Zurich University
of Applied Sciences

School of
Engineering

IDP Institute of Data Analysis
and Process Design

Bachelor thesis/Project work (Study programme, max. 2 lines)

Title (max. 4 lines)

Author	Ken Geeler Pascal Simon Bühler Philipp Rieser
Main supervisor	Marc Wildi
Industrial partner	Mobiliar
Date	11.06.2021

Please fill in the title sheet taking into account the following points:

→ Please do not change the font type or font size. Text should only be written over!
→ Please use only 4 lines max. per table row!
- Template: did you choose the right institute/centre? → Logo institute/centre
- Title: add your study programme directly after the word ‚Bachelor thesis / Project work' (max. 2 lines).
- Title: overwrite the running text with your Bachelor thesis title / Project work title (max. 4 lines).
- Author: fill in your first and family name (list alphabetical > family name).
- Supervisor: fill in your supervisor/s (list alphabetical > family name).
- Sup supervisor: if you do not have a sup supervisor → please delete this table row.
- Industrial partner: if you do not have an industrial partner → please delete this table row.
- External supervisor: if you do not have an external supervisor → please delete this table row.
- Date: please fill in current date.
- Finish: at the end please delete this description (grey) and safe the document in pdf format.

Zurich University
of Applied Sciences

School of
Engineering

DECLARATION OF ORIGINALITY
Bachelor's Thesis at the School of Engineering

By submitting this Bachelor's thesis, the undersigned student confirms that this thesis is his/her own work and was written without the help of a third party. (Group works: the performance of the other group members are not considered as third party).

The student declares that all sources in the text (including Internet pages) and appendices have been correctly disclosed. This means that there has been no plagiarism, i.e. no sections of the Bachelor thesis have been partially or wholly taken from other texts and represented as the student's own work or included without being correctly referenced.

Any misconduct will be dealt with according to paragraphs 39 and 40 of the General Academic Regulations for Bachelor's and Master's Degree courses at the Zurich University of Applied Sciences (Rahmenprüfungsordnung ZHAW (RPO)) and subject to the provisions for disciplinary action stipulated in the University regulations.

City, Date:	**Name Student:**
Winterthur, 11.06.2021	Geeler Ken
Winterthur, 11.06.2021	Pascal Simon Bühler
Winterthur, 11.06.2021	Philipp Rieser

Contents

Abstract

1. Introduction

Ken is testing working with Github.does it work now?

1.1. Intro

2. Theory

The following chapter is intended to provide the theoretical foundations necessary for our work. It is divided into a part that provides an overview of artificial neural networks. Followed by section 2.2. which shows the background and the ecosystem of Bitcoin. This knowledge should be kept in mind, which should help in understanding the price formation of Bitcoin.

2.1. Neural network

In the context of this work, artificial neural networks are used to answer supervised learning questions that focus on the classification of data. This means that a neural network finds a correlation between the data and their labels and optimizes its parameters to minimize the error for the next try. This process is called supervised training and is performed with a test data sample. An application example of classification is that a neural network is used for face recognition after it has learned the classification of different faces in the process of supervised training. Predictive analysis works similarly to the classification of labeled data. It estimates future values based on past events and can be trained with historical data. On the other hand, unsupervised learning (clustering) is applied to detect patterns from unlabeled data. Based on these patterns, for example, anomalies can be detected that are relevant in the fight against fraud (fraud detection). Unsupervised learning is not discussed further in this paper.

Section 2.1.1. will demonstrate the functioning of a neural network using a simple perceptron.

2.1.1. Perceptron

The construction of an artificial neural network is demonstrated using a perceptron. It is a simple algorithm for supervised learning of binary classification problems. This algorithm classifies patterns by performing a linear separation. Although this discovery was anticipated with great expectations in 1958, it became increasingly apparent that these binary classifiers are only applicable to linearly separable data inputs. This was only later addressed by the discovery of multiple layer perceptrons (MLP) [1].

Basically, a perceptron is a single-layer neural network and consists of the following five components and can also be observed in figure 1.

1. Inputs
2. Weights
3. Bias
4. Weighted sum
5. Activation function

Inputs are the information that is fed into the model. In the case of econometric time series, it is mostly the current and historical log returns (lags). These are multiplied by the weights and added together with the bias term to form the weighted sum. This weighted sum is finally passed on to the non-linear activation function, which determines the output of the perceptron.

