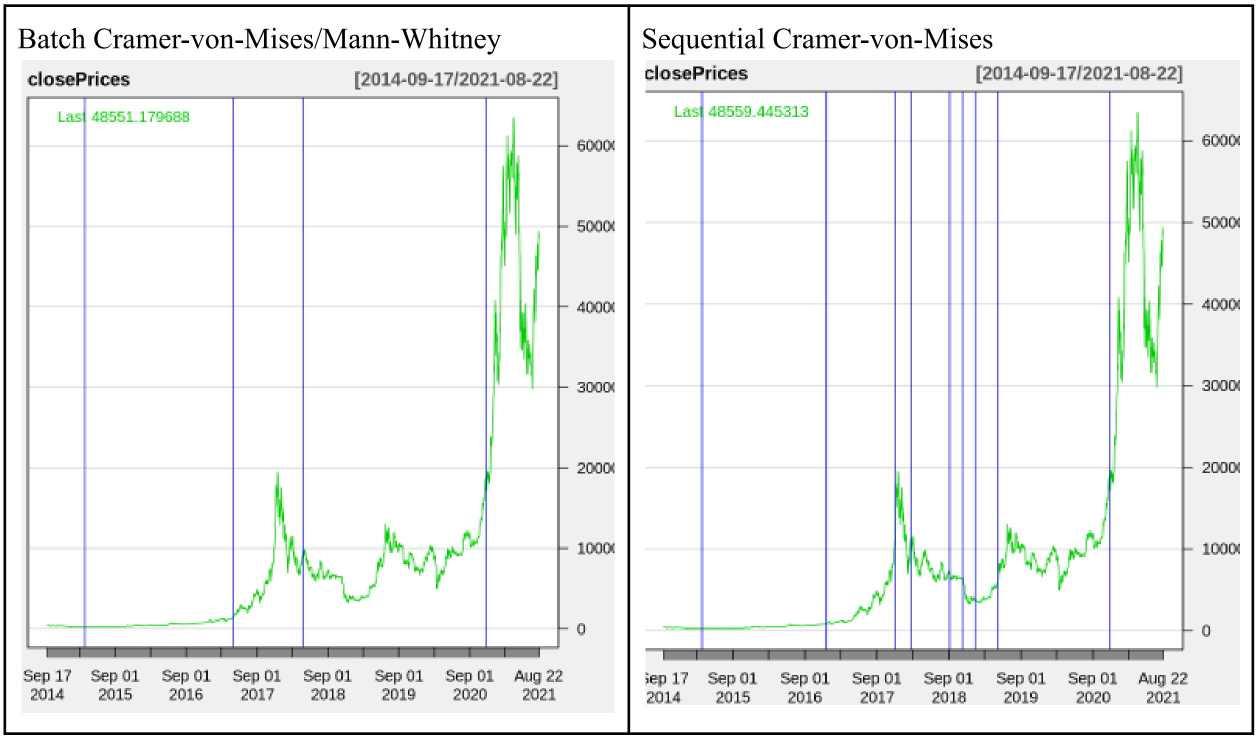


Figure 4. Applying Elisheva's Method to find change points of the log-returns time series preprocessed by finding the variances of windows of size 5. The Sequential trials have the largest allowed ARL0=50000, and the Batch one has alpha=0.01.

We observe that the changepoints detected using Batch Cramer-von-Mises/Mann-Whitney seems to be almost a subset of the changepoints detected using Batch Mood, for alpha = 0.01. This might suggest that changes in variance in the original series had transformed to changes in mean in the new series of variances.

B. Multivariate Time Series of Cryptocurrencies

Since the launch of Bitcoin in 2009 as the first decentralized cryptocurrency, many other cryptocurrencies have been created, some important ones such as Etherium, Litecoin, and Ripple, etc.

Analyzing the entire cryptocurrency market through the prices of multiple cryptocurrencies is helpful in understanding the dynamics of cryptocurrencies.

We convert four time series, including Bitcoin, Etherium, Litecoin, and Ripple, into log-return times series, then detect change points on this combined mixed log-return time series to represent the cryptocurrency market price change points. The ecp R package is able to perform multiple change-point analyses for multivariate time series. The method is able to estimate multiple change-point locations, without a priori knowledge of the number of change points. The procedures assume that observations are independent with finite αth absolute moments, for some $\alpha \in (0,2]$. We use the e.divisive method in the ecp R package which can compare permutations of the time series data before and after a point, finding the point with the greatest difference between its left and right subseries. An iterative approach similar to our recursion in batch detection is applied. Since the running time is quadratic, the method can become pretty slow as the data increases in size. [5] We generate the multivariate time series by combining the normalized log-return time series for Bitcoin, Etherium, Litecoin, and Ripple from 2015-08-07 to the present, normalizing with a min-max scaling of the time series (i.e. (x - min(x))/(max(x)-min(x)))). We do not normalize the prices directly since we need to take subsequent log returns, and normalization of prices would make all logs be between [-inf, 0], including -inf, which is not desirable.

Change-point and Log-Return of Cryptocurrencies

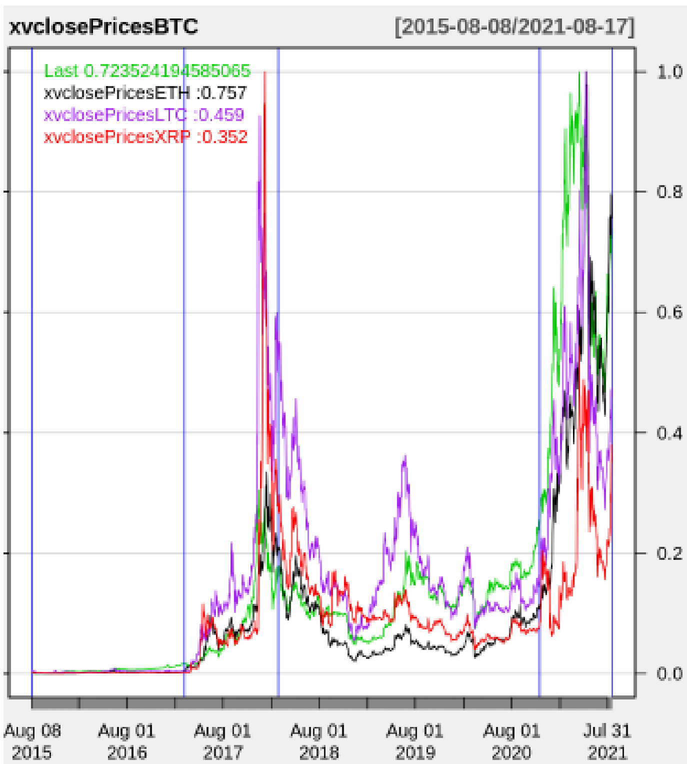
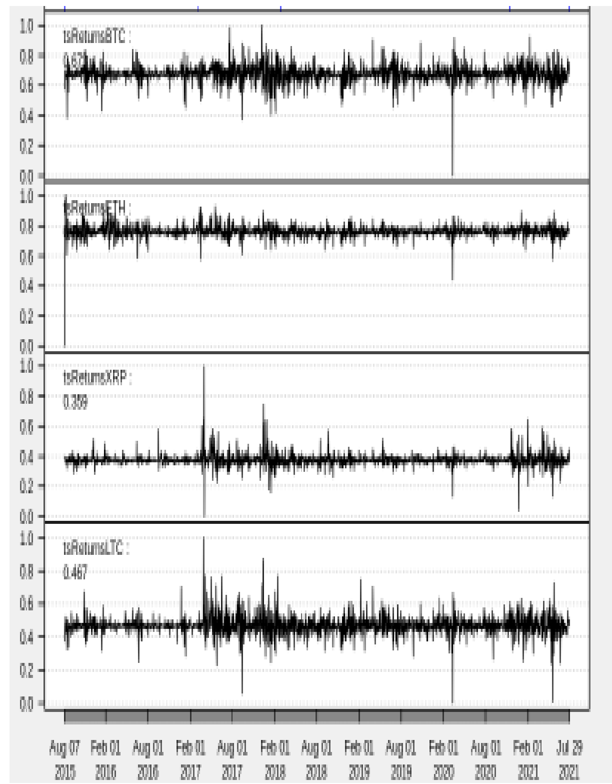
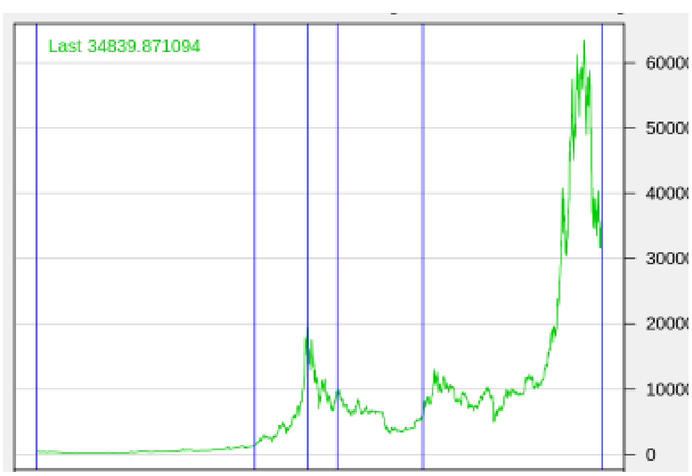
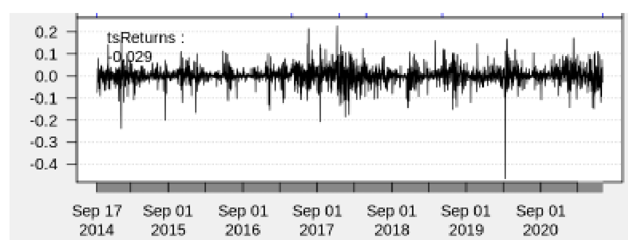
	Price	Log-Return(s)
ECP multivariate (2015-08-07/2021-06-30) (updated)	(normalized prices plotted; not normalized in calculations of CPs) 	
ECP univariate (with bitcoin) (2014-09-17/2021-06-30)		

