GHENT
UNIVERSITY

Department of Physics and Astronomy
Complex Systems Institute

Criticality and forecasting of the cryptocurrency market

Ciem Cornelissen

Supervisor Prof. Dr. Luis E. C. Rocha

June 1, 2022

Abstract

Because the cryptocurrency world is so young, less research has been done on this market. With the market increasing, it's time to study it better. Therefore, this thesis will contribute to this field by studying critical behavior and looking at some of its effects on, e.g. predictability of the price and the network. This critical behavior is found by looking at the non-Gaussian behavior of the difference between the closing prices of Bitcoin and Ethereum. For the predicting part, the machine learning methods used are autoregressive integrated moving average(ARIMA), Fbprophet, and recurrent neural networks(RNNs). First, an extensive search in the parameter space for all these methods, to find the optimal parameters for the variants of RNNs, is done. Ones these are found, every model is tested using the MASE error in combination with a cross-validation technique to make this comparison quantitative. The best model is then used to look at predictability during the found critical behavior. For the network analysis of the cryptocurrency market. The analysis is done over the crash of 2018 in combination with a wider overview analysis, to get a better idea of the market. Methods like motif analyses and degree distributions are looked at during the period of critical behavior. Phase transition-like behavior is found. This is over a period of a few months around November 13th, 2018, which had the clearest scale-free behavior. The error of the predictions showed a small correlation with the amount of correlation in the market. A correlation in both the degree distribution and the motif analyses with the amount of correlation in the market is found. Both the correlation in the market and the degree distribution showed a period of increase before the crash. If this does always happen, these two measures could be used as signals to signal a crash. The way of looking at the market through a physics lens, i.e. looking at the correlation in the price, can help risk assessment for investors and gives a new way of looking at the market and its condition.

Nederlandse Samenvatting

Deze thesis gaat over het zoeken van periodes waarin zich kritisch gedrag voordoet, en gaat dan op zoek naar de effecten dat dit gedrag heeft op dingen als voorspelbaarheid van de prijs en op het netwerk van de cryptocurrencies als een geheel. De thesis kan worden gezien als opgebouwd uit 3 delen.

Het eerste deel van de thesis gaat op zoek naar deze fase transitie achtige fenomenen door te kijken naar de correlaties in de sluitingsprijs over verschillende tijden, i.e. de tijd tussen de twee prijzen waartussen het verschil wordt gekozen wordt gevarieerd. Er wordt gezocht naar momenten waar de correlatie onafhankelijk wordt van de tijd. Dit schaal onafhankelijk gedrag doet denken aan een tweede orde fase transitie. Het zoeken wordt gedaan door, over een bepaald tijdsinterval een functie te fitten aan de distributie van de prijs verschillen. Deze parameter die gefit wordt, geeft aan hoe Gaussian of hoe niet Gaussian de distributie is. Hoe minder Gaussian, hoe meer niet random de prijs is en dus hoe meer correlatie er is tussen prijzen en dit over verschillende tijdspannen.

Het tweede deel gaat over, de voorspelbaarheid. Deze voorspelbaarheid wordt onderzocht door middel van machine learning methodes. Verschillende methodes worden gebruikt, maar de grootste focus ligt bij Recurrent Neural Networks(RNNs). Het best presterende van al deze methodes wordt gebruikt over de periode gevonden in deel 1, om te zien of er een correlatie is tussen dit kritische moment en de voorspelbaarheid. Naast de RNNs worden ook relatief makkelijkere modellen gebruikt zoals: ARIMA en Fbprophet. Voor de RNNs worden verschillende structuren getest, i.e. verschillende hoeveelheid lagen en verschillende types lagen, e.g. LSTM of GRU. Dit deel gaat ook in op het effect van extra data op de voorspelbaarheid van de methodes.

Het derde focust op een netwerk bestaande uit een circa 1800 cryptocurrencies, waarop meerdere netwerk analyse methodes worden toegepast. Het eerste deel van dit deel gaat meer in op de hele geschiedenis van het netwerk en het tweede deel gaat dieper in op de crash. De uitgezoomde analyse is er voor een breder beeld te krijgen van hoe het netwerk werkt en hoe alles varieert over langere periodes. Het tweede deel focust meer op de crash en kijkt naar correlaties tussen de crash en veranderingen in het netwerk door bijvoorbeeld te kijken naar de degree distributie.

Acknowledgement

Thank you Prof. Dr. Luis E. C. Rocha for being my promoter, for the feedback, for the help and insights, and the lectures. Thank you for letting me do my own thing. Thank you for the out-of-the-ordinary subject and for making it so interdisciplinary.

Thank you, colleague, Matthias Hooft, for helping me with the predicting part of the thesis. Thank you for your point of view, your written words, and your insights into machine learning

Contents

List of Figures

1 Introduction

> *If life were predictable it would cease to be life, and be without flavor.*
>
> **— Eleanor Roosevelt**

In the last 15 years, one could see the rise of a new market, i.e. the cryptocurrency market. This market came from nothing to a place where billions of dollars are being traded every day, and this is in a decade. It started around 2008 when Satoshi Nakamoto published a paper[37] on a purely peer-to-peer version of an electronic cash system. The power of the system is the fact that it does not need an overlooking financial institution. Instead of an institute, the decision power is in the hands of everyone who wants it. Everyone can look at everything that is happening and look if everything checks out. All based on cryptographic proof, giving a secure way of digital proving things. Nowadays, in 2022, there are over a 10000 different cryptocurrencies[12]. All these cryptocurrencies are different, belonging to certain groups or types coming with different and new ways of doing things. From Bitcoin, being purely money and a way of storing small amounts of data somewhere, it can not be changed, to networks that can do everything that a computer can do. For example, Ethereum[8], which has Turing complete code.

The increase in value of this asset comes with a lot of media attention. As the news of large value increases attracts more people who raise the valuation even more through buying into the asset. From being worth nothing to a total market cap, i.e. total sum of values of all coins, of almost 3 trillion dollars at the peak around the end of 2021. Apart from this increase in worth, the cryptocurrency market has also come with innovations trickling down to other sectors[44, 15, 34].

A way of looking at this market is through the lens of physics. One could see the cryptocurrency market as a complex system with its own dynamics. One can, with the help of physics, try to understand these dynamics better to get a better understanding of the market. A complex system like this could exhibit physical phenomena like phase transitions[25], it is in this thesis that these phenomena are looked for, with their effect on the ability to forecast and the network as a whole. It is expected that these phenomena will have an effect on the market and this will be tested in the thesis. In short, this thesis is about finding critical phenomena and looking at what the effects are on the predictability of the price of Bitcoin and the change in dynamics of the cryptocurrency network.

The cryptocurrency market is just like the stock market, a game between sellers and buyers, an attracting force and repelling force. With the forces being, time depended. Say Bitcoin gets a lot of good news, the price will go up as new buyers and existing buyers buy more. One could also see this as the attracting force getting stronger. When a system has attracting and repelling forces, the system can go through sudden changes. For some parameters of the forces or the system. The system might go through sudden change and change from a state to a new state. e.g. water that gets colder will suddenly go from liquid to ice. It's a sudden change that happens when the attracting force is stronger than the repelling force.

A market of buyers and sellers also can change states[42], i.e. a state of euphoria and good attention can suddenly change to a more negative and dormant state. For financial markets, these states get referred to as bull and bear markets. A bull market[41] means that the market is in an upward trend, investors are making money and the market gets good media attention. The bear market[23] is the opposite, i.e. investors are losing money as the price goes down and the media takes a point of view that is more negative.

One of the hardest problems in machine learning is forecasting the stock market[9] and, because of higher volatility, an even harder problem is predicting the price of cryptocurrencies[33]. This is the same problem as giving some values of a certain feature measured at a certain interval and then predicting what the next value of this feature will be at a later moment. The forecasting problem arises in many fields of research[2, 40, 61, 43, 59]. This thesis goes over the time series of the closing price of Bitcoin and compares different models on their predictive power. The focus lies mainly on Recurrent Neural Networks, with two other methods (autoregressive integrated moving average(ARIMA) and Fbprophet[21] as comparative models.

The hardness in problems like economic time series forecasting comes from the 'randomness' in the data. Many effects are influencing the trading behavior of people and it is impossible to build a model with 100 percent accurate prediction power. If the data is completely random (so there are no trends or seasonality), then the best prediction one can do is just to take the same value from the time step before, which will be used further on and will be called the naïve method. This is the reason many research in predicting time series compare their result to this so-called 'naïve method'. If the naïve method gives better results, the time series is completely random or the forecasting methods could not find any patterns in the series and will give bad predictions.

One of the biggest crashes in the world of cryptocurrencies happened on 13-11-2018. An interesting way of looking at this crash is with the help of network analysis. The measures and methods of the analyses might give insights into the dynamics of the crash and might lead to findings that might help predict a crash. More advanced methods like motive analysis and degree distribution fitting are used to look at the crash, in combination with more general measures for a broad overview. A more broad analysis is done to get a better view of the market and how it develops over longer periods.

Thesis Structure

This Thesis is about phase transition-like phenomena in the cryptocurrency market and seeing how predictability and the network of cryptocurrencies are affected. The thesis can be seen as made of 3 parts:

Part 1, is about critical behavior and the search for a moment in time this critical behavior occurs, i.e. a moment in time for which correlation length divergences. This critical behavior is reminiscent of second-order phase transitions. One could thus look at this as a search for a phase transition. This divergence is looked for in the price of Bitcoin and Ethereum, the two biggest cryptocurrencies(based on total worth, the sum of the value of all coins[12]). The way this behavior is looked for is by looking at the non-Gaussian behavior of made stationary closing prices of the two currencies.

Part2, is about predictability and tries to use machine learning to predict the price of Bitcoin. Different methods are used, but the primary focus is Recurrent Neural Networks. The best-performing method is then used through the period found in part 1. To see if there is a correlation between critical phenomena and predictability. It starts with some relatively simple models like Fbprophet and ARIMA. Then after the focus is on recurrent neural networks(RNNs) like Long Short Term Memory(LSTM) and Gated Recurrent Units(GRU). For the RNN-models different architectures of the neural network itself are used, e.g. the number of layers of different types in the network. There is also a version that is used that uses Empirical Wavelet Transforms(EWT)[29], i.e. the data is split up into different frequencies and on does frequencies RNNs are trained. The effect of extra data on the predictive power is also investigated, e.g. the amount a group of words is searched on google or just the highest and lowest price of that day.

Part 3, the last part, is about the effect of the period of critically on the network. A network made of around 1000 cryptocurrencies. On which multiple measures of network analysis are used and looked at to see what changes during these found phase transitions, e.g. the crash at the end of 2018. The first section of this part is about the more zoomed-out analysis, to lay a foundation for the analysis of the crash. This analysis makes use of more general measures. The period of the zoomed-out analysis is from around 2013 until the end of 2019. The second section of this part goes more into detail on the crash. For the crash, more advanced methods are used to get a better and more insight full look into the dynamics of the crash.

Theory 2

> *Everything has beauty, but not everyone can see.*
>
> — **Confucius**

2.1 Criticality in multiple markets

There is already some research done on phase-like transitions in financial systems, mainly of stocks in the stock market. Narrowing the field down to the cryptocurrency market leaves out most of the papers. This is to be expected as this financial system is not as old as, e.g. the stock market. This section goes over some of the most relevant related work which is used as inspiration for this research.

One paper that looks for critical behavior in the stock market is by Kiyono, K., Struzik, Z. R., & Yamamoto, Y.[25]. In this paper, they look at returns for the U.S. S&P 500, which are made stationary. The paper zooms in on the crash of 1887, also called black Monday. To this data, a PDF gets fit, which is chosen to be non-Gaussian to better describe the fat tails, i.e. a higher probability for large returning events. The results show an increase in correlation on larger time frames and a scale-free behavior of the fitting parameter at criticality. Suggesting that the correlation of returns over some time becomes independent of the time between the returns at this critical point. Because of this scaling-free behavior. This crash shows similarities with a second-order phase transition.

The same as the previous paper is done by F. Chudzyńskia and Z. Struzika[11]. But in this paper, they investigated the non-Gaussianity of the Bitcoin closing price instead of the U.S. S&P 500. They also did this by fitting a non-Gaussian probability density function to the distribution of the normalized return distribution of the Bitcoin closing price. The fitting parameter showed a much slower convergence to 0, i.e. a slower convergence to a Gaussian distribution. Showing that the correlation of price actions spans over a longer time window than that of the well-established stocks.

2.2 Forecasting

There is a lot of research on predicting the stock market using a wide variety of machine learning methods.

Prapanna Mondal, Labani Shit, and Saptarsi Goswami looked at the predictive power of the ARIMA model [36]. They used ARIMA to make predictions of 65 Indian stocks from different sectors. First, they searched for the optimal parameters of the model (this also depends on the used data set). They used the Mean Absolute Error (MAE) to test the accuracy of the predictions of ARIMA. The researchers found that ARIMA works very well for these stock predictions (an accuracy of at least 85%).

Another method is Fbprophet which is originally developed by Facebook. This method is, for example, studied by Bineet Kumar Jha and Shilpa Pande [21]. They used this model to make predictions of supermarket sales and compared the results with the predictions that ARIMA gives. The result was that Fbprophet has a better forecasting accuracy than ARIMA and other analog methods.

More advanced methods make use of Recurrent Neural Networks. For forecasting time series, the most commonly used methods are Long-Short term memory and Gated Recurrent Units. Extensive research on these methods is done by Khaled A. Althelaya, El-Sayed M. El-Alfy and Salahadin Mohammed [24]. They used LSTM and GRU on the Standard & Poor's 500 index from 01/01/2010 to 30/11/2017. They implemented these methods with different architectures (they looked at Stacked LSTM (SLSTM), Bidirectional LSTM (BLSTM), SGRU, BGRU) and they used the MAE, the Root Mean Squared Error (RMSE), and the coefficient of determination (R^2) error to quantify the prediction accuracy. The result was that the SLSTM was the best performing RNN.

The work of Weiling Chen et al. [55] started with the question: 'Is the stock market random?' in other words: 'Can one predict the stock market or is it completely random so that the best model for predicting it is a naïve model?'. Their answer is 'No, the stock market is not completely random' and their argument for this answer is that buyers do not always make rational decisions and are influenced by social networks. When many people are talking positively about some product, a lot more buyers will invest in the company that makes this product. In this paper, they found that including data from the news into a GRU model gave more accurate predictions. Thus, relevant features from social networks can improve one's forecasting model.

2.3 Networks and crashes

The stock market is not too different from the cryptocurrency market and has been analyzed with the use of networks. Insights can also come through looking through a different lens, i.e. physics[7]. Networks analysis has also been done on the cryptocurrency market. However, this is a smaller research field, which is to be expected, as the cryptocurrency market is decades younger than the stock market.

Tiziano Squartini, Iman van Lelyveld, and Diego Garlaschelli[50] looked at interbank networks. They looked at the changes in topology before and during a crisis. They also looked for precursors that could be used as signals to show a crash is coming. With their focus on motif analysis as a signal. They showed that as well for dyadic as for triadic motifs, a form of early warning can be observed. Both motifs show a change before the crisis, which suggests they can be used as signals. By looking at all triadic motifs before and during the crisis, they found a change in motif representation.

An example of a network analysis of the cryptocurrency market is done by Kin-Hon Ho, Wai-Han Chiu, and Chin Li[17]. In their paper, they studied a network of the top 120 cryptocurrencies over the period from 2013 to 2020. In their analysis, they mainly focused on some centrality measures, e.g. degree centrality, Node strength, closeness centrality, and eigenvector centrality. With these measures and more, they studied the correlation structure, stability of the market, and the most influential cryptocurrencies during different stages. Their study showed a weakening in correlation from 2013 to 2016 and after 2016, a strengthening in correlation. The study also showed that different periods got dominated by different cryptocurrencies.

3 Materials and methods

> *Imagination is everything. It is the preview of life's coming attractions.*
>
> — **Albert Einstein**

3.1 Data

The data that is used for finding the critical behavior is the Ethereum and Bitcoin closing prices. For Ethereum this data is from 2016 when the cryptocurrency got launched till 2022. For Ethereum in both data sets, all prices come from the same exchange, i.e. the Gemini exchange. With the difference between the two data sets the source that gathered the data from the exchange. One of the Bitcoin data sets is also used for the predictive part of the thesis and comes from the Bitstamp exchange [62]. The other Bitcoin data set is like a data set for Ethereum from the Gemini exchange. All data is of resolution 1 minute. This is needed for the methods that are used. For both coins, two data sets are used to see the effect of different data sets on the analyses.

For predicting, 2 different data sets are used: data set one [62] and data set two [53]. These data sets contain distinct features of Bitcoin like the 'Closing price', 'Open price', 'High price'. The first set contains data of these prices for every minute and takes the mean value over every day. The second data set contains just one value per day. For predicting only data of resolution day will be used. One of the goals is to predict the closing price for both data sets using different machine learning models and compare the results to see the effect of different data sets.

The closing price, from data set two [53], on every day is shown on figure (3.1). In this plot, one can see that there is a general upgoing trend with fluctuations on it. When looking closer, one can see that there are periods where the price varies relatively little (40%) while in other periods it changes a lot (600%). All the methods are tested over 2 different periods:

1. a calm period (from 1/3/2018 till 11/11/2018, which is 256 days in total, blue period).

2. a bullish period (from 18/7/2020 till 30/3/2021, which is 256 days in total, red period) that will be referred to as bullish period.

In figure (3.1), these calm and bullish periods are indicated by the colors blue and red, respectively. These two periods are chosen because there is a clear trend in the bullish while this is less in the calm period. This wild changing period is also called volatile because the price fluctuates around 10% more than in the calm, more stable period. This is tested by calculating the deviation on the mean for the two stationary time series. To make the time series stationary, first, the logarithm of the data is taken. Then a price of a day gets replaced by the difference of that day with the previous day.