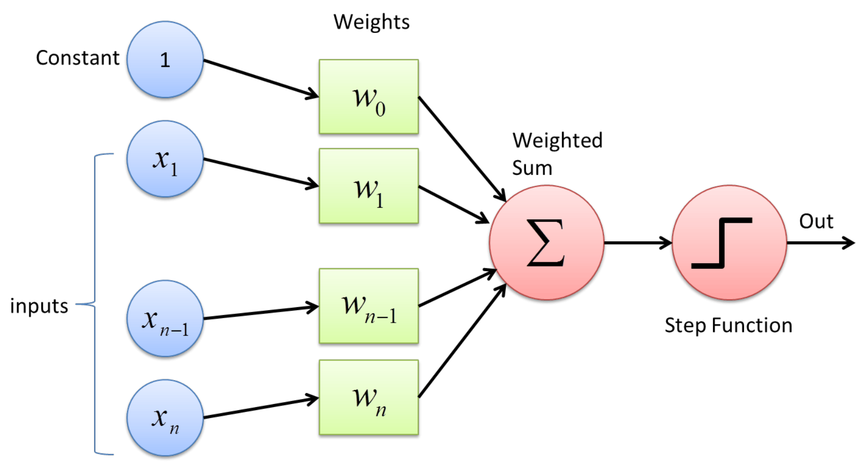


Figure 1: Schematic diagram of a perceptron.

The perceptron can also be represented as a function, which can be seen in equation 1. Analogous to the representation above, the inputs x_i are multiplied by the weights w_i in a linear combination. Then an error term is added so that the whole can be packed into the non-linear activation function $g(S)$. $\hat{y}$ is the binary output of this perceptron. With the aid of an activation function, binary output is obtained. The Heaviside step function shown in figure 1 is usually only used in single layer perceptrons, which recognize linear separable patterns. For the multi-layer neural networks presented later, step functions are not an option, because in the course of the backpropagation algorithm the gradient descent has to be minimized. This requires derivatives of the activation function, which in the case of this Heaviside step function equals 0. Because the foundation for the optimization process is missing, functions like the sigmoid function or the hyperbolic tangent function are used [2]. More about this topic is discussed in chapter 2.1.2.

$$\hat{y} = g(w_0 + \sum_{i=1}^{n} x_i w_i) \tag{1}$$

As just mentioned, the aim is to feed the perceptron with the training set and change the weights w_i with each cycle so that the prediction becomes more accurate. The output value is compared to the desired value. Finally, the sign of the difference $y - \hat{y}$ determines whether the inputs of that iteration are added to or subtracted from the weights. Ideally, the weights will gradually converge and provide us with a usable model [2].

2.1.2. Backpropagation algorithm

Finding the optimal weights of the neural network is achieved by finding the minimum of an error function. One of the most common methods for this is the backpropagation algorithm. This algorithm searches for the minimum of the error function by making use of a method called gradient descent. The gradient method is used in numerics to solve general optimization problems. In doing so, we progress (using the example of a minimization problem) from a starting point along a descent direction until no further numerical improvement is achieved. Since this method requires the computation of the gradient of the error function after each step, continuity and differentiability of this function must necessarily be given. The step function mentioned above in section 2.1.1. is therefore out of the question, but a non-linear function such as the logistic and the hyperbolic tangent functions (sigmoid) [3]. Both activation functions are visible in figure 2. While the target range of the 'ordinary' sigmoid function (equation 2) is between 0 and 1, the $\hat{y}$ of the hyperbolic tangent function (equation 3) ranges between -1 and 1. v_i equals the weighted sum including bias term.

