A Work Project, presented as part of the requirements for the Award of a Master's degree in Finance / Management from the Nova School of Business and Economics.

ALGORITHMIC TRADING WITH CRYPTOCURRENCIES

Jannik Nolden, Tim Schnülle, Tobias Röhnelt

Work project carried out under the supervision of:

Leid Zejnilovic

17-12-2021

Abstract

Since its inception in 2009, Bitcoin has gained popularity and importance in financial markets. The Bitcoin price is highly volatile entailing high risk and chances of high returns for traders. We define a holistic approach to build an intraday Bitcoin trading algorithm based on predictive analysis of ML models. The results show that our trading algorithm generates positive returns and to outperform its benchmark strategies after considerations for feasibility and profitability.

Keywords

Forecasting, Business Analytics, Cryptocurrency, Bitcoin, Social Media influencer, Price Prediction, Algorithmic Trading, Granger causality, Vader, Twitter, Trading strategy, Machine Learning, Deep Learning, Time Series Forecasting

Acknowledgements

The team gratefully acknowledges the help from supervisor Leid Zejnilovic as main representative of the Nova Data Science Knowledge Center and Nova SBE faculty.

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

List of Abbreviations

ADA	Cardano
ADF	Augmented Dickey Fuller
ARIMA	Auto-Regressive Integrated Moving Average
AVAX	Avalanche
BNB	Binance Coin
BUSD	Binance USD
DL	Deep Learning
DOGE	Dogecoin
DOT	Polkadot
ETH	Ethereum
FI	Feature Importance
GRU	Gated Recurring Unit
LSTM	Long Short-Term Memory
LTC	Litecoin
LUNA	Luna Coin
MC	Multicollinearity
ML	Machine Learning
RF	Random Forest
RMSE	Root mean square error
RNN	Recurrent Neural Network
SMBO	Sequential model-based optimization
SOL	Solana
TPE	Tree-structured Parzen Estimator
UNI	Uniswap

USDC	USD Coin
VADER	Valence Aware Dictionary and Sentiment Reasoner
XGB	Extreme Gradient Boosted trees
XRP	Ripple

Table of Contents

1 Introduction

Since its inception in 2009, Bitcoin has gained popularity and importance in the international financial landscape, attracting media coverage, the attention of regulators, government institutions, investors, academia, and the public (Sebastião and Godinho 2021). Following Bitcoin, other cryptocurrencies were introduced over the past decade. As of 14th of October 2021, there are 6,590 different cryptocurrencies on the market, amounting to a market capitalization of around $2.4 trillion for the entire cryptocurrency market (CoinMarketCap 2021). The opportunities to own and trade cryptocurrencies have increased significantly in recent years. With the rise of online wallet companies, trading is made easier and accessible to the public, which is reflected in higher trading volume and an increase in the number of wallets (Blockchain.com 2021).

On 14th of October 2021, the 24-hour trading volume of the entire cryptocurrency market amounted to $92 billion (CoinMarketCap 2021). On that day, the trading volume of Bitcoin was $43 billion compared a trading volume of $10 billion for Apple and $10 billion for Tesla as commonly known stocks (Wall Street Journal 2021). Bitcoin is the most relevant cryptocurrency on the market with a market capitalization of around $1 trillion (October 14th), accounting for around 46% (followed by Ethereum accounting for around 19%) of the total market capitalization of the cryptocurrency market (CoinMarketCap 2021). In addition to its high market capitalization and trading volume, the cryptocurrency market is characterized by high price fluctuations, i.e., high volatility. High volatility results in high risk and return because volatility is considered as an alternative measure for risk and risk has a positive and significant relation to returns (Bali and Lin, 2006). Cryptocurrencies do not follow the development of major financial asset classes but are driven by behavioral factors like for example herding factors where traders follow other people instead of relying on their own analysis (Sebastião and Godinho 2021). Machine Learning (ML) algorithms discover patterns and drivers for the financial development of an asset, enabling to develop a model that predicts future price movements, and generates returns superior to its benchmark if executed in the market (Tao, et al. 2021). Prior research has been conducted to analyze the applicability of different ML algorithms to predict the development of cryptocurrencies. We identified 73 papers that discuss the prediction of cryptocurrency prices (cf. Table 1). 63 of these papers (86%) analyze data from 2019 and previous years. 64 papers (88%) translate the prediction problem into either a Regression (42 paper, 58%) or Classification (22 paper, 30%) analysis. 53 papers (73%) use either Statistical or ML algorithms but do not compare both. The Feature Selection is characterized by endogen cryptocurrency features (69 paper, 95%). 21 papers (29%) consider a trading strategy to evaluate the model.

Due to recent developments in the Bitcoin market, patterns of the Bitcoin movement have changed. Before 2019, the highest trading volume ($120 billion) per week was accorded in the first calendar week in 2018. After 2019, the week with the highest trading volume grew by 538% to a total of $765 billion in calendar week eight in 2021 (Finance 2021). The state of research is limited as 63 papers do not incorporate data from 2019 on and dismiss the current pattern in the development of Bitcoin. Algorithms need to be trained with recent data. Research on the implementation of a real-time trading algorithm for Bitcoin is not covered by any paper. Limited research has been conducted regarding a holistic approach for the development of a trading algorithm, including both Regression and Classification problems, comparing several algorithms, considering endogen (Supply & Demand) as well es exogen features (Crypto market, Macro Financial, Political and Sentiment) and including a trading strategy for final evaluation. A holistic approach for algorithmic trading brings scientific novelty and can discover new insights for data-driven trading. Based on the identified gaps in the current state of research this work seeks to answer three major questions: (Study I) Does Twitter Sentiment impact short-term price fluctuations in Bitcoin? (Study II) What is the optimal modelling design for Bitcoin price and trend prediction? (Study III) How to translate multiple model predictions into an algorithmic trading strategy?

The main objective of this work is to build a Bitcoin trading algorithm based on the predictive analysis of ML models. Past research is analyzed and aggregated to develop a holistic approach, from Data Collection, Feature Engineering, Feature Selection, Model Implementation and Model Selection to the definition of a trading strategy. Using a simulation setting for real-time trading during a test period, the final evaluation is conducted on economic performance measures of trading strategies that combine multiple model predictions. The results are compared to benchmark strategies. The findings indicate that a trading algorithm derived from ML model predictions is able to generate positive returns and to outperform its benchmark strategies. Ensemble trading strategies that combine predictions of multiple Long Short-Term memory (LSTM) Regression models have the highest overall performance.

This work is organized in 5 major sections. In section 2, we review the related work. In section 3 we outline the methodology of this work and describe Problem Definition, Data, Modelling, and Trading Strategy. The individual studies (Study I, Study II, Study III) mentioned above represent self-contained analyses and are included in this section. In section 4 we report and discuss the results. Section 5 concludes this work and gives an outlook for future research opportunities.

Table 1: Coverage of major topics in the defined 73 papers

	Details	Number
Observation period	till 2019	63
	2019 - now	10
	Total	73
Prediction	Regression	42
	Classification	22
	Both	9
	Total	73
Algorithm	Statistical	20
	ML	33
	Both	20
	Total	73
Features	Supply & Demand	69
	Crypto market	0
	Macro-financial	12
	Political	6
	Sentiment	11
	Total (Higher due to duplication)	98
Trade strategy	Yes	21
	No	52
	Total	73

2 Literature Review

Systematic search of the literature ensures qualitative scientific work (Timmins and Mccabe 2005). We applied a forward and backward search process to identify relevant papers and used filter criteria to ensure a state-of-the-art literature base, as shown in figure 1.

Figure 1: Literature research approach

We used the EBSCO library, a collection of scientific databases, for our initial research search. EBSCO is a leading provider of research databases and includes paper, e-journals, magazines, and e-books (Williams and Foster 2011). EBSCO displays the peer review status of papers to ensure academic scientific quality. During forward search, we used the keywords in table 2 to find papers focusing on Bitcoin and cryptocurrency prediction or volatility.

Table 2: Combinations of the keyword search query

Keyword 1	Keyword 2
Bitcoin	Prediction
Bitcoin	Forecasting
Bitcoin	Volatility
Cryptocurrencies	Prediction
Cryptocurrencies	Forecasting
Cryptocurrencies	Volatility

We gathered 61 papers during the first step. We selected papers that focus on prediction or forecasting in the second step to reduce the literature base to 23 papers. We applied backward search, scanning related work for additional relevant paper. The scope totaled to 73 papers. We filtered the papers by content for ML algorithms and only included paper that were published in 2019 or later to ensure state-of-the-art. The remaining five papers represent the focus papers of our work, shown in table 3.

The focus papers provide guidance for our work in presenting the latest research results as well as represent a benchmark to compare our work. All papers that are included in this work are peer-reviewed to ensure high academic quality.

Table 3: Overview of focus paper

Author (Year)	Prediction	Forecast	Trade strategy	Supply & Demand	Crypto market	Macro-financial	Political	Sentiment
Chen, Li and Sun (2020)	Classification	5 minutes & 1 day	N/A	x	N/A	x	N/A	x
Cocco, Tonelli and Marchesi (2021)	Regression	1 day	N/A	x	N/A	N/A	N/A	N/A
Dutta, Kumar and Basu (2020)	Regression	1 day	x	x	N/A	x	x	x
Mudassir, et al. (2020)	Both	1 day, 1 week, 1 month	N/A	x	N/A	N/A	N/A	N/A
Sebastião and Godinho (2021)	Both	1 day	x	x	x	x	N/A	N/A

Focus papers are categorized according to the ML problem they are analyzing into Regression, Classification, and a combination of both learning problems. Prediction in general is defined as estimating the output for unseen data. Forecasting is a part of prediction and is concerned with time-series data (Matsuo 2003). In this work, we use the general wording “prediction”. The focus papers leverage different features which can be aggregated into five feature categories: Supply & Demand, Crypto market, Macro Financial, Political and Sentiment. The features categories will be explained in depth in section 2.2.

Chen, Li and Sun (2020) use high dimensional features of Supply & Demand, Macro-financial and Sentiment on a five-minute interval basis to predict the Bitcoin price trend in five minutes and for the next day. They highlight the importance of sample granularity and feature dimensions on ML model performance (Chen, Li and Sun 2020). Cocco, Tonelli and Marchesi (2021) and Dutta, Kumar and Basu (2020) predict the daily closing Bitcoin price. Cocco, Tonelli and Marchesi (2021) compare several ML frameworks to predict the prices of Bitcoin and Ethereum. They use five technical indicators that are calculated from the cryptocurrency price and provide insights how to build efficient trading frameworks (Cocco, Tonelli and Marchesi 2021). Dutta, Kumar and Basu (2020) investigate Feature Engineering of twenty features from Supply & Demand, Crypto Market, Macro-financial, Political and Sentiment for ML algorithms. They implement a simple trading strategy and demonstrate the possibility of financial gain through algorithmic cryptocurrency trading (Dutta, Kumar and Basu 2020). Mudassir, et al. (2020) and Sebastião and Godinho (2021) consider a Regression and Classification problem, predicting price and price trend. Mudassir, et al. (2020) predict Bitcoin volatility on a daily, weekly, and monthly base. They use 700 features based on technical indicators and show that it is possible to predict the daily Bitcoin price with low error rates (Mudassir, et al. 2020). Sebastião and Godinho (2020) predict the daily price of Bitcoin, Ethereum and Litecoin and implement trading strategies. They consider Supply & Demand, Crypto Market and Macro Financial features and find that ML is a good technique to predict cryptocurrency prices and price trends, enabling profitable algorithmic trading of cryptocurrencies.

None of the identified papers combines Regression and Classification, the usage of all five feature categories (Supply & Demand, Crypto market, Macro Financial, Political and Sentiment) and the implementation of a trading strategy.

3 Methodology

3.1 Problem Statement

The goal of this work is to develop a trading algorithm that generates superior returns to its benchmarks with automated trading of the cryptocurrency Bitcoin. Financial time-series data is challenging to analyze due to the dynamic, non-linear, non-stationary, highly volatile, and chaotic nature of financial markets. ML algorithms can be used to analyze large amounts of seemingly chaotic data, to discover patterns in the data and to predict future data. Additionally, an automated trading bot can react much faster to developments in the market than any human (Borges and Neves 2020). To build a trading algorithm that is based on a ML, we must convert the problem of profitable trading into a ML problem defining an output that can be generated by a ML model. In this paper, we conduct price and trend prediction. Price prediction represents a Regression problem and trend prediction a Classification problem. All learning problems represent a Supervised Learning Problem because the target variable, i.e., the Bitcoin price or the trend calculated from the Bitcoin price, is given, and can be tested against.

To evaluate our trading algorithm, we create a simulation design that is aligned to the prerequisites a real-time trading algorithm requires. The architecture of our work is represented in figure 2. Data is collected through API connections. The features are categorized in five feature categories: Supply & Demand, Cryptocurrency Market, Political, Macro Financial and Sentiment. The collected data is pre-processed, and Feature Transformation, Technical Analysis and Sentiment Analysis are applied to enrich the data and build the final dataset for modelling purposes. We implement Regression and Classification algorithms. Finally, trading strategies are developed that translate model outputs into trading actions.

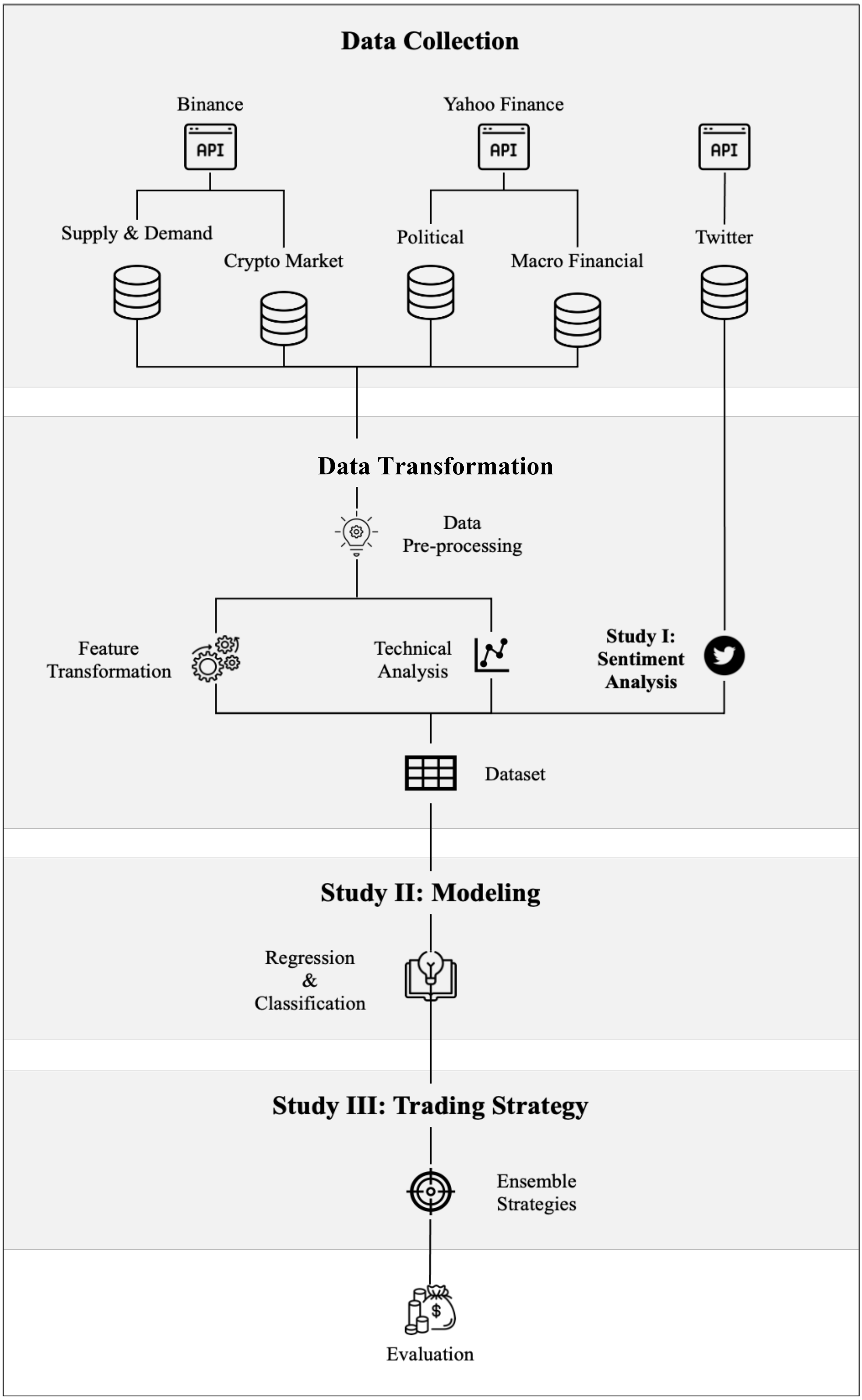


Figure 2: Architecture of the work project

We use Python 3.9 for this work. The training, validation, and evaluation of our ML models was executed through cloud computing provided by Genesis Cloud. We used two NVIDIA GPU GEDorce GTX 1080Ti cores. We used the computing power on demand for a two-week period. The main used Pytoch libraries are PyTorch 1.6 for DL algorithms, scikit-learn 1.0.1 for ML algorithms and Optuna 2.10.0 for Hyperparameter Tuning. The host operating system was Linux.

3.2 Data

3.2.1 Data Collection

A solid and valid data base is the prerequisite for any data analysis and the cornerstone of this project (Gupta, et al. 2021). Data Collection is a challenging process for algorithmic trading. Past data is needed to train the algorithm and real-time data must constantly be fed into the algorithm follow market fluctuations and adjust predictions accordingly. Data Collection requires time, restricting the selection of data sources.

We performed a detailed analysis of the Data Collection process introduced in our focus paper and identified two major characteristics. First, data is collected with different techniques and in different formats and second, features collected for the prediction of cryptocurrencies differ between papers. We observe three different techniques for the Data Collection process: Data retrieval from external files, e.g., CSV (Ahmed and Mafrachi, 2021), data retrieval using web scraping (Kim, et al. 2021), and Application Programming Interface (API) (Chen, Li and Sun 2020). Feature selection is performed differently in terms of categorization and number of features. Several studies examine for example the influence of S&P 500, Gold, other Cryptocurrencies and Sentiment on Bitcoin price fluctuations (Bouri, et al. 2017, Abraham, et al. 2018, Mallqui and Fernandes 2019). Factors influencing the Bitcoin price can be categorized in endogen and exogen features (Bouri, et al. 2017). We follow the approach of Sovbetov (2018) as the work provides the most comprehensive collection of factors influencing Bitcoin fluctuations. Sovbetov (2018) divides the features for cryptocurrency predictions into four categories: Supply & Demand, Cryptocurrency Market, Political and Macro Financial. In our work, we will extend the collection of Sovbetov (2018) by a fifth category: Sentiment. Considering the increasing importance of social media in recent years, prior research investigates a causal relationship between Online Sentiment and Bitcoin price fluctuations (Kraaijeveld and De Smedt 2020, Pano and Kashef 2020). Table 4 summarizes the five categories and gives examples for each. Endogen features are connected to supply and demand of cryptocurrencies. Exogen features are not directly connected to the observed cryptocurrency but measurements of other influencing factors.

Table 4: Overview of feature categories

Categories	Influence	Example Features
Supply & Demand	Endogen	Exponential Moving Average etc.
Cryptocurrency Market	Exogen	Ethereum, Solana, Cardano, Dogecoin etc.
Political	Exogen	CBOE Volatility
Macro Financial	Exogen	S&P 500, CAC40, DAX40, Nikkei 225 etc.
Sentiment	Exogen	Twitter

Within the focus papers (cf. table 3) Data Collection is mostly performed for daily data, only one paper analyzes intraday data. The access to past intraday data is limited compared to the access of daily data for most data sources. From the Yahoo Finance API, past intraday data can only be retrieved for a maximum period of 60 days (Aroussi 2021). There are data sources which can provide data for a longer period, which will imply further costs. Therefore, we decided to use the Yahoo Finance API. Four focus paper analyze data from 2019. No paper is based on data from 2021.

Table 5: Overview of Data Collection in the focus paper

Authors and Year	End Time	Frequency	Sources	Observations
Chen, Li and Sun (2020)	02.2019	5 minutes	1	50.000
Cocco Tonelli and Marchesi (2021)	04.2020	Daily	1	1.216
Dutta, Kumar and Basu (2020)	06.2019	Daily	9	3.469
Mudassir et al. (2020)	12.2019	Daily	N/A	N/A
Sebastiano and Godinho (2020)	03.2019	Daily	2	1.297

Binance provides past intraday data for cryptocurrencies nearly without limitations since the opening of the trading platform in 2017 (Binance 2021). The collection of Twitter data is different and depends on the arranged tweet limit. During our academic research the limit was set to 10.000.000 Tweets (Twitter, Developer Platform 2021a).

In our analysis, we exclusively use data sources that offer an API connection. APIs have an advantage over the other two Data Collection techniques because data is retrieved in a concise time. Downloading and reading CSV files or web scraping consumes more time. Uploading multiple files and scraping multiple websites increases the numbers of sources that need to be monitored, increasing the risk for changes in data formats which would interrupt the automated trading algorithm. Aiming to minimize the risk of unwanted changes in data formats, we use a minimum number of APIs that provide high data quality and ensure a data base that includes features from all five feature categories.