Figure 5. Bitcoin change points were identified by ecp R package e.divisive method using multivariate time series and univariate time series from 2015 to the present and Log-Return of Bitcoin. The first row shows the change point of bitcoin by detecting the multivariate time series combined by Bitcoin (BTC), Etherium (ETH), Ripple (XRP), and Litecoin (LTC) normalized log-return time series from 2015-08-07 to 2021-06-29, shown in that order. For the normalized prices plot of the four currencies, Bitcoin is green, Etherium is black, Ripple is purple, and Litecoin is red. The second row in the table shows the change point of bitcoin by detecting the univariate time series of bitcoin from 2014-09-17 to 2021-06-29. The green line is the log-return calculated by the price of bitcoin in the US dollar from Yahoo from 2014-09-17 to 2021-06-29.

We observe that both have detected a changepoint around the time before the 2017 boom, similar to the Mann-Whitney ARL0=1000 Sequential detection discussed above. Both also share one after the crash, though E.divisive seem to have found ones closer to the crash in contrast with Mann-Whitney whose one is found near the end of the downward price movement in 2019.

E.divisive multivariate found a changepoint late 2020 but did not find the 2019 changepoint found by univariate. This is perhaps because Ethereum and Ripple do not have the sustained price found in Bitcoin between 2019 and 2020, but only the spike in prices in 2020-201, as can be seen in the Figure 5.

We use the data from 2015-08-07 to 2021-06-29 to detect the mixed time series, since Etherium is available only after 2015-08-07. As a point of comparison, we also run e.divisive on Bitcoin from 2014-09-17. We find out that the detection of the longer univariate time series takes longer than shorter multivariate time series using the e.divisive method, for instance, univariate with only bitcoin (with earlier start date) takes about 24.8 minutes, while multivariate only takes 18.9 minutes. Since multivariate cannot reasonably be plotted, we only plot the price of bitcoin with the changepoints.

Part Two: Socioeconomic behind Bitcoin Change-point

Socioeconomic events can be the main drives behind Bitcoin dynamics, such as new regulations, robberies from exchanges, and even rumors spreading on Twitter. Any event may cause cryptocurrency market turmoil, or even lead to a financial crash. Having analyzed the main bitcoin change point, it is useful to put them into context and expose their key drivers, as well as the developments and events that promoted their nucleation or caused their sudden crashes.

Since there were too many sets of data, we first chose the data we detected from the Cramer-Vince-Mises method to analyze. Because it analyzes general changes in the distribution, without the knowledge of the distribution, it is a more general representation. For batch detection, we first look at the data when alpha=0.01, which has the highest confidence level out of all the alpha. We detected a total of three change-points, 2016-02-07, 2016-12-21, and 2018-03-04. The mean and variance of each interval between change points are as follows:

mean -0.000383505963917142, variance 0.00132558366449342, from 2014-09-19

mean 0.00228353427160357, variance 0.000505762125677388, from 2016-02-07

mean 0.00609053255317596, variance 0.00267865648252861, from 2016-12-21

mean 0.00102443609632902, variance 0.00153022878192985, from 2018-03-04

In November 2016, the Swiss Railway operator SBB (CFF) upgraded all their automated ticket machines so that bitcoin could be bought from them using the scanner on the ticket machine to scan the bitcoin address on a phone app. On 26 January 2018, Coincheck, Japan's largest cryptocurrency OTC market, was hacked. 530 million US dollars of the NEM were stolen by the hacker, and the loss was the largest ever by an incident of theft, which caused Coincheck to indefinitely suspend trading. On 7 March 2018: Compromised Binance API keys were used to execute irregular trades. After that, during Late March 2018, Facebook, Google, and Twitter banned advertisements for initial coin offerings (ICO) and token sales. Those all led to the 2018 cryptocurrency crash (also known as the Bitcoin crash and the Great crypto crash) was the sell-off of most cryptocurrencies from January 2018.[6][7]

For Sequential detection, when we use ARL=40000, which is when the confidence value is the highest and the false alarm is the lowest. We detect a total of 3 change points: 2017-04-24, 2018-04-29, and 2019-06-12, dividing the series into four periods. Between 2017-04-24 and 2018-04-29, the average and variance log-returns were the largest, around 0.0055 and 0.0030, respectively, suggesting more market volatility. This roughly correlates with the cryptocurrency 2017-boom and 2018-crash described in [7]. The fourth and last period also corresponds with a boom and crash, and has variance and mean log-returns higher than the two calmer periods (the first and third).

The ensuing period from 2018-04-29 to 2019-06-12 is the only period with negative average log-returns of -0.0004, and has variance very close to the first period from 2014-09-17 to 2017-04-24, below 0.0011. The negative returns may be related to the setbacks due to advertisement bans, hacks, and crackdowns [8] earlier in the year, and these events might have also resulted in the large variance of the previous period. Effects of the crash continued to the end of 2018, though the period is relatively more calm. If we take the interpretation from the Fear-and-Greed Index described below, the 2019 changepoint may be explained by the data suggesting that the investors are finally at a point where they are worried enough that they could hardly be more worried, so that the price could rise.

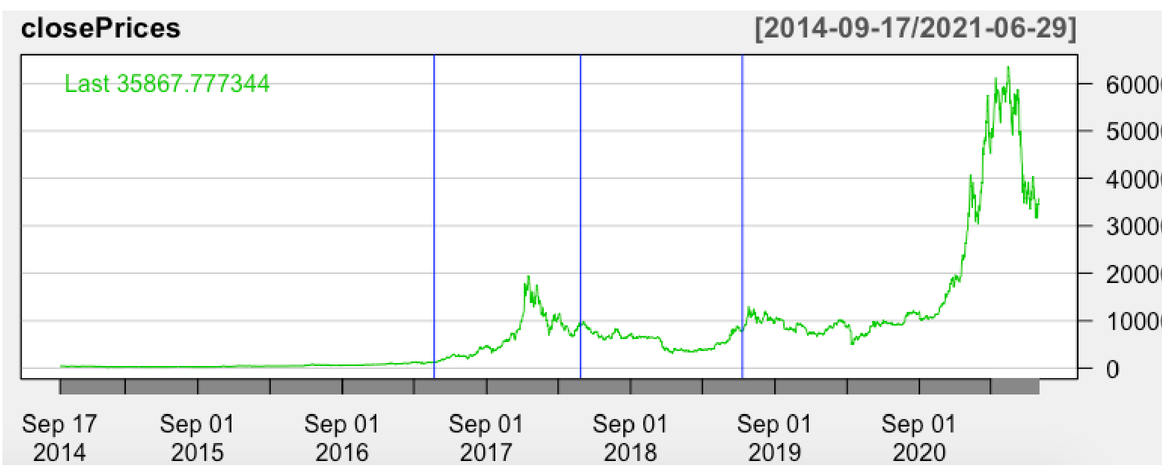


Figure 6. The changepoints detected using Sequential Cramer-von-Mises for ARL=40000.