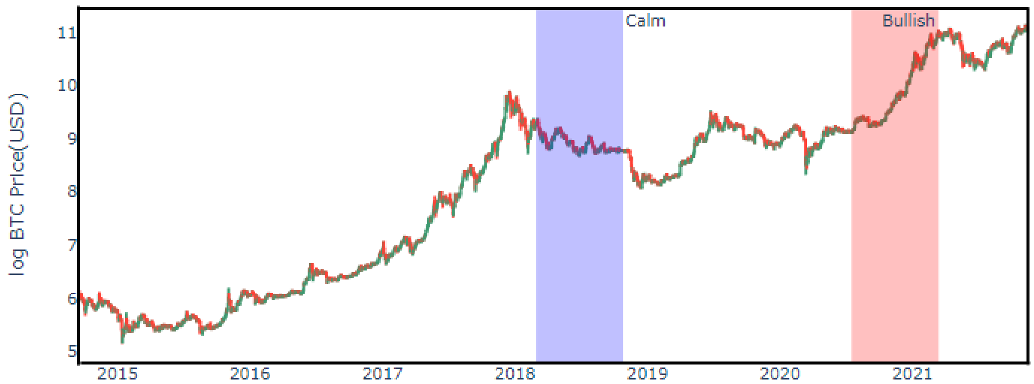


Fig. 3.1: A candlestick figure for Bitcoin of data set two [53]. The calm(blue) period and bullish(red) period are also indicated.

The data contains a few missing points. To fill in this empty data, the value of all features of the previous day in the series are used again. All the data points are also scaled, so they are between 0 and 1, as this makes an RNN converge faster[49]. For this, the MinMaxscalar function of sklearn is used.

For the network part, one data set is used, which is provided by the supervisor. This data set contains around 5000 different coins. First, the data is cleaned, all currencies with invalid data are removed. This leaves around half that can still be used. The data ranges from 2010 to 2020, but only from 2013 to 2020 is used, as there is not enough data before 2013 to make a good network. The data is the daily closing price of the different coins.

3.2 PDF fitting

The main method that is used for analyzing the data and for finding, phase transition-like behavior is to fit a non-Gaussian probability Distribution Functions(PDFs) to the data. Before this can be done, one first has to process the data. The first step of the process is to take the log of the data. The next step is to detrend the data by fitting linear functions to a window of a certain size $2s$ and then taking the deviation between the linear fitted function and the actual price. The window is chosen so that the last day of the window is t.

$$x(t) = ln(Z(t)) - f(a, b, t) \tag{3.1}$$

With $Z(t)$ the closing price at time t of either Ethereum(ETH) or Bitcoin(BTC). $f(a, b, t)$ is the fitted linear function with its two parameters a and b. The last step is then to take the difference of two $x(t)$ of distance s apart.

$$\Delta x_s(t) = x(t - s) - x(t) \tag{3.2}$$

Here its chosen to take the difference with a time of s in the past. This is done so that if some measure has certain behavior before a crash, then this measure could be used as a signal. This would not be possible if the difference was taken with s in the future. This is done over a sequence of periods given by:

$$s = 8 \times 2^i \tag{3.3}$$

Where i ranges from 0 to 13, so one has 13 different sets of $\Delta x_s(t)$. These 13 different sets make it possible to look at the non-Gaussian behavior over different time scales.

For the fitting part of the analysis the PDF:

$$P_\lambda(\Delta x_s) = \frac{1}{2\pi\lambda}\int_0^\infty e^{-\frac{\Delta x_s^2}{2\sigma^2}} e^{-\frac{ln^2(\frac{\sigma}{\sigma_0})}{2\lambda^2}} \frac{d\sigma}{\sigma^2} \tag{3.4}$$

is chosen. Eq.(3.4) was introduced by Castaing, B., Gagne, Y., & Hopfinger, E. J. in 1990[10]. The equation is the symmetric version of the equation that was first used to fit to velocity differences, in high Reynolds number turbulence, for the measurements in two points with distance r between them. Eq.(3.4) has the parameter λ used to fit the PDF to an observed distribution. One could also fit σ_0 to get a better fit to the measured distribution. If λ approaches 0, the PDF will approach a Gauss. The larger λ The larger the tails are of the distribution. This means that more unlikely events are happening on a more frequent basis than when one would compare it to a Gaussian distribution. One could see this as some correlation that is present which makes more unlikely events more probable.

Eq.(3.4) is fitted to the 13 different sets of $\Delta x_s(t)$ with there s coming from eq.(3.3). Looking at the 13 different fitted λ one would expect from looking at the related research that the first ones, so the λ corresponding to low s are higher and that λ gets smaller as s gets bigger. Intuitively this also works as one would expect more correlation at smaller time scales and less correlation at longer time scales. Looking for a phase transition then becomes looking for a moment of criticality for which this downslope of λ vanishes and one gets scale-invariant behavior, with the scale being time or the s.

3.3 Machine Learning

3.3.1 Optimisation, forecasting and validation

Before testing the used methods, a search for the best parameters is done for every RNN model in both periods and for the two data sets.

For these methods, the parameters that are being optimized are the batch size, days to predict, days to train, and the number of epochs. The days to train are the number of days the method uses to predict the next day. The days to predict are the number of days one tries to predict. The amount of epochs is the number of times one puts the whole data set through the network. Normally, after one part of the data goes through the network, the weights of the network are updated. This update depends on the error of the output of the network. To make the network learn faster, one can take some of these inputs together. Instead of updating after each passing, the network then updates its weights after a given amount of data going through it. The mean value of the errors is then used to update the weights. The amount of inputs one takes together before updating the weights and biases is called the batch size.

For the optimization of these parameters, the training data is split into a training part, a validation part, and a test part that will be left alone to compare the models. The validation part is the last days of the training data, i.e. the number of days equal to days to predict. Every RNN method is trained with different parameters on the training part. This model is then used to predict the closing price on the days to predict. For every predicted day, the method error is calculated. Which is the difference between the prediction of the method with the actual value of the data set also the error that would be made if one uses the naïve model is calculated. Then the mean value of the method error over all days to predict is taken and the same for the naïve errors. Finally, the ratio of both values is taken. This ratio is called the MASE error and is given by:

$$MASE = \frac{\frac{1}{J}\sum_J e_J}{\frac{1}{T-1}\sum_{t=2}^{T}|Y_t - Y_1|}. \tag{3.5}$$

For the RNN methods, there is an extra complication. One has to take some initial values for the weights and biases for the different layers. These weights are typically drawn randomly from a normal distribution. This makes that every time one retrains their method, the predictions one made with it will be slightly different. To overcome this problem, one can train their model several times. Then one calculates the mean value of the MASE errors and the standard deviation on this mean error. For ARIMA and Fbprophet, no optimization is done. However, ARIMA optimized itself with the use of auto ARIMA. The reason these two methods are not optimized is that these methods are used as a comparison for the other methods.

When one forecasts more than one day, one has to be careful. Say one has a model that uses the values of a feature of the previous 10 days to predict the value of the next day. When one wants to forecast the next day, one needs to give again 10 values from the past, but the last value should be the value one predicted in the previous step and not a known value from one's data set. Otherwise, one does not forecast the future, one just predicts one day, throws it away, and uses the actual data to predict again the next day.

To test the models, a version of cross-validation that is well suited for time series is used. One cannot simply split the data into pieces at random to do the common version of cross-validation because one wants to predict the future and not the past. The past might be better predicted by a different model than the future. The time-series version of cross-validation is well pictured in figure (3.2). The idea is that one takes, for example, the values of a certain feature over 50 days to predict the values on the next 4 days. On this fourth day, one calculates the method error and the naïve error. One then shifts everything one day later and one does this again. At the end of all these steps, one can calculate the multiple MASE errors and take the average. This will then be the final error in comparing the methods.

For the RNNs, first, the best parameters are taken. Then every model is trained 100 times on the training data and every time also the MASE error on the validation set is calculated. Then the 30 best models, 30 worst models, and 30 random models are taken (based on the error on the validation part of the training data) to test on the test data.

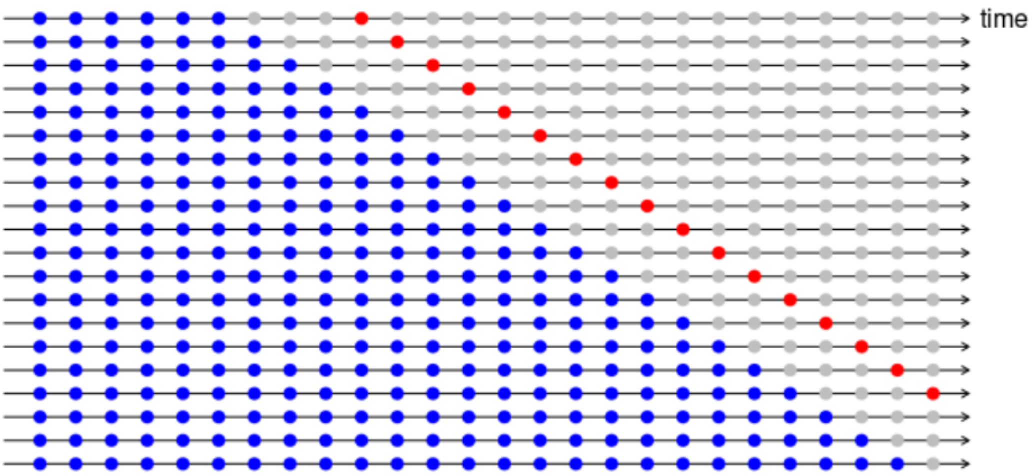


Fig. 3.2: Cross-validation for predicting time series [19]. The blue points are the input values for the model. The red points are the days that are predicted and are used for the forecast error using the MASE error function(3.5). The vertical axis depicts the shift over the time series.

3.3.2 ARIMA

ARIMA is an acronym for Auto Regressive Integrated Moving Average[30, 14, 5, 1, 58]. Auto Regressive models are models that take p days to predict the next day. The prediction for the value of a feature at time t is then just a linear combination of the values of this feature of the previous p time steps. Auto indicates that this is a regression of this feature against itself.

Moving Average models of order q uses the forecast error of the previous q time steps to predict the next value of a feature.

ARIMA is a combination of these two models. It considers the previous p values of the feature and the previous q forecast errors of this feature to predict the next value. The two parameters of this model are p and q. Typical values for these parameters are 3 to 7. This is mostly much smaller than the size of the data set and so this method cannot handle trends in the data well. Because of this, one first has to remove the trend and the seasonality to make the data stationary. This can be achieved by first taking the logarithm of every data point (this will stabilize the variance of the time series) and then taking the difference of every value with the previous value (to eliminate trends and seasonalitys).

3.3.3 Fbprophet

Fbprophet (or simply Prophet) is a widely used method[46, 27, 16, 35, 6], originally introduced by Facebook for forecasting daily data with weekly and yearly seasonality, plus holiday effects. The prediction of the value of a certain feature can formally be written as:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t, \tag{3.6}$$

Where $g(t)$ describes a piecewise linear-trend, $s(t)$ describes the seasonal patterns, $h(t)$ captures the effect of the holidays, and ϵ_t is just a white noise term. This model fits linear or logarithmic models to the data but with breakpoints (where one changes from one linear model to another). This method can be implemented using the Fbprophet package of python. The 'holidays-part' of the model is a parameter of the function as is the seasonality. Both can be turned off or on.

3.3.4 Long Short Term Memory

One of the most widely used RNNs for forecasting is Long Short Term Memory(LSTM) [13, 39, 48, 57, 31]. The advantage of this RNN is that it can remember information from long ago and at the same time, information from the recent past.

The architecture from a LSTM neuron is shown in figure (3.3). The X_t gate is the input from the data, e.g. a value of the Closing price for Bitcoin from the previous 60 days. This is for ordinary neural networks the only input. LSTM also uses information from the past. The Long memory information comes from the C_{t-1} state. This is the Long-term memory of LSTM that remembers all the relevant information from the past. The Short memory information is given by the output from the previous LSTM unit. The combination of these 3 inputs will give the final output for this LSTM unit.

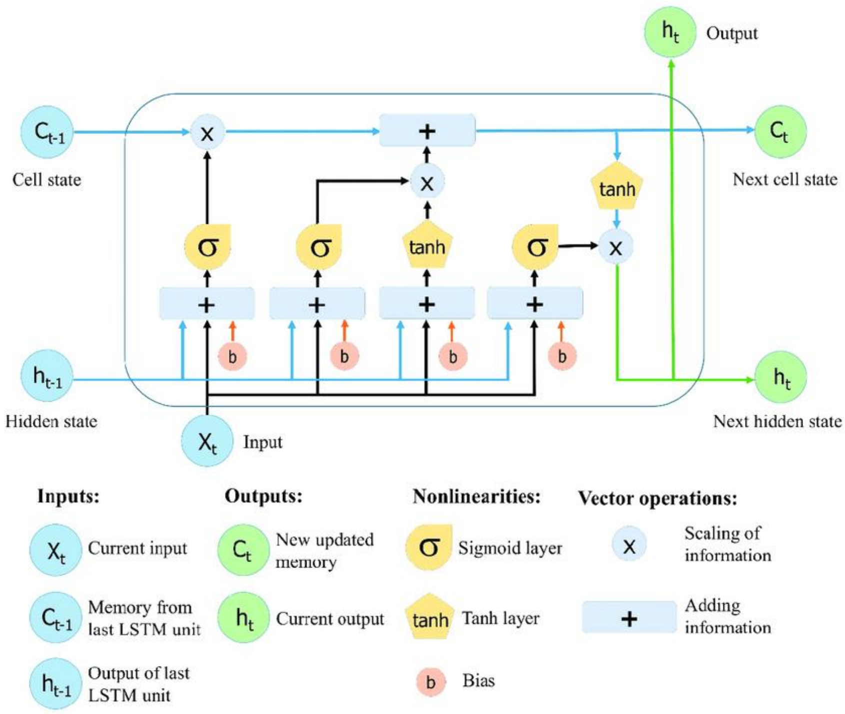


Fig. 3.3: The architecture of an LSTM neuron [26]

In figure (3.3) one can see that the memory (the C state) first gets updated by the input and the output h_{t-1} from the previous LSTM unit. This is done by summing these 3 inputs together and taking the Sigmoid function value. This value is then multiplied with the memory so the memory will shrink or stay constant (the Sigmoid function gives values between 0 and 1). The next step is to give recent memory to the C state. This recent memory is also given by the input and the output from h_{t-1}. The sum of these inputs gets first transformed to a value between -1 and 1 by a $\tanh$ function. This value will then be multiplied by a sigmoid value of this sum to give it more or less weight in the memory. Now the memory is updated and the LSTM unit can give an output. The output is again the sigmoid value of the sum of the input and the h_{t-1}. This value then gets multiplied by the $\tanh$ of the memory state and this value gives the final output from this LSTM unit and at the same time the h_{t-1} input for the next LSTM unit.

This RNN is implemented using the Keras packages. This method has the parameters days to predict, days to train, epochs, batch size, and the architecture of the RNN itself, e.g. the amount of LSTM layers. For the days to predict the optimal parameter (see section 3.2) are always taken. The Keras package already deals with the problem that the amount of input data is not a multiple from the batch size, so one does not have to worry about this. The architecture of the RNN itself is the number of layers, the types of layers, and the number of nodes in every layer.

3.3.5 GRU

Another widely used RNN method for forecasting problems is Gated Recurrent Units (GRU) [32, 18, 54, 22, 28]. The architecture of this method is shown in figure (3.4).

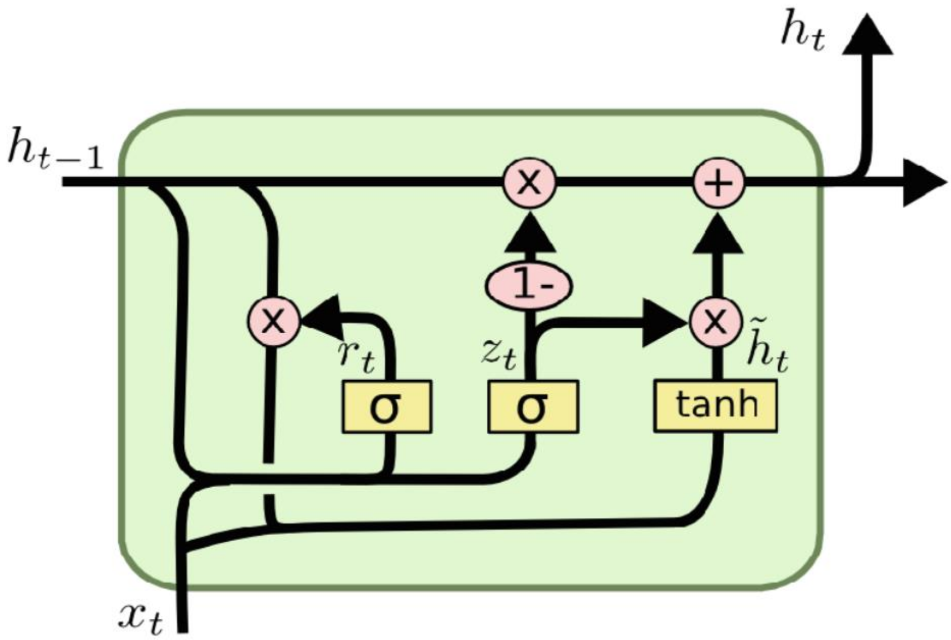


Fig. 3.4: The architecture of a GRU neuron [4]

This method is very similar to LSTM, but with some other mathematical operations in it, e.g. there is no memory state. The information from the past is given by the output of the previous cell, so there is no longer an individual output for the long-term memory. GRU has only 3 gates instead of the 4 like LSTM, with each gate having the same amount of parameters. Because of this, GRU takes less time to fit to the data as it has 75% of the number of parameters that LSTM has.