To determine the endogenous factors about the Bitcoin Price, the following features in the corresponding time interval were extracted. To examine the influence of other cryptocurrencies on the Bitcoin price, the 15 of the largest following Cryptocurrencies, based on market capitalization in October (coinmarketcap 2021) were included in the dataset. Along with the Bitcoin movement, 84,7% (coinmarketcap 2021) of the whole market capitalization is being tracked and analyzed.

As there is a relationship between Macro Financial Movement and the Price Development of Bitcoin (Walther 2019). The ten most important countries sorted by GDP were selected for further analysis, for each country the primary equity index and the currency for the respective country were extracted (Silver 2020). For countries with the same currency or a currency that is already included in the dataset the value has been skipped.

Furthermore, the most actively traded commodities according to Futures Industry Association were also added to the analysis (FIA 2021).

Table 6 gives an overview of the sum of the total number of features extracted. For example, if a feature is extracted like "Ethereum" it will be counted as one feature which will contain some more sub features like close price and trading volume. The Cryptocurrency Market (13 Features) and Macro-Financial (28 Features) category contain the most features. For Supply & Demand we extracted 12 features through technical analysis.

Table 6: Overview of collected features

Feature Category	Feature	Sub Features	Engineered Features
Supply and Demand	1	5	12
Cryptocurrency Market	13	78	0
Political	1	1	0
Macro-Financial	28	28	7
Sentiment	1	2	0
	Total	115	19

3.2.2 Exploratory Data Analysis

Figure 3 provides an overview of the observed data and gives a detailed picture of the development of the target variable. The data analyzed in this paper contains 8,929 observations, representing a time-period of 31 days from the 01.10.2021 00:05 to the 31.10.2021 23:55. The target variable is the closing price of Bitcoin measured in Tether (BTCUSDT), visualized in figure 3. Tether (USDT) is a cryptocurrency whose value is linked to the US dollar and called a stable coin. At the start of the observation period the price of Bitcoin is 43,981 USDT and reaches a price of 61,243 USDT at the last observation. For the analyzed period the average Bitcoin price was 57,653 USDT, while the median was 59,516 USDT. The price was subject to strong fluctuations and ranged from 43,361 USDT as a minimum to a highest price of 66,908 USDT within the period. The standard deviation was 5,285.

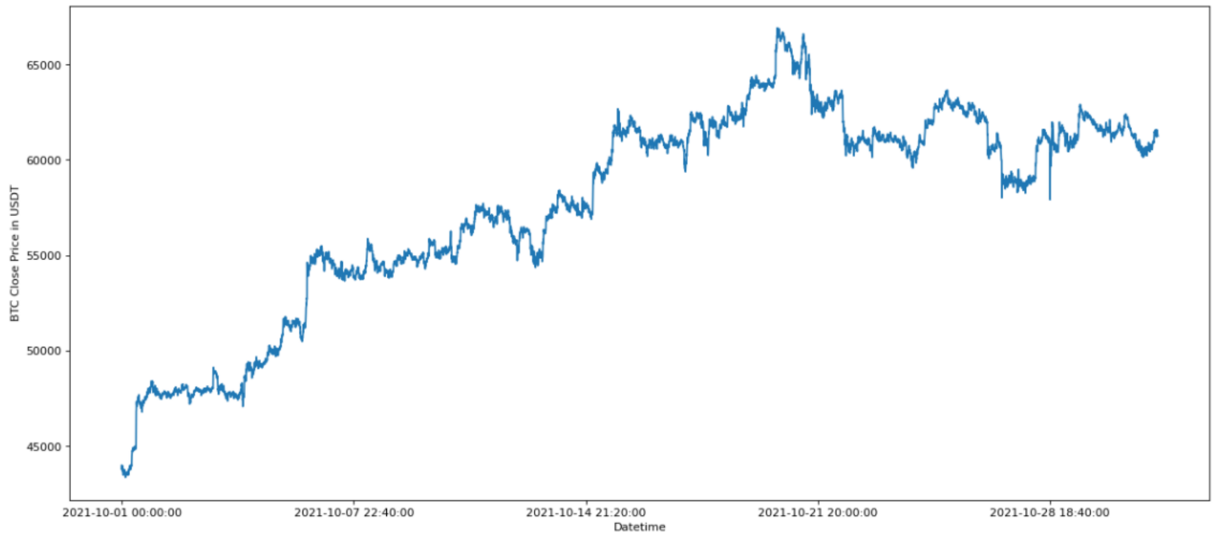


Figure 3: Development of Bitcoin price

We included 115 features which are divided in 5 categories as described in the data collection part. The current memory usage is 16,1 MB. In addition to the collected data, we add 19 features that are calculated using the collected data and which will be described in Feature Engineering part. An overview of the features is provided in Appendix 1.

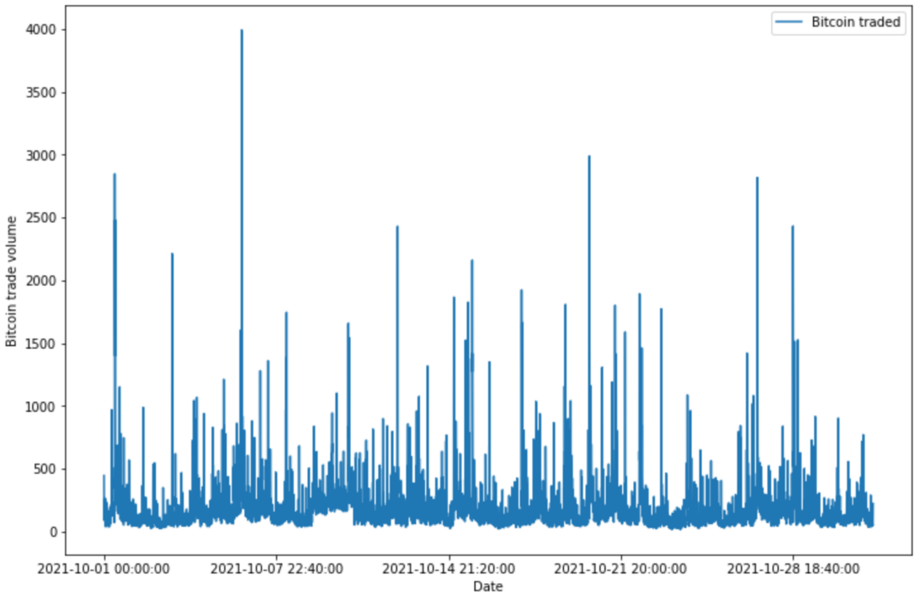


Figure 4: Bitcoin trade volume in 5-minute intervals

Figure 4 shows the trading volume of Bitcoin within a 5-minute time interval. On average, 175 Bitcoins are traded every 5 minutes on Binance, making the Binance platform a liquid trading venue.

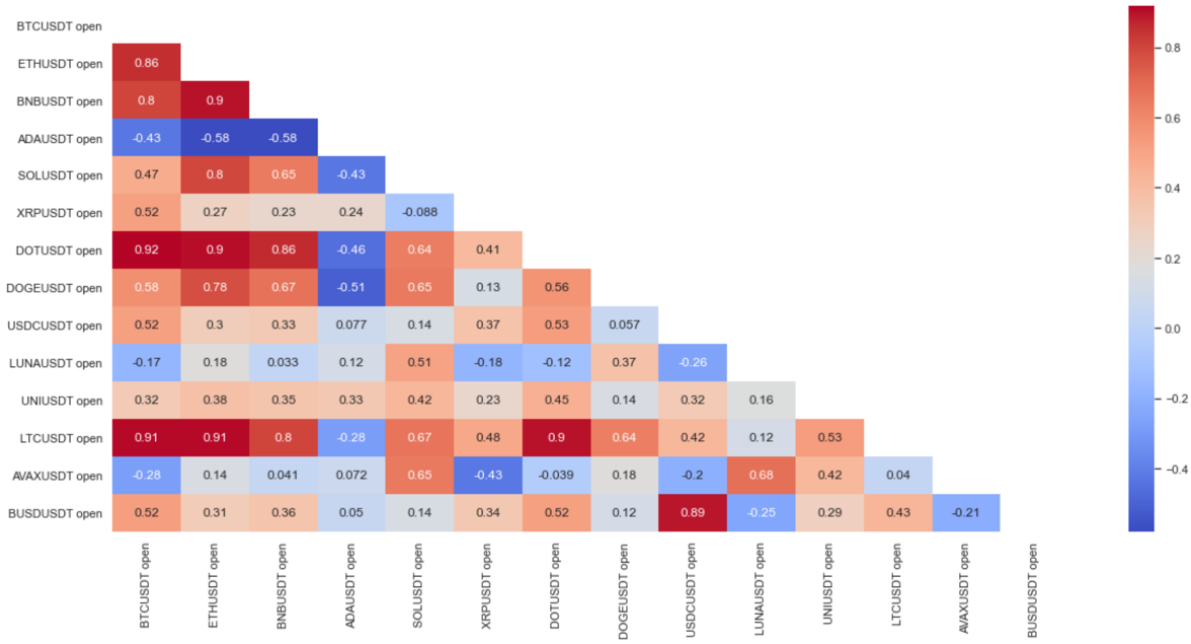


Figure 5: Correlation matrix of cryptocurrency market features

Figure 5 shows a correlation analysis of the cryptocurrencies observed in this work. The correlations range from -0.51 to 0.91. The differing correlations between cryptocurrencies indicate that the cryptocurrency market is not always moving in the same direction. The highest positive correlation with Bitcoin has Polkadot (DOT), i.e., 0.92, and the highest negative correlation Cardano (ADA), i.e., -0.43. In figure 6 the normalized price developments of the observed cryptocurrencies are visualized. The different developments support the findings of the correlation matrix. DOT experiences the highest increase in the observed period, ADA experiences the worst development.

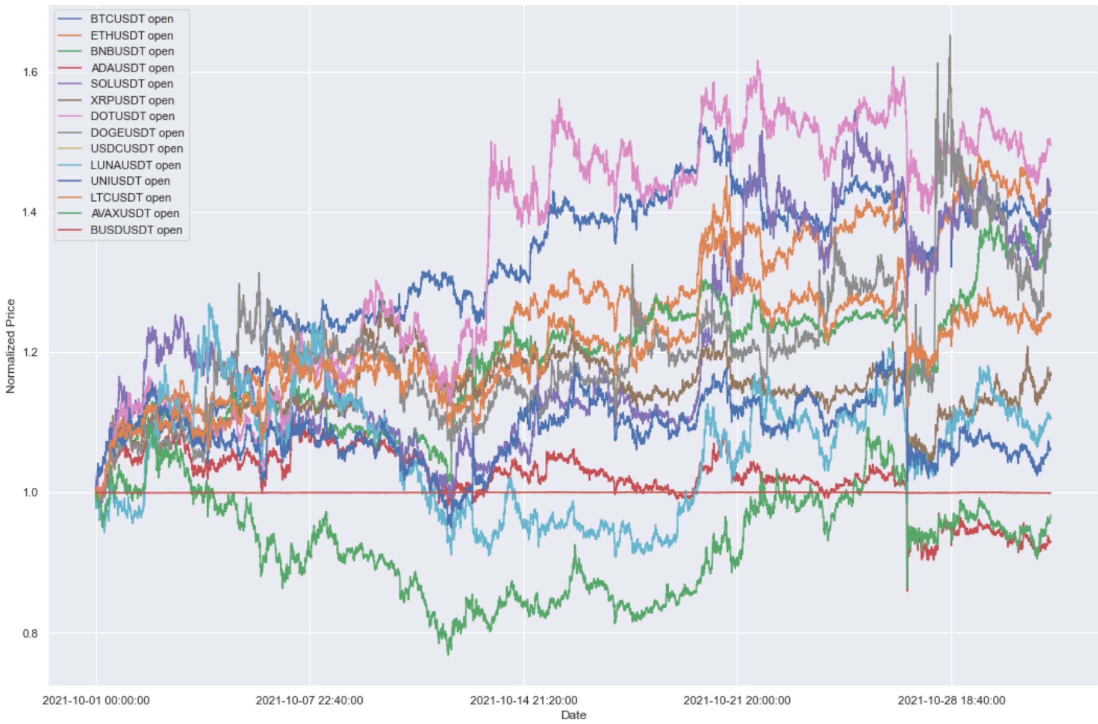


Figure 6: Normalized price development of observed cryptocurrencies

Cryptocurrency trading is not limited to opening hours of stock exchanges but is possible 24 hours, seven days a week. Data for cryptocurrencies exists for every 5-minute timestamp of the observed period as shown in figure 6. Data of the commodity market is received for opening hours of the market. This is visualized in figure 7.

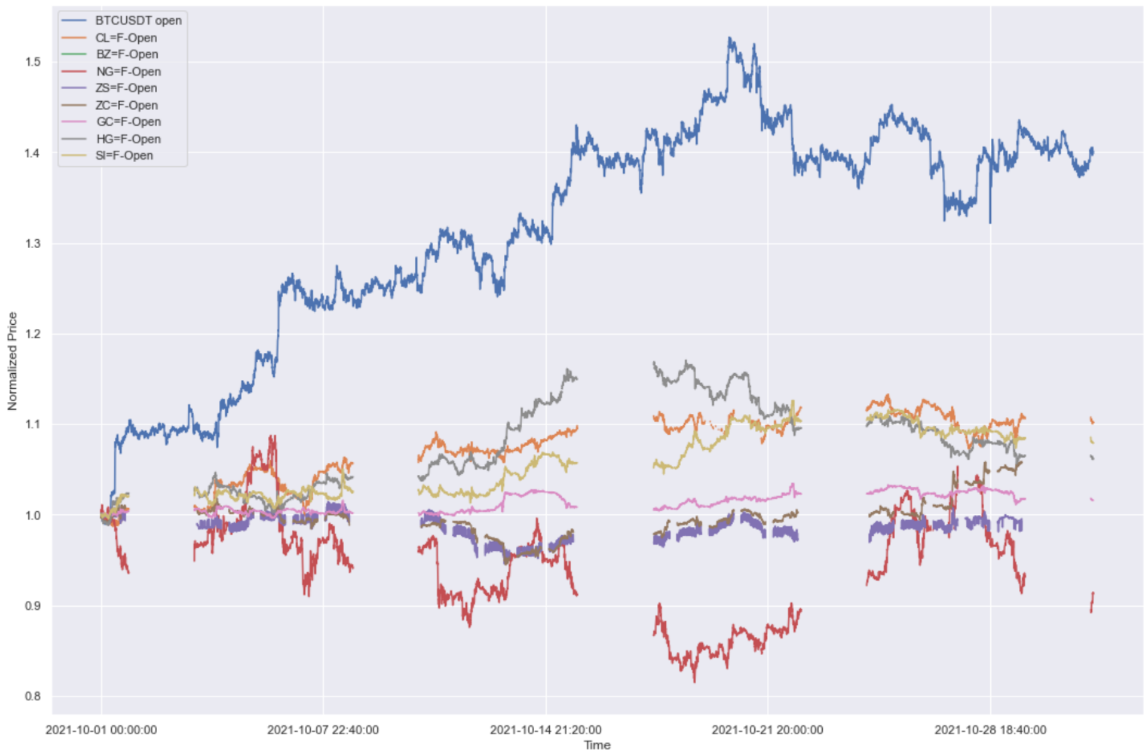


Figure 7: Normalized price development of commodity and Bitcoin prices

3.2.3 Data Transformation

3.2.3.1 Data Preprocessing

Different sources were used to retrieve data and the data was merged to build a comprehensive base for analysis and modelling purposes. As described in the previous section, we observe missing values in 98 of our 115 input features. As we are dealing with time-series data from the cryptocurrency as well as the general stock market data, missing values in our data have specific characteristics. While cryptocurrencies can be traded 24 hours a day, seven days a week, the trading of stocks, commodities and other securities is bound to opening hours of stock market exchange providers (Dutta, Kumar and Basu 2020). The number of missing values can differ between features, because of after-hours trading and differences in opening hours between Stock Exchanges. After-hours trading occurs after regular market hours. Due to after-hours trading and after-hours volatility, the opening price for a stock on the following day can differ quite extensively from the price at which it closed the previous day (Barclay and Hendershott

2003). Missing values need to be accounted for by deleting respective observations or features, or by imputation, because most of the existing ML algorithms don't work well with missing values.

Other researchers that have used general stock market data to predict cryptocurrency prices have used imputation methods to fill missing values. One of the simplest imputation methods that has been widely used is the Last Observation Carried Forward (LOCF) time-series imputation method (Vo and Yost-Bremm 2018). Therefore, it is assumed that stock, commodity, and other security prices do not change after closing hours, i.e., after-market trading is ignored (Dutta, Kumar and Basu 2020).

We use the LOCV imputation method in combination with Next Observation Carried Backward (NOCB) method to account for missing values that arise within time-horizons when the general stock market is closed. As for the LOCV, missing values are imputed as the previously observed value, i.e., the last observation is carried forward. In case there is no previous value the NOCV method is used. Thus, the follow-up value is used to impute the previous value. The combination of the observed and imputed data is then analyzed as there were no missing data.

3.2.3.2 Feature Transformation

The architecture of our model requires a transformation of our target variable, i.e., Bitcoin price. Sebastião and Godinho (2021) found out that model assembling enables profitable trading strategies. We formulate the analysis as a Regression and Classification problem with three different prediction horizons *ph = {"one hour": 1h, "two hours": 2h, "three hours": 3h}*. The horizons are selected, aiming to take advantage of intraday trading and avoid exaggerated transaction costs. The prediction is evaluated each five minutes as it is the most granular level, we can collect a comprehensive number of features. In addition to a Regression problem, we are dealing with a Classification problem, and we need to transform the Bitcoin price into a categorical variable, i.e., trend variable. For the Classification problem, we create a categorical variable that is equal to *0, 1* or *2.* The variable is set equal to *0* if the Bitcoin price is decreasing and set to *2* if the Bitcoin price is increasing. In all other cases the variable is set to *1*. We use a threshold of 0.5% to calculate the trend variable for the Classification problem. The distribution of the trend variables for each time lag is shown in figure 8. For the Regression problem, we create lagged variables for all prediction horizons *1h, 2h,* and *3h.* Each lagged variable represents the target variable for one prediction horizon and is used to validate and test the respective model.

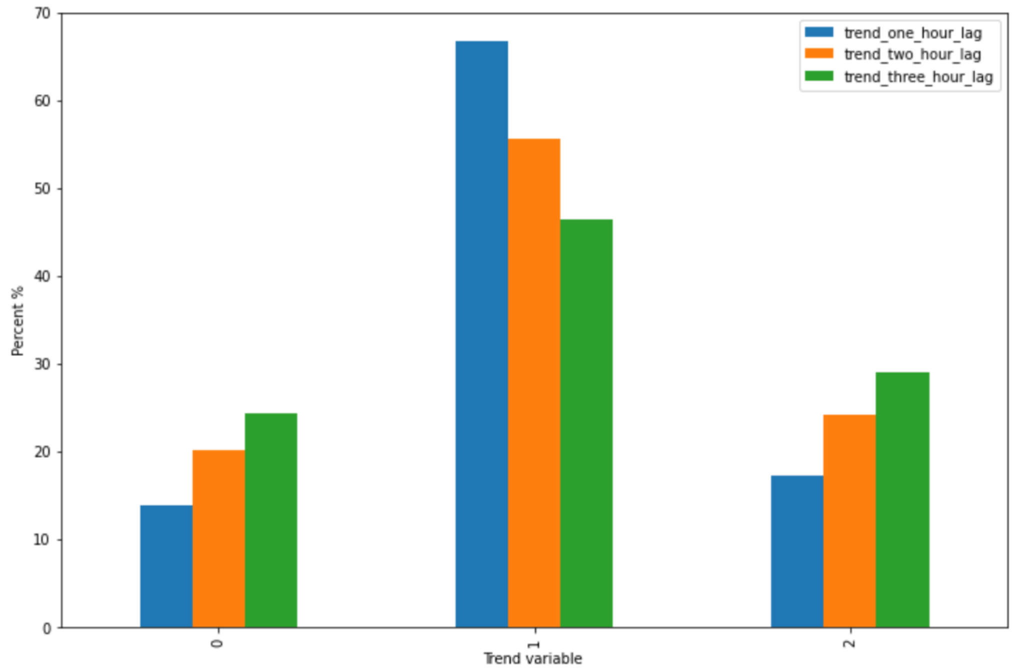


Figure 8: Distribution of trend variables

Aiming to improve modelling performance we add new features by transforming existing data. Lagged variables are common features to be included for predictive analysis of cryptocurrency prices and price trends (Sebastião and Godinho 2021). We include past lagged trend variables with time lags equal to those of the trend target variables, i.e., one, two and three hours. They show the trend of the Bitcoin price compared to the previous instance according to the defined time lags.

The day-of-the-week effect represents a well-known phenomenon in the study of financial markets where differing returns between days of the week are observed in a persistent way.

Indications for this anomaly are observed for many products on the financial market (Aharona and Qadan 2019). The price fluctuations of cryptocurrencies, especially Bitcoin, seem to depend on the day of the week (Sebastião and Godinho 2021). In this paper, we created daily dummy variables for each weekday to capture effects that are related to certain days of the week.