$$\hat{y}(v_i) = (1 + e^{-v_i})^{-1} \tag{2}$$

$$\hat{y}(v_i) = \tanh(v_i) \tag{3}$$

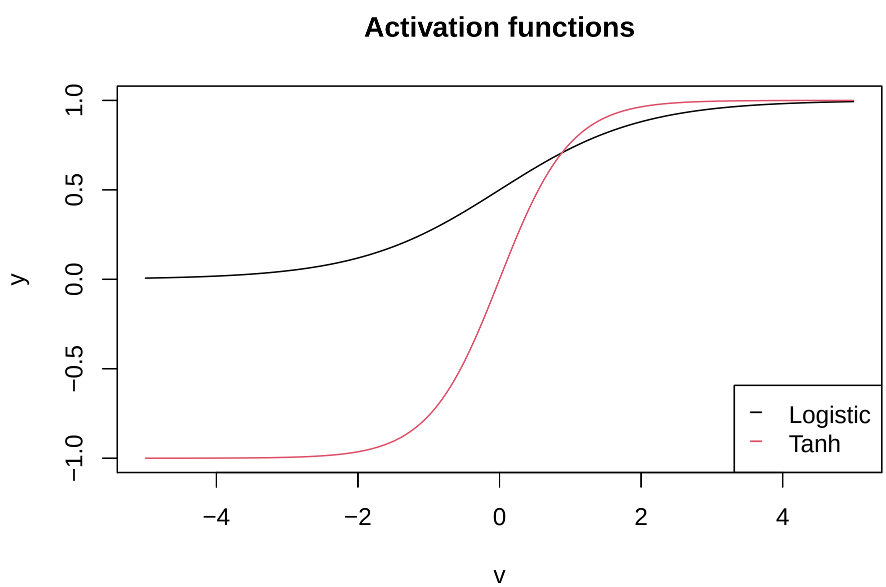


Figure 2: Two common sigmoid activation functions: logistic functions and hyperbolic tangent.

In the course of the error analysis, the output of the neural network respectively the result from the activation function in the output layer is compared with the desired value. The most commonly used error function E is the Mean Squared Error (MSE), which is seen in equation 4. y_i represents the actual value for the data point i, while $\hat{y}_i$ is the predicted value for data point i. The average of this error function is the average MSE, which is determined for a corresponding model. The learning problem is to adjust the weights w_i within the training sample so that $MSE(w)$ is minimized [4].

$$\begin{aligned} E = MSE &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - g(w_0 + x_i w_i))^2 \end{aligned} \tag{4}$$

As mentioned, this is searched for by the gradient descent method. The gradient of a function is a vector whose entries are the first partial derivatives of the function. The first entry is the partial derivative after the first variable, the second entry is the partial derivative after the second variable and so on. Each entry indicates the slope of the function in the direction of the variable to which it was derived. In this work, the notation ∇E is used when talking about the gradient for the error function E, which is displayed in equation 5 [3].

$$\nabla E = (\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_i}) \tag{5}$$

The weights get adjusted according to the following algorithm 6 where Δw_i is the change of the weight w_i and γ represents a freely definable parameter. In literature, this parameter is often called a learning constant [5]. The negative value is used because the gradient naturally points in the direction with the largest increase of the error function. To minimize the MSE, the elements in the gradient ∇E must be multiplied by -1.

$$\begin{gathered}\Delta w_i = -\gamma \frac{\partial E}{\partial w_i}, \\ \text{for } i = 1, 2, \ldots, n\end{gathered} \tag{6}$$

2.1.3. Multilayer perceptron

Multilayer perceptrons (MLP) are widely used feedforward neural network models and make usage of the backpropagation algorithm. They are an evolution of the original perceptron proposed by Rosenblatt in 1958 [1]. The distinction is that they have at least one hidden layer between input and output layer, which means that an MLP has more neurons whose weights must be optimized. Consequently, this requires more computing power, but more complex classification problems can be handled [6]. Figure 3 shows the structure of an MLP with n hidden layers. Compared to the perceptron, it can be seen that this neural network consists of an input layer, one or more hidden layers, and an output layer. In each layer, there is a different number of neurons, respectively nodes. These properties (number of layers and nodes) can be summarized with the term 'network architecture' and will be dealt with in this thesis.