3.3.6 Modifications to the RNN methods

In all the methods mentioned, the input data is the closing price of Bitcoin of the previous days. Also, multivariate data is used. With multivariate data being multiple layers: closing price, open price, highest price, ... There is also an attempt done to add some data of the google searches of Bitcoin. One could expect that the more people are talking about Bitcoin on a positive note, the more people will invest. This could give better predictions for the future Bitcoin price. It is expected that these multivariate methods would give different results for the 2 different data sets, more on this later.

An extra modification on the RNNs is bidirectional layers. Which are copies of the standard LSTM of GRU layers and pull the data through it in opposite directions. These modifications have also been tried in this analysis.

With these two modifications, one can build different RNN methods, e.g. LSTM VS GRU, multivariate VS univariate data, bidirectional layers or not, and even the amount of layers for each method can differ. However, the number of layers was always kept fixed: 1 LSTM/GRU layer with 40 nodes, 1 LSTM/GRU layer with 80 nodes, and 2 dense layers with both 80 nodes. A dense layer is just a fully connected layer, so all the nodes are connected with each node from the previous layer. The dependence of the dense layer (amount of layers and nodes) has also been investigated, and it is found that for 2 dense layers with 80 nodes, one achieved the optimal results.

3.3.7 EWT LSTM

The Empirical Wavelet Transform(EWT)[20, 60, 51] works by building a wavelet filter bank based on Fourier supports detected in the signal that is being processed[29], i.e. the boundaries of the filters used are chosen so that some maximum of the spectrum lays within. The number of boundaries and thus filers can be chosen and will be referred to as layers. The time series becomes a sum of subtracted signals, all filtered from the original signal through convolution with empirical wavelets. The first test of this method worked with 13 layers, so 12 frequencies and the noise/rest. These layers can be seen in fig.(3.5). In this figure, one sees at the top the actual closing price of Bitcoin from the second data set[53] and underneath that, one can see the sum of all the layers, which follows the general trend of the actual price. The other subplots of fig.(3.5) are all the different subtracted signals. The last filter(the lowest plot) has only one boundary and contains all the noise. The idea of this method came from[3], where they showed they got better results by doing this. The way this method works is by training an LSTM method on each one of the different signals except the noise. Then let this method make a prediction and at the end take the sum of the predictions to get the final prediction.

For a second test, the number of layers is changed to 100. This is done because, when using more layers, the sum of the layers more closely represents the original signal. It is also much easier to predict these signals. The MASE error of the signals is of the order 0.001. One does when doing this have more errors, but assuming that these errors are independent, the sum of the predicted signals will have an error that scales as a square root instead of a sum of the individual errors. This number of layers is computationally expensive, which makes it hard to optimize the parameter of the used model for all levels individually. For this method, no extensive search of the parameter space is done. But after testing, it is found that the most important parameters are the size of the data and the number of epochs. Both have to be large to get good predictions.

3.4 Network analysis

3.4.1 Making of the network

First of all, the data is made stationary. This is done by taking the log of the prices and taking the difference with the previous day. Doing this takes away trends, cycles,.. . This is important, as these trends will give an unreasonable correlation between coins. Two coins that are not correlated but increase in value over a period will be seen as correlated because of the same trend. To make the network, Pearson correlation matrices are made over a window of 60 days. These 60 days are the days taken before the dates mentioned. This makes it possible to use the measures in the present, as one would only need the information of the last 60 days. These matrices contain the correlation between every pair. The window gets moved in steps of 5 days from start +60 days to the end (2020). The matrix is used to build the network. All coins that have a correlation of more than 0.5 get an edge. Coins that do not have a single edge do not get added to the network. Coins with not enough data in the window also do not get added. This gives a varying network size from a small network of a few nodes around 2013 to a decent-sized network of around 2000 nodes around 2020[47].

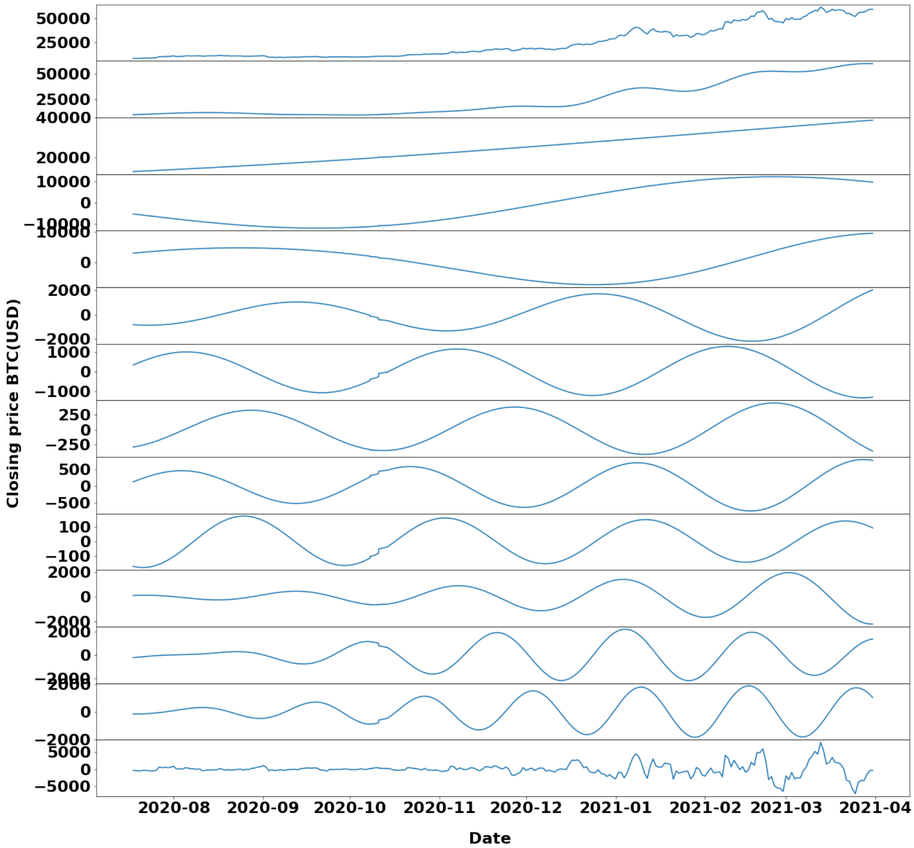


Fig. 3.5: An example of EWT for LSTM. The upper plot is the actual data. The second plot is the combination of the 12 signals, i.e. all filters except for the last one. The following 12 plots are the different signals filtered from the main signal and the last plot is the remaining noise.

3.4.2 Network measures

The betweenness is a centrality measure used in network analysis. It gives a fraction of how many of the shortest paths go to a specific node. The formula used for calculating the betweenness is given by:

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}. \tag{3.7}$$

In which V is the set of all nodes and $sigma(s,t)$ is the number of shortest paths from s to t. $\sigma(s,t|v)$ is the number of shortest paths from s to t going through the node v. The measure has a value range from 0 to 1 with 0 meaning that none of the shortest paths go through the given node and 1 meaning that all the shortest path goes through the node.

In the analysis, the average clustering is used which is the sum of all clustering coefficients divided by the total amount of nodes in the network. The clustering coefficient looks at how many triangles a node is a part of. So it looks as if the neighbors of a node are also connected. The formula of the clustering coefficient is given by:

$$c(u) = \frac{2T(u)}{deg(u)(deg(u)-1)}. \tag{3.8}$$

with $T(u)$ the number of triangles though node u and deg(u) the degree of u. This is for an unweighted graph. To see if this clustering has any statistical significance, clustering is compared to the clustering of randomized configuration models. This model generates random links between nodes respecting the sequence of degrees. This model is applied 5 times, over these 5 times the standard deviation is measured. The mean is taken and used to calculate the z-score with the formula:

$$Z_X = \frac{X - \langle X \rangle}{\sigma(X)}. \tag{3.9}$$

Here one takes the difference of X with the expected value of X or the average and divides this by the standard deviation of the average value.

The assortativity is a measure that reflects if there is a correlation between the degree of a given node and the degree of its neighbor. The values of the measure range from -1 to 1. An assortativity of 1 means that the degree of the neighbors is the same, so high degree nodes are connected to high degree nodes and small degree nodes are connected to small degree nodes. An assortativity of -1 means the degree of neighbors is opposite, so large degree nodes are connected with small degree nodes. The formula of how the assortativity is calculated is given by:

$$r = \frac{\sum_{xy} xy(e_{xy} - a_x b_y)}{\sigma_a \sigma_b}. \tag{3.10}$$

coming from [38]. e is the joint probability distribution of the degrees. a_x and b_x are the fractions of edges that start and end at vertices with values x and y.The two σ's are the standard deviations of the distributions a_x and b_x.

4 Results

> *Peace begins with a smile.*
>
> — **Mother Teresa**

4.1 Criticality

When looking at the result from both data sets for Bitcoin, there is almost no difference. One still sees the same phase transitions and at the same place. So the method is not data set dependent and thus has a certain stability. The dependence of dropping the empty data or linear interpolating is also explored, but for these two data cleaning methods, there is again found little difference. However, the interpolating method looked to be more stable, so will be used moving forward.

Fig.(4.1) is an example of the distributions. The distributions are those coming from the 13 different sets of $\Delta x_s(t)$, with s coming from eq.(3.3). The distributions and the PDFs have been multiplied with 0.1**i, with i the same i in eq.(3.3). The data used for this figure is that of Bitcoin from the year 2017, from the Gemini exchange. One can see that the fit agrees well with the observed distribution. One can also see a slow convergence to a Gauss, with only the last one beginning to resemble a Gauss. That is, with s equal to 32768, so the $\Delta x_s(t)$ is taken over 22.755 days.

In fig.4.2 one can see the fitted λs, from 2012 to 2021. The more white a line is, the higher the fitted λ is, so the fatter the tail of that distribution is. The darker the line, the more Gaussian the distribution is. When looking at the figure, one can see an overall decrease from left to right in the amount of white. This points to Bitcoin getting less correlated over time. This could be because of the increase in the market's size. With more and more people buying and selling Bitcoins. One can expect the influence of a single player to go down and thus correlation in the market also goes down. In the figure, one can see a few faint lines and one clear one. The clearest moment of criticality where correlation spreads over longer periods is around the end of 2018. To be specific, this is the crash of the 13th of November 2018, when the price of Bitcoin decreases rapidly. One can see this decrease in fig.

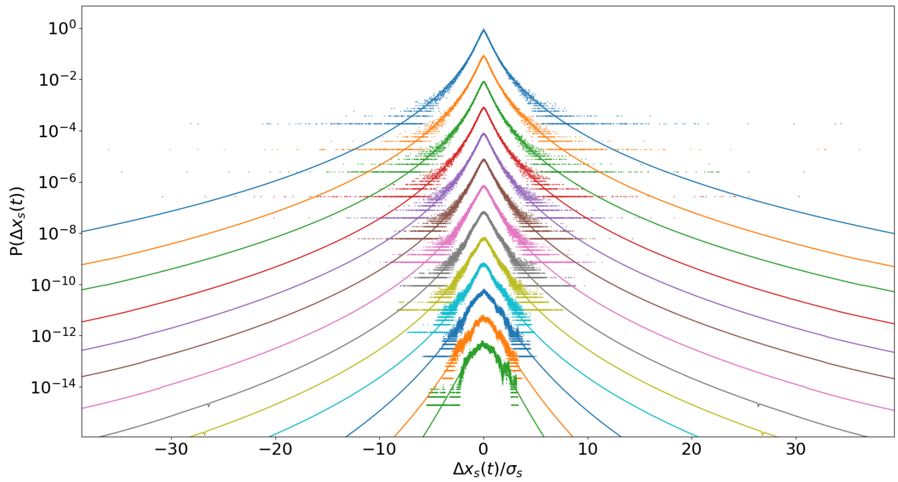


Fig. 4.1: P($\Delta x_s(t)$) versus $\Delta x_s(t)/\sigma_s$for the 13 distributions with their fitted PDF. s is increasing from top to bottom, following the sequence given by eq.(3.3). The distributions are multiplied with 10^{-i}, so that they appear underneath each other and are given multiple colours to make plot more clear.

3.1, just after the calm marked period. In around one month, the price of Bitcoin fell from around 6400 dollars to around 3200 dollars. This is a price decrease of 50%. When looking at the price, one could say that marks a transition from a phase of steady and rapid increase of the price, a bullish market, to a more flat market in which not too much happens, a bearish market. In between these two phases is a moment of criticality which is reminiscent of a phase transition.

Fig.4.3 is the same as fig.4.2 but now for Ethereum, from 2017 to 2022. One can see the same pattern as in fig.4.2, i.e. the overall amount of white goes down, which suggests the correlation over time goes down. Again, one can expect that this is because of the increase in the project's size. More and more players buy and sell the coin, so the price becomes more and more random. Again, as in fig.4.2, one can see the transition, is however for Ethereum less clear than for Bitcoin. but the line is still very much noticeable. Here, the transition is also around the crash of November the 13th 2018. However, the transition starts already before the crash, more on this later. For Ethereum, the price decreases from around 205 dollars to around 85 dollars. Which is a decrease of almost 60%. This transition is also as for Bitcoin, just before a period of not much movement, i.e. the price stays flat.

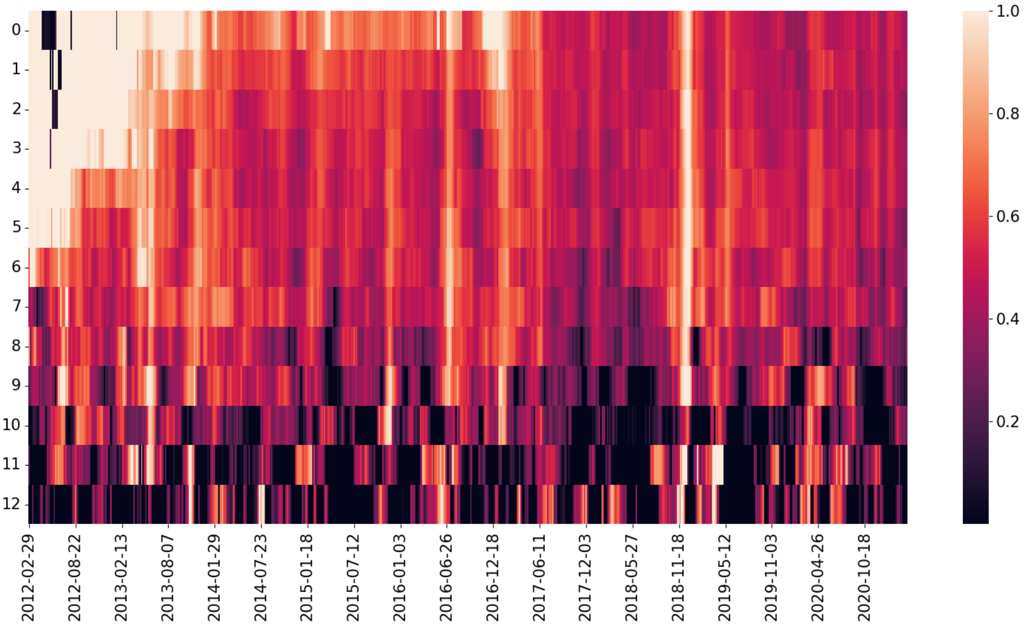


Fig. 4.2: The λs for the 13 exponents i from eq.(3.3) on the y-axis versus time from 2012 to 2021 for the closing price of Bitcoin. With the values going from 1 to 0. 0 being Gaussian behavior and 1 non-Gaussian behavior.

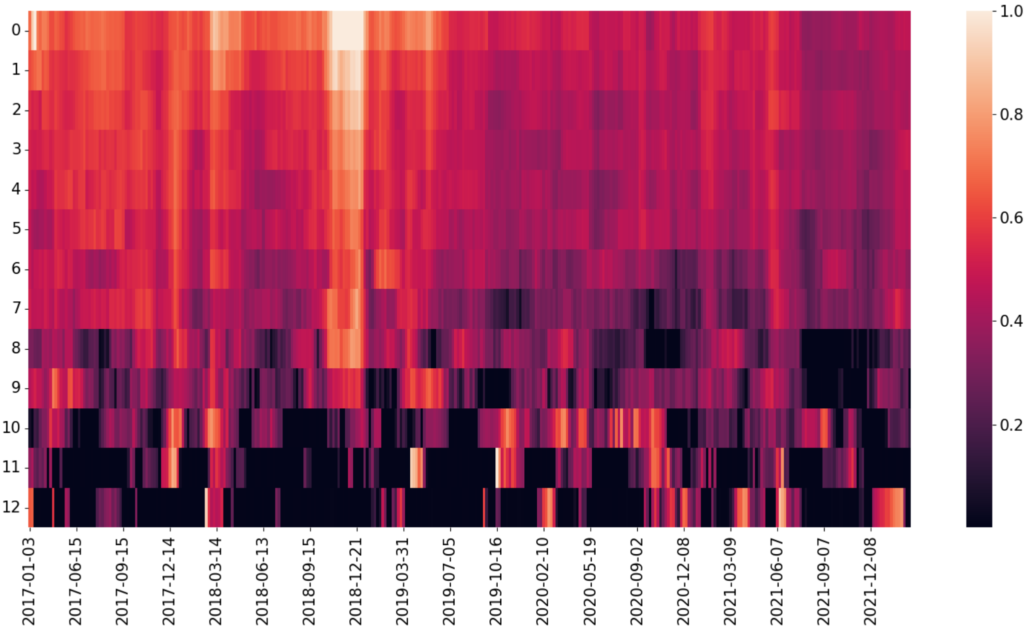


Fig. 4.3: The λs for the 13 exponents i from eq.(3.3) on the y-axis versus time from 2017 to 2022 for the closing price of Ethereum. With the values going from 1 to 0. 0 being Gaussian behavior and 1 non-Gaussian behavior.