3.2.3.3 Technical Analysis

Technical analysis is the study of historical prices and price movements in the market to get an estimation of the price or its trend in the future (Borges and Neves 2020). Technical analysis intends to identify specific rules like price trends, market cycles, momentum, volatility, or price chart patterns, under the assumption that prices move in trends and historic movements repeat themselves (Huang, Huang and Ni 2019). Extensive research regarding the impact of technical analysis of stocks has been conducted, constituting a high importance of technical features on future price predictions and trends (Fang, et al. 2020). For the cryptocurrency market, prior researchers have also used technical analysis for price prediction and have also concluded that technical features are an important factor to predict price movements (Kristjanpoller and Minutolo 2018) (Nakano, Takahashi and Takahashi 2018) (Huang, Huang and Ni 2019), (Abbad, Fardousi and Abbad 2014).

A vast amount of different technical indicators has been used in prior research with the intention to improve prediction of future price movements. Technical indicators differ in their purpose and can be divided in different categories like overlap study indicators, momentum indicators, cycle indicators, volatility indicators, and pattern recognition indicators. Prior research on the predictability of Bitcoin prices using a large set of 124 technical indicators has been conducted. (Huang, Huang and Ni 2019). Other researchers focus on a small number of technical indicators that are widely accepted. The most represented technical indicators that we identified in our research are Exponential Moving Average (EMA), Moving Average Convergence–Divergence

(MACD), Relative Strength Index (RSI), On Balance Volume (OBV) (Borges and Neves 2020) (Vo and Yost-Bremm 2018) (Nakano, Takahashi and Takahashi 2018).

In this paper, we calculated and included the following technical indicators: Exponential Moving Average (EMA), Moving Average Convergence–Divergence (MACD), Relative Strength Index (RSI), On Balance Volume (OBV) and Stochastic Oscillator. These technical indicators have comprehensively been used in prior research and are widely accepted by traders on the market. All indicators depend only on past Bitcoin prices. In the following paragraphs, we will provide more detailed information about the technical indicators that are used in this paper. Further elaborations and explanations of technical analysis and each technical feature can be found in (Murphy 1999).

Exponential Moving Average (EMA)

A moving average (MA) is a technical indicator that helps to smooth out the price data by dampen the effects of short-term oscillations, through a constantly updated average price. The MA is a trend following indicator that reacts to the market by announcing a trend that has already begun. An EMA is a variation of the MA that assigns more weight and significance the most recent data points, having the ability to react faster to recent price variations (Borges and Neves 2020). As this paper intends to analyze short-term predictions of the highly volatile Bitcoin price, we us the EMA. The EMA is calculated using the following equation:

$$EMA_t = EMA_{t-1} + \frac{smoothing\ factor}{n+1} + [Price_t - EMA_{t-1}] \qquad (1)$$

In equation *(1)*, *t* refers to the current period, *n* refers to the number of time periods the EMA is calculated on, and the *smoothing factor* represents a smoothing parameter that is set to the most common value of two for all calculations. For this study, we used 6 different time periods corresponding to *n = {12, 24, 48, 96, 288, 576},* representing time periods of one, two, four and eight hours, as well as one and two days. The first *n* values of EMA are set to an initial average of the first *n* time periods for each of the time periods, respectively.

Moving Average Convergence–Divergence (MACD)

The MACD is calculated using the difference between two trend following indicators, EMAs, of different time periods. As a trend-following momentum indicator it combines the purpose of trend-following and momentum (Borges and Neves 2020). The MACD is popular among traders thanks to its simplicity and effectiveness. It shows how the two EMAs converge and diverge and helps to understand whether the bullish or bearish movement in the price is increasing or decreasing (Vo and Yost-Bremm 2018). Traditionally, a 26- period EMA is subtracted from a 12-period EMA to calculate the MACD (Borges and Neves 2020). We use a 24- period EMA and a 12-period EMA representing a period of *1h* and *2h*, respectively. The equation to calculate the MACD is the following:

$$MACD_t = EMA_t \times 12 - EMA_t \times 24 \tag{2}$$

In equation (2), *t* refers to the current period and *12* and *24* refer to the EMA with the respective period *n* at time *t*.

Relative Strength Index (RSI)

The RSI measures the magnitude of recent price changes and is used to identify general price trends (Vo and Yost-Bremm 2018). It is a momentum oscillator that is used to evaluate whether a market is overbought or oversold. It represents a line that moves between two extremes and can take a value between 0 and 100 (Borges and Neves 2020). The RSI is calculated using the following equation:

$$RSI_t = 100 - \frac{100}{(1+[average\ gain_{\ t-14}/average\ loss_{\ t-14,t}])} \tag{3}$$

In equation (3), *t* refers to the current period. *Average gain* and *average loss* are calculated using the gains and losses of the past 14 observations, while losses are set to zero to calculate the *average gain* and gains are set to zero to calculate the *average loss*.

On Balance Volume (OBV)

While the previous indicators utilize prices and price movements, OBV is a technical momentum indicator that focuses on volume flow to predict price changes. OBV is built on the idea that volume movement precedes price movement and is a key factor behind markets. An increase of OBV signals a price move up while a decrease of OBV signals a decrease (Vo and Yost-Bremm 2018). The RSI is calculated using the following equation:

$$OBV_t = \begin{cases} OBV_{p-1} + Volume_p, & if\ Price_t > Price_{t-1} \\ OBV_{p-1} - Volume_p, & if\ Price_t < Price_{t-1} \\ OBV_{p-1}, & if\ Price_t = Price_{t-1} \end{cases} \tag{4}$$

In equation (4), t refers to the current period and *Volume* refers to the amount of trading volume in the past 5 minutes prior to t.

3.2.3.4 Study I: Sentiment Analysis

3.2.3.4.1 Introduction

Bitcoin is a speculative asset and its price is highly volatile (Valencia, Gómez-Espinosa and Valdés-Aguirre 2019). Trading Bitcoin is associated with high risk but also the chance for extraordinary profits. As a decentralized cryptocurrency, Bitcoin is trading 24 hours, 7 days a week, which makes it an exciting target for price speculations and predictions (Kraaijeveld and De Smedt 2020). Bitcoin does not have an inherent value and is not backed by any government (Bugár and Somogyvári 2020). Being detached from the characteristics of traditional assets, the value drivers of Bitcoin are continuously discussed and researched (Woebbeking 2021). The long-term development of Bitcoin represents a major discussion, where opinions range from Bitcoin being a bubble to being a solid investment (Valencia, Gómez-Espinosa and Valdés-Aguirre 2019). Therefore, the Bitcoin price is driven by behavioral factors where people do not follow their own analysis but, e.g., the opinion of a majority (Sebastião and Godinho 2021). Social media platforms are predominantly used to exchange information on Bitcoin and social media sentiment proved to have an influence on cryptocurrency prices (Kraaijeveld and De Smedt 2020). Additional, statements from influencers like Elon Musk have a significant influence on the Bitcoin price, leading to abnormal returns (Ante 2021).

The integration of high frequency features derived from Sentiment Analysis in algorithmic trading is barely represented in past academic literature. The important of sentiment for algorithmic trading of Bitcoin is analyzed by answering the following research question:

Does Twitter Sentiment impact short-term price fluctuations in Bitcoin?

In this work project, I perform a Sentiment Analysis for intraday sentiment extracted from Twitter data and investigate the impact on short-term price fluctuations of Bitcoin. I define a group of high influential accounts, i.e., VIP accounts, and analyze the difference between the impact of *Overall Sentiment* and *VIP Sentiment* on the Bitcoin price.

3.2.3.4.2 Related Work

Previous research observed the relationship between Twitter sentiment and the development of cryptocurrency prices. Kraaijeveld and Smedt (2020) observe *tweets* on a daily interval and find evidence that the observed sentiment has a Granger causal relation to Bitcoin returns. They formulate the hypothesis that an intraday analysis of Twitter sentiment further improves the results. The hypothesis is supported by findings of Pano and Kashef (2020), who identify a higher correlation for shorter time horizons if an optimal text pre-processing strategy is applied. Text pre-processing is a technique to reduce noise and is commonly used for Sentiment Analysis, especially for social media content. Multiple pre-processing methods for *tweets* are observed and compared by Pano and Kashef (2020) to extract the correct sentiment and categorize *tweets* in positive and negative. Sentiment can be extracted by two approaches. The first approach is lexicon-based and does not require labelled data and rely in predefined lexicons (Turner, Labille and Gauch 2021). The second approach includes ML and requires manually labelled data to create a training set. Where humans have to manually categorize the sentiment of a fraction of the *tweets* (Pant, et al. 2018). The VADER lexicon-based approach is the most popular method for Sentiment Analysis in social media, as described by Kraaijeveld and De Smedt (2020) and Stenqvist and Lönnö (2017). Dr. Rajab and Jahjah (2020) find a positive correlation between Twitter sentiment and the Bitcoin price of the next day. To further analyze a correlation for a "causal" relationship, the Granger causality test is a common procedure. The Granger causality test is used in prior research to investigate the relationship between stock data and investor sentiment (Chu, Wu and Qiu 2015) and the relationship between cryptocurrency data and social media sentiment (Kraaijeveld and De Smedt 2020) (Dastgir, et al. 2019).

The findings of Pant, et al. (2018) indicate that there is also a relationship of specific Twitter cryptocurrency news accounts and the Bitcoin price.

This work project performs an intraday Sentiment Analysis of tweets and investigates a Granger causal relationship of the observed sentiment and the Bitcoin price. Furthermore the observed news accounts from Pant, et al. (2018) will be extended and also tested for a Granger causal relationship.

After an introduction of Twitter, the project is divided into five different steps: (1) Data Collection, (2) Data Pre-processing, (3) Sentiment Analysis, (4) Filtering and a (5) Granger causality test. Further visualized in Figure 9:

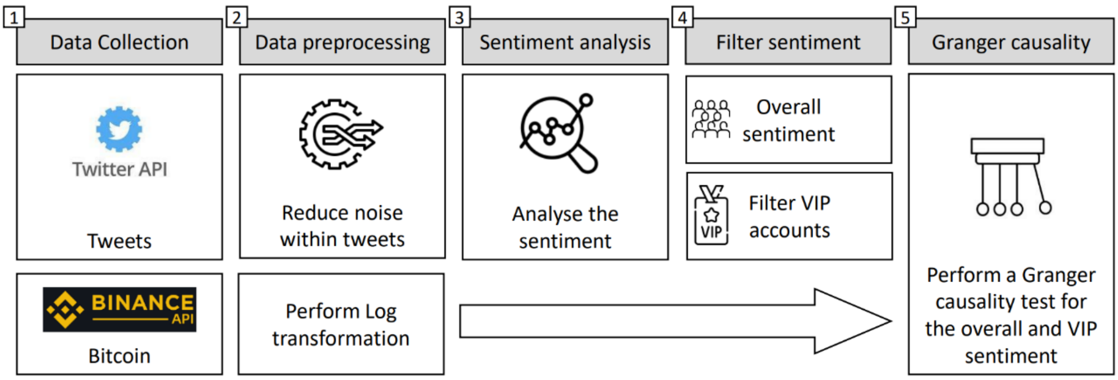


Figure 9: Structure of the Sentiment Analysis

3.2.3.4.3 Twitter

"Twitter is a global platform for public self-expression and conversation in real time. Twitter allows people to consume, create, distribute, and discover content and has democratized content creation and distribution." (Twitter, Fiscal Year 2020 Annual Report 2021, 6). Twitter has developed a rapid growth in users and popularity, becoming one of the most used social media networks globally (Wollams 2021). In 2020, average daily active user amounted to 192 million, increasing 27% compared to last year (Twitter, Fiscal Year 2020 Annual Report 2021). In 2021, Twitter was the fourth most visited website and the second most visited social network and online community globally (Neufeld 2021). It offers the users to share text, images, or short videos (Antonakak, Fragopoulou and Ioannidis 2021). A post on Twitter is called a *tweet.* It represents a short message addressed to a wide variety of receivers and is limited to a maximum of 280 characters (Twitter, Developer Platform 2021d). Therefore, users must be precise with their statements.

Due to the limitation in characters, the high number of users, the substantial increase of popularity, the publication of opinions and trends makes the Twitter database helpful in analyzing public opinions on specific topics (Kraaijeveld and De Smedt 2020).

3.2.3.4.4 Data

3.2.3.4.4.1 Data Collection

For this work, data was collected from 01.10.2021 up to 31.10.2021. Bitcoin prices are collected through the Binance API in a 5-minute frequency summing up to 8,893 observations in total (Binance 2021). *Tweets* are collected through the Twitter API (v2: Early access), launched in June 2020 (Twitter, Developer Platform 2021a). Counting 4,81 million *tweets* in English, queried for the keyword: “Bitcoin”. Queries in Twitter are case-insensitive for all characters (Twitter, Developer Platform 2021b). Which, for example, would also include the following notations: “bitcoin”, “BITCOIN”, or “BiTcOiN”. For each *tweet,* the following data is received:

Table 7: Received Twitter information

Information	Detail
Created_at	Timestamp in UTC when the post was published
Author_id	Unique author id from which the post was published
Public_metrics	Retweet, reply, like, and quote count
Source	The operating system was the post was made on
Text	The content of the tweet

If a tweet is retweeted, it will contain the same message as the original *tweet* but starts with: RT @username. Public metrics are measured at the point in time when the tweet was pulled. *author_id* is a unique number given to each account.

3.2.3.4.4.2 Data Descriptive statistics before data pre-processing

A total number of 864,968 unique *author_ids* were identified.

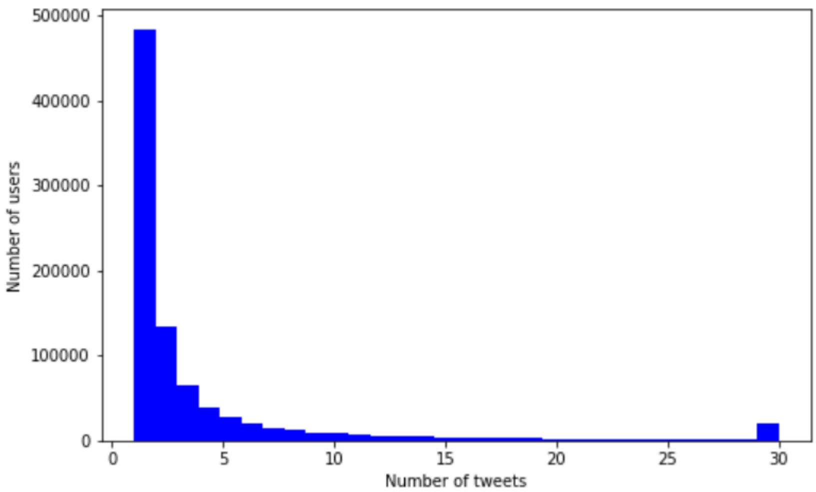


Figure 10: Distribution of tweets each user

Figure 10 shows how many times a unique *author_id* tweeted during October 2021. If the account tweeted more than 30 times, it was counted towards bin 30. Most of the accounts (478,000) only *tweet* once a month. As some accounts are tweeted 30 or more times indicate that the population is split into active tweeters and heavily active tweeters.

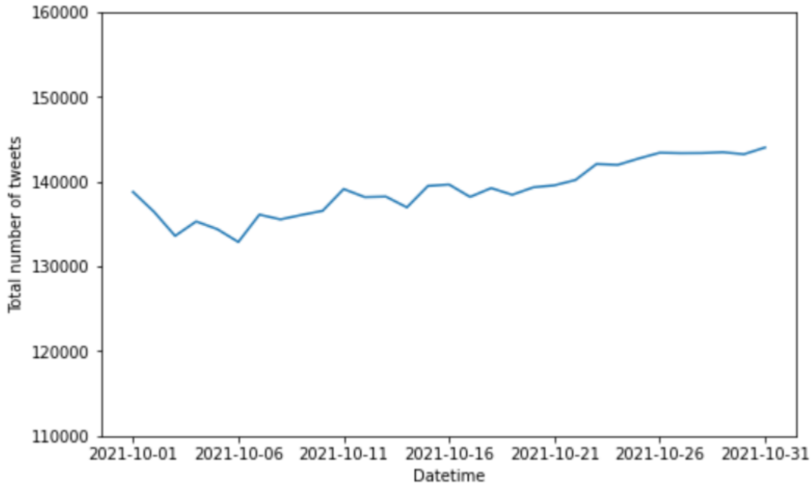


Figure 11: Development of total number of tweets

Figure 11 shows the number of total *tweets* observed per day, roughly 139,000 *tweets*. Towards the end of the month the number of *tweets* increases to 144,000 *tweets* a day.

3.2.3.4.4.3 Data Pre-processing

A *tweet* has a different structure than a newspaper article or a literary text from a book. The publication takes place without a correction. *Tweets* contain significantly more slang, emojis, or spelling mistakes (Pano and Kashef 2020). Hashtags (#), retweets (RT), mentions (@), or URLs, which are frequently used in *tweets*, add noise to the text (Antonakak, Fragopoulou and Ioannidis 2021). For example, the tagged username @love or @nice will influence the sentiment score of a sentence although it just represents a username without any polarity. A hashtag connected to a word #like or #love could also distort the analysis. It is recommended to clean *tweets* before performing the Sentiment Analysis (Elbagir and Yang 2019). To reduce noise in *tweets*, I performed the following data preparation steps, following the documentation proposed by (Pano and Kashef 2020). Table 8 shows all preprocessing steps performed.

Table 8: Preprocessing steps

Action	Example
Delete hashtags	#
Delete tags of users	@sampleusername
Delete URLs	https://www.sampledomain.com/
Delete HTML entities	&amp, \n, etc.
Delete stopwords	the, is, a, etc.
Delete numbers	1-9

Figure 12 visualizes the step of Preprocessing for a sample tweet.

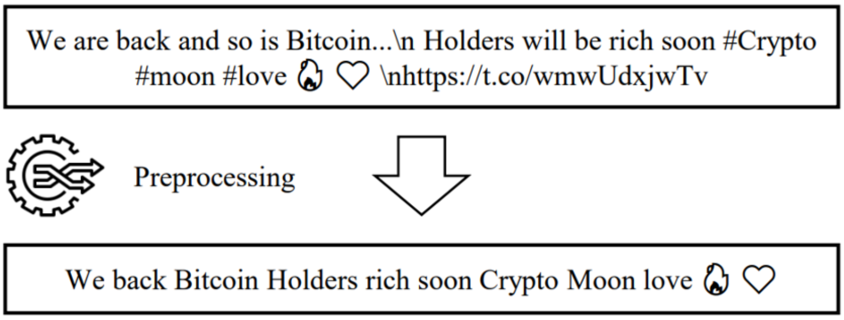


Figure 12: Preprocessing visualization

3.2.3.4.4.4 Descriptive Statistics

After cleaning with the above-mentioned techniques, a common technique to visualize insights about textual data is a Word Cloud (Heimerl, et al. 2014). The most used words in connection with Bitcoin are identified and first interpretations about important topics connected to Bitcoin can be made. The more giant and bolt a word is represented in the graphical output, the higher is the frequency of the observed text. Figure 13 presents the Word Cloud for the search conducted in this work. Next to Bitcoin, the most frequently used words are: Crypto, BTC, gift, Price, and Ethereum. This shows that people that tweet about Bitcoin also tweet about other cryptocurrencies and the price. This is a first indication that the price is linked to the content of the observed *tweets*.

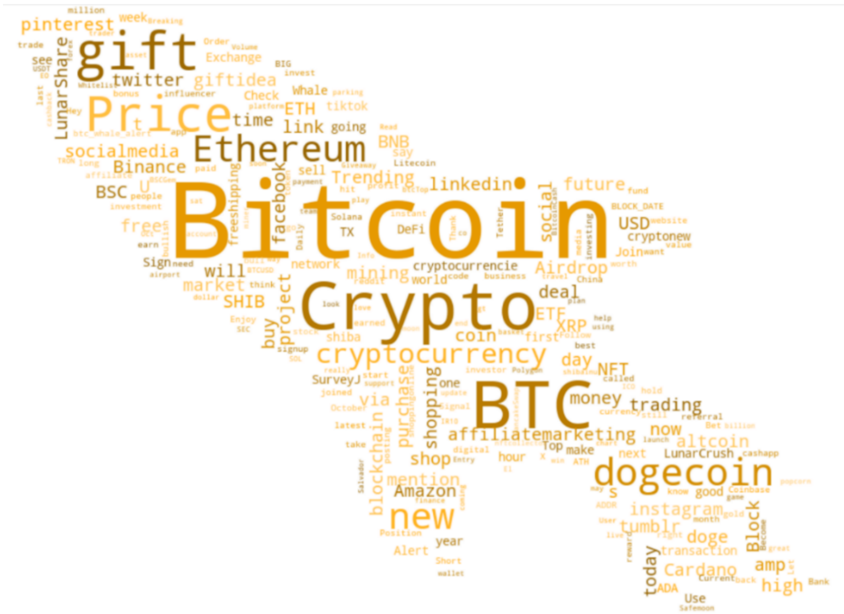


Figure 13: Tweet Word Cloud

3.2.3.4.5 Methodology

3.2.3.4.5.1 Sentiment Analysis

Sentiment Analysis is at the forefront of Natural Language Processing as it possesses the most beneficiary abilities to analyze textual data (Medhat, Hassan and Korashy 2014). Sentiment Analysis deals with analyzing the polarity of texts. It processes Sentiment by computing polarity for each word in the analyzed text (Srishthi and Seba 2020). The growing field of Sentiment Analysis provides a wide range of tools specifically developed for many kinds of textual data. These tools differ in their method and usage but can be overreachingly divided into a lexicon-based approached and a ML based approach. Lexicon-based tools are most often used in literature, as they possess the most accessible and most understandable way of analyzing sentiment in data (Khoo and Johnkhan 2018). ML tools are less optimal to interpret and require manually labeled data to create a train set (Mitra 2020). The ML and the lexicon-based approach are explained in more detail in the following section.