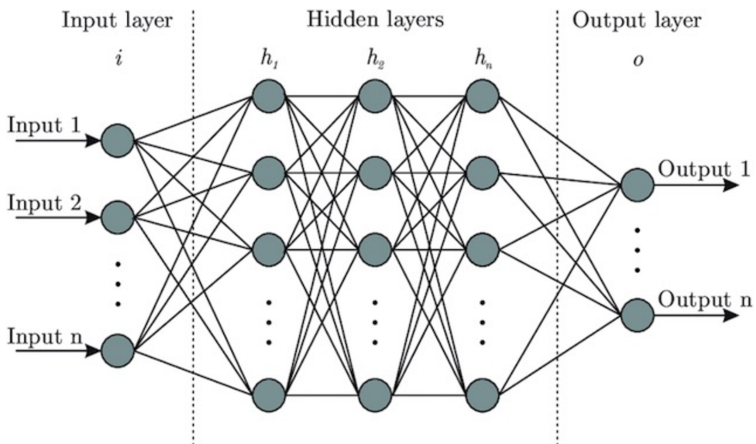


Figure 3: Schematic diagram of a multilayer perceptron

Every neural network has an input layer, which consists of one or more nodes. This number is determined from the training data and tells us how many features should be delivered to the neural network. In the case of bitcoin prices, we could use today's price and the prices of the last 10 days (lags 1-10), so the input layer would consist of 11 nodes. Some configurations also require a bias term to adjust the output along with the weighted sum, which is also added to the input layer. In contrast to the scheme of the MLP, this setup can be seen in figure 1 where the bias term is defined as 'constant.' Similarly to the input layer, each neural network has exactly one output layer. This can consist of one or more nodes. In this thesis, MLP is used as a regressor and therefore only one neuron is needed in this layer.

In between are the hidden layers, whose number and size can be configured as desired. The challenge is to find an optimal and efficient configuration without causing overfitting of the training data. The number of hidden layers depends primarily on the application area of the neural network. For example, working with image recognition would require more layers since the image file is broken down into individual pixels. Subsequently, the layers are used to optimize from rough outlines to the smallest detail. In our research, we came across several methods or 'rules of thumb' to optimize the model. A frequently suggested method is explained by Andrej Karpathy (director of the AI department of Tesla, Inc.). His GitHub entry recommends the approach of starting with a model that is too large that causes overfitting. Subsequently, the model is reduced by focusing on increasing training loss and improving validation loss [7].

2.1.4. Recurrent neural networks (RNN)

2.1.5. Long-short term memory (LSTM)

2.1.6. Challenges

- Early stopping required to avoid overfitting to in-sample data

- Gradient vanishing problem

Sigmoid suffers from the problem of Vanishing Gradient. The gradients of the NN's output with respect to the parameters become so small, that the NN takes smaller steps towards the minima of the loss function and eventually stop learning.

- Dying Relu Problem (?)

2.1.7. Model comparison

- Plot of performance
- Sharpe Ratio
- Diebold Mariano

2.2. Bitcoin

In this section bitcoin as a crypto-curreny is introduced. The historical data is analyzed and commented. Further the technology in and around crypto-currencies is briefly explained. A detailed explanation would require a paper itself, therefore the explanation is done as simple as possible.

In the following work bitcoin as a cryptocurreny is mentionend in its short term BTC, by the meaning of US Dollars per Bitcoin.

2.2.1. Historical analysis

The story of bitcoin began with a paper published by the name of Satoshi Nakamoto [8]. The publisher of the document cannot be assigned to a real person, therefore the technology inventor remains mysteriosly unknown until today. In 2009 the first bitcoin transaction was executed. On account of the opensource technology of bitcoin, lots of alternative currencies were created. Until 2013 the cryptocurrencies operated under the radar of most regulatory institutions. Because of the anonymity of the transactions, criminals were attracted by the newborn payment method. Headlines, such as the seizure of 26,000 bitcoins by closing the "Dark-Web" Website Silkroad through the Drug Enforcement Agency, followed moreoften in the newspapers. Nevertheless in 2014 more companys,such as :Zynga, D LasVegas Casinos, Golden Gate Hotel & Casino, TigerDirect, Overstock.com, Newegg, Dell, and even Microsoft [9]. began to accept bitcoin as a payment method. In 2014 the first derivative with bitcoin as an underlying was approved by the U.S.Commodity Futures Trading Commission. 2015 an estimated 160000 merchants used bitcoin to trade . It is observed that the value of bitcoin is very volatile, we will discuss this in a FURTHER XYXY section. Let us first look a the price in 5 and the log(price)4 and get a sense of the chart. Note: The data in the charts start in 2014 where it was listed in coinmarket, events between 2009 and 2014 are described without visualization.