The mean and variance of each interval between change points are as follows:

2014-09-18 to 2017-04-24 Calm period:

mean: 0.00102281680569398, variance 0.00107358674118345

2017-04-24 to 2018-04-29: Dangerous/bull period:

mean: 0.00553216738823242, variance 0.00301944343273643

2018-04-29 to 2019-06-12 Calm/bear:

mean: -0.000403053904460364, variance 0.00108104997453162

2019-06-12 to present: higher risk, slightly higher returns:

mean: 0.00197873658732267, variance 0.00171304592845981

We note that a more continuous analysis of market sentiments might reveal other interpretations of the changepoints and the periods between them, instead of a rough look at significant events. For example, comparing the Fear and Greed Index hosted by the website Alternative.me (https://alternative.me/crypto/fear-and-greed-index/) and the Sequential Cramer-von-Mises (ARL=1000) changepoints after 2018 we found above seems to suggest the changepoints are near the extrema/turning points of the Greed/Fear values, especially the 2018-04-29 and 2019-06-12.

The low values represent "Fear", and high values "Greed", according to the website. "Fear" corresponds with sudden rise in volatility, rise of Bitcoin share of the crypto market, or increase in worried searches in Google Trends, and "Greed" corresponds with "high buying volumes in a positive market", high interactivity with a coin and its hashtag in Twitter, or increasing market share for an alt-coin. The authors argue that "Fear" could be a buying opportunity, and "Greed" signals that the market is due for correction.

A more exact version (no data before Feb 2018, so I made them 0) is below, for ARL = 1000 and 40000 respectively.

Sequential Cramer-von-Mises, ARL=1000	Sequential Cramer-von-Mises ARL=40000

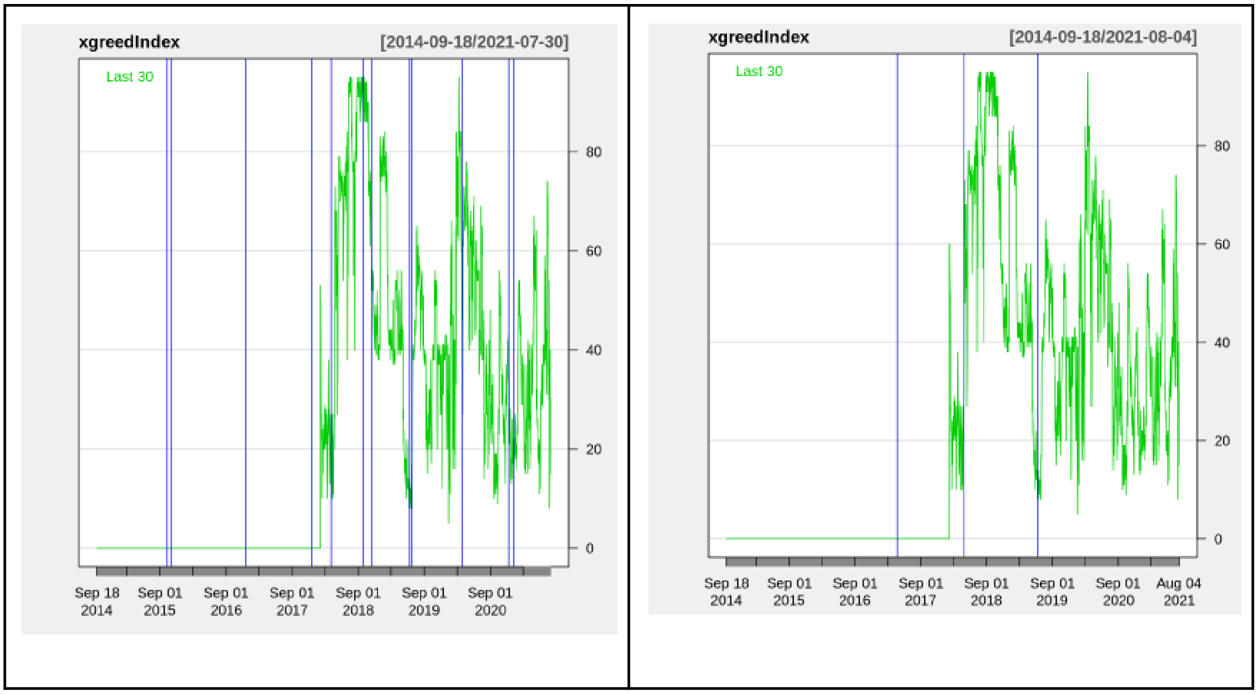


Figure 7. The changepoints we found from investigating the log-returns of the closing prices of Bitcoin in the context of the Fear and Greed Index.

Part Three: Topological Data Analysis and CPD

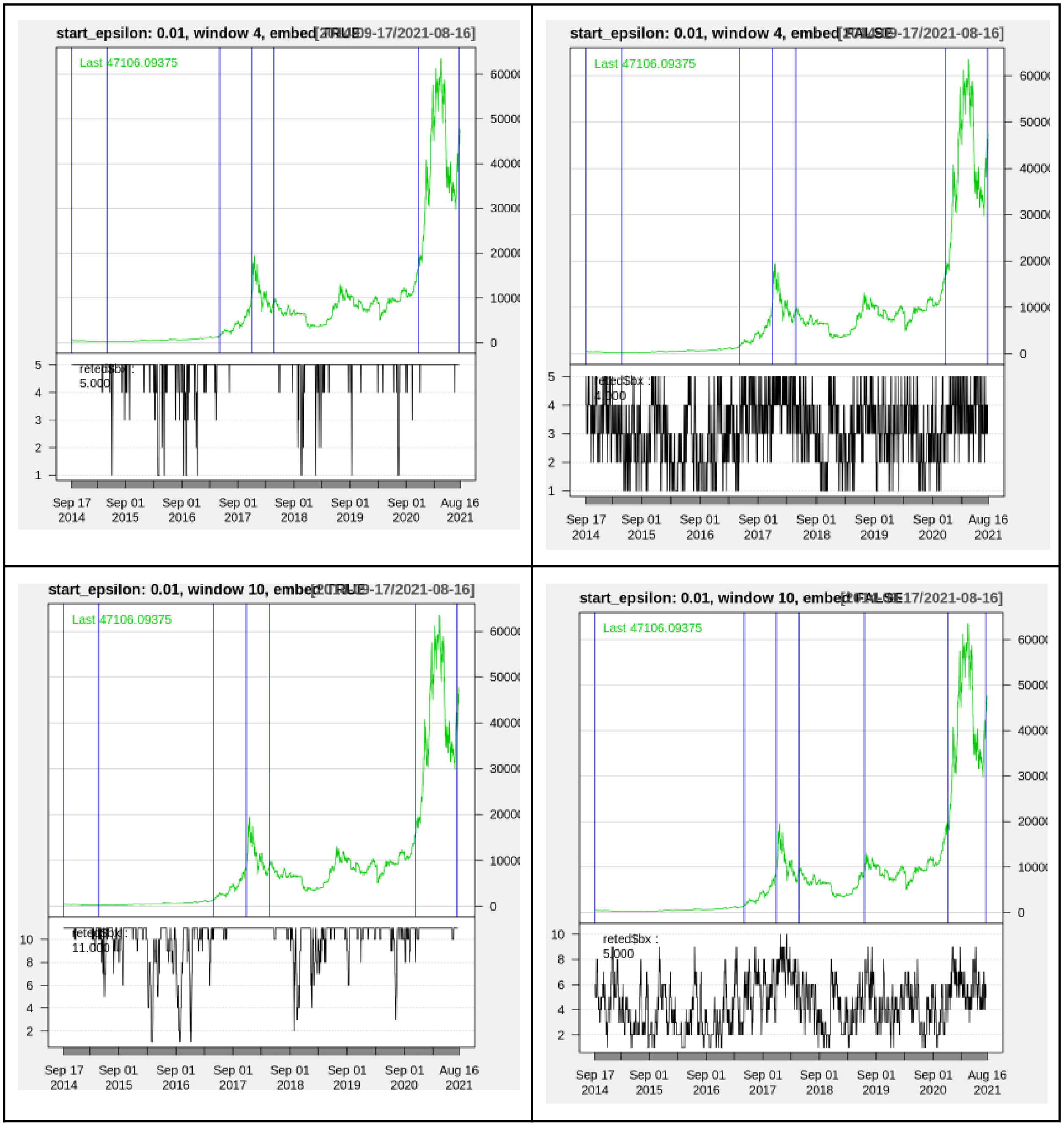
Islambekov et al. [10] introduced the idea of using Topological Data Analysis (TDA) as a tool to enhance changepoint detection. The basic idea is to first apply TDA to get a sequence of betti numbers, which measures the "n-dimensional holes" of the topology of the data, and then estimate the changepoints of the new resulting time series.