3.2.3.4.5.2 Machine Learning based Tools

The process of Sentiment Analysis, with ML tools, can be interpreted as a sentiment Classification. A given ML algorithm is trained on a data set with a label/class for each data point (Liu, Bi, and Fan 2017). The model is tested on the entire data set if its accuracy on a labeled validation data set is high enough (Jain and Jain 2019).

A problem with this kind of method is the need for labeled data. As the current dataset contains 4.81 million *tweets* it would take a huge amount of time and people to label a train set manually. Furthermore, it could lead to unwanted biases in the results if the data has been classified by only a few people. ML based tools are also less interpretable because, respective of the given model, the tool acts as a kind of black box on the data (Carvalho, Pereira and Cardoso 2019). This problem arises for traditional ML algorithms and increases for Deep Learning algorithms.

3.2.3.4.5.3 Lexicon-based Tools

Lexicon-based approach is based on lexicons of features – in this case: words – with their respective polarity. These lexicons were already filled and labeled. Texts are analyzed by querying the words and aggregating the produced polarity values (Turner, Labille and Gauch 2021). A challenge for lexicons is that some words in English do not possess a single polarity score but influence the polarity of other words, intensifiers like “very” or “most” or downtowners such as “slightly” or “somewhat” (Taboada, et al. 2011).

Taboada, et al. (2011) also show that negators and intensifiers (boosters) are words that drastically change the sentiment of a sentence without having a polarity value on their own such as "not", "don’t" or “nobody”. However, they do change the sentiment of a sentence or a word they are linked to.

Furthermore, in social media posts, emotions are often expressed in emoticons rather than in words which also need to be measured to extract the sentiment. In the following paragraph two lexicon-based sentiment analysis tools are presented and compared.

3.2.3.4.5.3.1 SentiArt

SentiArt is a lexicon-based Sentiment Analysis tool that has produced good results on literary texts. It was developed by (A. M. Jacobs 2019). The tool consists of lexicons from three different languages (English, German and Dutch). These lexicons have different feature columns that express the computed score for six emotions such as “fear” or “joy”. The feature which compared to the compound score of VADER is the AAP (Affective-Aesthetic Potential) score (Jacobs, et al. 2020). While SentiArt proved to produce good results on sentiment of novels and literary texts, it does not possess the ability to analyze the sentiment of emojis or punctuation (A. M. Jacobs 2019).

3.2.3.4.5.3.2 VADER

The lexicon-based tool Valence Aware Dictionary and Sentiment Reasoner (VADER) was first introduced by (Hutto and Gilbert 2014). VADER consists of a lexicon of words that have been developed by researchers using qualitative and quantitative methods (Elbagir and Yang 2019). The lexicon’s features were established using a wisdom-of-the-crowd approach where different people assess a word's sentiment, and all assessments are aggregated to form a precise assessment (Hutto and Gilbert 2014). The developed lexicon is then combined with grammatical and syntactical rules to establish sentences' estimates in different combinations. (Hutto and Gilbert 2014). VADER has demonstrated to improve social media data results compared to traditional sentiment lexicons, due to the following reasons (Hutto and Gilbert 2014): A major part of the VADER model is its ability to take symbols into account. For instance, emoticons play a significant role on Twitter as they convey a user’s sentiment ((Eisner, et al. 2016). VADER possesses an emoji dictionary that maps an emoticon to a fitting description. For example, the emoji 😊, which conveys a positive sentiment, has the description “smiling face with smiling eyes”, such that VADER evaluates the sentiment of the emoji using only its description.

Another part of *tweets* are punctuation marks. Using many punctuation marks, users convey boosted opinions and sentiments. The sentiment of "I really like this" differs from the sentiment of "I really like this!!!". The second sentence is much more intense than the first. Similarly, the use of ALL-CAPS modifies a sentence's sentiment (Hutto and Gilbert 2014). Using the example from above, "I REALLY like this" conveys more intensity than the original sentence. These problems are tackled in VADER by using an empirically derived mean sentiment intensity rating increase and multiplying this scalar value with the given sentence or word as established in (Hutto and Gilbert 2014). Additionally, VADER is accurate in catching the negations mentioned above (Swarnkar 2020). VADER returns a compound score ranging from -1 (negative) to +1 (positive), which is further defined as the Sentiment score (Boldt and Borg 2020). Figure 14 visualizes the Vader Sentiment Analysis of the above cleaned tweet:

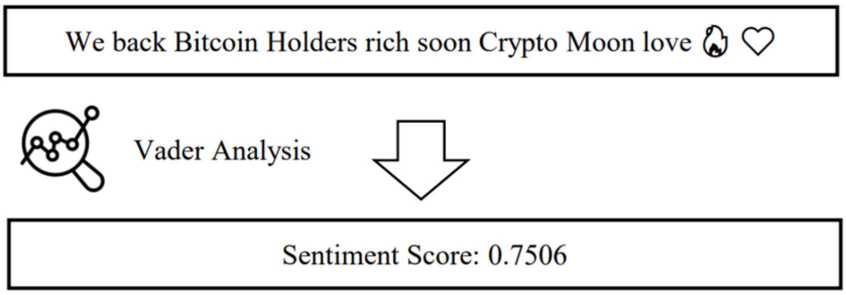


Figure 14: VADER Sentiment Analysis procedure

3.2.3.4.6 VIP Sentiment

A post from someone who is an expert or has a high number of followers is more valuable than a post from an average person, which is supported by the study from (Pant, et al. 2018) (Pant, et al. 2018).

To give these accounts a different weight, high influential Twitter accounts from people and organizations are identified, considering four different influential groups. First, the seven Bitcoin news accounts identified by Pant, et al. (2018) are included. The second group consists of the ten most followed news accounts on Twitter, according to (McCabe 2019). The third group is represented by the account of Elon Musk, chosen due to the observed abnormal returns in the work performed by Ante (2021). Finally, the list is completed by the most influential people on Twitter regarding cryptocurrencies and Bitcoin, according to Coinbound (2021). Appendix 2 shows all accounts that were added and analyzed within the VIP Sentiment.

3.2.3.4.7 Results and Discussion

3.2.3.4.7.1 Sentiment Results

In this section, I present the results of the performed VADER Sentiment Analysis and compare the differences between the *Overall Sentiment* and the *VIP Sentiment*.

Table 9: Results of the Sentiment Analysis

Measure	Overall Sentiment	VIP Sentiment
Mean	0.254	0.166
Standard deviation	0.055	0.355
Minimum	-0.087	-0.908
Maximum	0.536	0.917

Table 9 describes the *Overall Sentiment* results of the Sentiment Analysis. The mean is higher for the *Overall Sentiment* (0.254) than the *VIP Sentiment* (0.166), but both are positive. The standard deviation is significantly higher for the *VIP Sentiment* (0.355) compared to the *Overall Sentiment* (0.055). In line with the higher standard deviation, minimum and maximum values are more extreme for the *VIP Sentiment*. The minimum value of the *Overall Sentiment* is particularly striking because it is only (-0.087), which indicates that the average Sentiment nearly never gets negative.

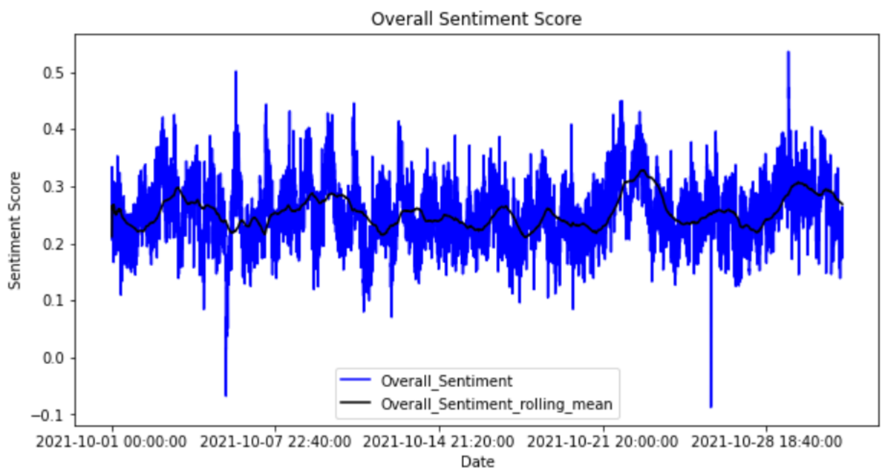


Figure 15: Development Overall Sentiment score

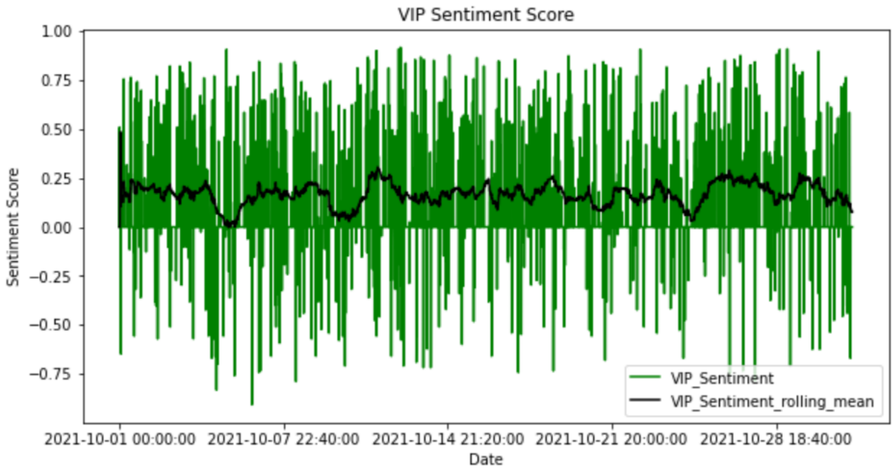


Figure 16: Development of the VIP Sentiment score

Figures 15 and 16 visualize the development of the *Overall Sentiment* and the *VIP Sentiment* during October 2021. The black line is the daily rolling average for each respective Sentiment which is also compared in one graph in figure 17. The *Overall Sentiment* has four outstanding peaks during October, with two highs and two lows.

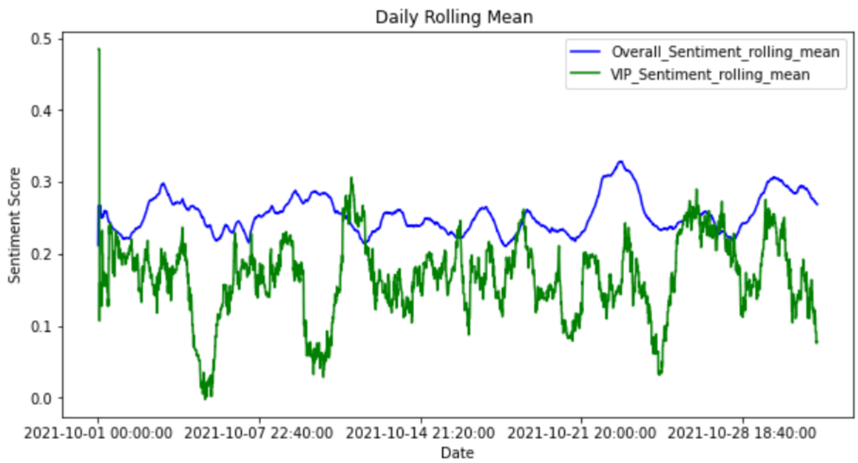


Figure 17: Comparison between Overall Sentiment and the VIP Sentiment

3.2.3.4.7.2 Granger Causality

"For a strictly stationary bivariate process $\{(X_t, Y_t)\}$, $\{X_t\}$ is a Granger cause of $\{Y_t\}$ if past and current values of X contain additional information on future values of Y that is not contained in past and current Y-values alone." (Diks and Pancheko 2006). A variable is helpful if the included variable X reduces a prediction error of the target variable Y (Clower 2021). Testing for Granger causality within a Sentiment Analysis is critical to check whether the sentiment Granger causes the Bitcoin price or if the Bitcoin price Granger causes a sentiment (Kraaijeveld and De Smedt 2020). It is the most common "causality" test for Sentiment Analysis and is widely used in a broad range of papers for cryptocurrencies and stocks (Kraaijeveld and De Smedt 2020) (Chu, Wu and Qiu 2015) (Behrendt and Schmid 2018).

One of the requirements for testing for Granger causality is that all variables need to be stationary (Granger 1969), which can be tested with the Augmented Dickey Fuller (ADF) Test (Dickey and Fuller 1981) that is commonly used in prior research (Li 2020).

If the p-value is <= 0.05, the null hypothesis, that the series is nonstationary, is rejected.

Table 10: Augmented Dickey Fuller Test

Data	**ADF Statistics**	**p-value**
Bitcoin Price	-2.734	0.068
Overall Sentiment	-9.236	0.000
VIP Sentiment	-37.173	0.000

For the ADF Test, a p-value > 0.05 is observed for the Bitcoin price. The null hypothesis cannot be rejected. The *Overall Sentiment* and the *VIP Sentiment* both have a p-value of 0.00 and do not need to be further transformed. A common technique to transform skewed data is taking log values (Li 2020). After calculating the log values and reperforming the ADF Test the p-value is less than 0.05.

Table 11: ADF Test after transformation

Data	**ADF Statistics**	**P-Value**
Log Bitcoin Price	-3.088	0.0274

After transforming, the Granger causality test can be performed, testing the null hypothesis:

$$H0 : \{Xt\}\ is\ not\ Granger\ causing\ \{Yt\}$$

The graph below plots the p-values for each time lag *t* from 1 to 40 in 5-minute steps for the respective sentiment. The null hypothesis is rejected when a p-value smaller <= 0.05 is observed, visualized as the red line in figure 18.

The blue and green highlighted areas represent the time intervals for which the null hypothesis can be rejected. Blue highlights the *Overall Sentiment* and green the *VIP Sentiment*.

For the *Overall Sentiment* the null hypothesis can be rejected with a time lag of one interval (5 minutes). The *VIP Sentiment* has a significant p-value given a time lag of nineteen intervals (95 minutes).

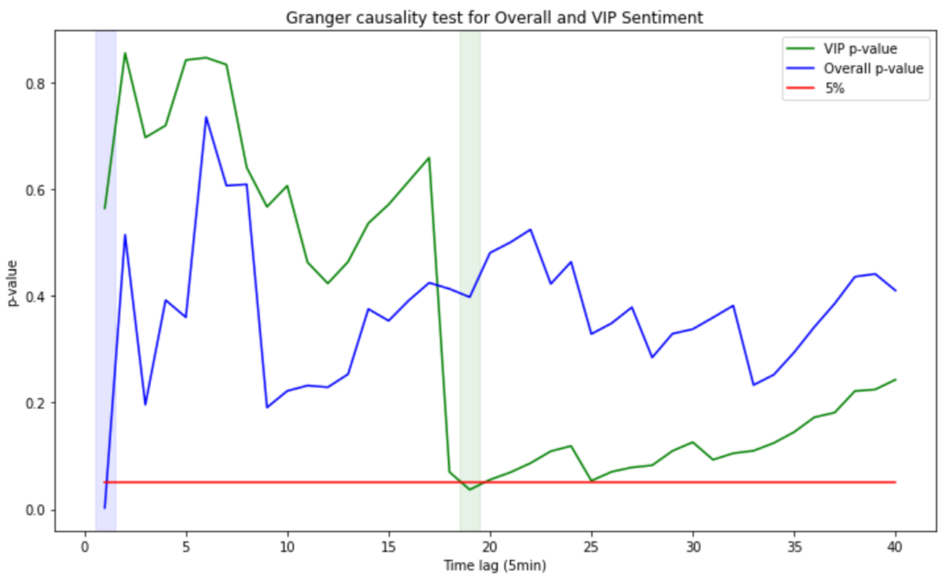


Figure 18: Granger causality test

3.2.3.4.8 Conclusion and Future Work

The relationship of *Overall* and *VIP Sentiment* to the intraday Bitcoin log price was analyzed using a Granger causality test. I find a Granger causal relationship of both Sentiments to the Bitcoin price. The *Overall Sentiment* demonstrates predictive influence in 5 minutes. The results of the *VIP Sentiment* show a Granger causal relationship to the Bitcoin log price in 95 minutes. The results show that Twitter Sentiment can reduce the predictive error. The effect of the Twitter *VIP Sentiment* is lagged against the *Overall Sentiment*. *Overall* and *VIP Sentiment* will be included as a feature for algorithmic trading.

Further work can focus on analyzing the interactions between *Overall* and *VIP Sentiment* and reveal the interrelationship. There could exist a group of accounts which are Granger causing the Overall Sentiment. In addition, the number of followers and the number of likes could be integrated into the analysis.

3.3 Study II: Modelling

3.3.1 Introduction

Bitcoin gained importance in financial markets, attracting media coverage, as well as attention of regulators, investors, and the public in general. Volatility in the cryptocurrency market is higher compared to volatility in traditional asset markets (Cocco, Tonelli and Marchesi 2021). Cryptocurrency trading is speculative as cryptocurrencies do not have an inherent value. In general, high model prediction performance increases the algorithmic trading profitability (Vo and Yost-Bremm 2018). Machine Learning (ML) techniques are promising in time-series prediction as they enhance algorithmic trading performance (Nakano, Takahashi and Takahashi 2018). For cryptocurrency time-series prediction, Deep Learning (DL) techniques have been shown to outperform traditional time-series algorithms (Dutta, Kumar and Basu 2020, Vo and Yost-Bremm 2018, Borges and Neves 2020). DL is a type of ML that imitates the learning process by which humans gain knowledge and is beneficial for analysing large amounts of data (Lim and Zohren 2020). Prior research has been conducted to test the ability of ML algorithms to predict cryptocurrency prices and price trends. A uniform design for Bitcoin price and trend prediction is missing that describes a comprehensive ML process and contains all required ML modelling steps. Consequently, a related research issue can be formulated as:

What is the optimal modelling design for Bitcoin price and trend prediction?

The objective of this work is to generate insights on how to build an optimal Bitcoin price and trend prediction model for algorithmic trading. I compare ML algorithms for Regression and Classification using Data Sampling, Scaling, Feature Selection and Hyperparameter Tuning. I conduct Hyperparameter Tuning for DL algorithms and evaluate the model performance in a bullish and bearish market, which adds novelty to previous research. The findings indicate that Long Short-Term Memory (LSTM) is best suited for Bitcoin price prediction (Regression) for all prediction horizons and Gated Recurring Unit (GRU), LSTM and Recurrent Neural Network (RNN) are the best models for trend predictions (Classification) in one, two and three hours, respectively. This work is divided in five sections. The 2nd section provides an overview of previous research in Bitcoin price and trend prediction. In the 3rd section, I outline the modelling methodology. In the 4th section, I discuss validation and evaluation results and conclude my work in the 5th section.

3.3.2 Related Work

Early cryptocurrency research classifies Bitcoin as a speculative asset due to high volatility and bubble-like behavior (Cheah and Fry 2015). Since cryptocurrencies have no inherent value, research focuses on factors influencing cryptocurrency price fluctuations (Fang, et al. 2020). Research investigated that Bitcoin is driven by endogenous (e.g., technical aspects of Bitcoin) and exogenous factors (e.g., cryptocurrency market, economic, political and sentiment indicators) (Bouri, et al. 2017). The research interest in cryptocurrency price and trend prediction has increased recently, simultaneously with the rise of market price and trading volume (Sebastião and Godinho 2021). Prior work proves that ML algorithms are effective in cryptocurrency price and trend prediction (Fang, et al. 2020). Based on a literature review of 136 cryptocurrency price and trend prediction papers, 53 papers apply ML algorithms. Table 12 contains the five most recent papers, predicting Bitcoin price or trend using ML and data input until at least 2019.

Table 12: Overview related works

Author (Year)	Problem	Sampling	Scaling	Feature Selection	Algorithm	Tuning	Best Results
Chen, Li and Sun (2020)	Classification	Train, Test	N/A	FI	Statistical & ML	N/A	67% accuracy (LSTM)
Cocco, Tonelli and Marchesi (2021)	Regression	CV	Standard	N/A	ML	Manual	2.66 MAPE (LSTM)
Dutta, Kumar and Basu (2020)	Regression	Train, Validate, Test	N/A	MC	Statistical & ML	Manual	0.017 RMSE (GRU)
Mudassir et al. (2020)	Both	CV	MinMax	MC & FI	ML	Manual	Regression: 1.58 RMSE (SANN) Classification: 60% accuracy (SANN)
Sebastião and Godinho (2021)	Both	Train, Validate, Test	N/A	N/A	Statistical & ML	Auto	Regression: 3.36 RMSE (LR) Classification 51% accuracy (SVM)

Based on table 12 seven ML modelling steps can be identified to compare the related work: Problem Statement, Sampling, Scaling, Feature Selection, Algorithm, Hyperparameter Tuning. The identified papers have specific focus areas. No paper links all seven steps into one model. The Bitcoin price prediction can be stated as a Regression or Classification problem. Sebastião and Godinho (2021) and Mudassir, et al. (2020) prove high predictive performance of both cryptocurrency price (Regression) and trend prediction (Classification). Sebastião and Godinho (2021) built trading strategies using a combination of Regression and Classification algorithms and find that model assembling achieves the most prosperous trading results. Cocco, Tonelli and Marchesi (2021) and Mudassir, et al. (2020) conduct subsampling and scaling of the dataset. Cocco, Tonelli and Marchesi (2021) compare different ML frameworks to predict the daily closing Bitcoin price and investigate robust model performance validation using 3-fold Cross-Validation. They find high prediction performance of two-staged DL models. Mudassir, et al. (2020) use MinMax scaler in their work. The results indicate high performance of ML models for predicting Bitcoin price movements in short (5 minutes) and medium (1 day) terms.