Around 2010 bitcoin had the first increase in price as it jumped a 100% from 0.0008 USD to 0.08 Dollar [10]. In 2011 the price rose from 1 USD to 32 USD within 3 months and recessd shortly after to 2 USD this can be referred as a first price bubble in bitcoin, for the next year the pri climbed to 13 Dollars and reached a never seen level of 220 USD, only to plunge to 70 USD within a half month in April 2013. By the end of the year a rally brought btc up to a peak of 1156 USD. The following year brought bad news and the price slowly decreased to 315 USD in 2015 after an observed drop of 20% after news from the trial of Ross Ulbricht, founder of Silk road marked in Letter **A**.From this point in time, things began to change, more volume was flushed in the market and the price of BTC began to ascend and the real rally began ,the BTC rose up to 20k USD / BTC on 17th September 2017 **B**. After the rise comes the fall and BTC lost value for more than a year until **C** 2018-12-15 the trend reverted and found its peak after 6 months in **D** 2019-06-26, but oncemore it was not lasting for long as bitcoin lost **D** 2020-03-12 nearly half its value in 4 days. But the story wasn't over by now after the drop, the price of the cryptocurrency regained value, passed previous levels and shortly after exploded, after companies like tesla and signal bought a big chunk of bitcoins, into a maximum of 58000 USD per bitcoin.

Figure 4: Schematic diagram of a perceptron.

Figure 5: Schematic diagram of a perceptron.

2.3.2. SHA256 Hash

- Block
- Blockchain
- Distributed Blockchain
- Token
- Coinbase Transaction
- Public/Private Key -> Signing
- Signature (sign, verify)
- Transaction

3. Methodology

The focus of this thesis is to predict historical prices of bitcoin using the models listed in Chapter XX. The predictive accuracy of these obtained predictions are compared using loss functions (Annualized Sharpe, Diebold Mariano Test, MAE, MSE, RMSE, Mincer-Zarnowitz Regressions). Then, based on the best models with the most accurate predictions, trading strategies are worked out to compare with a buy-and-hold strategy. Finally, we would like to venture into the topic of explainability and attempt to explain why the chosen models lead to these outcomes. The procedure of this quantitative study is described in this chapter.

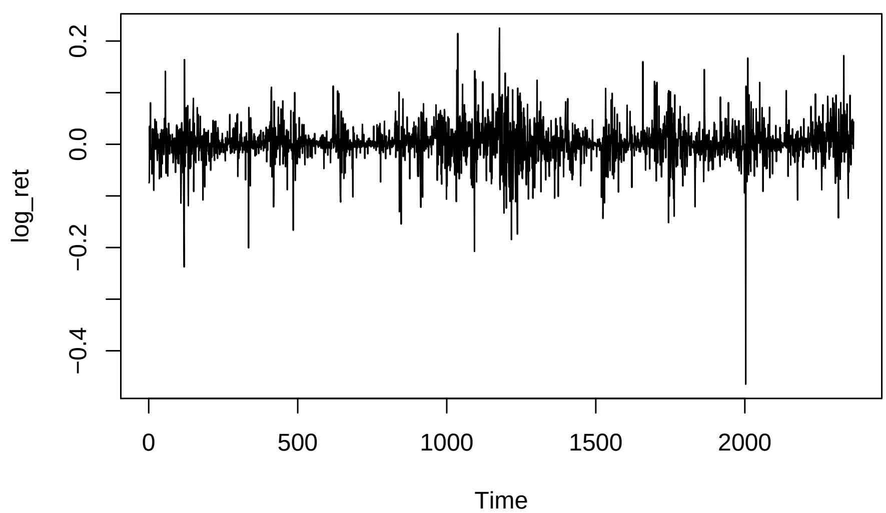
- Data and Analysis of Bitcoin (BTC/USD)
- Defining the train and test samples (including description about calm and volatile phases).
- Calculate predictions with the defined models (AR, NN, RNN, LSTM, Emponential Smoothing + NN (Slavek Smyl)).
- Compare Predictions / performance with Realized Data (Annualized Sharpe, Diebold Mariano Test, MAE, MSE, RMSE, Mincer-Zarnowitz Regressions)
- Explain trading strategies
- Explainability for the best models
- (Backup: Which models work well in which market phases?)