If the given data is univariate, then we might want to embed the data in higher dimensional spaces (here we are doing 3 dimensional space) by moving a window of the size of the embedding dimension across the time series to generate the points in the higher dimensional space. Then we move a window of size n across the resulting time series to get sets of points each with cardinality n (since indexing in R is inclusive, the window size in the code should be n-1).

We impose a simplicial complex on each set using a specific epsilon value, such that the mutual distance between every point in a simplex is less than epsilon. This is called a Rips complex, and we use the Rips diagram, which actually calculates a sequence of such complexes, called a Rips filtration, in the TDA R-library for such calculations. Naturally, for different epsilons, the betti numbers for simplicial complexes would differ based on the value of the epsilon. As Islambekov et al. observed, for small window sizes, topological features of dimension greater than 0 are not frequently observed, so we limit ourselves to 0th dimensional betti numbers. We compute the 0-th dimensional betti numbers for a range of epsilons using a function the CosmoBetti package (https://rdrr.io/github/gonzalezgouveia/CosmoBetti/man/compute_betti_number.html), and produce a multivariate time series, each time has a vector of 0 betti numbers for standard intervals of epsilons up to epsilon = 0.5.

Finally, we apply the e.divisive method to the time series to calculate the 5 most significant changepoints. This bound is arbitrary since the calculation might produce a great number of changepoints, which costs a lot of time and is not feasible for analysis.

Below are the results for embedding dimension = 3 and 1, and window size = 5 and 11, with 50 epsilons (starting from 0.01) or 5 epsilons (starting 0.1), producing a total of 8 graphs.



start_epsilon: 0.01, window 4, embed TRUE
Last 47106.09375
start_epsilon: 0.01, window 4, embed FALSE
Last 47106.09375
start_epsilon: 0.01, window 10, embed TRUE
Last 47106.09375
start_epsilon: 0.01, window 10, embed FALSE
Last 47106.09375
reted$bx :
Sep 17 2014
Sep 01 2015
Sep 01 2016
Sep 01 2017
Sep 01 2018
Sep 01 2019
Sep 01 2020
Aug 16 2021

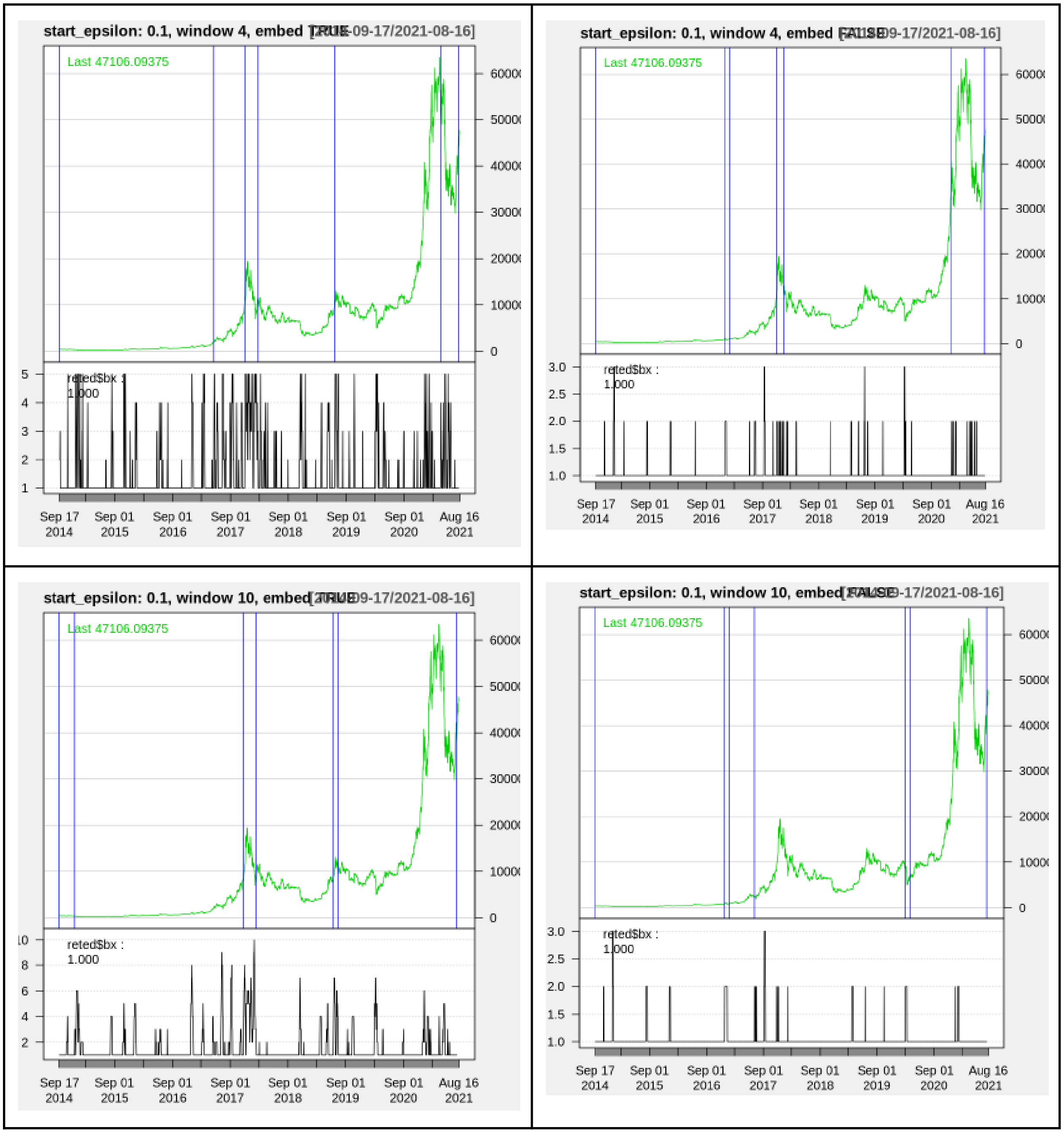


Figure 8. The black graphs are the 0th dimensional betti numbers calculated using epsilon = 0.01 if there are 50 epsilons, and epsilon = 0.1 if there are 5 epsilons, and the corresponding window size. The changepoints shown are the 5 most relevant ones. It seems with larger window size, the betti numbers are smaller.

We observe that with larger initial epsilon, 0.1, the changepoints agree less, whereas with smallest epsilon = 0.01, the changepoints agree more with each other, with a difference of only a few days, with the exception only being the one with no embedding and window size 11, where the disagreement is over the location of the first changepoint and if there should be a changepoint in 2019. This changepoint is more in line with smallest epsilon =0.1, and with embedding.