Three of the focus papers compare ML algorithms to statistical benchmarks to classify the predictive performance. Chen, Li and Sun (2020), Dutta, Kumar and Basu (2020) and Mudassir et al. (2020) utilize Feature Selection. Chen, Li and Sun (2020) investigate Feature Importance (FI) while applying ML algorithms to samples with different data structures and dimensional features. The results indicate the importance of the sample dimension in ML. Dutta, Kumar and Basu (2020) and Mudassir et al. (2020) delete features with Multicollinearity (MC) in the dataset. Dutta, Kumar and Basu (2020) identify MC in analyzing correlation coefficients. The results show that GRU performs better than LSTM and RNN to predict future Bitcoin prices. The work of Sebastião and Godinho (2021) is the only one that implements automated Hyperparameter paper implements Hyperparameter Tuning for DL algorithms and evaluates the final model performance in a bullish and bearish period, which will be investigated in this work.

3.3.3 Methodology

The architecture is represented in figure 19 and is intended to guide through the Bitcoin price and trend prediction modelling section. Section 3.3.3.1 describes the data and section 3.3.3.2 deals with the ML problem. In section 3.3.3.3, I subsample the dataset and scale it in 3.3.3.4. In section 3.3.3.5, feature selection is analyzed, and the algorithms are introduced in 3.3.3.6 and tuned in 3.3.3.7.

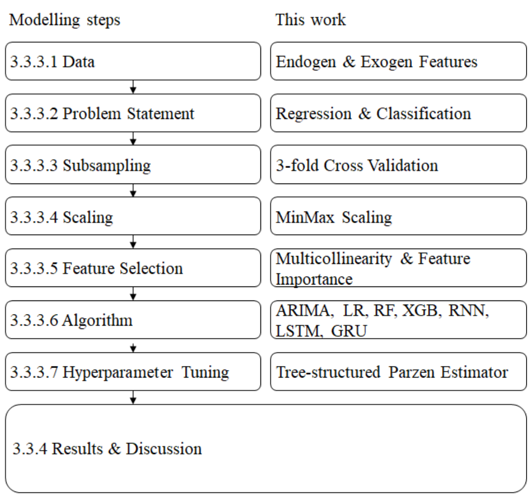


Figure 19: Modeling structure

3.3.3.1 Data

Solid and valid data input is a prerequisite for ML algorithms (Gupta, et al. 2021). The 5-minute interval-based database totals 8,893 observations, from 01-10-2021 00:05 to 31.10.2021 23:55 and is provided by Yahoo, Binance and Twitter Developer API. The dataset contains collected endogenous (Supply & Demand) and exogenous (Cryptocurrency Market, Political and Macro Financial) data which have already been studied in literature and generated features through feature engineering and sentiment analysis. The dataset comprises 134 features.

3.3.3.2 Problem Statement

The focus of this work is Bitcoin time-series price and trend prediction. Time-series is a set of data with successive moments in time. Time-series prediction is the prediction of the target´s future development by supervised time-series analysis (Mudassir, et al. 2020). Sebastião and Godinho (2021) found out that model assembling enables profitable trading strategies. I formulate the analysis as a Regression and Classification problem with three different prediction horizons *ph = {"one hour": 1h, "two hours": 2h, "three hours": 3h}*. The horizons are selected, aiming to take advantage of intraday trading and avoid exaggerated transaction costs. The prediction is evaluated each five minutes as it is the most granular level, we can collect a comprehensive number of features. The goal is to define the best performing model for each combination of problem and prediction horizon. The criteria to select the best performing model differs between Regression and Classification.

<u>Regression</u>

The target variable of the Regression analysis is the Bitcoin closing price (USDT) in *1h*, *2h* and *3h*. I use root mean square error (RMSE) as a performance metric for Regression algorithms. RMSE calculates the square root of the average prediction error and assesses the quality of a prediction. RMSE is the preferred Regression metric as it assigns high weights to large errors (Bohte, Rossini 2019). The objective is to identify the Regression model with the lowest RMSE value. Appendix 3 describes the RMSE formula.

Classification

The target variable of the Classification analysis is the trend of the Bitcoin price (USDT) BT = *{"down": 0, "equal": 1, "up": 2}* according to a 0.5% threshold in $1h$, $2h$ and $3h$. Appendix 4 shows the distribution of the Classification target variable for 1h, 2h and 3h. The Classification algorithms are intended to analyze the jumps of Bitcoin price. I use accuracy as a performance metric for Classification algorithms. Accuracy measures the fraction of right predictions (Chen, Li and Sun 2020). The goal is to select the Classification model with the highest accuracy (cf. Appendix 5).

3.3.3.3 Subsampling

ML relies on i) training data to build a model, on ii) validation data to tune the hyper-parameter and select the best performing model and on ii) test data to evaluate the model performance (Henckaerts, et al. 2021). The validation and test data are kept isolated during the training process to prevent data leakage (Borges and Neves 2020). Data leakage occurs when information that would not be known to that point in time is used to build a model (Hannun, Chuan and Laurens 2021). Time-series data must be given special attention as the data must be split into subsets according to the chronological order of observations (Borges and Neves 2020). Prior work splits the data into a three-subset logic: train, validation, and test data (Dutta, Kumar and Basu 2020, Sebastião and Godinho 2021). Only a few papers in prior research make use of Cross-Validation. Cocco, et al. (2021) use 3-fold Cross-Validation on a total dataset of 1,216 observations. K-fold Cross-Validation is a data split procedure that enables robust Model Selection (Bergmeier and Benítez 2012). Building several training and validation sets give an accurate representation of the model performance (Kuhn and Johnson 2013). K-fold Cross-

Validation splits training data into k groups to average the performance across all splits (Schafer 1993). I split the data into a train dataset, consisting of 7,893 observations (89% of the data) and a test dataset, comprising 1,000 observations (11% of the data). The train dataset starts from 01.10.2021 00:05 and represents a time-horizon of 27.5 days. The test dataset represents a time-horizon of 3.5 days. I use time-series Cross-Validation to build and validate the model in different time-horizons to account for time-specific movements (Ji, Kim and Im 2019). I split the train dataset into three subsets, each consisting of 1,000 validation observations (cf. Appendix 6). For each split, the train set increases in the number of observations to prevent data leakage. I execute Scaling and Feature Selection on data that is purely used for training.

3.3.3.4 Scaling

Feature Scaling enables the same range of values for each feature to guarantee stable convergence of weights and biases. Else, features with a wide range dominate other features (Borges and Neves 2020). The scaler trains on the subset of data which is purely used for training and transforms the entire dataset (Saxena and Sukumar 2018). Prior research normalizes data through MinMax scaling. Mudassir et al. (2020) demonstrate high prediction performance implementing the Minmax scaler. Minmax Scaling shifts the data between 0 and 1, maintaining the relative magnitude of outliers. The equation in Appendix 7 expresses the scaler. I use Minmax Scaling for the input features as well as the target variable.

3.3.3.5 Feature Selection

Feature Selection reduces high dimensionality of features to improve the generalization of prediction models (Niu, et al. 2020). Feature Selection is split into MC and FI. MC arises when features are correlated to each other and cause overfitting (Farrar and Glauber 1967). Mudassir, et al. (2020) used Variance Inflation Factor (VIF) which measures collinearity in a Multiple Regression model as well as cross correlations to identify interdependencies between features. FI methods can be categorized into i) filter-based, ii) wrapper-based and iii) embedded methods.

Dutta, Kumar and Basu (2020) implement a filter-based method and use statistical tests to identify the correlation of features with the target variable. Chen, Li and Sun (2020) as well as Mudassir, et al. (2020) implement a wrapper-based method and use different subsets of features to train models and keep features with the best results. Embedded-based methods use voting of multiple ensemble methods to identify useful features but are not used in literature (Niu, et al. 2020). I use MC for Random Forest (RF), Logistic Regression (LR), Extreme Gradient Boosted trees (XGB) and DL and FI for RF, LR and XGB (cf. Appendix 8). I measure the FI using the intersection of Filter (Correlation), Wrapper (RF) and Embedded (LR) based Feature Selection. Combining multiple Feature Selection methods will improve the model performance (Tsai and Hsiao 2010).

3.3.3.6 Algorithm

Established time-series prediction is based on statistical algorithms (Elsaraiti and Merabet 2021). Statistical models use historic data of the target variable to predict future developments (Mudassir et al. 2020). The focus of my work are ML algorithms. They incorporate non-linearity into prediction models for non-stationary financial time-series (Nakano, Takahashi and Takahashi 2018). DL is a subcategory of ML and is inspired by the structure and function of the brain called artificial neural networks (Lim and Zohren 2020). DL algorithms are an attractive alternative to existing ML time-series prediction models as they extract features from data and identify hidden nonlinear relationships without relying on econometric assumptions and human interaction (Chong, Han, and Park 2017). It turns out that DL algorithms can capture the non-stationary behavior of cryptocurrencies (Mudassir, et al. 2020). That coherence will be further investigated. This work focuses on comparing Statistical and ML algorithms (cf. Appendix 9). For ML algorithms, I use nonlinear RF and XGB which recently dominate applied ML competitions (Brownlee 2021). In addition, I consider DL algorithms: Recurrent Neural Network (RNN), Long Short-Term memory (LSTM) and Gated Recurring Unit (GRU) as they showed high performance in the work of Chen et al. (2020) and Dutta, Kumar and Basu (2020). Statistical algorithms are the benchmark of this work as they are effective for short-term price prediction and represent a challenging benchmark for ML models (Elsaraiti and Merabet 2021). Statistical algorithms are used to compare model evaluation and trading results. The Regression benchmark of this work is the Auto-Regressive Integrated Moving Average (ARIMA) algorithm. The Classification benchmark is the LR, which outperforms other algorithms in the work of Chen et al. (2020) and Sebastião and Godinho (2021). The following contains the implementation of different algorithms.

ARIMA is a combination of autoregressive (AR), integrated (I) and moving average (MA) (Ibrahim, et al. 2020). AR is a univariate method that models the relationship of a variable at a specified time with its previous values. MA models the error terms of a variable at a specified time with the error terms at a previous time. I is used to transform a non-stationary time-series into a stationary time-series (Munim, Shakil and Alon 2019). Appendix 10 represents the ARIMA equation.

LR is a multiple variate Regression method for Classification problems. LR estimates the probability of occurrence, not the target variable itself (Chen, Li and Sun 2020)). Appendix 11 represents the LR equation.

RF is an ensemble method that is built on decorrelated decision trees that are trained individually on random data subset (Breiman, Bagging Predictors 1996) (Breiman , Random Forests 2001). Ensemble methods consist of a collection of predictors to provide better prediction performance. Decision trees adopt a tree structure to recursively partition the feature space. The prediction result is average of each decision tree (Chen, Li and Sun 2020).

XGB is a decision-tree-based ensemble ML algorithm and is an improved version of a decision tree because each tree is approximated by Regression functions. XGB parallelizes the growth of gradient boosted trees in a forest and speeds up the time to grow trees. The prediction result is the weighted average of each decision tree (Chen, Li and Sun 2020).

RNN is a neural network suitable for time-series modelling (Bernal, Fok and Pidaparthi 2012). RNN uses internal state memory to persist information (Dutta, Kumar and Basu 2020). It consists of input, hidden and output layers. Each layer has multiple information processing units called neurons (Nakano, Takahashi and Takahashi 2018). The hidden layer allows information flow from one step to the next. A hidden state is a representation of previous inputs. When the length of the input sequence is too large, long-term information is not considered, which results in the vanish gradient problem (Ji, Kim and Im 2019). If the gradient shrinks to a small value, RNN fails to learn longer past sequences (Hochreiter 1998). Appendix 12 shows the RNN structure.

LSTM is an RNN architecture, designed to learn long-term dependencies (Hochreiter 1997). It addresses the vanishing gradient problems (Chung and Shin 2018). LSTM consists of several LSTM cells, composed of three gates: i) input, ii) forget and iii) output gate. Gates control the information flow in and out of the cell (Dutta, Kumar and Basu 2020). Each gate is based on a sigmoid layer and a point-wise multiplication operation which outputs a number between 0 and 1 and indicates how much information should be passed. LSTMs can decide to remove or add information to the LSTM cell state (Staudemeyer and Morris 2019). Appendix 13 shows the LSTM structure.

GRU is similar to LSTM but combines the forget and input gates into one single update gate and merges the cell state and hidden state (Dutta, Kumar and Basu 2020). GRU is a simpler version of LSTM and uses fewer parameters and is faster to train than the LSTM (Ji, Kim and Im 2019). GRU comprises an update and reset gate as well as current memory content (Dutta, Kumar and Basu 2020). The reset gate determines the amount of previous state data used with current input data. The update gate determines the amount of data collected from the previous state (Phaladisailoed and Numnonda 2018). Appendix 14 shows the GRU structure.

3.3.3.7 Hyperparameter Tuning

Hyperparameter Tuning optimizes the model performance by defining hyperparameter. Hyperparameter are algorithmic parameters for initialization prior to training the algorithm. Algorithms are trained with Hyperparameter combinations and tested on the validation set to measure performance according to the loss metric. The Hyperparameter with the optimized loss metric will be chosen for model deployment (Borges and Neves 2020). Grid search is a widely used Hyperparameter optimization algorithm (Xiaolei, Mingxi, and Zeqian 2020). It searches through a specified range of values to find the best Hyperparameter (Raschka 2015). Sebastião and Godinho use grid search to find best Hyperparameter for ML algorithms (Sebastião and Godinho 2021). Prior research implements DL Hyperparameter Tuning using manual search. Coco, Tonelli and Marchesi build “for loops” to test Hyperparameter for ANN, LSTM and BNN. Manual Hyperparameter Tuning is limited in terms of the number of Hyperparameter and range of specified values (Cocco, Tonelli, and Marchesi 2021). I will use Tree-structured Parzen Estimator (TPE), Hyperparameter Tuning for DL algorithms and grid search for ML algorithms. TPE is a sequential model-based optimization approach (SMBO). SMBO builds models to approximate the performance of Hyperparameter based on the observed performance and chooses new Hyperparameter accordingly (Bergstra, et al. 2011). TPE is designed to provide the best possible latency improvement and is the state of the art in terms of Hyperparameter Tuning to allow higher model performance (Claesen, Simm and Popovic 2014). Appendices 15 and 16 contain the grids used for each hyper-parameter of each learning algorithm.

3.3.4 Results and Discussion

This section presents the results of the price (Regression) and trend (Classification) prediction. First, an overview of the Validation results is provided. For evaluation, the best performing Regression and Classification model is selected for *1h*, *2h* and *3h*. Second, I will discuss the evaluation results. Appendix 17 shows the detected features of MC. Appendices 18-23 display the selected features for LR, RF and XGB. Appendices 24-25 contain the 3-fold Cross-Validation results and best Hyperparameter for Regression and Classification respectively.

3.3.4.1 Validation

<u>Regression</u>

The LSTM models outperform the other ML models and show the lowest predictions errors for *1h*, *2h* and *3h* (cf. table 13*)*. RF *3h* Bitcoin price prediction exhibits the highest error value for the Regression analysis with an RMSE of 0.15366. LSTM shows the lowest error measure for *1h* Bitcoin price predictions with an RMSE of 0.03875. DL algorithms in general show lower RMSE values than the RF and XGB. The ARIMA benchmark shows comparatively but slightly worse RMSE values than the LSTM. My findings are in line with the research by Ji, Kim and Im (2019), who prove the predictive performance of LSTM compared to other ML algorithms. Dutta, Kumar and Basu (2020) find that the GRU shows a higher prediction performance than the LSTM for daily Bitcoin prediction. For intraday trading, we can disprove this statement and find that the LSTM has a higher prediction performance than the GRU.

Table 13: Validation Results Regression (RMSE)

Horizon	ARIMA	RF	XGB	RNN	LSTM	GRU
1h	0.03946	0.13123	0.10691	0.11939	**0.03875**	0.05756
2h	0.05923	0.12336	0.10308	0.07449	**0.05656**	0.0733
3h	0.06381	0.15366	0.13779	0.06693	**0.05959**	0.05998

<u>Classification</u>

GRU performed best for *1h* prediction, LSTM performed best for *2h* prediction and RNN performed best for *3h* prediction (cf. table 14). LR *3h* Bitcoin trend prediction exhibits the lowest accuracy (36%) for the Classification analysis. GRU exhibits the highest accuracy (73%) for *1h* Bitcoin trend prediction. DL algorithms outperform RF and XGB again for all algorithms and all prediction horizons. The accuracy decreases with an increasing prediction horizon for all algorithms except for XGB, as the class distribution of the target variable converges (cf. Appendix 4). Making the right decisions becomes more difficult, the longer the prediction horizon. GRU, LSTM and RNN outperform the LR benchmark for *1h*, *2h* and *3h*. The DL models show high prediction performance and a higher accuracy than the statistical benchmark model for *1h*, *2h* and *3h*. The results disprove the statements of Chen et al. (2019) that statistical models like a LR outperform complex ML algorithms for Classification (Chen, Li and Sun 2020).

Table 14: Validation Results Classification (Accuracy)

Horizon	LR	RF	XGB	RNN	LSTM	GRU
1h	0.63667	0.67667	0.608	0.71849	0.71746	**0.72746**
2h	0.41067	0.47867	0.356	0.54477	**0.5544**	0.54891
3h	0.36367	0.43967	0.36733	**0.49637**	0.47461	0.48508

3.3.4.2 Evaluation

We refit LSTM (*1h, 2h* and *3h*) for Regression and GRU (*1h*), LSTM (*2h*) and RNN (*3h*) for Classification on the training and validation dataset so that the ML models can use the latest observations up to the test period (Borges and Neves 2020). To avoid biased results, I evaluate the final models on the test dataset, as well as on bearish and bullish time sections of the test dataset. Bearish and bullish markets can be categorized according to the market trend (Cohen, Zinbarg and Zeikel 2014). The bullish period lasts from 29.10.2021 8:05 to 29.10.2021 16:35, containing 103 observations and is characterized by a Bitcoin USDT growth of 3.55%. The bearish period lasts from 31.10.2021 1:10 to 31.10.2021 12:45, containing 140 observations and is characterized by a Bitcoin USDT decline of 3.43%.

Regression

Figure 20 and Appendices 26-27 show the true values as well as the ARIMA and LSTM Bitcoin USDT Price prediction for *1h*, *2h* and *3h* respectively.

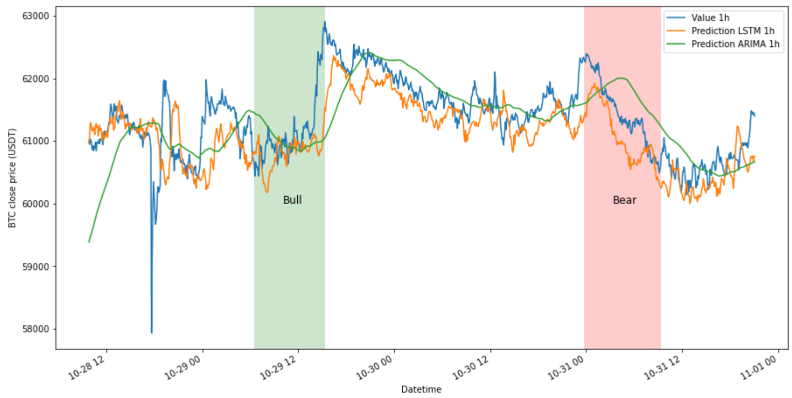


Figure 20: Plot Evaluation Results Regression 1h

Table 15 shows the evaluation results of the refitted LSTM model for *1h*, *2h* and *3h*. The LSTM model outperforms the ARIMA benchmark model for all time horizons. The RMSE value of the LSTM in the bullish and bearish period is lower than the RMSE value of the ARIMA, except for the *3h* prediction horizon. The ARIMA (0.05516) in the bullish period performs better than the LSTM (0.05821), which indicates further optimization opportunities. The LSTM demonstrates the lowest values in the bearish period compared to the whole test data and the bullish period. The results show high predictive performance of the regression LSTM model especially in bearish market conditions. The high predictive model performance indicates model properties for a prosperous trading strategy.