Fig.4.4 shows a zoomed-in version of fig.4.2 on the crash of the 13th of November 2018. The figure shows the significant increase of λ starting before the crash. With the λs reaching a maximum just after the crash. This increase before the crash can be of interest as it can be used as a precursor for other crashes that are yet to happen. This can also be seen in fig.4.5, an increase before the crash and a maximum after and during, and then a decrease again. Fig.4.5 is done with data from the Gemini exchange, but the same things as for the other data can be seen. Even before the crash, there is some kind of maximum, i.e. a moment of criticality. When comparing this to the price of Ethereum, which looks the same as Bitcoin for that period, it looks very flat. The daily prices that are, but on the shorter time frame, there must already have been some build-up to the crash.

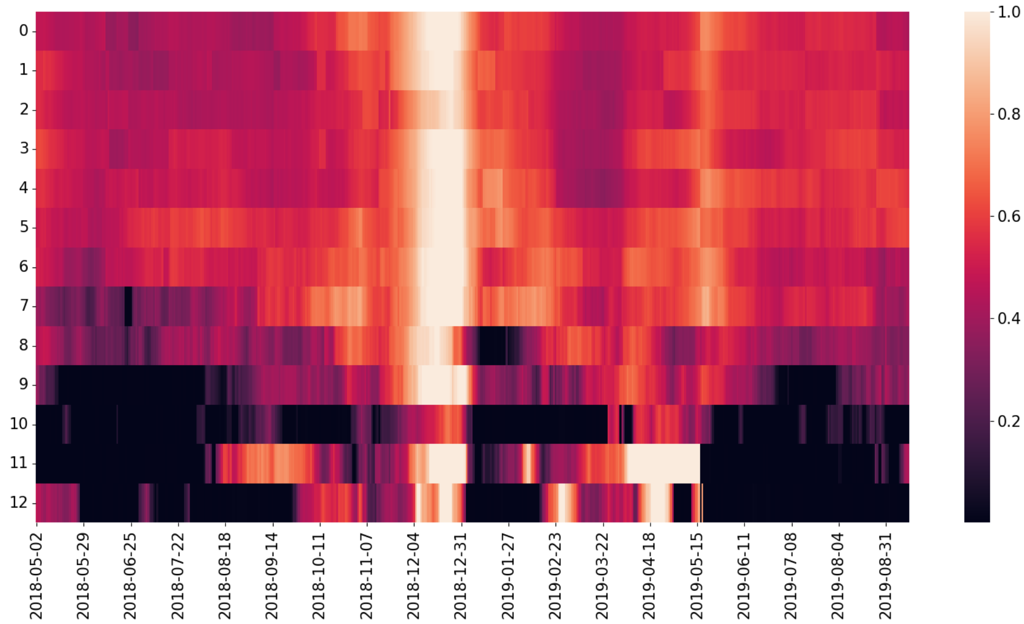


Fig. 4.4: The λs for the 13 exponents i from eq.(3.3) on the y-axis, zoomed in on the crash of the end of 2018, for the closing price of Bitcoin. With the values going from 1 to 0. 0 being Gaussian behavior and 1 non-Gaussian behavior.

Fig.4.6 shows the averages of the λs from Bitcoin and Ethereum throughout the crash. In this figure, one can see similar maximums for both Bitcoin and Ethereum, with Ethereum having a smaller peak before this maximum. In this figure, one can also better see the increase of the correlation, i.e. an increase in the λs, even before the crash. This shows there is some change in the dynamics of the market even before the crash, some build of non-Gaussian behavior. One can also see that from beginning to end, the moment of criticality takes approximately 4 months before the average of the λs gets lower than 0.5 again. This points to a lasting shock through the system.

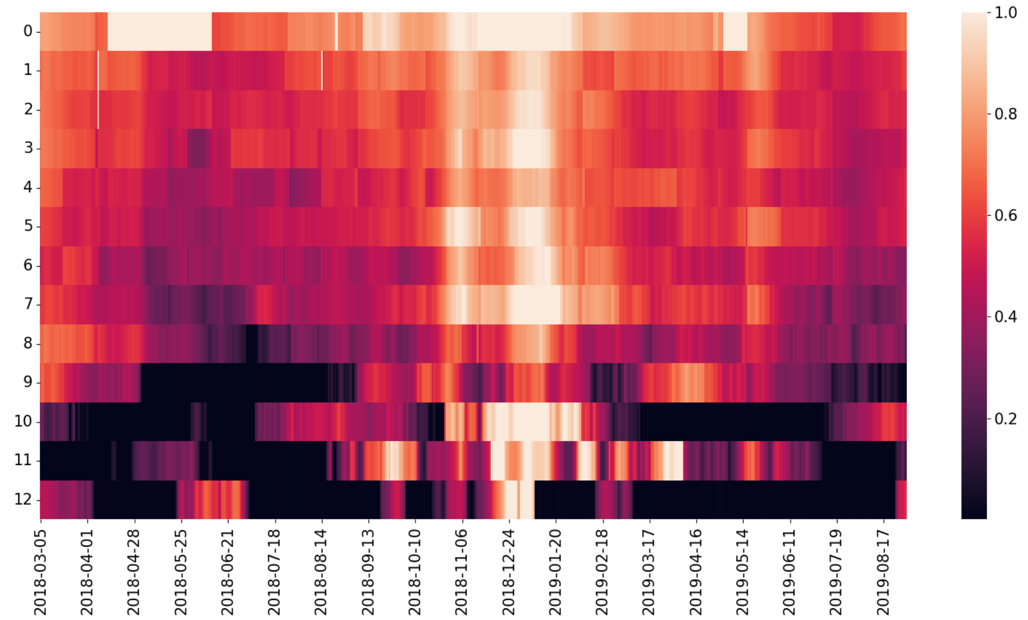


Fig. 4.5: The λs for the 13 exponents i from eq.(3.3) on the y-axis, zoomed in on the crash of the end of 2018 for the closing price of Ethereum. With the values going from 1 to 0. 0 being Gaussian behavior and 1 non-Gaussian behavior.

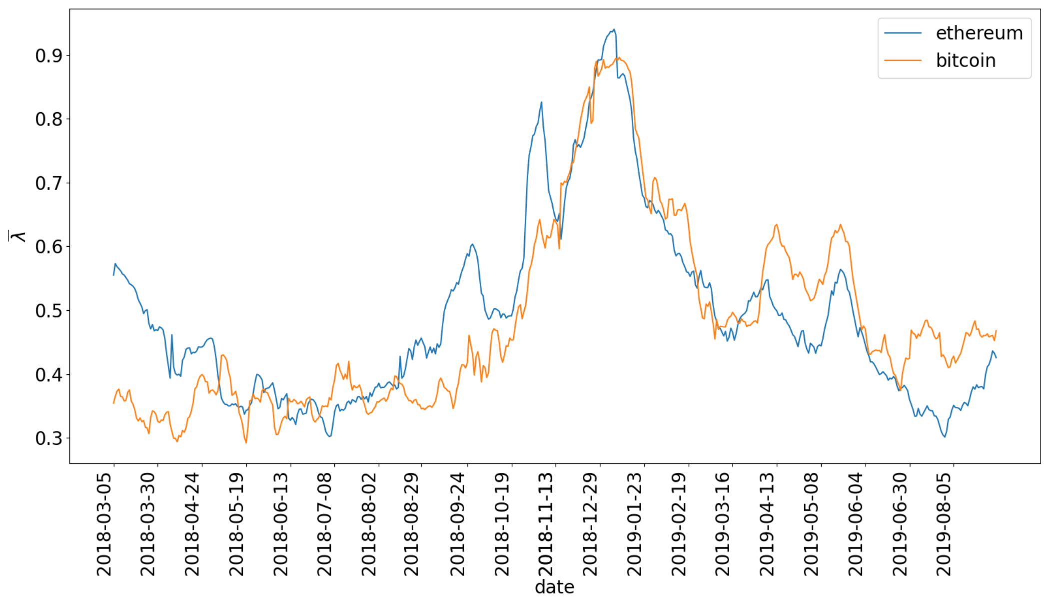


Fig. 4.6: The average of the different λ of the different layers of Bitcoin and Ethereum in function of time zoomed in around the crash of November the 13th.

4.2 Predictability

4.2.1 Optimising the parameters of RNN methods

Before the optimization is done, the dependency of the MASE on the amount of training data is looked at. An example of LSTM can be seen in figure (4.7). The results for these dependencies were about the same for every RNN. Here, one can see that the optimal amount of training data lies around 240 days. This means that the training data for the calm and bullish period have the perfect length.

The optimization plot for the parameters of the LSTM method is shown in figure (4.8). Here, one can see that the MASE is most influenced by the days to predict parameter. The minimum is around 10 days, which could be expected. It is impossible to predict every day precisely because of the big fluctuations in the data. After a lot of days, one's predictions will also become less accurate. Being in between, so around 10 days, one tries to more or less predict the trend in the data, which should be easier. So the optimal amount of days should lay somewhere in between 1 and a couple of weeks. This is exactly what one can see in the figure.

4.2.2 Testing the models

Once the optimal parameters of the RNNs are found, 10 models are ready to be tested. First, the models are tested on the test data (containing 16 days) using a time-series cross-validation combined with the MASE error (for the RNN methods this is done for 3 times 30 models, the 30 best, the 30 worst, and random 30). The results for the first data set are shown in figure (4.9) and (4.10) and the results for data set two are shown in figure (4.11) and (4.12).

The error bars in figures (4.9), (4.10), (4.11), and (4.12) are the deviations of the MASE for the RNN methods. ARIMA and Fbprophet don't have these error bars because there is no need for initializing weights for these methods. These methods will always give the same predictions. For the second data set, there has also been looked at EWT LSTM with 13 layers and the results for this method are also shown in figures (4.11) and (4.12).

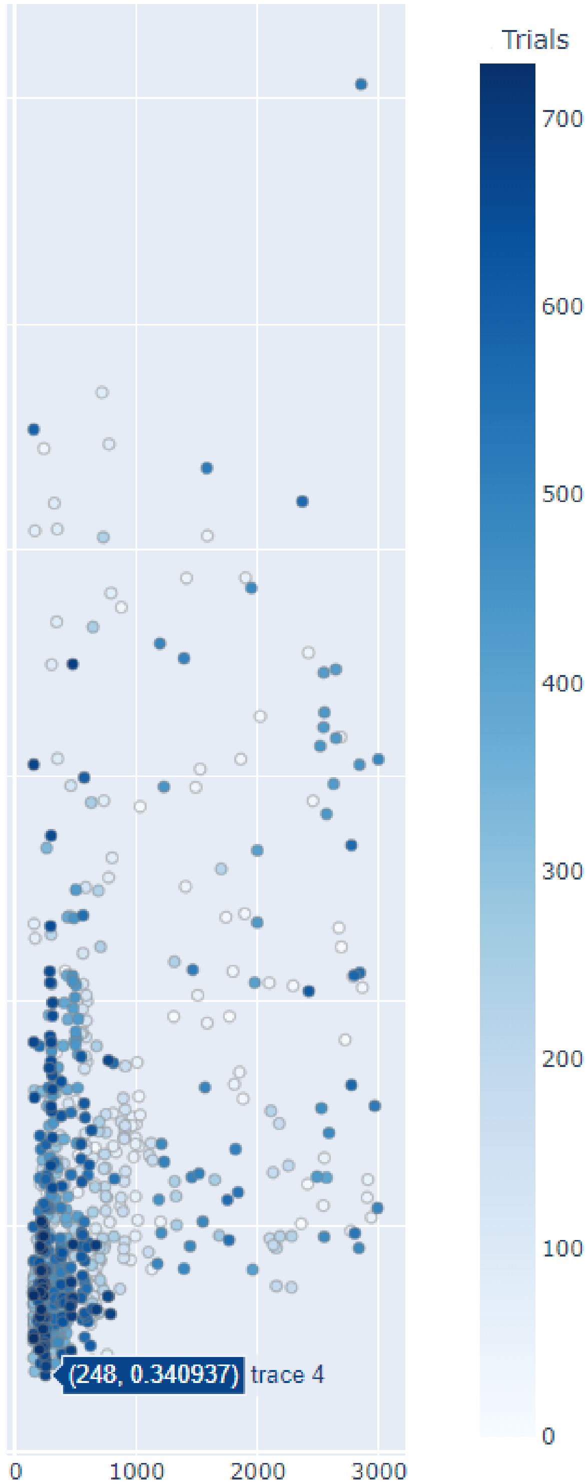


Fig. 4.7: The MASE error vs the size of the data. A minimum can be found around 240 days, which justifies the chosen lengths for the bullish period and calm period. The bar shows the order of the trails.

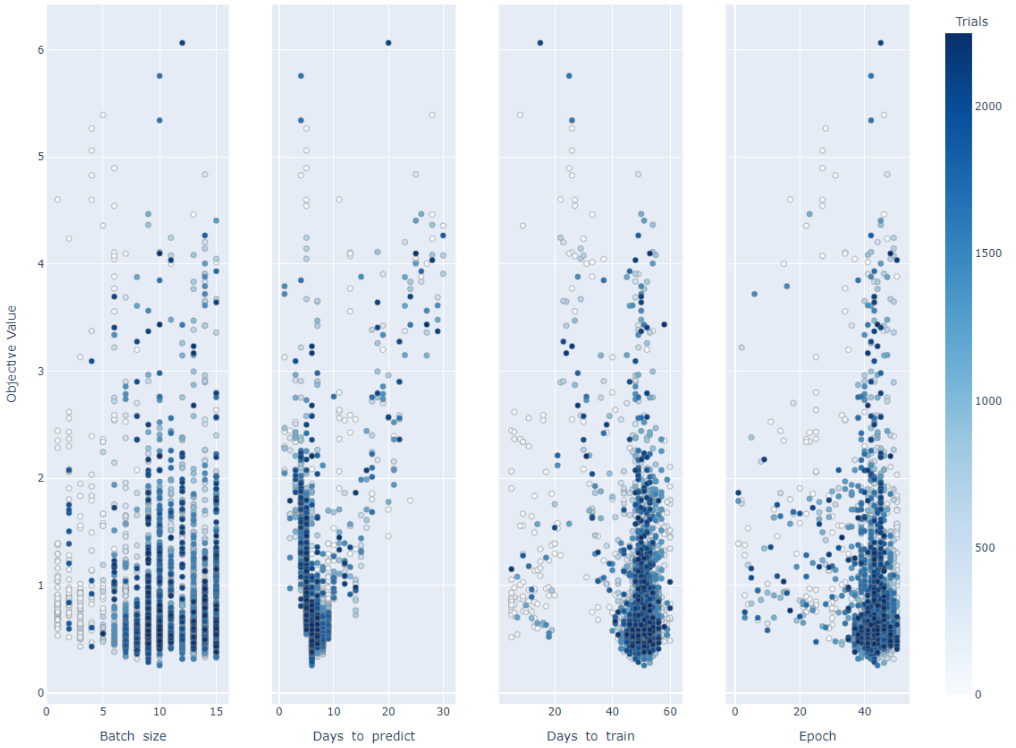


Fig. 4.8: The optimization plots for LSTM. On the y-axis, the mean value of the MASE plus its standard deviation is plotted, this is the objective value. Every point is the mean value of 10 runs. The bar shows the order of the trails.

The tests for the second test of EWT are not as extensive as the other methods, mainly due to how computationally expensive this method is. To test this method, the method is used in the same way as the other EWT method with fewer layers apart from the extensive parameter search. For this method, the number of epochs and data set size is chosen to be large, e.g. 500 epochs and 500 days in the data set, as this improves the predictive power of frequencies. The model is tested on random 8 random chosen days, chosen by a random generator over the whole data set minus the days the method needs to make predictions. Then the MASE for these days is measured. With the days one has to predict, in the filtering of the frequencies process, the model has predictability. As almost all errors are lower than 1. However if one takes away the days one has to predict, the method return error of around 3. So higher than some of the better predicting methods but significantly lower than the values of the method with fewer layers.

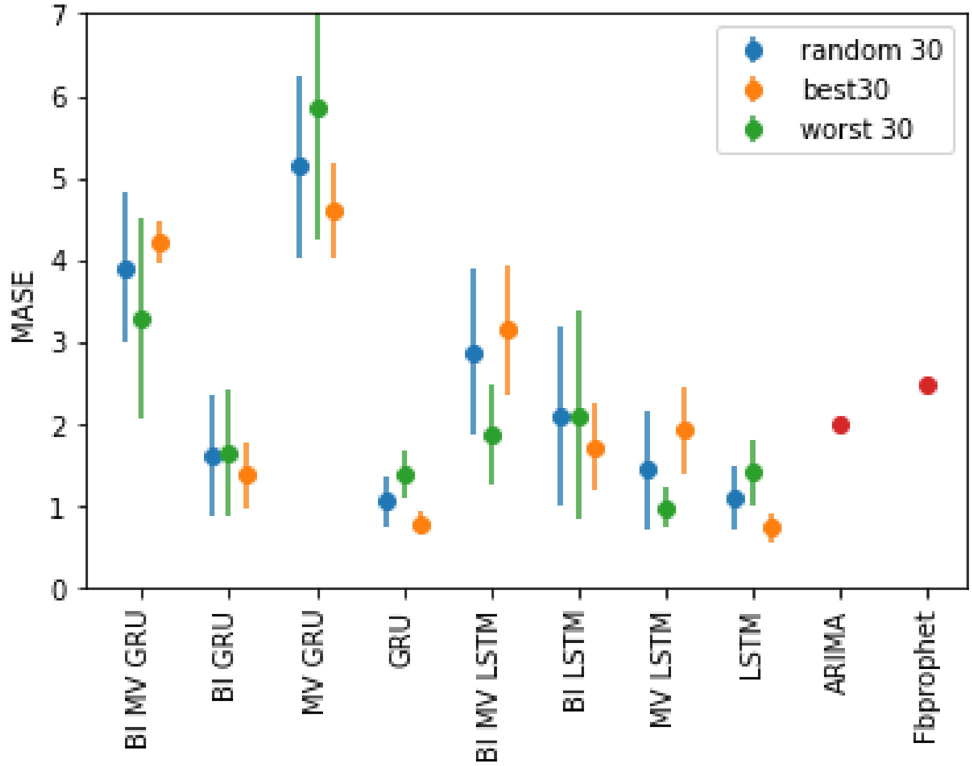


Fig. 4.9: The resulting mean value of the MASE after cross-validation. The deviation of the errors is also plotted in the figure as error bars. These are the results for the first data set over the bullish period.