We tried applying PCA to the 50 epsilons, 3-embedding, window = 5 case to reduce the data to a univariate time series, and then apply univariate analysis using Cramer-von-Mises with alpha=0.001. Due to the threshold of the test statistics is still too low and too many changepoints are produced, we limit the Batch recursion to 2 levels (0th level with 1 changepoint, 1st level with at most 2 changepoints, and 2nd level with at most 4 changepoints, so there are at most 7 changepoints).

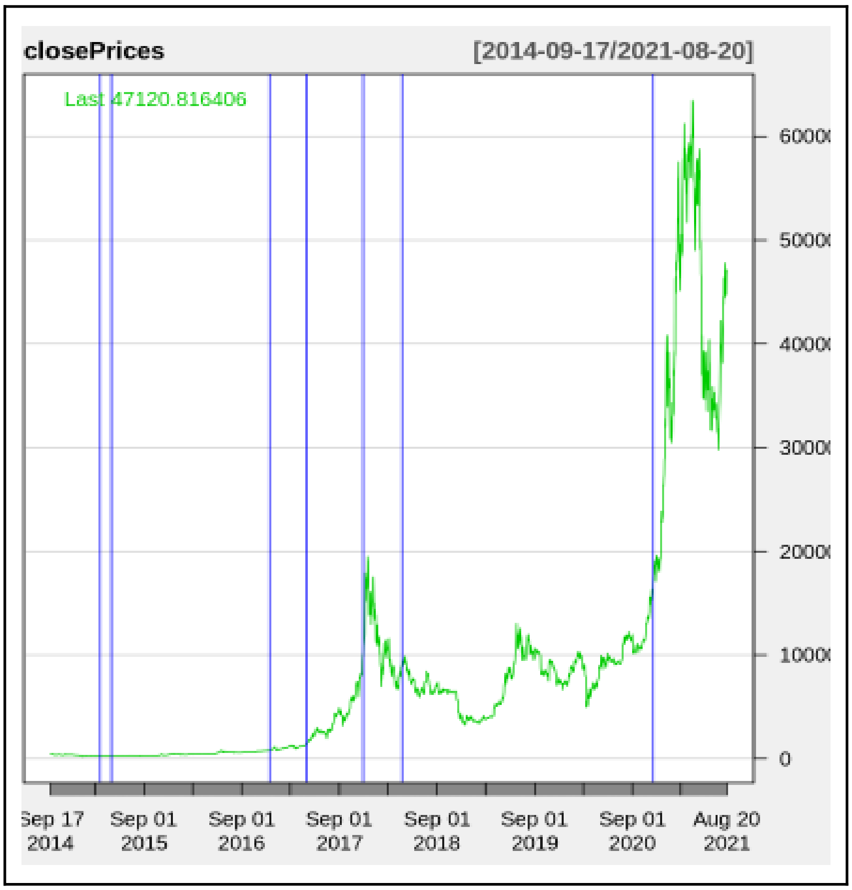
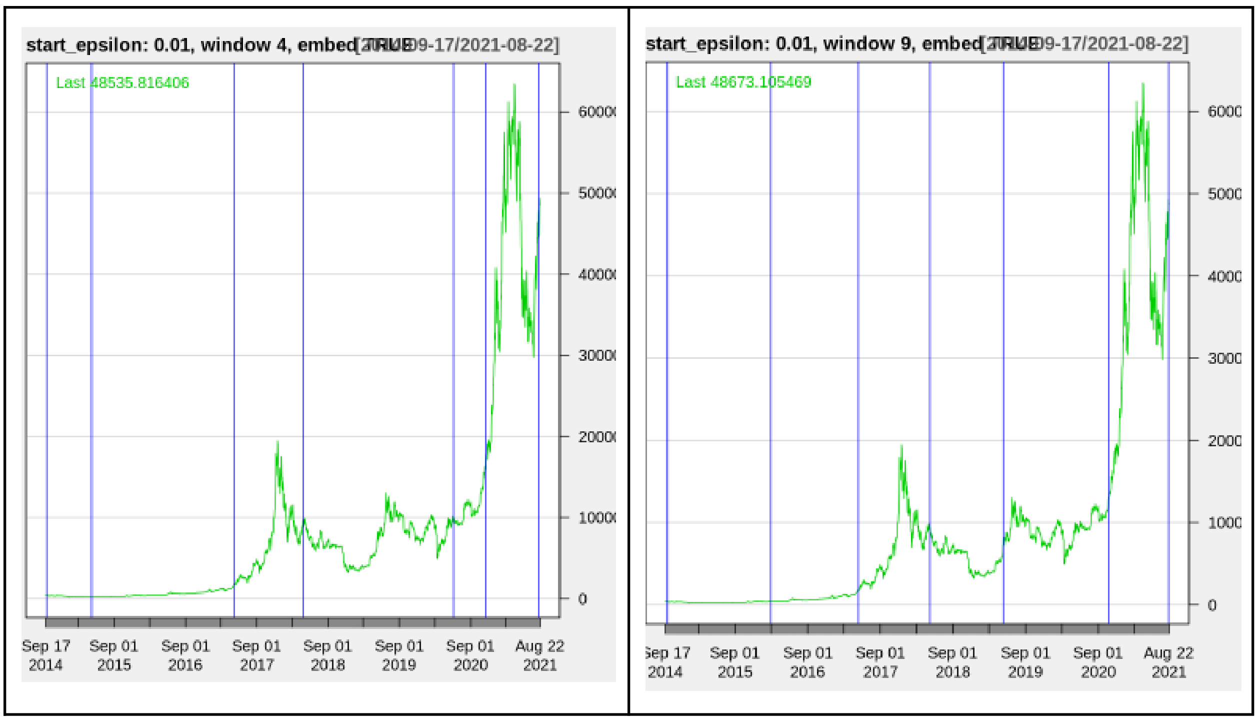


Figure 9. Results of applying Batch recursively after applying PCA to reduce the 2522x50 dimensional time series to a 2522x1 dimensional time series (plotted). For information about the thresholds, reduced time series, and Mood and Mann-Whitney trials, see Appendix Table D. The recursive sequential analysis with ARL=40000 gives even more changepoints, and I cannot reasonably limit the number of iterations.

`Non-overlapping Windows

Elisheva (private communication) pointed out that the problem with sliding a window over the data such that a data point can be included in multiple windows, which is what is done in Islambekov et al. [10] is that the resulting time-series would be weakly dependent, while our changepoint detection methods assume that the data are independent and identically distributed between the changepoints.

However, in Islambekov et al. [10]'s method we may need to take windows twice, once for embedding to 3D with overlapping sliding windows. This may be problematic as the resulting time series may still be dependent. Hence we may prefer to work without embedding, and use non-overlapping windows for the persistence diagrams and betti-0 numbers. We add the results for 3D embedding with overlapping windows for comparison.