Table 15: Evaluation Results Regression (RMSE)

Horizon	**All ARIMA**	**All Model**	**Bull ARIMA**	**Bull Model**	**Bear ARIMA**	**Bear Model**
1h	0.02711	0.02521 (LSTM)	0.03975	0.03381 (LSTM)	0.02628	0.02032 (LSTM)
2h	0.03039	0.02909 (LSTM)	0.04835	0.04177 (LSTM)	0.03177	0.01771 (LSTM)
3h	0.03385	0.03183 (LSTM)	0.05516	0.05821 (LSTM)	0.03763	0.01195 (LSTM)

Classification

Table 16 shows the evaluation results of the refitted GRU (*1h*), LSTM (*2h*) and RNN (*3h*) models. Appendices 28-45 show the confusion matrix for LR and GRU (*1h*), LSTM (*2h*) and RNN (*3h*). The GRU (*1h*) and RNN (*3h*) model outperform the LR benchmark model. The LSTM (*2h*) shows worse performance than the LR benchmark model. All DL models show higher performance in the bearish period than in the bullish period or in the whole dataset. The results indicate optimized ML models for Bitcoin trend prediction for *1h* and *3h* and further need for investigation for Bitcoin trend prediction for *2h*.

Table 16: Evaluation Results Classification (Accuracy)

Horizon	All LR	All Model	Bull LR	Bull Model	Bear LR	Bear Model
1h	0.627	0.712 (GRU)	0.4466	0.64078 (GRU)	0.74286	0.75 (GRU)
2h	0.552	0.534 (LSTM)	0.48544	0.36893 (LSTM)	0.47143	0.55 (LSTM)
3h	0.395	0.438 (RNN)	0.35922	0.37864 (RNN)	0.33571	0.45 (RNN)

3.3.5 Conclusion and Future Work

Several ML Regression and Classification models for Bitcoin price and trend prediction were compared to the ARIMA and LR benchmark models using Data Sampling, Scaling, Feature Selection and Hyperparameter Tuning. The results indicate that LSTM yields the best prediction performance for intraday Bitcoin price prediction. The DL models GRU, LSTM and RNN demonstrate the best performance for Bitcoin trend prediction in *1h*, *2h* and *3h*, respectively. The high predictive performance of Regression and Classification models build the basis for a profitable trading strategy. The optimal modelling design is characterized by endogen (Supply & Demand) and exogen features (Crypto market, Macro Financial, Political and Sentiment) for Regression and Classification, 3-fold Cross-Validation for subsampling, MinMax Scaling, Feature Selection through Multicollinearity detection, TPE Hyperparameter Tuning to outperform statistical and other ML models.

Further research can extend the work by implementing Online Learning and resampling. Online learning is a promising technique for learning from continuous data streams to increase model performance. A resampling method could enhance the predictive performance of Classification models in learning feature pattern for bearish and bullish periods.

3.4 Study III: Trading Strategy

3.4.1 Introduction

Algorithmic trading is defined as any form of trading using algorithms to automate all or most parts of trading (Lu, et al. 2020). Driven by the rise in computing power and the emergence of artificial intelligence (AI) in the financial industry, algorithmic trading is receiving more and more attention, not only by institutional investors but also by retail traders (Mordor Intelligence 2020). According to a study of *Mordor Intelligence* (2020)*,* algorithmic trading accounts for around 60-73% of the overall United States equity trading and they expect the algorithmic trading market to witness a CAGR of 11% from 2021 to 2026 (Mordor Intelligence 2020).

Extensive computing power creates an informational advantage in terms of capacity to receive and process information and enables the development and use of Machine Learning (ML) for trading algorithms that have proven to be able to outperform human traders (Lu, et al. 2020) (Petukhina, Reule and Härdle 2021). Predictive analysis represents only one part of automated algorithmic trading. Predictions that are derived from ML algorithms need to be translated into a set of rules to generate trading decisions, i.e., a trading strategy. Past research focuses on the development of ML algorithms to predict future prices and price trends and derives financial performance from simple strategies of individual models.

The purpose of this work is to expand prior research and to answer the following question:

How to translate multiple model predictions into an algorithmic trading strategy?

I use the output of multiple price and trend prediction models as developed and described in this work to generate trading strategies for algorithmic trading of Bitcoin. Trading strategies are analyzed and evaluated with the aim to democratize algorithmic trading, i.e., making it accessible for retail traders. The main contributions of this work are the analysis of ensemble trading strategies that combine different prediction horizons. The findings indicate that ensemble strategies improve the performance of individual strategies and outperform benchmark strategies.

This work is divided into five major sections. The 2nd section provides an overview of previous research in the field of trading strategies derived from predictive analysis. In the 3rd section, I outline the methodology to build and test trading strategies. In section 4, results of the trading strategies are compared and evaluated. In the last section, I conclude this work and give an outlook for future research opportunities.

3.4.2 Related Work

Fang, et al. (2020) show that 85% of the research paper for cryptocurrency trading have appeared since 2018, illustrating the topicality of the subject. ML represents an important aspect of this research as ML algorithms can discover relationships between data inputs that are difficult or impossible to observe by humans (Fang, et al. 2020). Research has proven the applicability of ML algorithms to predict cryptocurrency prices or price trends (Nakano, Takahashi and Takahashi 2018). Being a valuable technique for algorithmic trading, ML represents only the first step on the way to automatic trading of cryptocurrencies. A set of rules needs to be defined to translate ML predictions into actual trading signals (Atsalakis, et al. 2019). Accurate predictions of ML models are not necessarily a guarantee for the generation of high profits (Chevallier, Zhu, and Zhang 2021).

In previous research on ML algorithms for cryptocurrency trading, the objective was to optimize prediction results of the respective models. While prediction results are the basis for trading algorithms and essential for the performance of automated trading, a trading strategy that translates predictions into trading signals is equally important (Chevallier, Zhu, and Zhang 2021). Researchers use different trading strategies to generate trading actions and evaluate the economic performance of their ML models. While there exists research on portfolio strategies trading multiple assets simultaneously, I focus on one-asset strategies that define trading rules for one asset only. From the literature examined in this work, research paper that consider a one-asset trading strategy and are published no later than 2015 are considered. An overview of these papers is provided in Appendix 46.

The strategies reviewed for this work can be allocated to two major categories: Buy and Sell (BS) strategies and Long and Short (LS) strategies. BS strategies consider the creation of long positions and trigger trading signals that lead to the purchase or sale of an asset (Sebastião and Godinho 2021). As capital is not unlimited available and assets can only be sold if holding a long position for the LS strategy, some rules need to be defined for the translation of trading signals. Borges and Neves (2020) and Dutta, Kumar and Basu (2020) for example define an initial portfolio value that is either fully invested or fully devested in the asset. A sell signal triggers the sale of the entire portfolio, and a buy signal triggers the investment in the asset using all available resources (Borges and Neves 2020) (Dutta, Kumar and Basu 2020). LS strategies establish both long and short positions, where a buy signal will lead to buying the asset and a sell signal to short selling the asset (Dutta, Kumar and Basu 2020). Short selling speculates on the decline of the asset and means selling an asset that is not owned by the seller. The short seller borrows money to sell the asset and must repurchase the respective asset at some point, i.e., close the position (Nakano, Takahashi and Takahashi 2018). The amount that can be borrowed for short selling is usually limited by the amount of capital the trader holds and is associated with additional trading costs and legal regulations (Garcia and Schweitzer 2015). LS strategies must define when the repurchase of the asset will be undertaken. Dutta, Kumar and Basu (2020) use daily trading signals and cover short positions on the end of the day.

Three of the identified papers introduce a trading strategy that uses the output of multiple models. Chu, Chan and Zhang (2020) use an Exponential Moving Average (EMA) crossover strategy with different time periods to calculate EMA. The final trading signal is generated by taking the average of the individual signals triggered by the different time periods (Chu, Chan, and Zhang 2020). Borges and Neves (2020) use four different ML algorithms (Logistic Regression, Random Forest, Gradient Tree, Support Vector) to predict cryptocurrency prices and price trends. They compare the performance of trading signals derived from individual model signals to the performance of an ensemble model that takes the unweighted average of the individual signals, concluding that the ensemble model produces the best results (Borges and Neves 2020). Sebastião and Godinho (2021) use six different learning algorithms (Linear, Linear binary, Random Forest, Random Forest binary, Support Vector Machine, Support Vector Machine binary) to implement three model ensembles, where a minimum of four, five or six algorithms must agree on the trading signal to be translated into a final trading signal (Sebastião and Godinho 2021). They conclude that the trading strategies derived from the model ensembles significantly outperform the market (Sebastião and Godinho 2021).

3.4.3 Methodology

3.4.3.1 Problem Statement

Rational traders make unemotional trading decisions and base trading actions of any financial asset on the present value of all future cash flows (Ahn and Kim 2021). Ahn and Kim (2021) conduct a sentiment analysis to capture emotional factors and find evidence for emotional trading in the cryptocurrency market. Emotions can lead to unprofitable trading decisions and emotional investing is a main reason why many retail traders are buying at peaks and selling during the valleys of a cycle (Ahn and Kim 2021). Defining a trading strategy that is making trading decisions based on prespecified rules takes away the emotional component of investing. Although there is extensive information regarding trading strategies and trading algorithms available to the public, the development of a trading strategy based on advanced analytics is complicated and time consuming (Chan 2021). Retail traders often do not have the time or the resources to build their own algorithmic trading strategy.

The goal of this work is to translate the outputs of multiple predictive models into an algorithmic trading strategy for retail traders. I define algorithmic trading strategies for Bitcoin using different techniques to combine predictions from Regression and Classification models with three different prediction horizons *ph = {"one hour": 1h, "two hours": 2h, "three hours": 3h}*. The prediction horizons are selected to take advantage of intraday trading and avoid exaggerated transaction costs. Predictions are evaluated every five minutes. The characteristics of the trading strategies are selected to provide an algorithmic trading design that is understood by retail traders. This paper elaborates an example to translate multiple model predictions into ensemble strategies. The entire process from the output of predictive analysis to the translation into trading strategies, including the economic evaluation of the strategies, is included in this work, and visualized in figure 21.

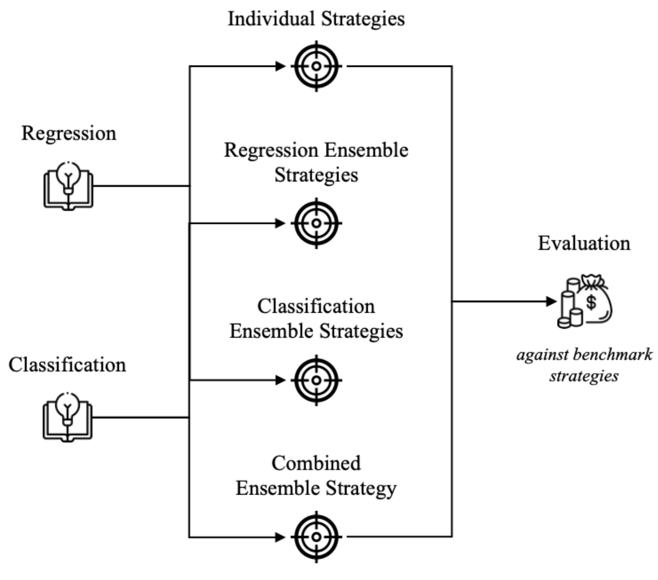


Figure 21: Trading Strategies overview

3.4.3.2 Data

Model predictions build the basis to develop trading strategies that earn superior returns to its benchmarks. In addition to using different ML algorithms for Classification and Regression problems, prior research on algorithmic trading with predictive analysis differs on prediction frequency and prediction horizon. Prediction frequencies depend on the frequency of input data that is collected and used. Atsalakis et al. (2019), Dutta, Kumar and Basu (2020) and Sebastião and Godinho (2021) make daily predictions, while Borges and Neves (2020) and Shintate and Pichl (2019) make predictions every minute. Vo and Yost-Bremm (2020) use a RF algorithm and compare trading intervals of 5-, 15- and 360-minutes. They find that models with a 5- and 15-minute trading frequency have significantly better overall results than the model with a trading frequency of 360 minutes (Vo and Yost-Bremm 2018).

The models that are used in this work predict future prices or future price trends of Bitcoin on a frequency of five minutes for a prediction horizon of *1h*, *2h* and *3h*. LSTM is identified to be the best Regression algorithm for all prediction horizons. GRU, LSTM and RNN are identified to be the best Classification algorithms for *1h, 2h* and *3h* predictions, respectively. In accordance with the evaluation of modeling performance, we include ARIMA and LogReg as benchmark models for economic performance. The models are trained during the period from 01.10. 0:05 to 28.10 09:45, consisting of 7,893 observations and evaluated in the test period from 28.10. 9:50 to 31.10 21:05, consisting of 1,000 observations. The economic evaluation of the strategies in this work is based on the predictions in the test period to avoid a look-ahead bias. A look-ahead bias refers to the translation of trading decisions from information, that was not available at the time (Chan 2021). The evaluation is based on 5-minute instances where a trading signal is triggered at every instance. The evaluation period consists of 1,000 instances, corresponding to a total period of 3.43 days.

Predictions of Classification models (i.e., LogReg, GRU, LSTM, RNN) are defined as trend signals *s = {"down": -1, "equal": 0, "up": 1}*. Regression models (i.e., ARIMA, LSTM) predict future prices of Bitcoin. Price predictions are transformed into trend signals, where an up or down trend is recognized when the predicted price difference, in absolute terms, is higher than the transaction fee associated with this trade, i.e., 0.1% of the current Bitcoin price. The trend signals derived from the Regression models predict a down trend for more than 60% of all instances, while the trend signals from the Classification models predict an equal trend for more than 75% of all instances. This is in accordance with the higher threshold for up and down trend signals for Classification (0.1%). A summary of the trend signals derived from the model predictions is shown in table 17.

Table 17: Overview of trend signals from predictive analysis

	down [in %]	equal [in %]	up [in %]
reg_1h	80.2	10.2	9.6
reg_2h	62.0	12.9	25.1
reg_3h	76.8	12.0	11.2
class_1h	1.3	95.0	3.7
class_2h	8.4	84.3	7.3
class_3h	15.5	76.3	8.2

3.4.3.3 Trading Strategies

A trading strategy is a set of rules that define the position on the market or trigger a trading action (Sebastião and Godinho 2021). Trading strategies are necessary to recommend or execute trading decisions and to analyze financial performance (Atsalakis, et al. 2019). According to the Efficient Market Hypothesis (EMH), it is impossible to make a profit by predicting prices. However, prior research has demonstrated that superior returns can be generated from algorithmic trading strategies that are derived from predictive analysis (Nakano, Takahashi and Takahashi 2018) (Sebastião and Godinho 2021).

For this work, trading signals *ts = {"sell": -1, "hold (position)": 0, "buy": 1}* are triggered every five minutes. I use a BS strategy, where a buy signal triggers an investment in Bitcoin, if no Bitcoin is held at that time, and a sell signal triggers the sale of the current position, if a current investment position is held. The trading signals of each trading strategy are visualized in Appendix 47. The BS strategy is chosen over the LS strategy because this paper intends to provide retail traders with a comprehensible strategy. LS strategies are more difficult to understand because short selling is bound to additional trading costs and legal regulations (Garcia and Schweitzer, 2015). In previous research, trading strategies are derived from individual prediction models, with only some paper combining predictions of multiple models to generate trading signals. The results in the literature indicate that ensemble strategies, which combine the output of multiple models, generate superior financial performance to strategies that use predictions of one model (Borges and Neves 2020) (Chu, Chan, and Zhang 2020) (Sebastião and Godinho 2021). In this work, I combine predictions of different algorithms and different prediction horizons, i.e., *1h, 2h* and *3h*. I economically evaluate the performance of seven ensemble trading strategies, that combine the predictions of Classification and Regression models for three different prediction horizons. For preparation of the ensemble trading strategies, price predictions from Regression models are translated into up and down trend signals *s = {"down": -1, "up": 1}* if the difference to the previous instance, in absolute terms, is higher than the transaction costs associated with this trade, i.e., *0.1%*. Otherwise, the price prediction is translated into an equal trend signal *s = {"equal": 0}*.

The ensemble strategies consist of seven strategies. The first six strategies combine prediction horizons of *1h*, *2h* and *3h* of one learning problem, either Regression or Classification. The strategies, i.e., *tendency*, *majority*, and *consensus*, require different levels of agreement between the signals of the individual models. Considering the signals *s = {"down": -1, "equal": 0, "up": 1}* from individual models, a trading signal is generated when the sum of the individual signals is at least one, two or three, in absolute terms, for the *tendency, majority* and *consensus* strategy, respectively. The *tendency* strategy is considered the riskiest because it follows the signal of one model when the other two models suggest holding the position, or when two models suggest the same direction while the third model suggests the opposite trend. The *consensus* strategy only generates a trading signal if all individual models agree on the future up or down trend. The trading strategies *tendency, majority,* and *consensus* are visualized in Appendix 47. The last ensemble strategy *ensemble_best* combines the best performing regression model with the best performing Classification model. The strategy is structured in two layers. Trading actions follow the signals of the Regression model (first layer) but are overruled by up or down trend signals of the Classification model (second layer).

3.4.3.4 Back-testing

Back-testing is a procedure where an investment strategy is tested on past data to see how it would have performed. Historical trades are created in accordance with the respective investment strategy, using information that was available at each trigger point in the past (Chan 2021). Two factors are very important to receive reliable results when implementing a back-test: data leakage and test metrics. Data leakage, also called Look-Ahead Bias, refers to the usage of data that was not available at the time. This leads to inflated performance of investment strategies that will not be obtained when using the respective strategy for real-time trading (Chan 2021). The selection of test metrics is important when combining different investment strategies because many evaluation metrics have characteristics that falsely benefit trading strategies or time periods (Nakano, Takahashi and Takahashi 2018).

In the observed research paper, Portfolio Value (PV), Return on Investment (ROI), Sharpe Ratio (SR) and Maximum Drawdown (MDD) are the most common metrics used for back-testing of algorithmic trading strategies. PV shows changes in portfolio value over time and is well suited to graphically compare the development of different algorithmic trading strategies against each other or against the underlying asset (Dutta, Kumar and Basu 2020). ROI, which is closely related to the PV, is calculated as the net gain from the trading strategy against the initial investment and is used to show the financial performance over a specific time horizon, in relative terms (Atsalakis, et al. 2019). PV and ROI show the performance over a specific period, but do not consider volatility, i.e., risk. The SR is a risk adjusted metric, that compares the generated returns to the standard deviation of these returns (Sebastião and Godinho 2021). MDD is the largest observed loss from a peak in the PV of the portfolio and is a measure of downside risk as it shows how much the PV has dropped from a peak in the observed period (Nakano, Takahashi and Takahashi 2018). It represents an important metric, because a major drawdown, i.e., the difference between the current PV and the global maximum of the PV, is an emotional sensitive scenario for any trader (Chan 2021).

In this work, I use PV, ROI, SR and MDD as the most common metrics to analyze and evaluate different trading strategies. The formulas of these metrics are shown in Appendix 48. I use the SR as the primary metric to evaluate the trading strategies because it represents a risk adjusted return and therefore is more comparable for different periods in time (Chan 2021). I also report the *Number of trades, Periods invested* and *Profitable positions* to present a general overview of the trading actions. Following the EMH, it is futile to make profits by predicting prices. I compare the metrics to the *buy_and_hold* strategy, where an investment in Bitcoin is made in the first instance of the test period and held until the end (Atsalakis, et al. 2019). In addition, simple statistical prediction algorithms are used as a benchmark. While institutional investors hire well educated people to develop complicated trading algorithms, *Chan* (2021) states that simple algorithms can generate returns equal to or even higher than complex algorithms (Chan 2021). Therefore, I benchmark the proposed strategies against trading strategies entirely derived from signals generated from an ARIMA and a L model, which are time-series prediction models.

3.4.4 Results and Discussion

In this section, the results obtained from the described trading strategies are reported. First, an overview of the results from the trading strategies is provided. The results are compared to benchmark strategies. Finally, I discuss the results.

Aligned with evaluations in previous work, results were obtained through back-testing and are subject to limitations of this technique as described in this work. Nevertheless, the design of this work is specifically built to reproduce a realistic environment as closely as possible to extrapolate its results into an actual active trading system. According to Statista (2021), Binance is the largest exchange platform for cryptocurrencies, accounting for a 24-hour trading volume of $31.79 billion on November 22, 2021. Settings are chosen to represent a reflection of the parameters on the Binance platform. This allows for an evaluation of results that could have been achieved in real-life. Performance metrics are calculated using a start balance of 10,000 USDT, which is equal to around $10,000. Seeking to develop trading strategies for retail traders, a 30-day trading volume lower than 50 BTC is assumed, which results in a transaction fee of 0.1% on Binance (Binance 2021). No taxes are included in the calculation because gains on cryptocurrency trading are not subject to taxes in Portugal (Cointaxlist 2021). Results are calculated for a period of 3.43 days, from 28.10.2021 9:50 to 31.10.2021 21:05. During this time Bitcoin achieved a total return of 0.13% and had 46.9% profitable positions, representing a balanced movement of up and down trends.