4.2.3 Interpreting the results

In figures (4.9), (4.10), (4.11), and (4.12) one can see that the non-RNN methods are not the best methods. This is as expected because these methods try (one or multiple) simple fits, which is not enough to describe the chaotic time series of the Bitcoin (closing-) price. The non-RNN methods work better in the bullish period (for both data sets). The reason for this could be that the bullish period has a more dominant trend, which is easier to predict. The same conclusion holds for the ordinary LSTM and GRU.

If one looks at the RNNs with bidirectional layers, one can see that the non-multivariate methods have about the same performance in the Calm period as in the bullish period. There is also no difference between the first data set and the second data set. From this, one can conclude that the bidirectional layers are not

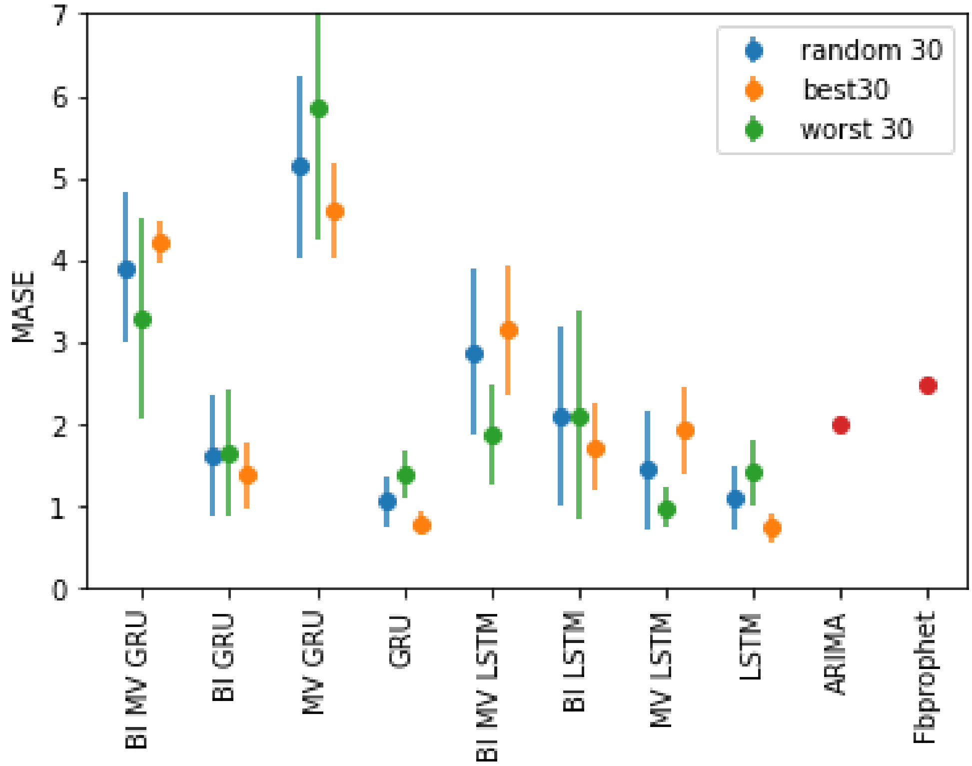


Fig. 4.10: The resulting mean value of the MASE after cross-validation. The deviation of the errors is also plotted in the figure as error bars. These are the results for the first data set over the calm period.

giving more predictability.

When looking at the multivariate methods, one can see that these mostly do a better job in the Calm period than in the bullish. The reason for this is that these methods get a lot of data (Closing price, open price, low price, high price, google search data, ...). In the bullish period, not only the closing price but all these features are changing faster than in the calm period, which makes it harder to optimize the weights of the neural networks. In the Calm period, these features are more stable and so it is easier to fit the model on this enormous amount of data.

An even more apparent effect of these methods can be seen when one compares these methods in the two different data sets. The multivariate RNNs perform better in data set two than in data set one. To explain this, one can look at fig. (4.13). Here, one can see the correlations between the distinct features (after the data is made stationary). One can see that even when removing the trend and seasonalities,

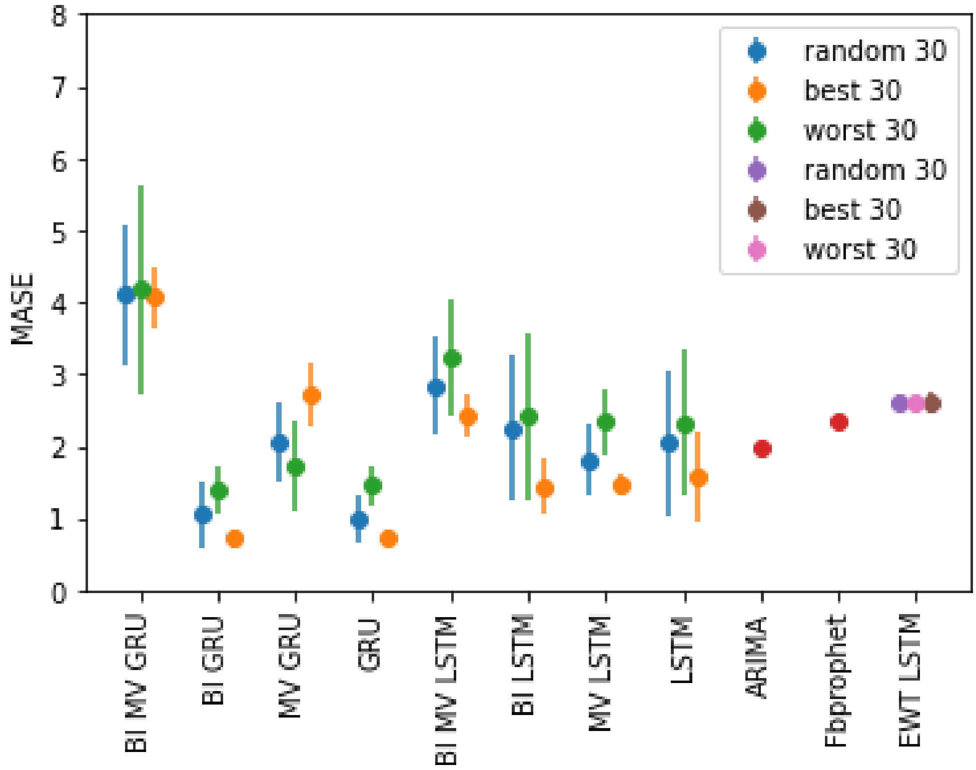


Fig. 4.11: The resulting mean value of the MASE after cross-validation. The deviation of the errors is also plotted as error bars. Also, the results of the EWT LSTM are shown. These are the results for the second data set over the bullish period.

the open, high, low, and closing prices in data set 1 are very highly correlated. The reason for this is that in this data set one does not just give all the features after one looked at the whole day (like one did in data set two). One looked at these prices every minute and at the end, one took the mean value of all these prices. Per-minute, these prices are about the same. This means that these 4 features in data set one contain almost the same information. This means that if one gives all of this data to the MV RNNs, one will do a lot of over-fitting. In data set two 4 prices are given after waiting a whole day, so here one has 4 different values and so one does not overfit the model. This can also be seen in fig. fig. (4.14).

The mean goal here was to predict the Closing price of Bitcoin giving values from previous days. Has this been achieved? To answer that question, one has to look at which method has a MASE smaller than one because if it's larger than one, the naïve model makes a better prediction. The naïve model just predicts the same value as the previous day and if this method gives better predictions, the data is just too

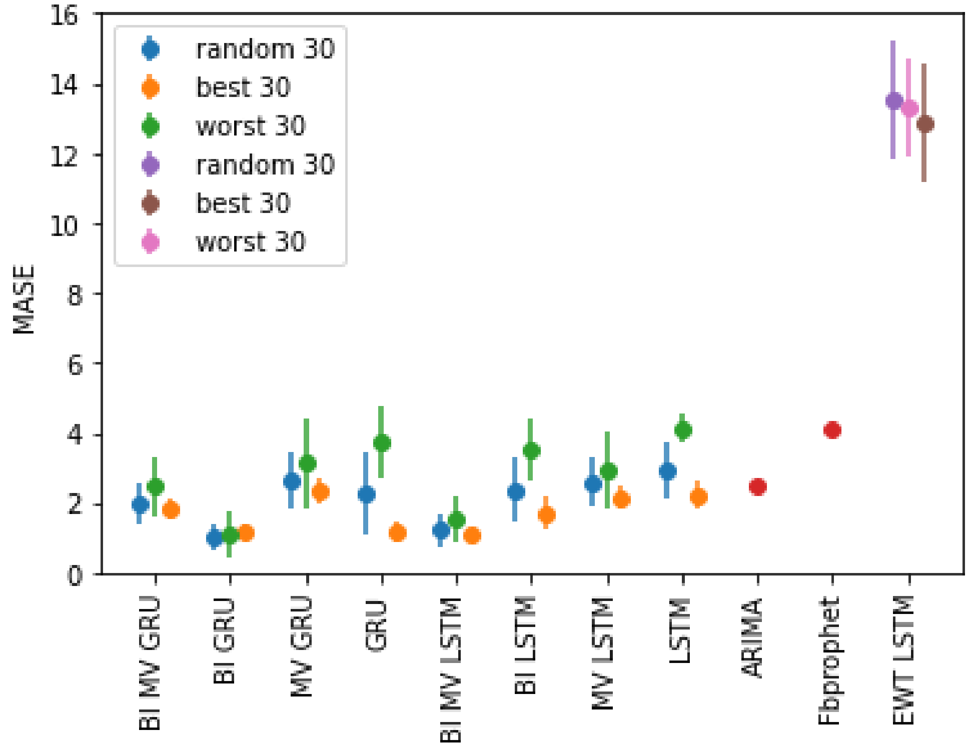


Fig. 4.12: The resulting mean value of the MASE after cross-validation. The deviation of the errors is also plotted as error bars. Also, the results of the EWT LSTM are shown.These are the results for the second data set over the calm period.

random for the more advanced method to make a good prediction. If one looks again at figures (4.9), (4.10), (4.11), and (4.12). One can see that some methods have a MASE smaller than one. In the bullish period of the data set, GRU and LSTM have some predictive power, i.e. if one takes the best 30 models after training and validating 100 models. In the calm period, it is BI GRU, which has a MASE smaller than one. For the second data set, it is GRU and BI GRU that do a great job in the bullish, but in the calm period, there is no model with a MASE smaller than one, i.e. the naïve model is the best one can do here.

If one looks at the EWT LSTM, then one can see that this method gives dire predictions. This method can be improved (see next section below) but this is not done due to time constraints.

To get a more visual view of how well a suitable method performs and how bad a poor model performs, the predictions of an example of both cases are plotted.

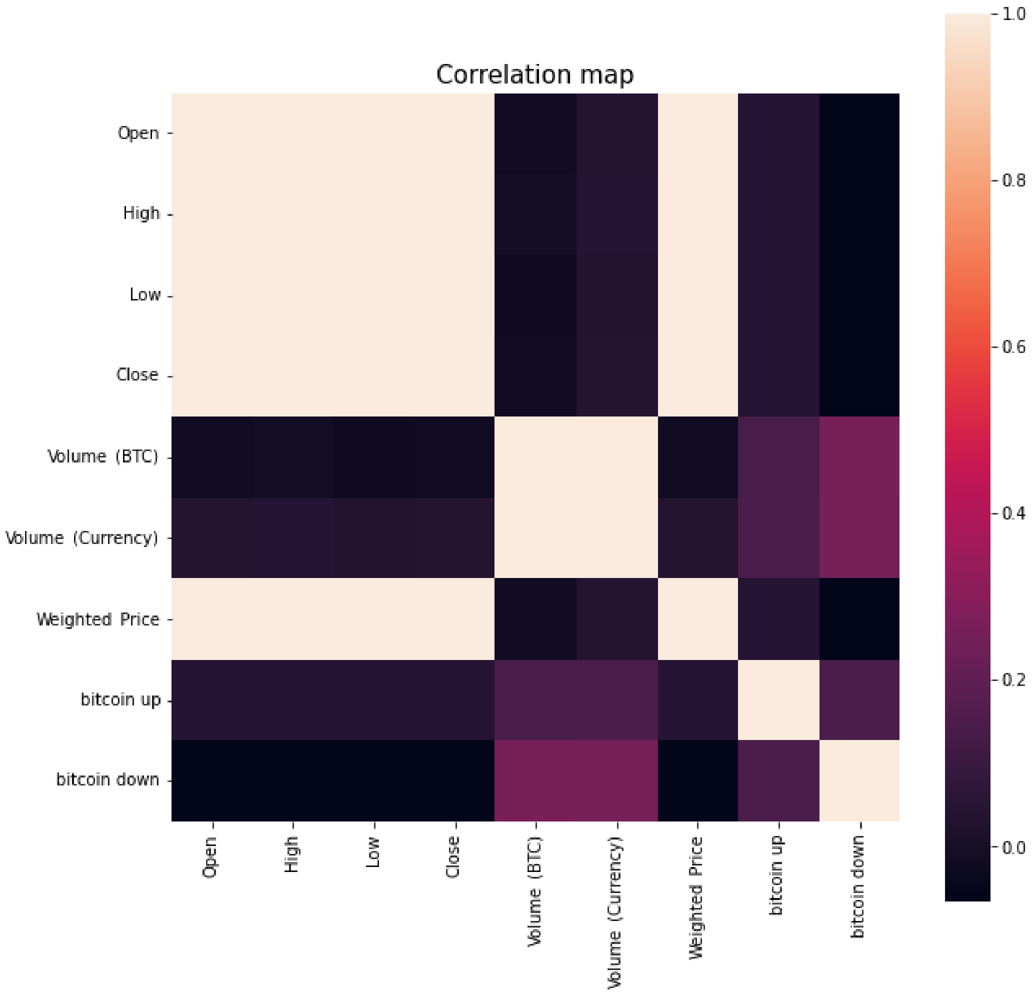


Fig. 4.13: The colour bar shows the colours with the corresponding Pearson correlation.The correlations between the distinct features that are given to the multivariate RNNs. Before making this plot, the data is made stationery, i.e. to remove the correlation because of the trend and seasonalities. These are the results for the first data set.

In fig.(4.15) one can see these predictions. Here, one can see clearly that the predictions lay far from the actual data for the BI MV GRU method. On the lower plot of fig.(4.15) the predictions are shown of the ordinary GRU with the second data set. Here, one sees the predictions lay much closer to the actual data. The actual trend is well predicted.

4.2.4 Correlation criticality and predictability

Now the different means of predicting are tested, the best model found in the previous sub-sections is used to look at the predictability over the period of the

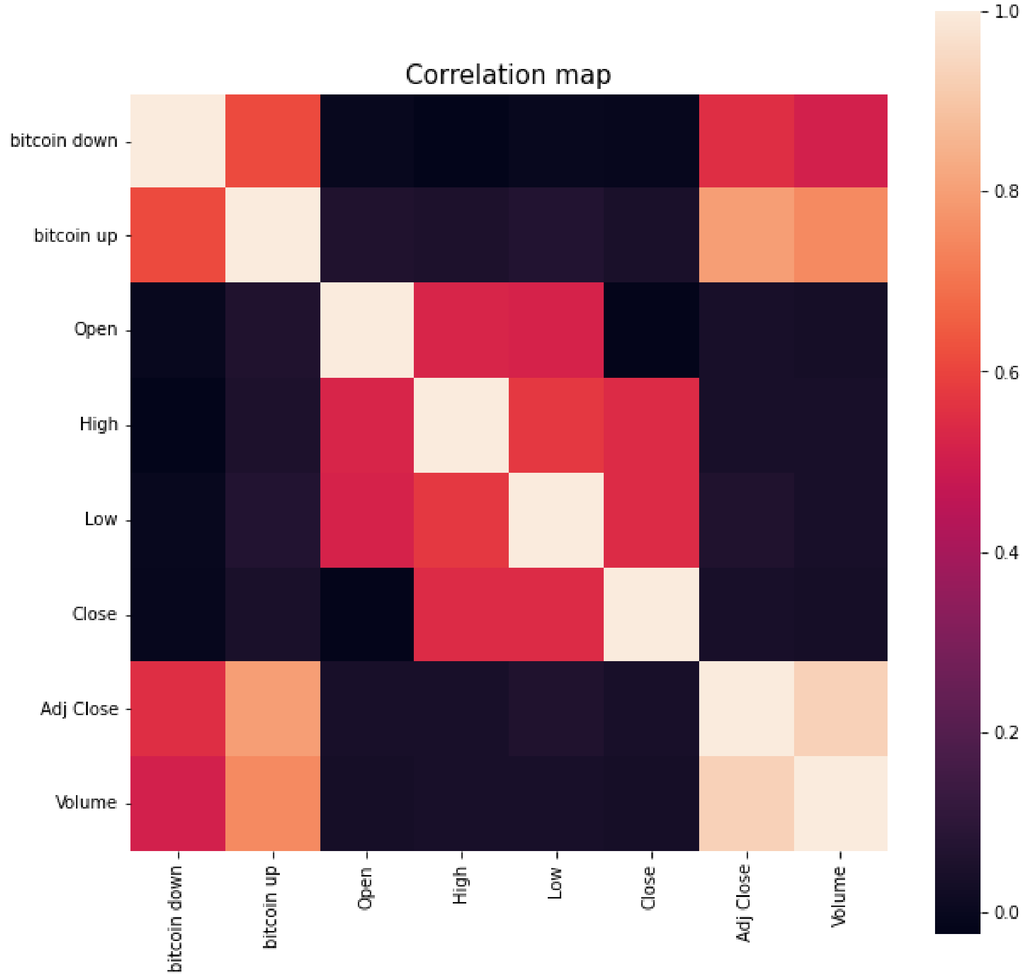
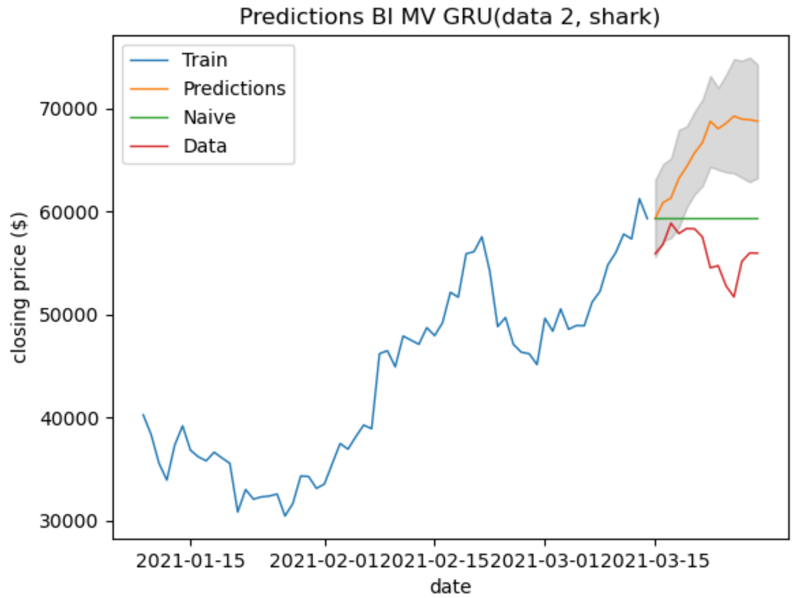


Fig. 4.14: The colour bar shows the colours with the corresponding Pearson correlation.The correlations between the distinct features that are given to the multivariate RNN. Before making this plot, the data is made stationery, i.e. to remove the correlation because of the trend and seasonalities. This is the second dataset.

phase transition found in the criticality part of the thesis(4.1). Looking at the results of the previous subsection, one can see that GRU performed the best or second-best in almost all scenarios. So in this section, GRU is used as a predictor.