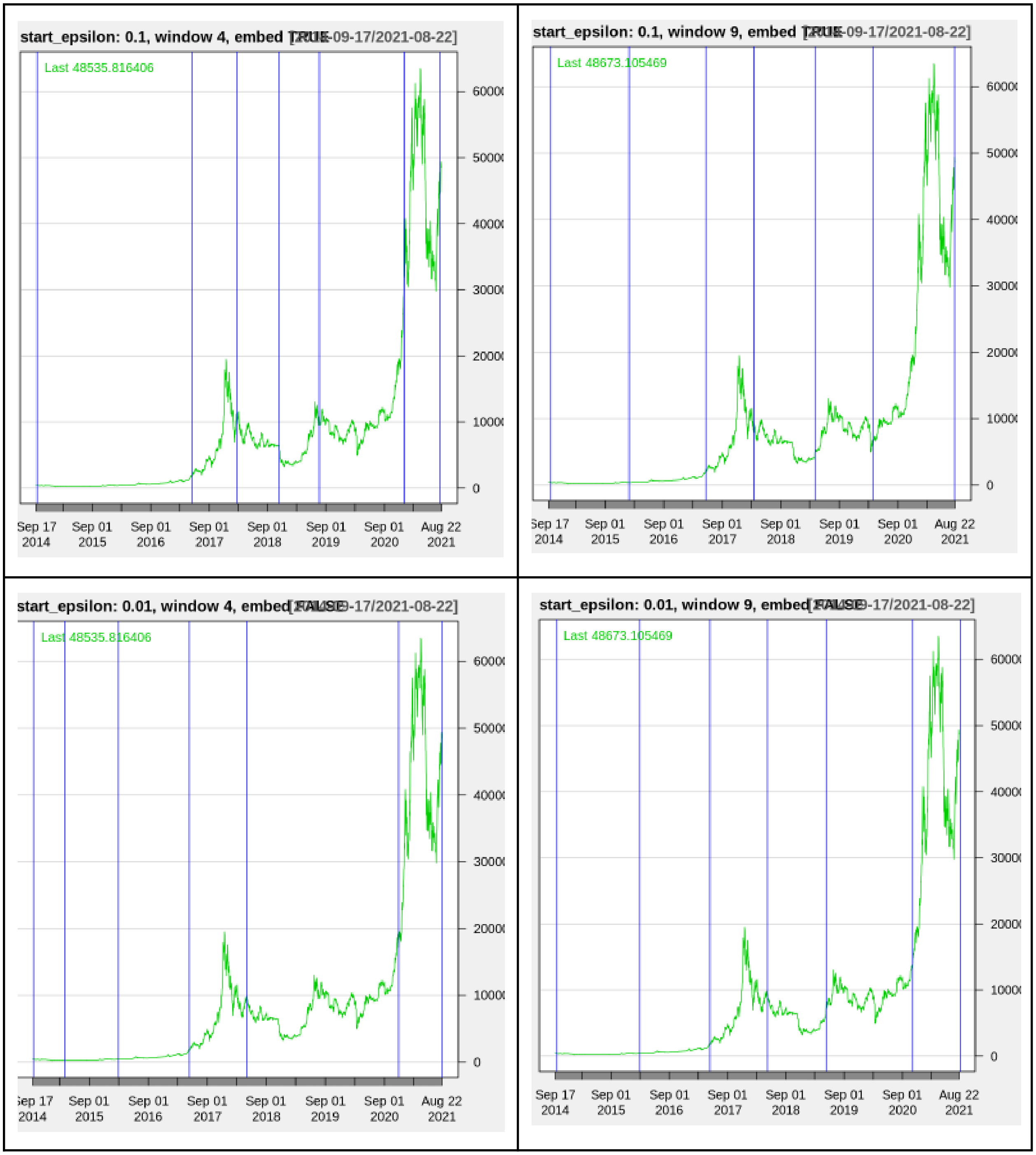

start_epsilon: 0.1, window 4, embed TRUE [2014-09-17/2021-08-22]
Last 48535.816406
start_epsilon: 0.1, window 9, embed TRUE [2014-09-17/2021-08-22]
Last 48673.105469
start_epsilon: 0.01, window 4, embed FALSE [2014-09-17/2021-08-22]
Last 48535.816406
start_epsilon: 0.01, window 9, embed FALSE [2014-09-17/2021-08-22]
Last 48673.105469
Sep 17 2014
Sep 01 2015
Sep 01 2016
Sep 01 2017
Sep 01 2018
Sep 01 2019
Sep 01 2020
Aug 22 2021

Figure 10. Applying TDA with non-overlapping windows, though for the embeddings, the embedding windows still overlap. The changepoints are calculated after multiplying by the window size.

We also applied PCA to the no-embedding version (since embedding might be guilty of being weakly dependent), with 50 epsilons and a window size of 5 (non-overlapping). The number of changepoints produced is now much more manageable, so there is no need to limit the amount of iterations for Batch.

Batch (alpha=0.01)	Sequential (ARL=40000)

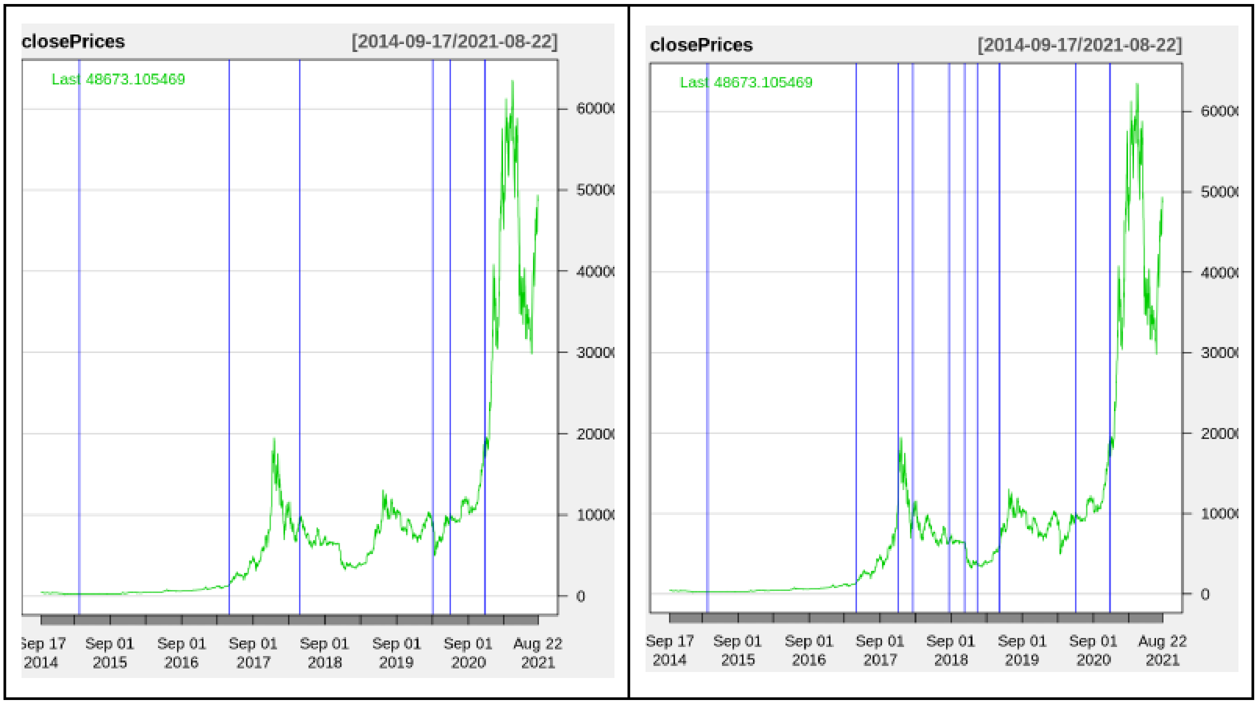


Figure 11: Cramer-von-Mises applied to the time series after PCA reduction from the 50 dimensional time series of betti-0 number on windows of 5 from the univariate time series of Bitcoin log returns from 2014-09-17 to present.

TDA test with Gaussian Series

To compare with Elisheva results, we also try to apply the TDA method to a Gaussian time series of length 200, with a jump of variance from 1 to 2 at 100. We run it for 1000 trials by using seeds from 1 to 1000 for each time series.

On the base case, applying Cramer-von-Mises Bath detection with alpha = 0.05 to the original Gaussian series produces a detection rate of 0.086, and a Mean Average Error (calculated by adding 100 to the total error for each trial without a changepoint detected, and the absolute distance between the changepoint detected, when detected, and 100) of 95.516. E.divisive (limiting the number of changepoints to 1) applied to the series does produce a detection rate of 1, and a MAE of 28.348. Since detection rate cannot be improved for e.divisive, we did not use it to test the TDA+PCA method.

E.divisive did not produce optimal results when TDA is applied without PCA (without embedding), for both epsilons {0.01,0.02,...,0.5} and {0.1, 0.2, …,0.5}, and window size from 5 to 10-- in fact, no changepoint is detected at all, besides the two points that mark the beginning and end of the sequence which it usually detects. Islambekov et al. [10] were able to produce slightly improved MAE by using e.divisive in PCA+TDA.