Table 18: Overall results of trading strategies during the test period

Strategies	PV [USDT]	ROI [in %]	Sharpe Ratio	MDD [in %]	Number of trades	Periods invested [in %]	Profitable positions [in %]
buy_and_hold	10,013	0.13	0.58	5.96	0	100.0	46.9
ARIMA_1h	9,844	-1.56	-1.87	5.51	43	54.3	71.3
ARIMA_2h	9,824	-1.76	-2.19	5.51	43	54.4	71.2
ARIMA_3h	9,817	-1.83	-2.29	5.56	43	54.5	71.1
LogReg_1h	9,978	-0.22	0.11	5.96	1	97.6	48.3
LogReg_2h	10,056	0.56	1.19	5.96	1	33.9	81.3
LogReg_3h	10,076	0.76	1.45	5.96	27	52.5	71.2
reg_1h	9,895	-1.05	-1.50	5.57	37	13.4	91.5
reg_2h	10,016	0.16	0.74	5.53	43	28.5	83.3
reg_3h	10,440	4.40	12.17	2.08	39	15.5	90.2
class_1h	9,811	-1.89	-3.13	5.96	3	10.0	**94.7**
class_2h	10,176	1.76	3.05	5.96	3	25.2	86.7
class_3h	10,332	3.32	5.39	5.96	5	27.3	85.1
ARIMA_tendency	9,817	-1.83	-2.29	5.56	43	54.5	71.1
ARIMA_majority	9,824	-1.76	-2.19	5.51	43	54.4	71.2
ARIMA_consensus	9,844	-1.56	-1.87	5.51	43	54.3	71.3
LogReg_tendency	10,105	1.05	1.86	5.96	25	52.6	71.4
LogReg_majority	10,327	3.27	4.90	5.96	7	56.5	70.7
LogReg_consensus	10,013	0.13	0.58	5.96	0	100.0	46.9
reg_tendency	10,019	0.19	0.80	4.97	47	14.8	90.5
reg_majority	10,722	7.22	17.76	**1.71**	25	15.5	90.9
reg_consensus	10,732	7.32	**18.91**	**1.71**	9	11.5	93.7
class_tendency	10,027	0.27	0.76	5.96	7	23.9	86.7
class_majority	10,176	1.76	3.05	5.96	3	25.2	86.7
class_consensus	10,013	0.13	0.58	5.96	0	100.0	46.9
ensemble_best	**10,763**	**7.63**	18.90	2.15	9	15.5	91.4

The results obtained from the trading strategies are shown in table 18. The development of the PV of the trading strategies, including the triggered trading actions are visualized in Appendix 49. In general, strategies derived from Regression models have a higher trading frequency, i.e., a higher number of trades, than strategies derived from Classification strategies. This is consistent with a more conservative approach for Classification models that have a higher defined threshold (0.5%) to trigger an up or down signal than the Regression models (0.1%).

Individual strategies, i.e., strategies derived from the predictions of one model alone, that are based on models with a prediction horizon of *1h* have a negative return of -1.05% and -1.89% for Regression and Classification, respectively. The other individual strategies derived from predictive analysis generate positive returns and outperform *buy_and_hold*, *ARIMA* and *LogReg* as their benchmark strategies. The performance of individual strategies increases with the prediction horizon. Using model predictions with a prediction horizon of *3h* yields the highest ROI among the individual strategies with a ROI of 4.40% for Regression and 3.32% for Classification, compared to a ROI of 0.13% for the *buy_and_hold* strategy. The individual strategies based on models with a prediction horizon of *3h* also have the highest SR (*reg_3h:* 12.17; *class_3h:* 5.39) and, for Regression, the lowest MDD (*reg_3h:* 2.08%) among the individual strategies and outperform the *buy_and_hold* strategy that has a SR of 0.58 and a MDD of 5.96%. Considering the substantial difference in performance between the individual strategies, and the finding in previous work that model assembling improves the profitability, we define ensemble strategies (Ji, Kim and Im 2019) (Borges and Neves 2020) (Sebastião and Godinho 2021). While previous research combines models of different ML algorithms, we extend this research by assembling models from the same algorithm with different prediction horizons. The effect of model assembling on the performance of strategies differs between Regression and Classification.

Combining Regression models to an ensemble strategy increases the performance if at least a majority voting is defined. The Classification ensemble strategies do not increase the performance of the individual strategies. The Regression ensemble strategies *reg_majority* and *reg_consensus* achieve a ROI of 7.22% and 7.32%, and a SR of 17.76 and 18.91, respectively. They outperform *buy_and_hold* (ROI: 0.13%; SR: 0.58) as their benchmark strategy. *ARIMA_consensus* represents the model among the ARIMA models, but none of the ARIMA models earns positive returns during the test period. The strategies reg*_majority* and *reg_consensus* have a better risk performance which is observed in a lower MDD. This results in better risk adjusted returns, i.e., an increase in the SR (Nakano, Takahashi and Takahashi 2018). While the performance of individual strategies derived from Classification models is similar to strategies derived from Regression models, the performance of the ensemble strategies is significantly worse. The ensemble strategies derived from the classification models do not increase the performance. The best performing Classification ensemble strategy *class_majority* has an ROI of 1.76% and a SR of 3.05, identical to the individual strategy *class_2h*, but worse than the individual strategy *class_3h*. Individual and ensemble strategies of Classification models are not able to decrease the MDD of the buy-and-hold strategy (5.96%). The ensemble strategy *ensemble_best* combines the best performing Regression strategy (i.e., *reg_consensus)* and the best performing Classification strategy (i.e., *class_3h*). The *ensemble_best* strategy has a higher ROI than its assembled strategies, representing the highest ROI (7.63%) among all strategies, but it has a lower SR (-0.01) and a higher MDD (+0.44) than the *reg_consensus* strategy. The developments of the PV for the *ensemble_best* strategy, the best performing ensemble strategy from Regression (*reg_consensus)* and Classification (*class_consensus)*, the best performing individual Regression (*reg_3h)* and Classification (*class*_3h) strategy, the best performing LogReg (*LogReg_majority)* and ARIMA (*ARIMA_consensus)* strategy as well as for the *buy_and_hold* strategy, are shown in figure 22. The figure shows that the high performance of the strategies *reg_3h, reg_consensus,* and *ensemble_best* is substantially influenced by selling the Bitcoin position when the Bitcoin price experience the first major drawdown. This finding is supported by the significantly lower MDD of these three strategies as shown in table 18.

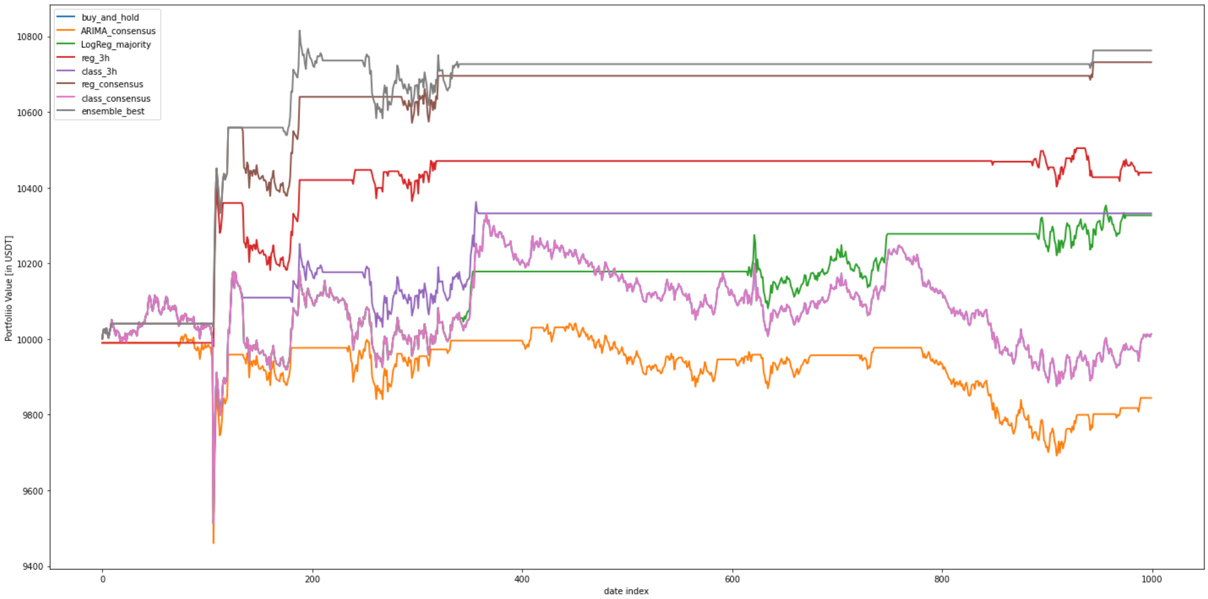


Figure 22: PV development of best performing Regression and Classification strategies

Overall, the ensemble strategies *reg_consensus* that combines predictions of Regression models with different prediction horizons and the *ensemble_best* strategy that combines the best performing Regression and Classification strategy have the best overall results, with similar performance metrics. The strategy *reg_consensus* has the highest SR (18.91%) and the lowest MDD (1.71). The *ensemble_best strategy* has the highest ROI (7.63%) and the highest final PV (10,763 USDT). Both strategies trigger nine trades during the test period. Both, *reg_consensus* and *ensemble_best,* are mostly not invested in Bitcoin with periods invested of 11.5% and 15.5%, respectively.

The results presented in this work are based on a BS strategy, which represents only one possibility to trade on the cryptocurrency market. The reported profitability measures are subject to the specifics of a BS strategy. Implementing a LS strategy with the possibility to create short positions might affect the results of the trading strategies differently. Considering additional financial products like Bitcoin futures, where short positions are easily created and transaction costs lower, might also lead to better results (Sebastião and Godinho 2021).

3.4.5 Conclusion and Future Work

Several trading strategies for Bitcoin that combine the predictions of ML algorithms for Regression and Classification problems were described and their profitability compared to benchmark strategies.

The results obtained in this work indicate that, in general, trading strategies that are derived from predictive analysis outperform the *buy_and_hold* strategy, i.e., beat the market. Regression ensemble strategies that combine the predictions of Regression models with different prediction horizons have the best overall results, if at least a majority voting is implemented. During the observed test period, they outperform other trading strategies in all performance metrics. Given the limitations of using back-testing to evaluate the trading strategies, our research proved a superior profitability of ensemble trading strategies to the market in the highly speculative environment of cryptocurrencies.

Direct extensions to this work could include looking into a wider range of financial products. For example, the creation of short positions or futures trading could be included in trading strategies. Further, additional cryptocurrencies could be added to create portfolio strategies.

4 Results and Discussion

In this section, we describe the findings from the simulation of a real time Bitcoin trading algorithm. First, we provide an overview of the generated insights regarding the influence of twitter sentiment on the Bitcoin price (Study I), the optimal modeling design for Bitcoin price and trend prediction (Study II) and the best performing trading strategy derived from the predictive analysis (Study III). Second, we evaluate the profitability of the trading strategy including costs associated with running a real time trading algorithm. Finally, we discuss feasibility of implementation and limitations of financial evaluation.

We introduce a real time Bitcoin trading algorithm that covers the entire process from Data Collection to the translation of trading signals, analyzing the influence of Twitter, the modeling design and trading strategies. We investigate that *Overall Sentiment* and *VIP Sentiment* for Twitter have a Granger causal relationship to the Bitcoin price. We find that LSTM yields the best price prediction performance for Bitcoin price prediction in *1h*, *2h* and *3h*. DL outperforms RF and XGBoost for Classification. GRU, LSTM and RNN provide the best trend prediction of the Bitcoin price for *1h*, *2h* and *3h* trend predictions, respectively. We find evidence that algorithmic trading of Bitcoin using predictive analysis of ML algorithms can earn positive returns. Strategies derived from regression models have a higher financial performance than strategies derived from Classification models. We identify the ensemble strategy *reg_consensus* that combines the predictions of LSTM regression models with three different prediction horizons to be the best performing strategy. The *reg_consensus* strategy generates a ROI of 7.32% for the test period, which is superior to the ROI of the *buy_and_hold* strategy (0.13%). The ARIMA represents the benchmark for our Regression models. The *ARIMA_consensus* strategy has the best overall results among the ARIMA models but is not ablet to trigger profitable trading decisions. The PV development of the *reg_consensus* strategy is visualized and compared to *buy_and_hold* and *ARIMA_consensus* strategy in figure 23.

Figure 23: PV development of buy_and_hold, ARIMA_consensus and reg_consensus

Profitability

The calculation of the ROI that is performed in this work, includes trading associated costs based on the cost-settings of Binance. Costs that arise for the development, implementation and deployment of the ML models are not factored into the calculation. For a final evaluation of the trading algorithm, costs for Data Collection and Computing Power need to be included. Several platforms offer the integration of algorithmic trading strategies, providing API access, computing power, back-testing analysis, and other services (Fang, et al. 2020). The usage of these services presents multiple opportunities to structure costs but this work project intends to provide a stand-alone approach for algorithmic trading. The Data collection of this work is aimed to incur a minimum of costs. While the Yahoo Finance and Binance API are available free of charge, a paid API is required to collect Twitter data. For academic research like our work, the Twitter Developer API is freely available after application review and signing a non-commercial use agreement. The implementation of a trading algorithm requires a commercial Twitter API that costs $2.499 per month and allows to retrieve a maximum of five million tweets per month (Twitter, Developer Platform 2021c). The costs for the commercial Twitter API during the period tested in this work, would amount to $280. Due to the calculational complexity of Hyperparameter Tuning for DL algorithms, GPU computation is required (Cocco, Tonelli and Marchesi 2021). External computing power needs to be purchased to train and deploy the developed algorithms. A GeForce GTX 1080Ti with two GPU cores is required to train the LSTM Regression model for predictions in *1h*, *2h* and *3h*. Monthly costs total $876. The costs for the computing power during the period tested in this work, amount to $100 (Genisis, 2021). Additional costs include the development and monitoring of the algorithm as well as the connection to the platform API for real-time trading. These costs are difficult to quantify, and we do not include them in the following evaluation. Gains from cryptocurrency trading are not subject to taxes in Portugal and therefore not included in the calculation

(Cointaxlist 2021). The ROI and the total profit of *buy_and_hold*, the *ARIMA_consensus* strategy as the best performing strategy among the ARIMA strategies, and the strategy *reg_consensus,* are shown with and without the inclusion of costs for Twitter API and Computing Power in table 19. The performance metrics are calculated for the test period from 28.10.2021 09:50 to 31.10.2021 21:05. During this period our trading algorithm, based on the *reg_consensus* strategy, earns a total profit after costs of 352 USDT, equal to a ROI of 3.52%. *ARIMA_consensus* and *buy_and_hold* only achieve a ROI of 0.13% and -1.56%, respectively. Considering costs for Data Collection and Computing Power our trading algorithm outperforms its benchmark strategies.

Table 19: ROI of *reg_consensus* strategy and benchmark strategies including costs

All Amounts in USDT	Buy_and_hold	ARIMA_consensus	reg_consensus
Start Balance	10,000	10,000	10,000
Final Balance	10,013	9,834	10,732
+ Profit	13	-156	732
- Twitter API	0	0	280
- Computing Power	0	0	100
Profit (after costs)	13	-156	352
ROI (after costs)	0.13%	-1.56%	3.52%

Feasibility

Automated trading based on the developed trading algorithm requires considerations for feasibility of a real-time implementation. While previous research fails to address components of real-time implementation, we discuss limitations of the trading algorithm when it comes to real-time trading. Training of ML algorithms requires computing time dependent on the provided computing power.

LSTM algorithms have a high computational complexity (Cocco, Tonelli, and Marchesi 2021). Using computing power from *Genesis* cloud it took around eight hours for each LSTM algorithm to train, limiting the frequency of applying a newly trained model when Computing

Power is used as described in this work. An alternative to the presented modeling design, i.e., batch learning, is online learning. Online learning is a promising technique for learning from continuous streams of data but requires a different modeling architecture. For online learning, the algorithm takes the current model and subsequently uses new observations to further adjust the weights of each parameter. Online learning is faster to train but more difficult to maintain as the algorithms rely on a constant flow of data points (Hoi, et al. 2021). Constantly collecting data in the same format is challenging. The data for this work project is entirely collected using APIs. Although APIs are specifically designed to support Data Collection, APIs are subject to changes. For example, Twitter updated its API in June 2020 with an Early Access for the *v2 API*. In November 2021, the usage was published for all developers, causing changes in the Data Collection process. The four main variations are: Endpoint URLs, app and project requirements, response data format, and request parameters (Twitter, Twitter Developer 2021). Changes in API structure need to be monitored and code needs to be adjusted to prevent prediction errors. Trading signals triggered by the trading algorithm can either notify the trader or directly execute a trading action. An automated execution of trading signals requires the implementation of a real-time connection between the trading algorithm and a trading platform, e.g., Binance. While this work presents the foundation for a real-time implementation of the trading strategy, the connection to the Binance platform is not in the scope of this work. Considering these feasibility issues, we find that our developed trading algorithm has the prerequisites to be used for real-time trading.

5 Conclusion and Future Work

We define a holistic approach to build an intraday Bitcoin trading algorithm derived from predictive analysis of ML models and test the developed trading algorithm in a simulation setting. Special focus is placed on the impact of Twitter Sentiment (Study I), the Modeling Design (Study II), and the Trading Strategy (Study III). Finally, we evaluate profitability and feasibility of the trading algorithm in real-time implementation, which previous research fails to address.

We combine Regression and Classification models, with features from five feature categories (Supply & Demand, Crypto market, Macro Financial, Political and Sentiment). Study I identifies a Granger causal relationship between the overall and VIP Twitter Sentiment on the Bitcoin price. Study II concludes that LSTM models yield the best prediction performance for Bitcoin price prediction and GRU, LSTM and RNN generate the best Bitcoin trend predictions in *1h*, *2h* and *3h*, respectively. Study III finds superior profitability of ensemble trading strategies over individual trading strategies and identifies a Regression ensemble strategy to achieve the best overall results. Combining the findings of Study I, II, and III, we provide a holistic design of a trading algorithm. Finally, we evaluate the profitability of our trading algorithm for a real-time implementation considering costs for Data Collection and Computing Power and evaluate feasibility concerns. Our findings indicate that our intraday trading algorithm can be implemented for real-time trading and generates positive returns that exceed the returns of benchmark strategies.

Direct extensions to this work can investigate the real-world implementation of the presented design for an intraday Bitcoin trading algorithm. Special focus should be placed on the deployment of online learning for continuous model development. Further work can elaborate on developing additional business plans for monetarizing the presented trading algorithm.

List of Appendices

List of tables

List of figures

Appendix

Appendix 1: Feature Overview

Feature	Category	Interval	Sources	Start	End
BTC	Supply & Demand	5 min	Binance API	01.10.2021	31.10.2021
ETH	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
BNB	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
ADA	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
SOL	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
XRP	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
DOT	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
DOGE	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
USDC	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
LUNA	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
LTC	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
AVAX	Cryptocurrency Market	5 min	Binance API	01.10.2021	31.10.2021
S&P 500	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
SSE Composite	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
Nikkei 225	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
Dax 40	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
BSE Senex	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
FTSE 100	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
CAC 40	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
BOVESPA	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
FTSE MIB	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
TSX Composite	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
CNY	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
JPY	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
EUR	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
INR	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
GBP	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
BRL	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
CAD	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
CBOE Volatility	Political	5 min	Yahoo API	01.10.2021	31.10.2021
Brent	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
Natural Gas	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
Soybeans	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021

Corn	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
Gold	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
Copper	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
Silver	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
WTI	Macro Financial	5 min	Yahoo API	01.10.2021	31.10.2021
Twitter	Sentiment	5 min	Twitter API	01.10.2021	31.10.2021

Appendix 2: VIP List

Name	**Twitter**	**ID**
CNN Breaking News	@cnnbrk	428333
The New York Times	@nytimes	807095
CNN	@cnn	759251
BBC Breaking News	@bbcbreaking	5402612
BBC World	@bbcworld	742143
The Economist	@theeconomist	5988062
Reuters Top News	@reuters	1652541
The Wall Street Journal	@wsj	3108351
Time	@time	14293310
BitcoinNews	@BTCTN	3367334171
CryptoCurrency	@cryptocurrency	216304017
CryptoYoda	@CryptoYoda1338	852256178021294080
BitcoinMagazine	@BitcoinMagazine	361289499
CoinDesk	@coindesk	1333467482
Roger Ver	@rogerkver	176758255
Erik Voorhees	@ErikVoorhees	61417559
Ty Smith	@TyDanielSmith	961971412528517120
Tone Vays	@ToneVays	2577886615
CryptoCobain	@CryptoCobain	2259434528
Tyler Winklevoss	@Tyler	24222556
Vitalik Buterin	@VitalikButerin	295218901
CryptoWendyO	@CryptoWendyO	935742315389444096
StackingUSD	@StackingUSD	431243238
Girl Gone Crypto	@Girlgone_Crypto	1150790822813560833
David Gokhshtein	@davidgokhshtein	170049408
Hailey Lennon	@HaileyLennonBTC	3740778132
Justin Sun	@justinsuntron	902839045356744704
Ivan on Tech	@IvanOnTech	390627208
Kenn Bosak	@KennethBosak	4693571508

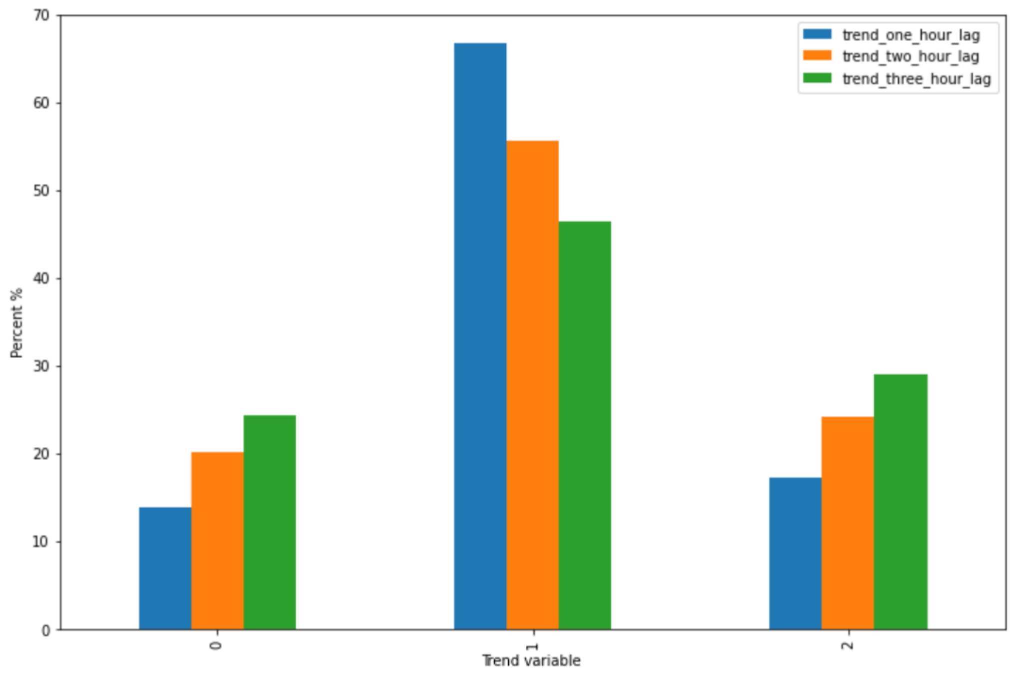
Scott Melker	@ScottMelker	17351167
TheCryptoDog	@TheCryptoDog	887748030304329728
BitBoy Crypto	@Bitboy_Crypto	954005112174862336
Dan Held	@DanHeld	1598709350
LayahHeilpern	@LayahHeilpern	455937214
Elon Musk	@elonmusk	44196397

Appendix 3: Equation RMSE

$$RMSE_i = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2}$$

y_i denotes the actual value of Bitcoin price i and $\hat{y}_i$ is the predicted value thereof (Mudassir, et al. 2020).