In fig. 4.16 one can see the averages, like in fig. 4.6 throughout the crash. In this figure, one can also see the errors in the predictions made by the model. These are the RMSE errors instead of the MASE errors, as here it's the correlation that is being looked at. Because the MASE is a scaled error, concerning another error, this makes it harder to compare predictive power and the moment of criticality. The errors indicated by the points with different colors have error bars because these are averages of the top 5 performing GRU models over the training data. The errors are



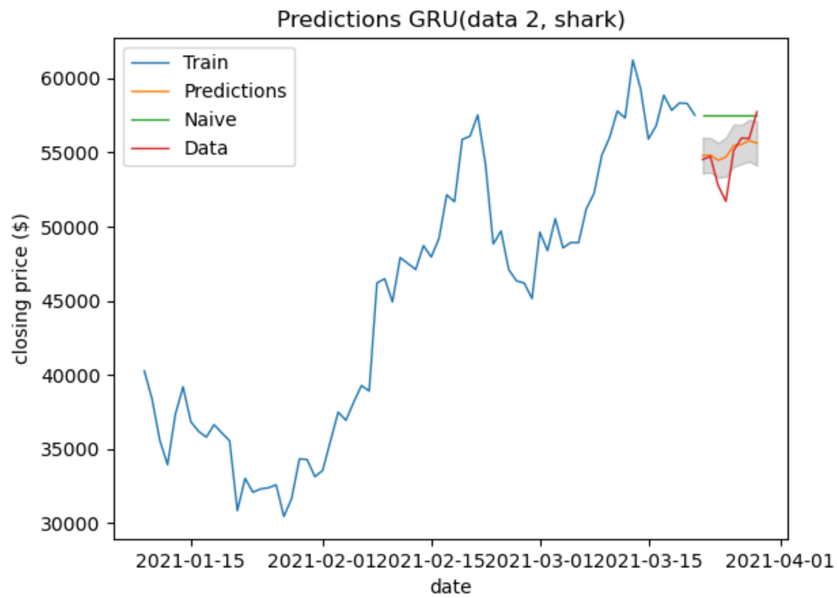


Fig. 4.15: The upper plot shows the predictions of BI MV GRU for data set two in the bullish period. The lower plot shows the predictions of GRU for data set two in the bullish period.

indicated with different colors because each color is a different model, i.e. a model that is trained just before that period. The red dashed line one can see in the figure is the moment of the crash. One can see the increase of the error at the moment of the crash. This is probably because the model has a hard time predicting the crash. Calculating the Pearson correlation coefficient, between the error and the average of 0.31, is obtained. This indicated a small positive correlation between the increase of correlation in the market and a decrease in predictive power. When one changes the experiment so that instead of predicting all-new days with the model and on instead moves the days that are predicted only one day to the future. So some training data is being predicted. Then the correlation becomes around 0.7, which is an increase of 0.4 from the other correlation coefficient.

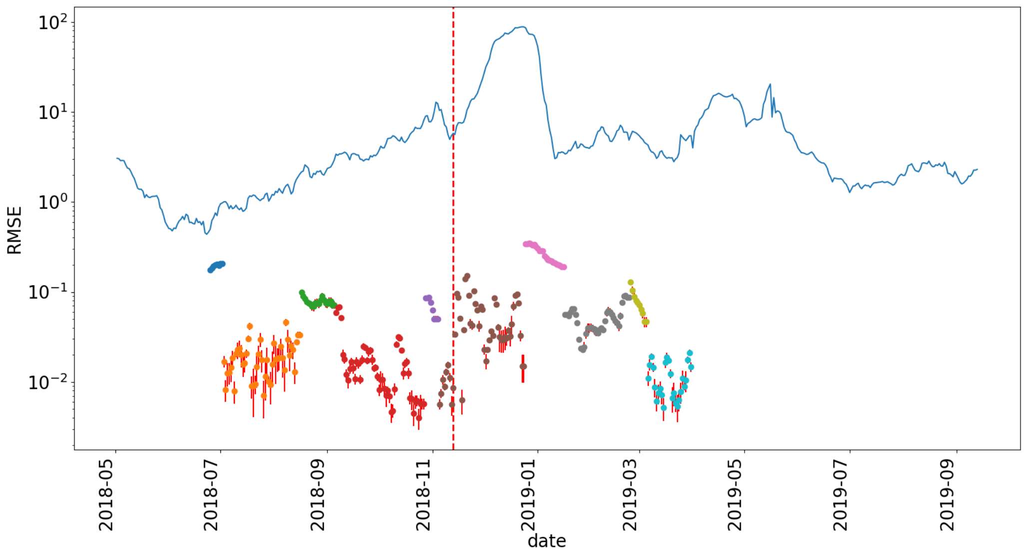


Fig. 4.16: The average of the different λs of the different layers in function of time together with RMSE errors of predictions of the chosen model. The red line indicates the crash. The different colours of the errors depict that the model is trained over different periods.

In fig. 4.17 one can see the same as for fig. 4.16, however, now it is the MASE error that can be seen instead of the RMSE error. In this figure, the horizontal dashed red line is a line for the y-axis being equal to one. This is done because this line indicated predictive power. As being lower than this line, the model predicted better than just taking the last day and taking the same value for the closing price of Bitcoin. The Pearson correlation for these errors and the average λs is around 0. In the figure one can see the errors are often underneath 1, some models more than others, showing some predictive power.

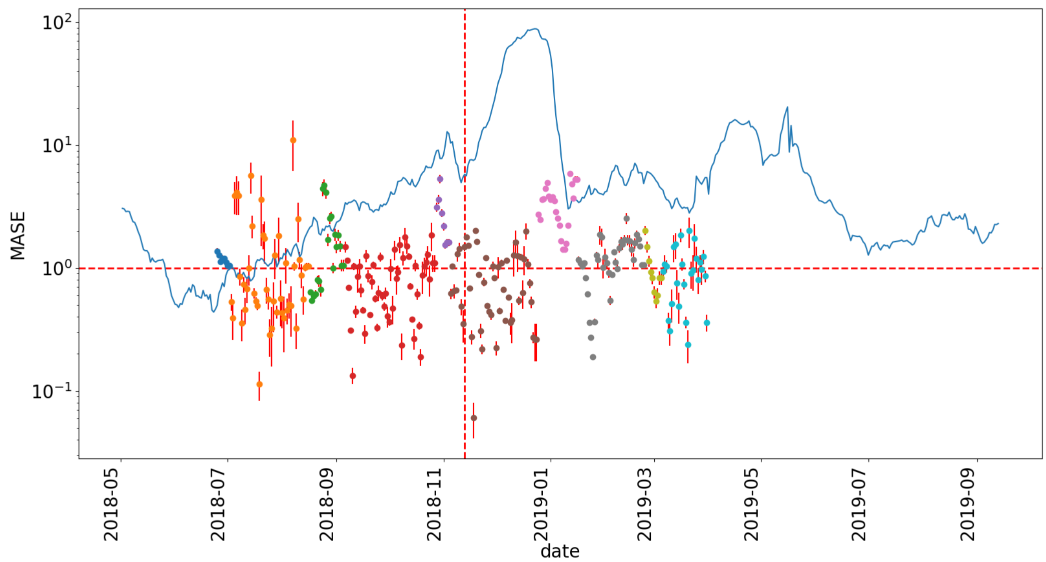


Fig. 4.17: The average of the different λs of the different layers in function of time together with MASE errors of predictions of the chosen model. The vertical line indicates the crash and the horizontal line indicates 1 on the vertical axis. The different colors of the errors depict that the model is trained over different periods.

4.3 The cryptocurrency network

A visualization of the network can be seen in fig.(4.18). This network contains 1853 different coins and 366530 edges connecting them. The size of the nodes depends on their degree and the color is depending on their betweenness. The core is not red; it is mostly the coins between the core and the outliers that correlate the most with both groups. These get a higher betweenness, as seen in the figure as they are redder.

4.3.1 Overall view of the network

This first part is about looking at the entire timeline. To see how the network changes over longer periods and also get a feeling about how much it changes. This will be done with some basic measures, e.g. the clustering coefficient.

In fig.(4.19) one can see the number of vertices and edges for the whole of the data set. One can see a strong increase from the beginning till the end. This is because the cryptocurrency market is so new that a lot of cryptocurrencies are still being

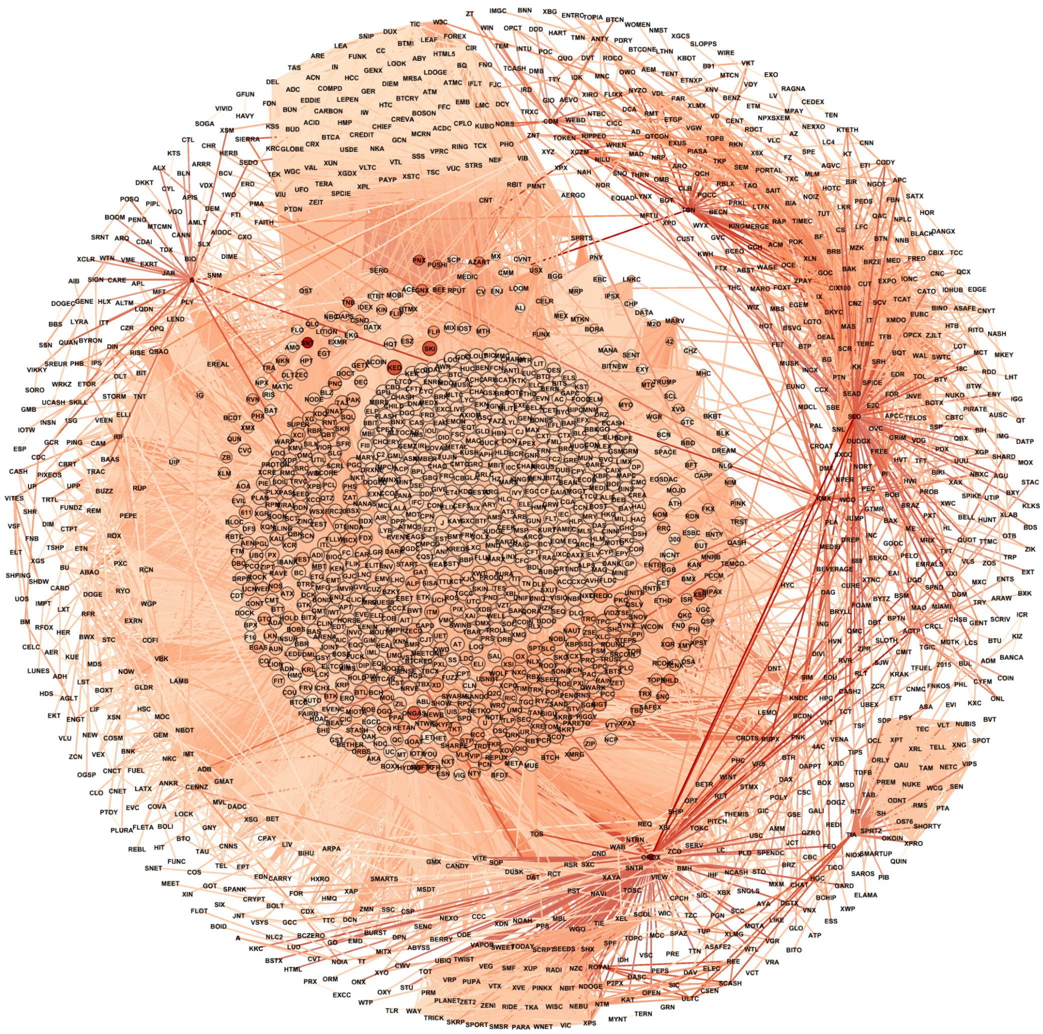


Fig. 4.18: Visualization of the network on 21-12-2019.

created. Each one with its new technology or idea, which can trickle down to other branches[44, 15, 34].

From fig.(4.19) one might expect that the average correlation of the market increases as the number of edges increases much more than the number of vertices. However, the number of edges is scalable, as the square to the number of edges. Looking at fig.(4.20) one sees the correlation has an overall downtrend. This is probably because, as the market gets bigger, the average individual influence of cryptocurrencies on each other gets smaller as the market gets diluted. In fig.(4.20) one can also see the betweenness and the number of connected components. The amount of connected components increases as expected as the network increases fast, but it does not exceed 30 and is mostly around 10. With most of the components being pairs that are only correlated with each other. This means a connected compo-

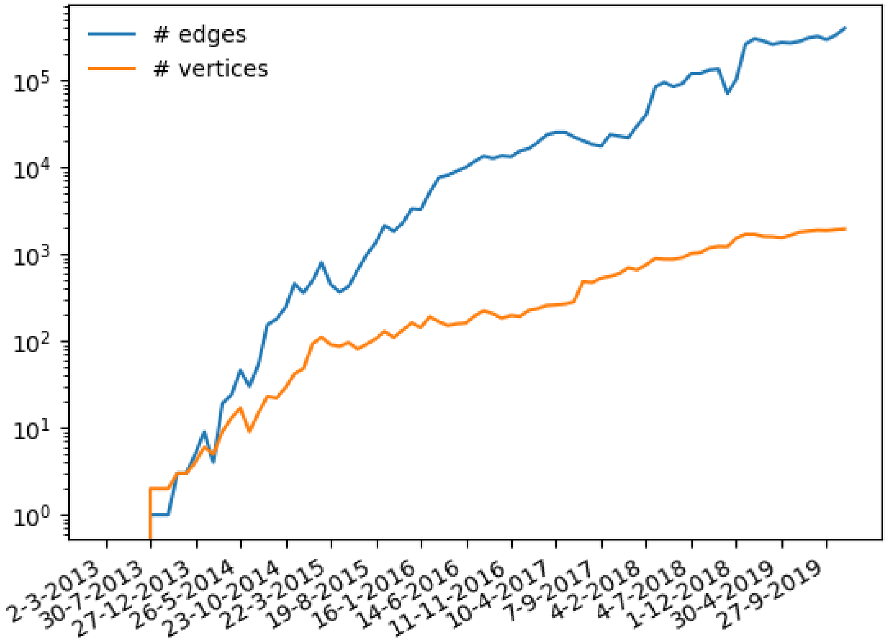


Fig. 4.19: The number of vertices(cryptocurrencies) and edges in the network from 2/3/2013 till 21/12/2019.

nent analysis can not be done. The network is thus mostly a well-connected giant component without many minor components. The figure shows that the average betweenness decreases. This is as expected as the size of the network increase and thus the number of shortest pads increases.

In fig.(4.21) one sees that the clustering does not have a trend and is around 0.65, which means there is some clustering happening in the network but it does not change with time. However, if one looks closely at the end of 2018 when the crash happened, one can see a decrease in clustering.

From fig.(4.21) one sees that the clustering is there not by chance as the Z-score is around 20. The z-score of the clustering even shows an upward trend, this could be because of the increase in the network's size, which could make the results of the configuration models be more close to each other and thus give a smaller standard deviation. By looking at eq.(3.9) one can see that this would increase the z-score.