3.1. Data and analysis of Bitcoin

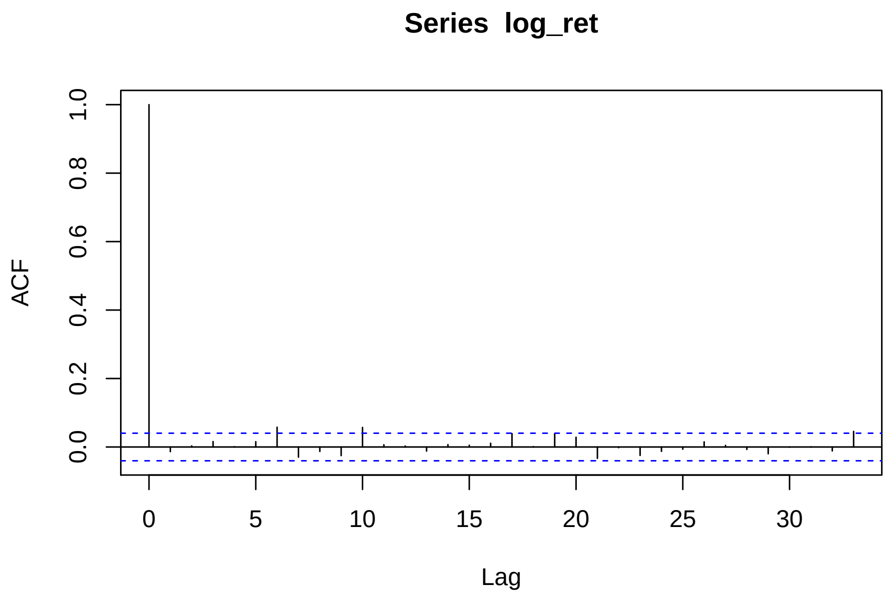
The data in this paper is accessed via yahoofinance provided by coinmarket https://coinmarketcap.com/. We use the daily "closing price" of bitcoin in US Dollars with the ticker BTC-USD. Cryptoassets are tradeble 24 hours a day 256 days a year, there is no real "closing price" for the bitcoin, therefore the "closing-Price" is just the last price of the day evaluated at last timestamp with timeformat UTC.

In chapter 2.3. the bitcoin price is visualized. For processing and analyzing the data in order to fullfill the weak stationarity assumptions we transform the data into logreturns

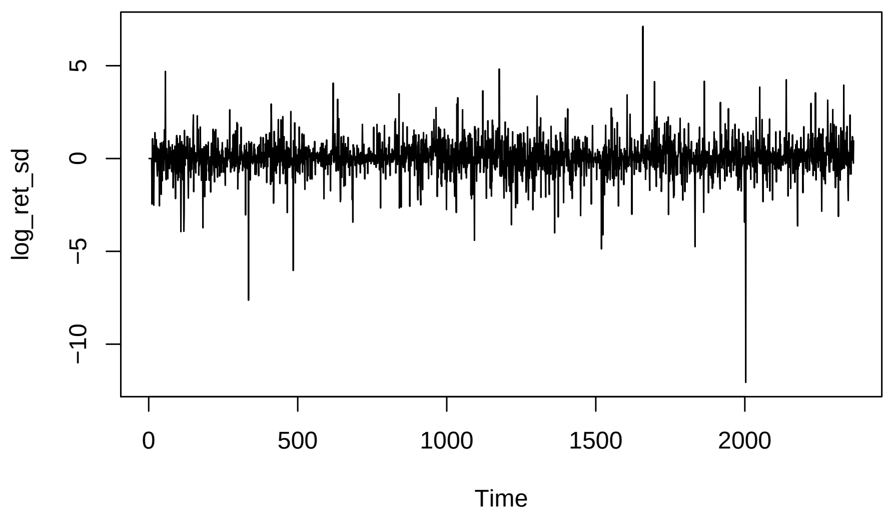
$$\text{LogReturn} = \log(x_t) - \log(x_{t-1})$$