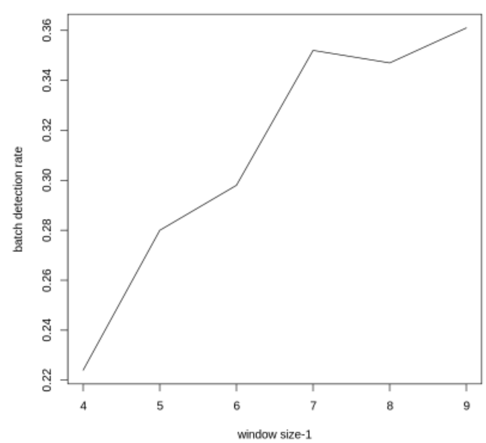
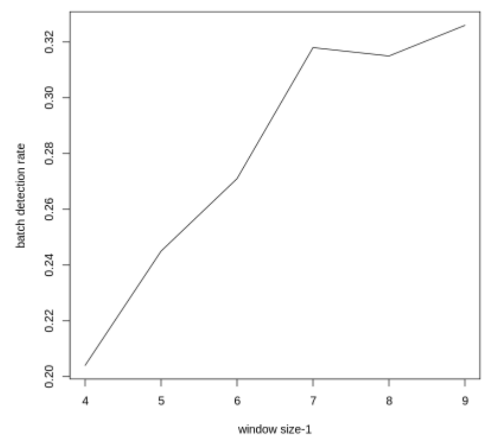
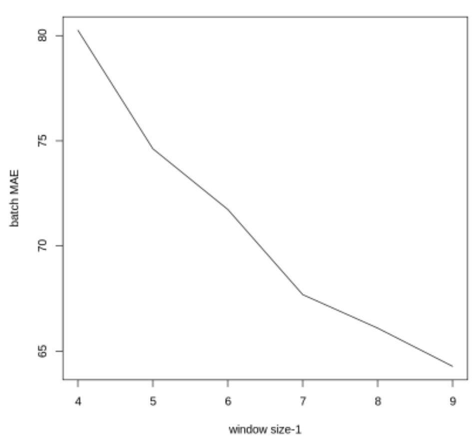
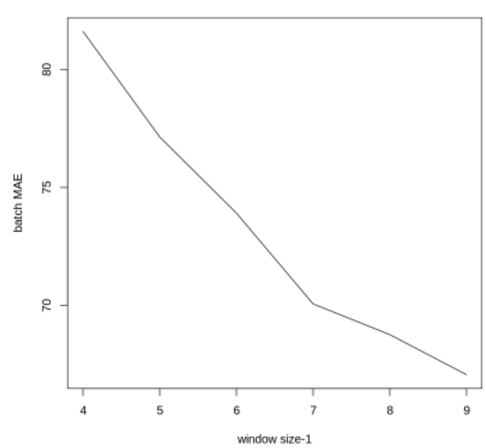
	Epsilons {0.01,0.02,...,0.5}	Epsilons {0.1, 0.2, …,0.5}
Detection rate		
MAE		

Figure 12. The graphs above shows the data for applying Cramer-von-Mises Batch detection to the time series after Principal Component Analysis reduced the multidimensional series of betti-0 numbers to a one dimensional series. The window sizes are offset by 1 (so when it's 4, it is actually 5) for indexing reasons.

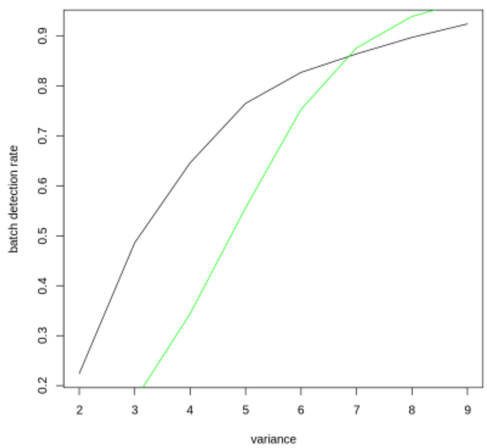
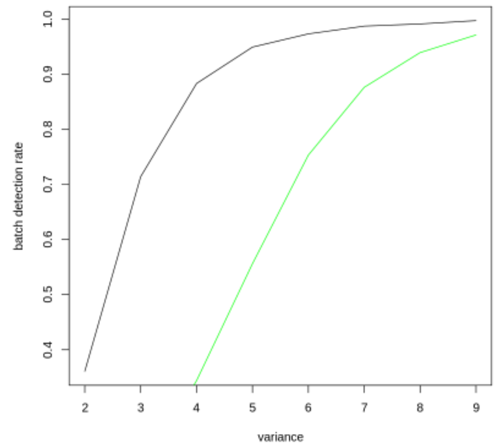
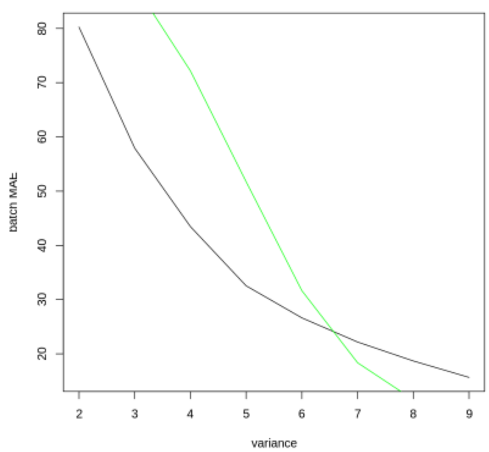
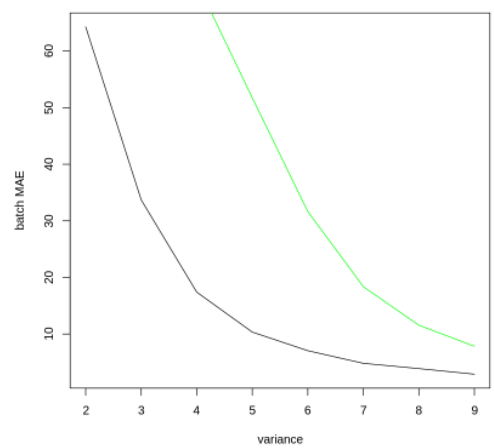
<table>
<tr><td></td><td>Window = 5</td><td>Window = 10</td></tr>
<tr><td>Detection rate</td><td>
</td><td>
</td></tr>
<tr><td>MAE</td><td>
</td><td>
</td></tr>
</table>

Figure 13. The detection rates and MAE for increasing variances of the second part of the time series, from 2 to 9, for windows =5 and 10. The variance of the first half is 1. The green lines are the data for the original series of the corresponding variances, for comparison.

Bibliography

[1] Aminikhanghahi, Samaneh, and Diane J Cook. “A Survey of Methods for Time Series Change Point Detection.” *Knowledge and information systems* vol. 51,2 (2017): 339-367. doi:10.1007/s10115-016-0987-z. (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5464762/#R1)

[2] Hayes, Adams. “What is a Time Series?”. *Investopedia.* https://www.investopedia.com/terms/t/timeseries.asp

[3] Ross, Gordon. "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package." *Journal of Statistical Software [Online]*, 66.3 (2015): 1 - 20. Web. 3 Sep. 2021. https://www.jstatsoft.org/article/view/v066i03

[4] Hudson, Robert and Gregoriou, Andros, Calculating and Comparing Security Returns is Harder than you Think: A Comparison between Logarithmic and Simple Returns (February 7, 2010). Available at SSRN: https://ssrn.com/abstract=1549328 or http://dx.doi.org/10.2139/ssrn.1549328

[5] James, Nicholas, & David S. Matteson. "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data." *Journal of Statistical Software [Online]*, 62.7 (2015): 1 - 25. Web. 3 Sep. 2021. https://www.jstatsoft.org/article/view/v062i07

[6] Wikipedia, “History of Bitcoin.” https://en.wikipedia.org/wiki/History_of_bitcoin