Fig.(4.22) is the same as fig. (4.21) but here, the measure can also be negative. If one looks at the z-score, one sees that, from the beginning, the assortativity

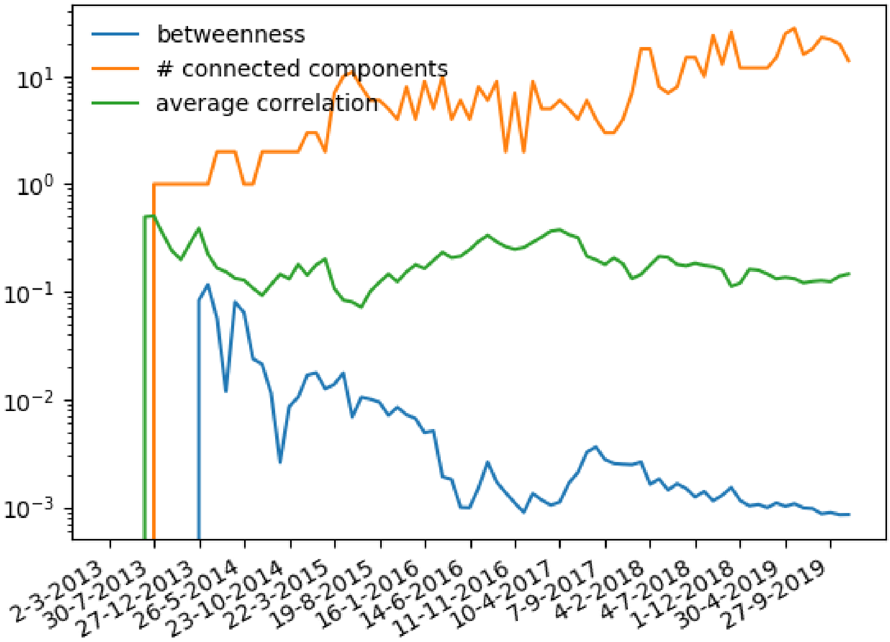


Fig. 4.20: The betweenness, the number of connected components, and the average correlation from beginning to end.

differs from a randomized configuration model. So there is a correlation between a cryptocurrencies influence, i.e. its degree, and the type of neighbors they have. In 2018, right around when the crash happened, which will be better explored in part 2, the assortativity changed signs, from negative before the crash to positive after the crash. So during this period, cryptocurrencies changed from more influencing smaller coins to more influencing larger coins that also have a lot of influence. One can see this effect of having a positive assortativity in fig. (4.18), here one can see that coins with a high degree are close together as most of their links are to high degree coins.

4.3.2 A better look at the crash

Degree distribution analysis

This section is about a degree distribution analysis of the network around the time of the crash on 13-11-2018. This was one of the biggest and fastest crashes in the history of the cryptocurrency market. The two figures of this section are fig.(4.23)

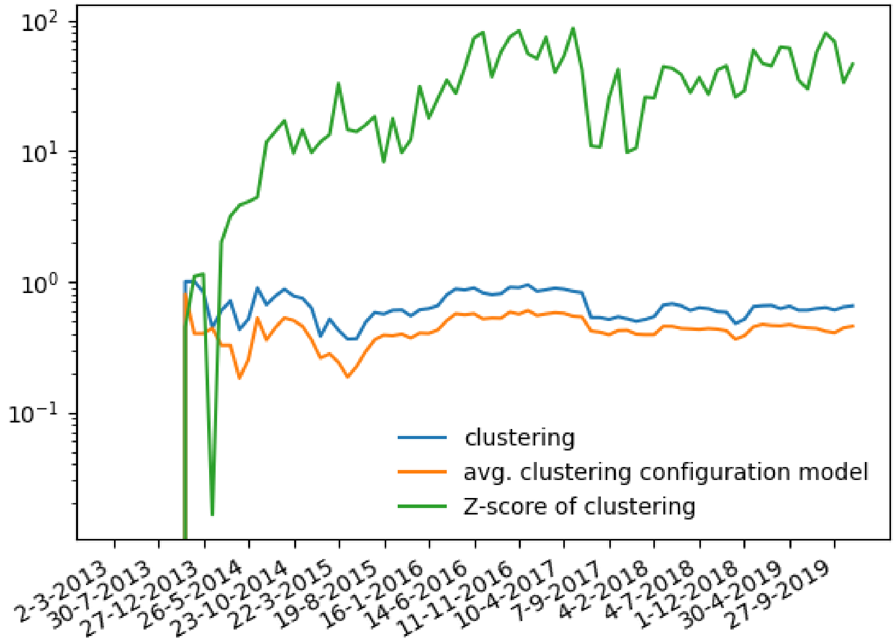


Fig. 4.21: The clustering, the average clustering of randomised configuration models, and the Z-score of the observed clustering from beginning to end.

and fig.(4.24). In the first figure, one can see three distributions. One before the crash, one during the crash, and one after the crash. The one before was around 5 weeks before the crash and the one after around 8 weeks after. If one looks at the first one, one can see that the network is heavenly biased towards vertices with large degrees. Which differs from the usual degree distribution form[45], or a stock market degree distribution [52], which both follow better a power-law distribution. This trend persists over the entire timeline and means that the network is made mostly of the large core with vertices having a high chance of having a high degree. In fig.(4.18) one can already see this large core in the center of the figure.

When comparing the different distributions of fig.(4.23). One sees that both before and after the crash, the distributions are more shifted to higher degrees. Through the shock of the crash, the cryptocurrencies became less correlated, and thus the cryptocurrencies lost edges. When looking at fig.(4.19) one can indeed see a dip in the number of edges.

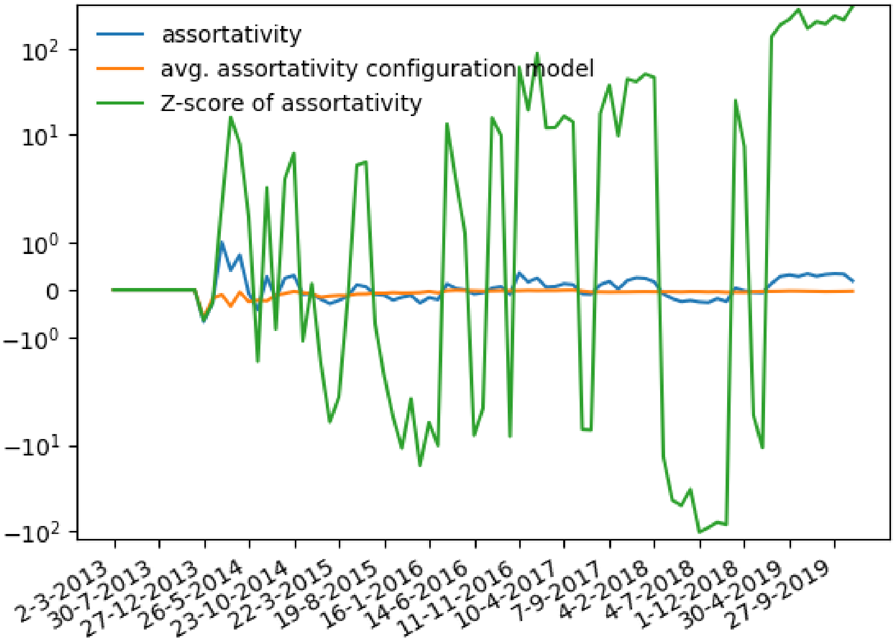


Fig. 4.22: The assortativity, the average assortativity of randomised configuration models, and the Z-score of the observed assortativity from beginning to end.

When looking at the fig. (4.24) for a more wide view of the distribution. This is a fit to the distributions, but on a wider scale. This fit was made by fitting the equation:

$$y = \frac{\beta}{x^{\alpha}}. \tag{4.1}$$

Fig.(4.24) shows the α of eq.(4.1). One can indeed again see that around the crash one gets larger α. This means that distribution around this time was shifted more to smaller degree vertices. Fig.(4.24) is interesting because one can see the increase in the fit even before the crash, which could mean this could be a signal used to signal that there is a crash coming.

Motif analysis

This part is about a motif analysis, which is done by using the program FANMOD[56]. This program goes over the network and looks for motifs of a selected size. For this analysis, these are motifs made of 3 vertices. For an undirected network there are two different triadics, that is the fully connected one and one with a missing

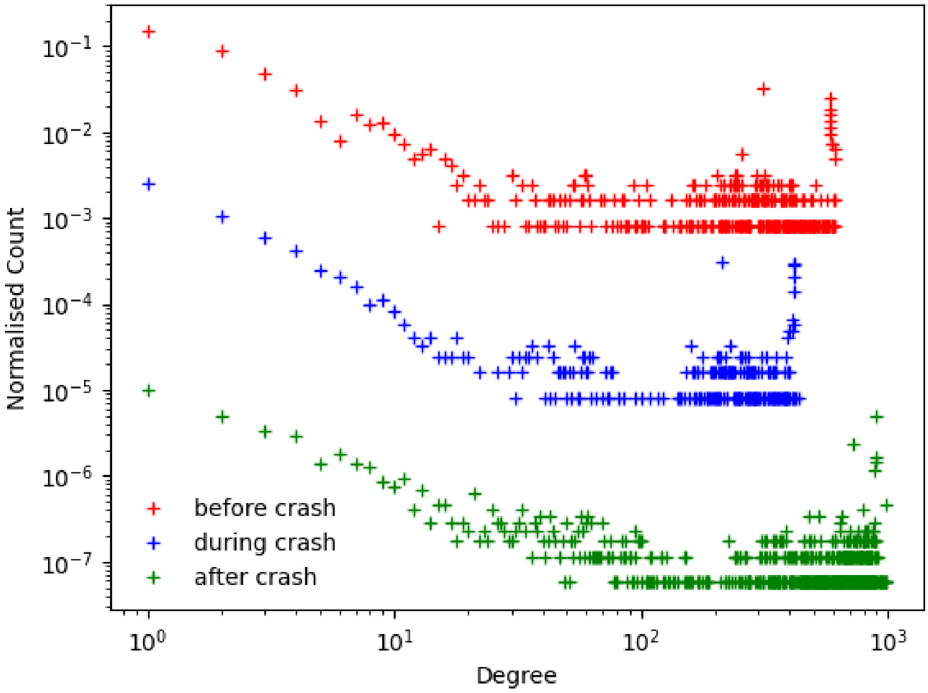


Fig. 4.23: The degree distributions before, during, and after the crash. The distributions are multiplied with 10^{-i} so that the plot is more clear.

link. The values found here for the one with a missing edge are the same for the fully connected ones but negative. This means that the not fully connected ones are found exactly the amount the fully connected ones are found more than in a randomized network but then less than in the randomized network.

In fig.(4.25) one can see the z-score of the fully connected triadic motifs, compared to 7 randomized networks. A positive z-score means that the frequency of the motif that was found is higher than the observed motifs in the randomized networks. This thus means that the neighbors of a cryptocurrency are more likely to be neighbors as well. This could be because of the well-connected core central in the network. One can also see that the z-score changes over time, getting higher around the crash. The increase happens already before the crash and is not an effect after the crash. This is interesting as this might be used as a signal if this would indeed always be the case.

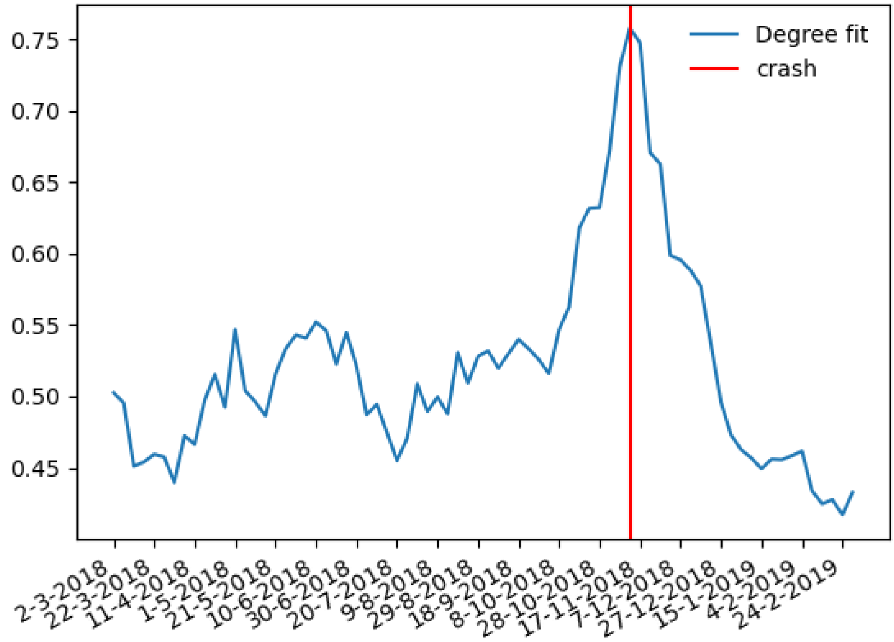


Fig. 4.24: α of eq.(4.1) of the fit to the degree distribution during the period around the crash.

correlation and cycle analysis

An attempt is made to find a correlation between the price of a cryptocurrency and the k-score value it has. This is done by making a list of the k-core values of all cryptocurrencies in the network at some moment and then calculating the Pearson correlation with a list of the cryptocurrency prices. But no correlation is found. The values are around 0.03. This correlation is measured over the period of fig.(4.24) but does not exceed 0.05. There is thus no correlation between the k-core of a coin and its prices. This is as expected, as the prize of a cryptocurrency is random and can be chosen at the start of its creation. Because the number of circulating coins can be chosen. More interesting would be to compare the k-core with a coins market cap. But the database does not have this information.

In fig.(4.26) one can see α of eq.(4.1) fitted to the cycle distribution. Cycle distributions show a better power-law distribution than degree distributions. But one can see in fig.(4.26) α does not predict the crash. α increases on the crash. But it is not out of the ordinary when comparing it to the data around the crash. One can thus

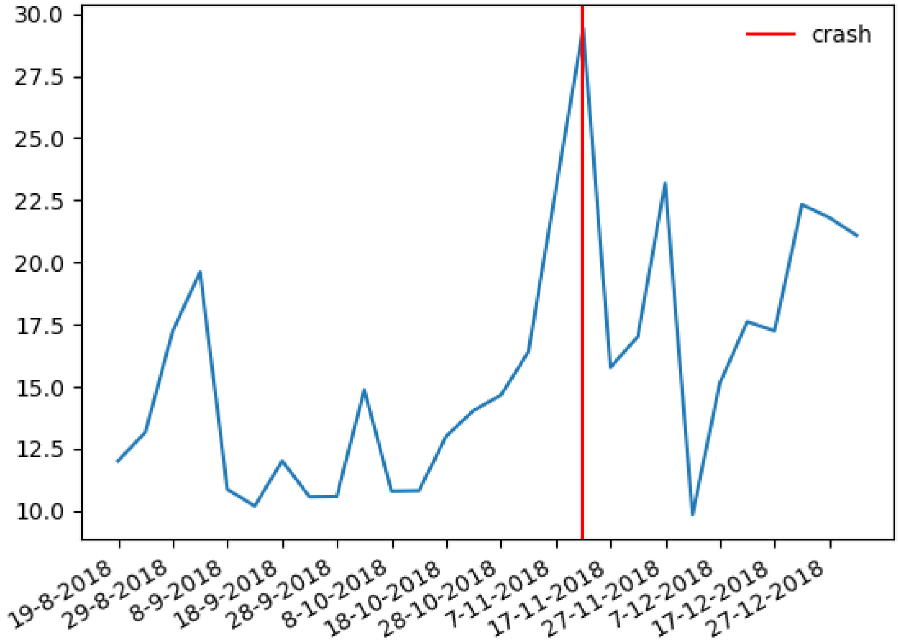


Fig. 4.25: The z-score of the triadic motifs of the network around the period of the crash.

conclude that here, the cycle can not be used as a clear signal. The shock of the network thus does not change the cycles too much.

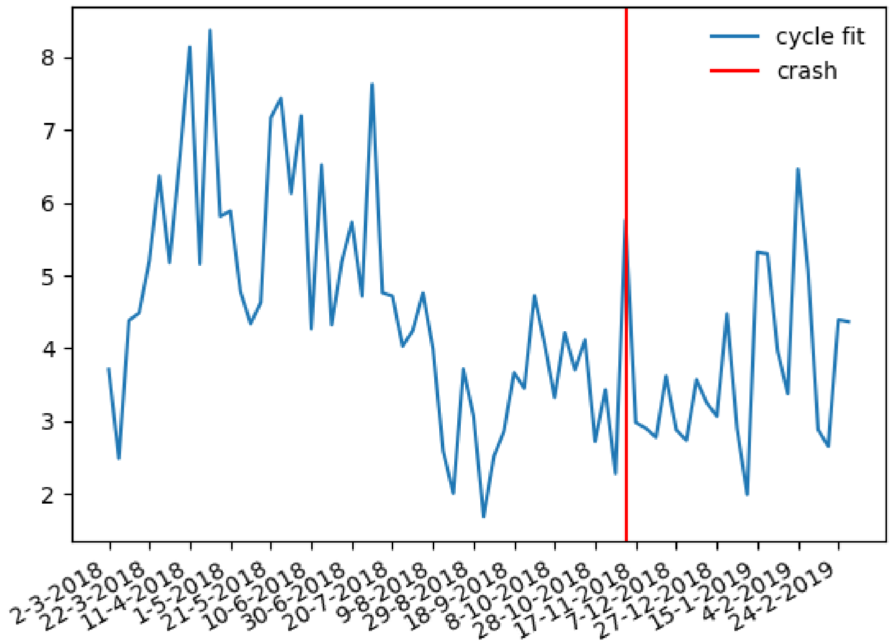


Fig. 4.26: α of eq.(4.1) of the fit to the cycle distribution during the period around the crash.

5 Conclusion

The thesis is about finding a moment of criticality resembling a phase transition and looking at its effects on the market. This moment has been found, i.e. a moment of scale-free behavior has been observed. The fitting parameter that tells one how Gaussian a distribution is, is during this period independent of the length of time between two observations. This measure could help, for example, investors with risk analyses of cryptocurrencies or stocks. More research should be done on this way of looking at the market. The measure of the average lambda could help signal a crash. One could see a build of correlation in the market before the crash happened. However, it is not shown that this measure is consistent with all crashes. A way of improving this way of looking at the data would be by combining the results of different data sets, as fluctuations because of the way the data is gathered cancel each other out. Or combine the results of many currencies so that one gets a better view of what the entire market is doing instead of just one cryptocurrency.