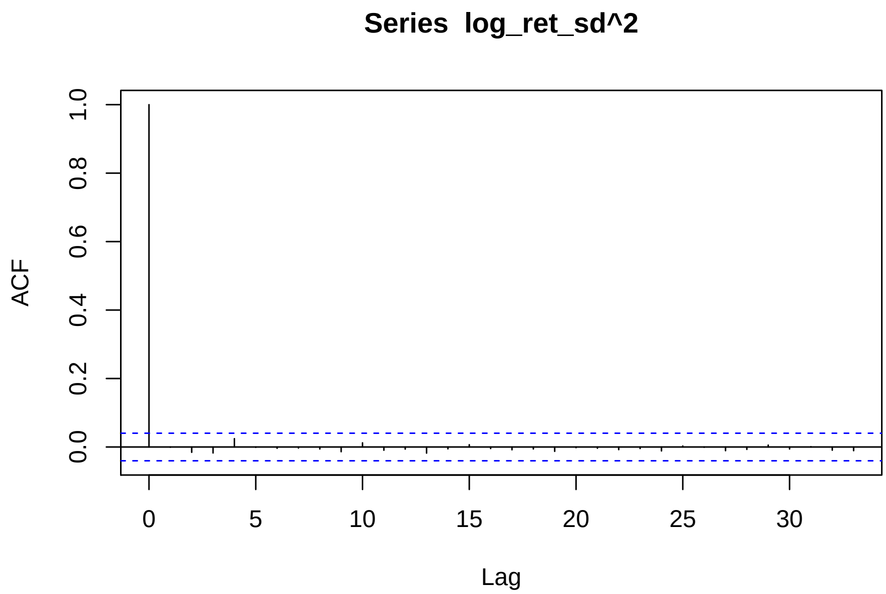


By computing the autocorrelation oft the log_returns, there is still dependence visible in lag 6 and 10. This indicates dependency in volatility-cluster, to cancel out the effect an ARMA-GARCH model is fitted to the data and the residuals are standardized by the model standard-deviation.



To check the dependencies the standardized





3.2. Defining train and test samples

- Describe different phases
- Explain why we set train and test sample like this
- Describe stable and volatile phases and why we should keep that in mind for predictions

3.3. Forecasting

- Autoregressive process (AR)
- Deep learning neural network / multi layer percepton
- Recurrent neural network (RNN)
- Long short-term memory (LSTM)

3.3.1. In-sample

- Compare Predictions with Realized Data (Annualized Sharpe, Diebold Mariano Test, MAE, MSE, RMSE, Mincer-Zarnowitz Regressions)

3.3.2. Out-of-sample

- Compare Predictions with Realized Data (Annualized Sharpe, Diebold Mariano Test, MAE, MSE, RMSE, Mincer-Zarnowitz Regressions)

3.4. Trading strategies

- Define trading strategies
- Sign-trading (daily)
- Vola-gewichtet trading
- Define realistic fee structure for trading (Coinbase Pro, Binance, Kraken etc.)

3.4.1. Other cryptocurrency

- Test our best model with another time series

3.5. Explainability

- Performing the predictions with the two (?) best models
- Include variations to find possible starting points for explainability (number of nodes, layers)

3.6. (Relationship between accuracy and market phase)

- Test

4. Results

4.1. Results chapterino

5. Conclusion

Best Trading Algorithm ever!

5.1. Get rich or die tryin

Neque volutpat ac tincidunt vitae semper quis. At elementum eu facilisis sed odio morbi quis commodo odio. Eget dolor

5.2. Be GME stock, or not to be GME stock

Tellus at urna condimentum mattis pellentesque id nibh. Morbi tempus iaculis urna id volutpat lacus laoreet. Sem fringilla

References

[1] F. Rosenblatt, *The perceptron: A probabilistic model for information storage and organization in the brain.* Psychological Review, 1958, pp. 386–408.

[2] P. L. B. Martin Anthony, *Neural network learning: Theoretical foundations.* Cambridge University Press, 1999.

[3] R. Rojas, *The backpropagation algorithm.* Springer Berlin Heidelberg, 1996, pp. 149–182.

[4] G. B. O. Yann Lecun Leon Bottou, *Efficient BackProp.* Image Processing Research Department AT& T Labs, 1998, pp. 1–44.

[5] L. Hunsberger, "Back propagation algorithm with proofs." https://www.cs.vassar.edu/~hunsberg/cs365/handouts-etc/backprop.pdf (accessed Mar. 21, 2021).

[6] M. A. J. I. Hassan Ramchoun Youssef Ghanou, *Multilayer perceptron: Architecture optimization and training.* International Journal of Interactive Multimedia; Artificial Intelligence, 2016, p. 26.

[7] A. Karpathy, "A recipe for training neural networks." https://karpathy.github.io/2019/04/25/recipe/ (accessed Mar. 24, 2021).

[8] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system.* online: www.bitcoin.org, 2008, p. 9.

[9] U. W. Chohan, *A history of bitcoin.* University of New South Wales, Canberra, 2017.

[10] J. Edwards, "Bitcoins price history." https://www.investopedia.com/articles/forex/121815/bitcoins-price-history.asp (accessed Mar. 01, 2021).

Attachment

This project work is created with R-4.0.2 , RStudio Version 1.4.904 and RMarkdown in collaborative working via Git / Github