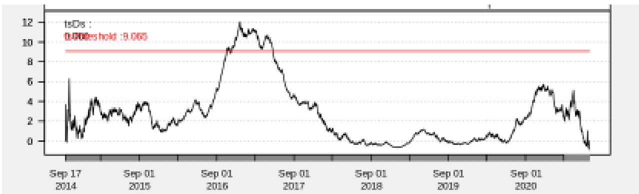
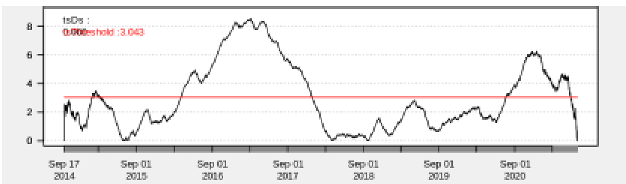
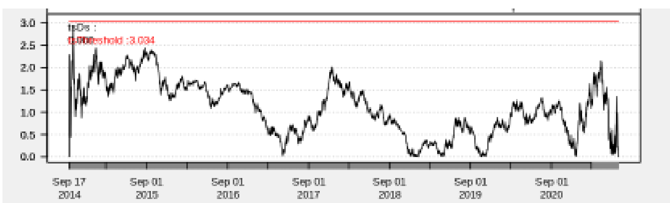
[7] Wikipedia, “Cryptocurrency bubble.” https://en.wikipedia.org/wiki/Cryptocurrency_bubble#2018_crash

[8] Kript Team. “Influence of Different Events on Bitcoin Price.” *Medium*. Aug 22 2018. https://medium.com/kriptapp/influence-of-different-events-on-bitcoin-price-c9b2c0dfffce

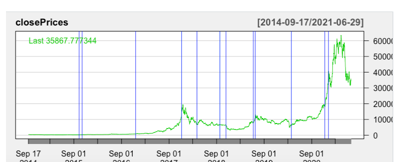
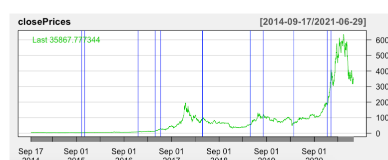
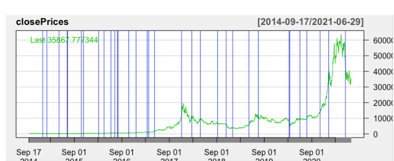
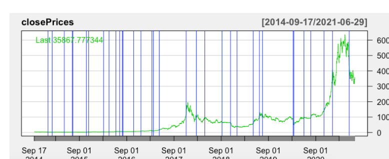
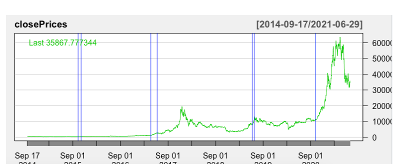
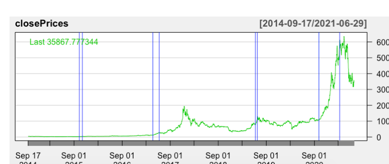
[9] BitcoinWiki. “Bitcoin History.”. https://en.bitcoinwiki.org/wiki/Bitcoin_history#Bitcoin_in_2019

[10] Islambekov, U, Yuvaraj, M, Gel, YR. “Harnessing the power of topological data analysis to detect change points”. *Environmetrics*. 2020; 31:e2612. https://doi.org/10.1002/env.2612

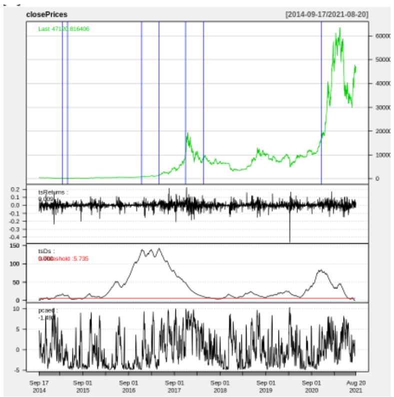
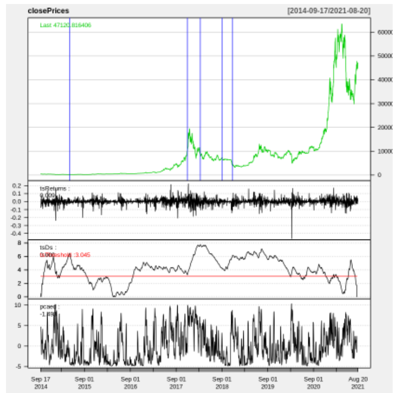
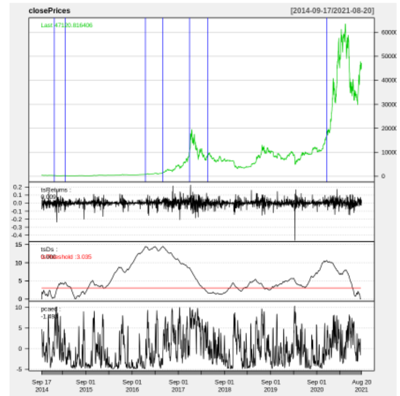
Appendix

threshold




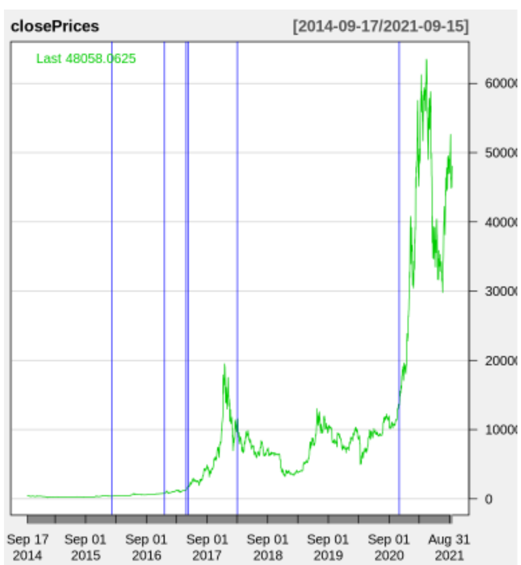
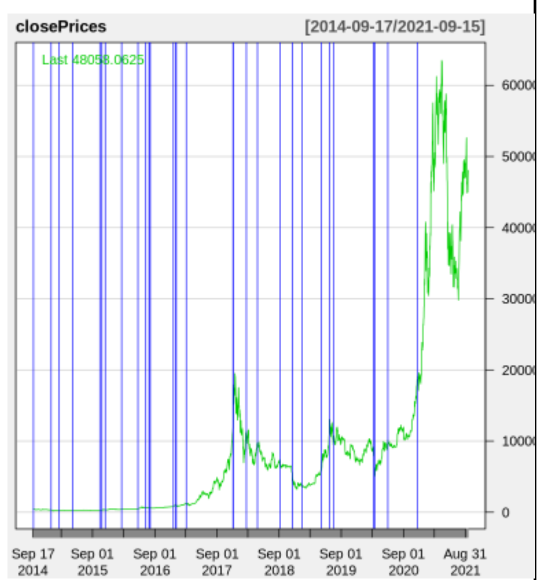
Appendix table B

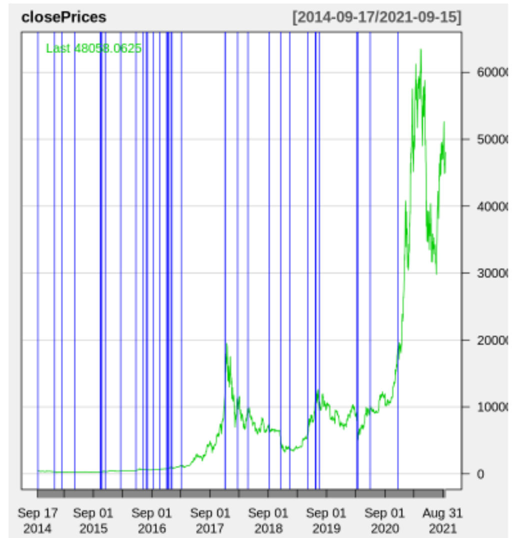
ARL	900	800
Cramer-von-Mises		
Mood		
Mann-Whitney		

Appendix Table C

Cramer	Mood	Mann-Whitney
		

Appendix Table D

alpha	Cramer	Mood
0.05		

0.1	closePrices [2014-09-17/2021-09-15] Last 48058.0625 0 Sep 17 2014 Sep 01 2015 Sep 01 2016 Sep 01 2017 Sep 01 2018 Sep 01 2019 Sep 01 2020 Aug 31 2021	closePrices [2014-09-17/2021-09-15] Last 48058.0625 0 Sep 17 2014 Sep 01 2015 Sep 01 2016 Sep 01 2017 Sep 01 2018 Sep 01 2019 Sep 01 2020 Aug 31 2021