Some models showed evidence of predictive power. The results showed a dependence on data and in which kind of period one is trying to make predictions, i.e. a calm period or a bullish period. It is found that with the first data set, GRU and LSTM have a MASE smaller than one in the bullish period, while it is BI GRU that has this in the Calm period. For the second data set, there is predictive power with BI GRU and GRU in the bullish period, while none of the models did well in the calm period. When comparing the indicator of the phase transition, i.e. the average of the different λ's, over time with the predictability, i.e. the error RMSE, there is a correlation. This correlation is, however, low, i.e. 0.3. This could be because of the difficulty of making predictions because of the randomness of the data. As random predictions would give a correlation of 0.

Different measures are used in different ways to analyze the network of cryptocurrencies. One of the more useful ones was the degree distribution. Which showed the potential of being a signal, as α(the slope of the distribution) increased even before the crash. The motif analysis showed possible usable behavior, as the z-score increased even before the crash. Some further work can be done on the correlation between market cap and different measures, e.g. k-core. The market cap could be a better indicator of influence as it represents the size of a project. Also, one could look at different motif sizes and further research can be done on the signaling potential of the degree fit and motif analysis by looking at more crashes.

One can do some improvements to get better predictions. Especially for the EWT part, one can do a lot of optimizations. One can, for example, make more EWT layers. The more filters one considers, the more accurate the decomposition represents the actual price. For the optimization of the EWT LSTM model, the parameters of the EWT layers for all the LSTM layers are optimized at once. Better would be that one optimized the parameters of the EWT layers for every LSTM layer individually because the optimal parameters could differ between the different LSTM layers. One can also use this EWT procedure for other RNN methods than the ordinary LSTM. GRU gives most of the time better predictions so maybe one can improve this with the EWT modification. A different way of optimizing could be beneficial, i.e. one can try to optimize the structure of the neural network instead of the here optimized parameters. One could also use methods that consider better the opinion of the people, by looking, for example, at the sentiment of tweets. Another improvement can be done in the multivariate methods. The improvement is just finding better and more data that gives relevant information. Here there is only a focus on Bitcoin, but there are other cryptocurrencies that one could look at with the used methods. Maybe these other coins are more predictable.

In this thesis, one could see the discovery of phase transition-like behavior. This was for around 2 months around November 13th, 2018, which had the clearest scale-free behavior. After this finding, a model has been trained to predict the closing price. The error of these predictions showed a small correlation with the average of the different layers of the λs. In the cryptocurrency network, one could see the effect over the period of criticality(before and during the crash) in both the degree distribution and the motif analysis.

References

[1] Maher Ala'raj, Munir Majdalawieh, and Nishara Nizamuddin. "Modeling and forecasting of COVID-19 using a hybrid dynamic model based on SEIRD with ARIMA corrections". In: *INFECTIOUS DISEASE MODELLING* 6 (2021), pp. 98–111 (cit. on p. 14).

[2] Aytac Altan, Seckin Karasu, and Enrico Zio. "A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer". In: *APPLIED SOFT COMPUTING* 100 (Mar. 2021) (cit. on p. 2).

[3] A Altana, S Karasu, and S Bekiros. *Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques*. 2019 (cit. on p. 19).

[4] R. Anderson. *RECURRENT NEURAL NETWORKS*. 2019. URL: `https://technopremium.com/blog/rnn-talking-about-gated-recurrent-unit/` (visited on Apr. 27, 2022) (cit. on p. 17).

[5] K. E. ArunKumar, Dinesh V. Kalaga, Ch. Mohan Sai Kumar, et al. "Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-Regressive Integrated Moving Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average (SARIMA)". In: *APPLIED SOFT COMPUTING* 103 (May 2021) (cit. on p. 14).

[6] Ersin Aytac. "Forecasting Turkey's Hazelnut Export Quantities with Facebook's Prophet Algorithm and Box-Cox Transformation". In: *ADCAIJ-ADVANCES IN DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE JOURNAL* 10.1 (2021), pp. 33–47 (cit. on p. 15).

[7] Marco Bardoscia, Paolo Barucca, Stefano Battiston, et al. "The physics of financial networks". en. In: *Nat Rev Phys* 3.7 (July 2021), pp. 490–507 (cit. on p. 7).

[8] Vitalik Buterin et al. *Ethereum: A next-generation smart contract and decentralized application platform*. 2014 (cit. on p. 1).

[9] Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Antonio Sanna. "Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting". In: *EXPERT SYSTEMS WITH APPLICATIONS* 164 (Feb. 2021) (cit. on p. 2).

[10] B Castaing, Y Gagne, and E J Hopfinger. "Velocity probability density functions of high Reynolds number turbulence". en. In: *Physica D* 46.2 (Nov. 1990), pp. 177–200 (cit. on p. 11).

[11] F Chudzyński and Z Struzik. "Criticality of Bitcoin Market". In: *Acta Phys. Pol. A.* 139.4 (Apr. 2021), pp. 447–450 (cit. on p. 5).

[12] *CoinMarketCap*. 2022. URL: https://coinmarketcap.com/ (visited on Apr. 27, 2022) (cit. on pp. 1, 3).

[13] Ammar H. Elsheikh, Vikrant P. Katekar, Otto L. Muskens, et al. “Utilization of LSTM neural network for water production forecasting of a stepped solar still with a corrugated absorber plate”. In: *PROCESS SAFETY AND ENVIRONMENTAL PROTECTION* 148 (Apr. 2021), pp. 273–282 (cit. on p. 15).

[14] Dongyan Fan, Hai Sun, Jun Yao, et al. “Well production forecasting based on ARIMA-LSTM model considering manual operations”. In: *ENERGY* 220 (Apr. 2021) (cit. on p. 14).

[15] Turusha Ghimire, Atharva Joshi, Samgeeth Sen, et al. “Blockchain in additive manufacturing processes: Recent trends & its future possibilities”. en. In: *Mater. Today* 50 (2022), pp. 2170–2180 (cit. on pp. 1, 40).

[16] Liang Guo, Weiguo Fang, Qiuhong Zhao, and Xu Wang. “The hybrid PROPHET-SVR approach for forecasting product time series demand with seasonality”. In: *COMPUTERS & INDUSTRIAL ENGINEERING* 161 (Nov. 2021) (cit. on p. 15).

[17] Kin-Hon Ho, Wai-Han Chiu, and Chin Li. “A network analysis of the cryptocurrency market”. In: (Dec. 2020) (cit. on p. 7).

[18] Guoyan Huang, Xinyi Li, Bing Zhang, and Jiadong Ren. “PM2.5 concentration forecasting at surface monitoring sites using GRU neural network based on empirical mode decomposition”. In: *SCIENCE OF THE TOTAL ENVIRONMENT* 768 (May 2021) (cit. on p. 17).

[19] R J Hyndman and G Athanasopoulos. *Forecasting: principles and practice*. OTexts; Melbourne, Australia, 2021 (cit. on p. 14).

[20] C. K. Jha and M. H. Kolekar. “Empirical Mode Decomposition and Wavelet Transform Based ECG Data Compression Scheme”. In: *IRBM* 42.1 (Feb. 2021) (cit. on p. 18).

[21] B K Jhu and S Pande. *Time Series Forecasting Model for Supermarket Sales using FB-Prophet textitProceedings of the Fifth International Conference on Computing Methodologies and Communication, IEEE Xplore Part Number: CFP21K25-ART*. 2021 (cit. on pp. 2, 6).

[22] Seungmin Jung, Jihoon Moon, Sungwoo Park, and Eenjun Hwang. “An Attention-Based Multilayer GRU Model for Multistep-Ahead Short-Term Load Forecasting (dagger)”. In: *SENSORS* 21.5 (Mar. 2021) (cit. on p. 17).

[23] Jochen Jungeilges, Elena Maklakova, and Tatyana Perevalova. “Asset price dynamics in a “bull and bear market””. In: *STRUCTURAL CHANGE AND ECONOMIC DYNAMICS* 56 (Mar. 2021), pp. 117–128 (cit. on p. 2).

[24] A Khaled and M Althelaya. *El-Alfy, M.Salahadin (n.d.) Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU)*. Dhahran 31261, Kingdom of Saudi Arabia (cit. on p. 6).

[25] Ken Kiyono, Zbigniew R Struzik, and Yoshiharu Yamamoto. “Criticality and phase transition in stock-price fluctuations”. en. In: *Phys. Rev. Lett.* 96.6 (Feb. 2006), p. 068701 (cit. on pp. 1, 5).

[26]Le, Ho, Lee, and Jung. “Application of Long Short-Term Memory (LSTM) neural network for flood forecasting”. en. In: *Water (Basel)* 11.7 (July 2019), p. 1387 (cit. on p. 16).

[27]Yuanjiang Li, Yi Yang, Kai Zhu, and Jinglin Zhang. “Clothing Sale Forecasting by a Composite GRU-Prophet Model With an Attention Mechanism”. In: *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS* 17.12 (Dec. 2021), pp. 8335–8344 (cit. on p. 15).

[28]Yuanjiang Li, Yi Yang, Kai Zhu, and Jinglin Zhang. “Clothing Sale Forecasting by a Composite GRU-Prophet Model With an Attention Mechanism”. In: *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS* 17.12 (Dec. 2021), pp. 8335–8344 (cit. on p. 17).

[29]Wei Liu, Siyuan Cao, and Yangkang Chen. “Seismic time–frequency analysis via empirical wavelet transform”. In: *IEEE Geosci. Remote Sens. Lett.* 13.1 (Jan. 2016), pp. 28–32 (cit. on pp. 3, 18).

[30]Xiaolei Liu, Zi Lin, and Ziming Feng. “Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM”. In: *ENERGY* 227 (July 2021) (cit. on p. 14).

[31]Xiaolei Liu, Zi Lin, and Ziming Feng. “Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM”. In: *ENERGY* 227 (July 2021) (cit. on p. 15).

[32]Xiaolei Liu, Zi Lin, and Ziming Feng. “Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM”. In: *ENERGY* 227 (July 2021) (cit. on p. 17).

[33]Ioannis E. Livieris, Niki Kiriakidou, Stavros Stavroyiannis, and Panagiotis Pintelas. “An Advanced CNN-LSTM Model for Cryptocurrency Forecasting”. In: *ELECTRONICS* 10.3 (Feb. 2021) (cit. on p. 2).

[34]D D F Maesa and P Mori. “Blockchain 3.0 applications survey”. In: *JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING* 138 (2020), pp. 99–114 (cit. on pp. 1, 40).

[35]Lorenzo Menculini, Andrea Marini, Massimiliano Proietti, et al. “Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices”. In: *FORECASTING* 3.3 (Sept. 2021), pp. 644–662 (cit. on p. 15).

[36]Prapanna Mondal, Labani Shit, and Saptarsi Goswami. “Study of effectiveness of time series modeling (Arima) in forecasting stock prices”. In: *Int. J. Comput. Sci. Eng. Appl.* 4.2 (Apr. 2014), pp. 13–29 (cit. on p. 6).

[37]Satoshi Nakamoto. “Bitcoin: A peer-to-peer electronic cash system”. In: *Decentralized Business Review* (2008), p. 21260 (cit. on p. 1).

[38]M E J Newman. “Mixing patterns in networks”. en. In: *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 67.2 Pt 2 (Feb. 2003), p. 026126 (cit. on p. 21).

[39]H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad. “Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management”. In: *INTERNATIONAL JOURNAL OF INFORMATION MANAGEMENT* 57 (Apr. 2021) (cit. on p. 15).

[40] Konstantinos Nikolopoulos, Sushil Punia, Andreas Schafers, Christos Tsinopoulos, and Chrysovalantis Vasilakis. “Forecasting and planning during a pandemic: COVID-19 growth rates, supply chain disruptions, and governmental decisions”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 290.1 (Apr. 2021), pp. 99–115 (cit. on p. 2).

[41] Stephanos Papadamou, Nikolaos A. Kyriazis, Panayiotis Tzeremes, and Shaen Corbet. “Herding behaviour and price convergence clubs in cryptocurrencies during bull and bear markets”. In: *JOURNAL OF BEHAVIORAL AND EXPERIMENTAL FINANCE* 30 (June 2021) (cit. on p. 2).

[42] Stephanos Papadamou, Nikolaos A. Kyriazis, and Panayiotis G. Tzeremes. “Non-linear causal linkages of EPU and gold with major cryptocurrencies during bull and bear markets”. In: *NORTH AMERICAN JOURNAL OF ECONOMICS AND FINANCE* 56 (Apr. 2021) (cit. on p. 2).

[43] Cameron Roach, Rob Hyndman, and Souhaib Ben Taieb. “Non-linear mixed-effects models for time series forecasting of smart meter demand”. In: *JOURNAL OF FORECASTING* 40.6 (Sept. 2021), pp. 1118–1130 (cit. on p. 2).

[44] Michael Rogerson and Glenn C Parry. “Blockchain: case studies in food supply chain visibility”. en. In: *Supply Chain Manage.: Int. J.* 25.5 (June 2020), pp. 601–614 (cit. on pp. 1, 40).

[45] Andrey Rzhetsky, Jacob Foster, Ian Foster, and James Evans. “Choosing Experiments to Accelerate Collective Discovery”. In: *Proceedings of the National Academy of Sciences of the United States of America* (in press) (Nov. 2015) (cit. on p. 43).

[46] Christophorus Beneditto Aditya Satrio, William Darmawan, Bellatasya Unrica Nadia, and Novita Hanafiah. “Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET”. In: *5TH INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND COMPUTATIONAL INTELLIGENCE 2020*. Ed. by W Budiharto, A Kurniawan, D Suhartono, et al. Vol. 179. Procedia Computer Science. 5th International Conference on Computer Science and Computational Intelligence (ICCSCI), ELECTR NETWORK, NOV 19-20, 2020. 2021, pp. 524–532 (cit. on p. 15).

[47] Christophe Schinckus, Dang Pham Thien Duy, and Nguyen Phuc Canh. “Interdependences between cryptocurrencies: A network analysis from 2013 to 2018”. en. In: *J. interdiscip. econ.* 33.2 (July 2021), pp. 190–199 (cit. on p. 19).

[48] Farah Shahid, Aneela Zameer, and Muhammad Muneeb. “A novel genetic LSTM model for wind power forecast”. In: *ENERGY* 223 (May 2021) (cit. on p. 15).

[49] J Sola and J Sevilla. “Importance of input data normalization for the application of neural networks to complex industrial problems”. In: *IEEE Trans. Nucl. Sci.* 44.3 (June 1997), pp. 1464–1468 (cit. on p. 10).

[50] Tiziano Squartini, Iman van Lelyveld, and Diego Garlaschelli. “Early-warning signals of topological collapse in interbank networks”. en. In: *Sci. Rep.* 3.1 (Nov. 2013), p. 3357 (cit. on p. 7).

[51] Yuan Tao, Chao Liu, Chunyang Liu, Xingwang Zhao, and Haojie Hu. “Empirical Wavelet Transform Method for GNSS Coordinate Series Denoising”. In: *JOURNAL OF GEOVISUALIZATION AND SPATIAL ANALYSIS* 5.1 (June 2021) (cit. on p. 18).

[52] Chi K Tse, Jing Liu, and Francis C M Lau. “A network perspective of the stock market”. en. In: *J. Empir. Finance* 17.4 (Sept. 2010), pp. 659–667 (cit. on p. 43).

[53] Arpit Verma. *Bitcoin Data*. 2022. URL: `https://www.kaggle.com/varpit94/bitcoin-data-updated-till-26jun2021` (visited on Apr. 27, 2022) (cit. on pp. 9, 10, 18).

[54] Jun Wang, Junxing Cao, Shan Yuan, and Ming Cheng. “Short-term forecasting of natural gas prices by using a novel hybrid method based on a combination of the CEEMDAN-SE-and the PSO-ALS-optimized GRU network”. In: *ENERGY* 233 (Oct. 2021) (cit. on p. 17).

[55] C Weiling, Y Zhang, C K Yeo, C T Lau, and B L Lee. *Stock Market Prediction Using Neural Networks through News on Online Social Networks* (cit. on p. 6).

[56] Sebastian Wernicke and Florian Rasche. “FANMOD: a tool for fast network motif detection”. en. In: *Bioinformatics* 22.9 (May 2006), pp. 1152–1153 (cit. on p. 44).

[57] Rui Yan, Jiaqiang Liao, Jie Yang, et al. “Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering”. In: *EXPERT SYSTEMS WITH APPLICATIONS* 169 (May 2021) (cit. on p. 15).

[58] Hanyu Yang, Xutao Li, Wenhao Qiang, et al. “A network traffic forecasting method based on SA optimized ARIMA-BP neural network”. In: *COMPUTER NETWORKS* 193 (July 2021) (cit. on p. 14).

[59] Hanyuan Zhang, Haiyan Song, Long Wen, and Chang Liu. “Forecasting tourism recovery amid COVID-19”. In: *ANNALS OF TOURISM RESEARCH* 87 (Mar. 2021) (cit. on p. 2).

[60] Bobai Zhao, Qinglong Li, Qian Lv, and Xiameng Si. “A Spectrum Adaptive Segmentation Empirical Wavelet Transform for Noisy and Nonstationary Signal Processing”. In: *IEEE ACCESS* 9 (2021), pp. 106375–106386 (cit. on p. 18).

[61] Jian Zhou, Yingui Qiu, Shuangli Zhu, et al. “Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate”. In: *ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE* 97 (Jan. 2021) (cit. on p. 2).

[62] Zielak. *Bitcoin Historical Data*. 2021. URL: `https://www.kaggle.com/mczielinski/bitcoin-historical-data` (visited on Apr. 27, 2022) (cit. on p. 9).