Appendix 4: Distribution of trend variable



Appendix 5: Equation Accuracy

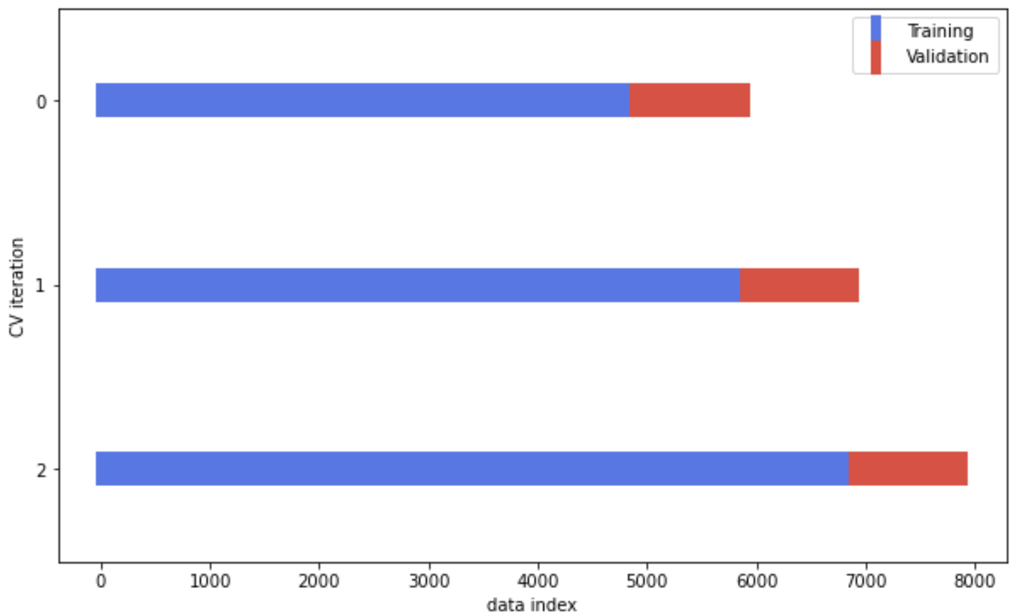
		Predicted label		
		0	1	2
	0	TP_0	FP_1	FP_2
True label	1	FP_0	TP_1	FP_2
	2	FP_0	FP_1	TP_2

In an imbalanced multi-class Classification scenario, micro-average is preferable to consider each individual prediction equally. A micro-average will aggregate the contributions of all classes to compute the average metric. Accuracy, Precision, Recall and F1 will be the same for micro averaging. (Liu, et al. 2015).

$$Accuracy = Precision_{Micro} = Recall_{Micro} = F1_{Micro} \frac{TP_0 + TP_1 + TP_2}{TP_0 + FP_0 + TP_1 + FP_1 + TP_2 + FP_2}$$

Accuracy is obtained from the overall number of correct predictions (TP_i) divided by all predictions ($TP_i + FP_i$) (Wang and Chiang 2007)

Appendix 6: Cross-Validation split



Appendix 7: Equation MinMax scaler

$$z_i = \frac{x_i - \min(x)}{max(x) - \min(x)}$$

Here, x represents the raw data of observation i whereas and z denotes the scaled data of observation i (Mudassir, et al. 2020).

Appendix 8: Feature Selection Structure

Name	**Statistical**	**RF & XGB**	**DL**
Feature Selection	*Not needed*	*MC: VIF* *FI: Wrapper method (RF), Embedded method (LR) ad Filter method (Correlation)*	MC: VIF

Appendix 9: Algorithm Pre-Selection

	Regression	**Classification**
Benchmark (Statistical)	ARIMA	LR
ML	RF, XGB, RNN, LSTM & GRU	

Appendix 10: Equation ARIMA

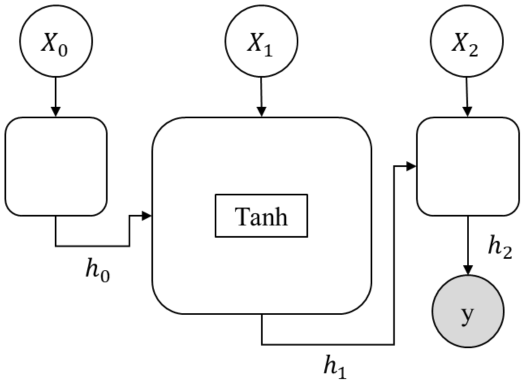
$$\Delta z_t = \sum_{i=1}^{p} \emptyset_i \Delta z_{t-i} + \sum_{i=1}^{p} \theta_i \varepsilon_{t-i} + \varepsilon_t$$

Appendix 11: Equation LR

$$max \sum_{i=i}^{n} \log p(y_i | x_i, \theta)$$

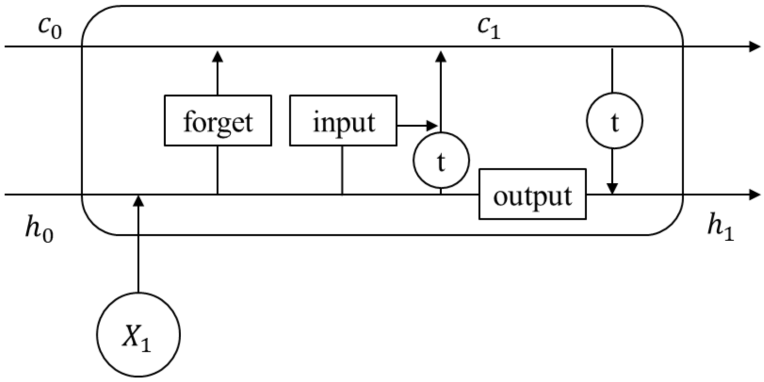
$$p(y = 1|x) \quad = \sigma(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)}$$

Appendix 12: Architecture RNN



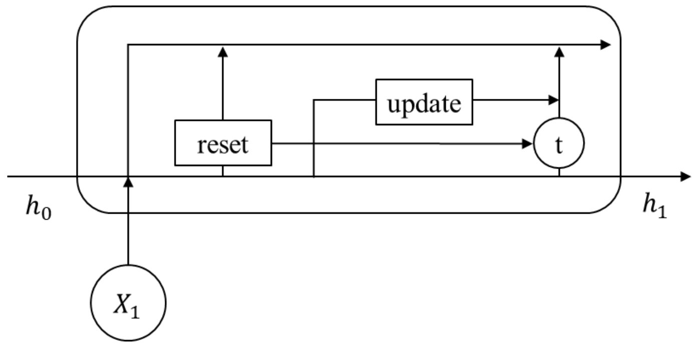
The RNN Tanh activation calculates the current hidden state $h_1$using a combination of the input X_1 and the previous hidden state h_0. The Tanh activation regulates the values flowing through a network.

Appendix 13: Architecture LSTM



The forget gate merges the input from the previous hidden state h_0 along with the input from the current state X_1. The input gate updates the cell state. The sigmoid function decides which information to keep from the Tanh output (t) (Chung and Shin 2018). The output gate controls whether the information of current cell state c_1 is visible. The previous hidden state and current values are passed through a sigmoid function and the cell state values are passed through the tanh function. The tanh output and sigmoid output are multiplied to produce the new hidden state (Dutta, Kumar and Basu 2020).

Appendix 14: Architecture GRU



Appendix 15: ARIMA

	Hyperparameter	Explanation
ARIMA	p	The number of lag observations included in the model, also called the lag order
	d	The number of times that the raw observations are differenced, also called the degree of differencing
	q	The size of the moving average window, also called the order of moving average

Appendix 16: Range Hyperparameter ML Algorithms

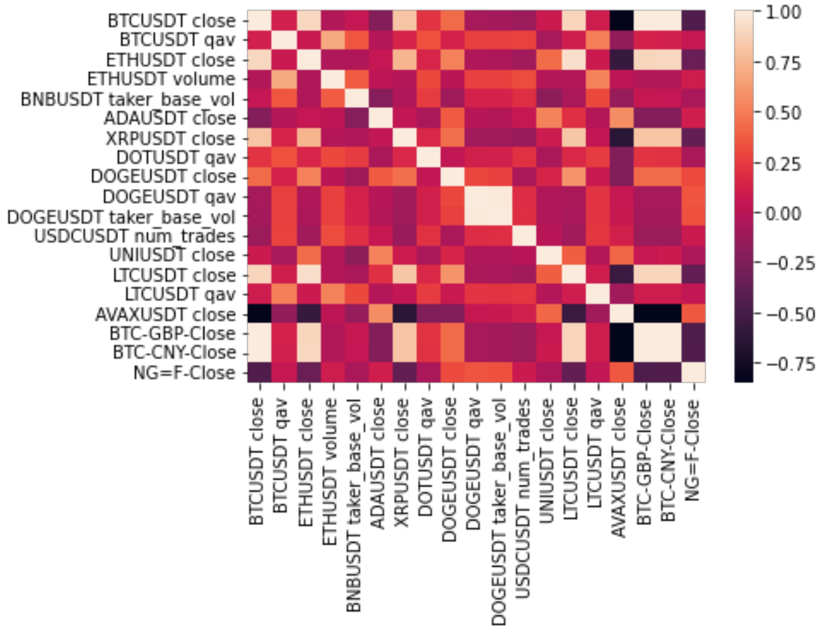
	Hyperparameter	Explanation	Values
RF & XGB	max_depth	The maximum depth of the tree	range(10,100,10)
	n_estimators	The number of trees in the forest	range(10,60,100)
LR	C	Inverse of regularization strength	
	penalty	Specify the norm of the penalty	
RNN, LSTM & GRU	n_epochs	Number times an entire dataset is passed forward and backward through the neural network	range(200,400,25)
	batch_size	Total number of training examples present in a single batch	range(40,60,10)
	optimizer	Algorithm optimizer	range(200,400,25)
	input_dim	Features in the input	range(200,400,25)
	output_dim	Features in the output	range(200,400,25)
	hidden_dim	Features in the hidden state h	range(200,400,25)
	layer_dim	Recurrent layers	range(200,400,25)
	dropout	Dropout layer on the outputs of each layer except the last layer	range(200,400,25)
	learning_rate	Change of the model in response to the estimated error each time the model weights are updated	range(200,400,25)

	weight_decay	Weight regularization provides an approach to reduce the overfitting	range(200,400,25)

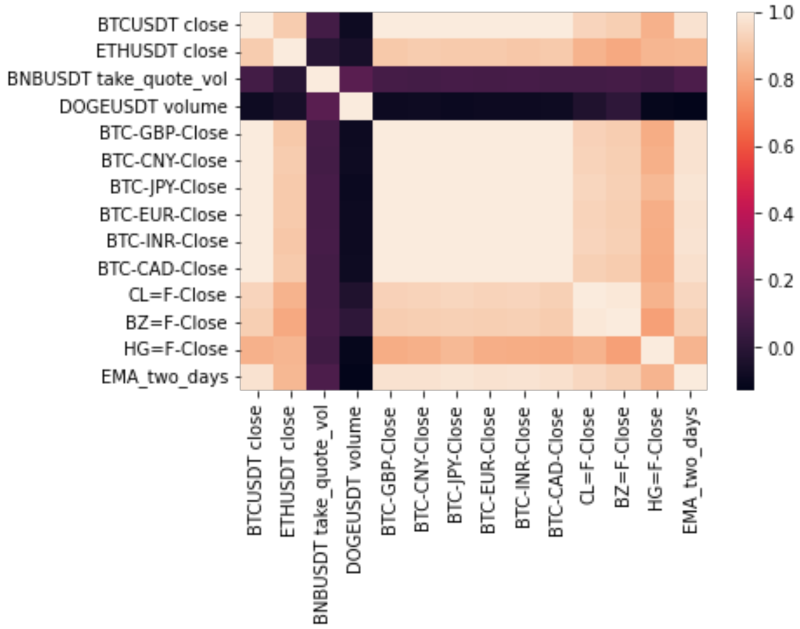
Appendix 17: Detected Features Multicollinearity

Features
USDCUSDT volume
USDCUSDT qav
USDCUSDT taker_base_vol
USDCUSDT take_quote_vol
BUSDUSDT volume
BUSDUSDT qav
BUSDUSDT taker_base_vol
BUSDUSDT take_quote_vol
NYSE
NASDAQ
LSE
EUREX
EMA_one_hour
EMA_two_hours
EMA_four_hours
MACD
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday

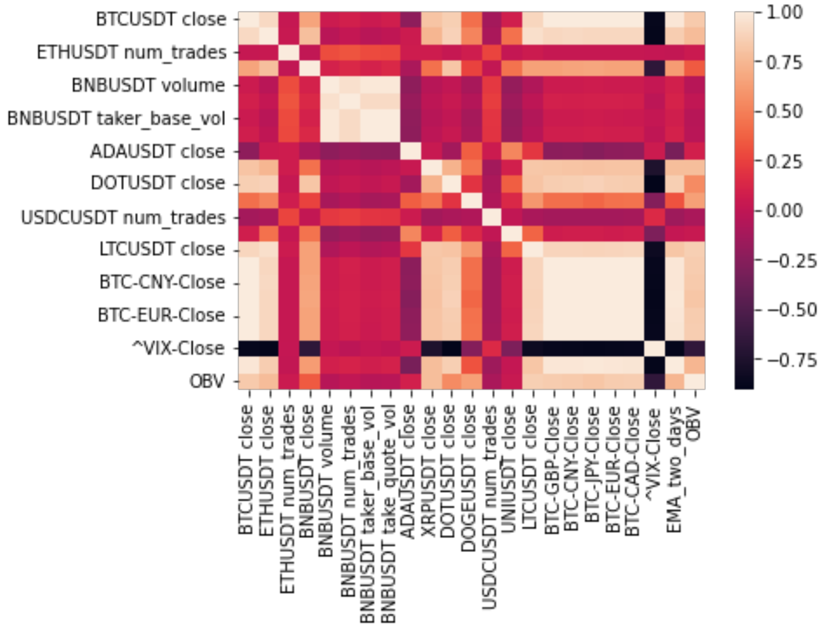
Appendix 18: Correlation matrix selected Features *Classification 1h*



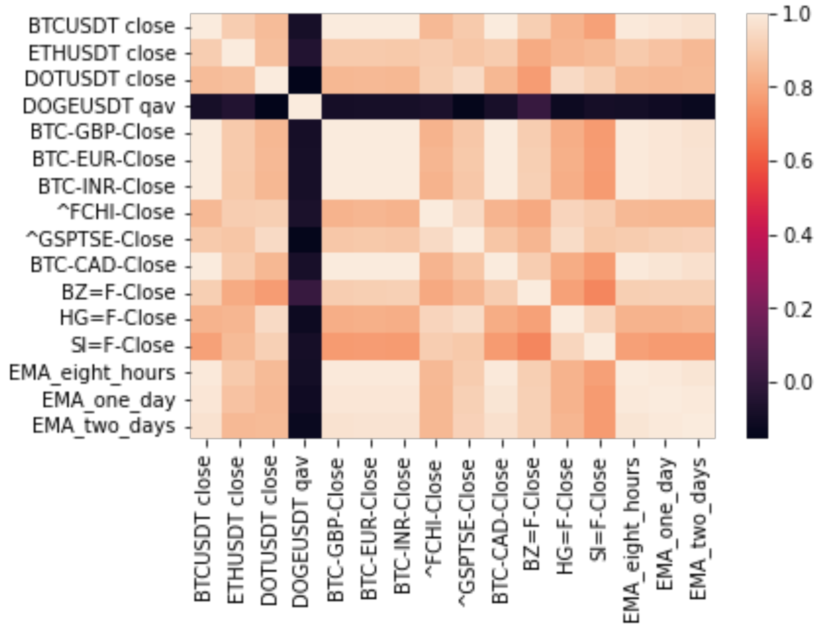
Appendix 19: Correlation matrix selected Features *Regression 1h*



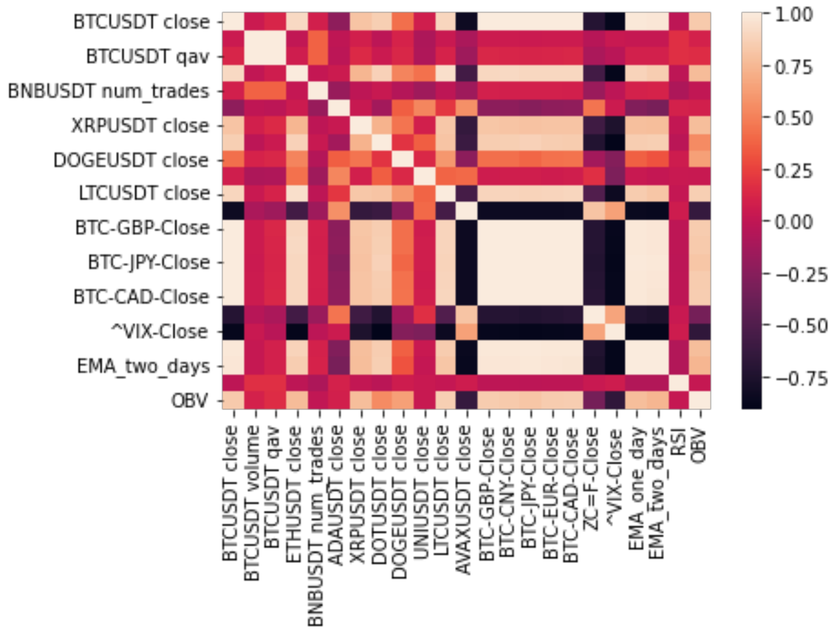
Appendix 20: Correlation matrix selected Features *Classification 2h*



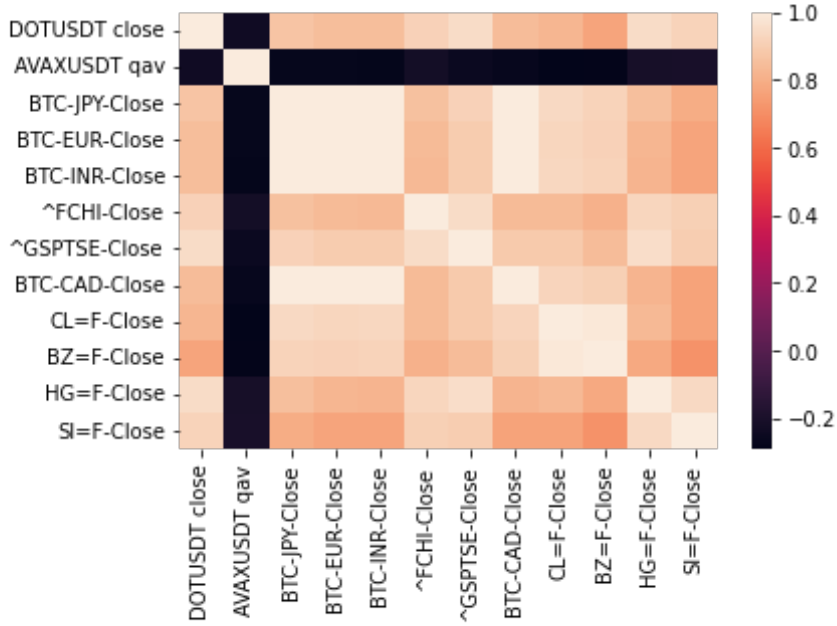
Appendix 21: Correlation matrix selected Features *Regression 2h*



Appendix 22: Correlation matrix selected Features *Classification 3h*



Appendix 23: Correlation matrix selected Features *Regression 3h*



Appendix 24: Tuned Hyperparameter Classification models

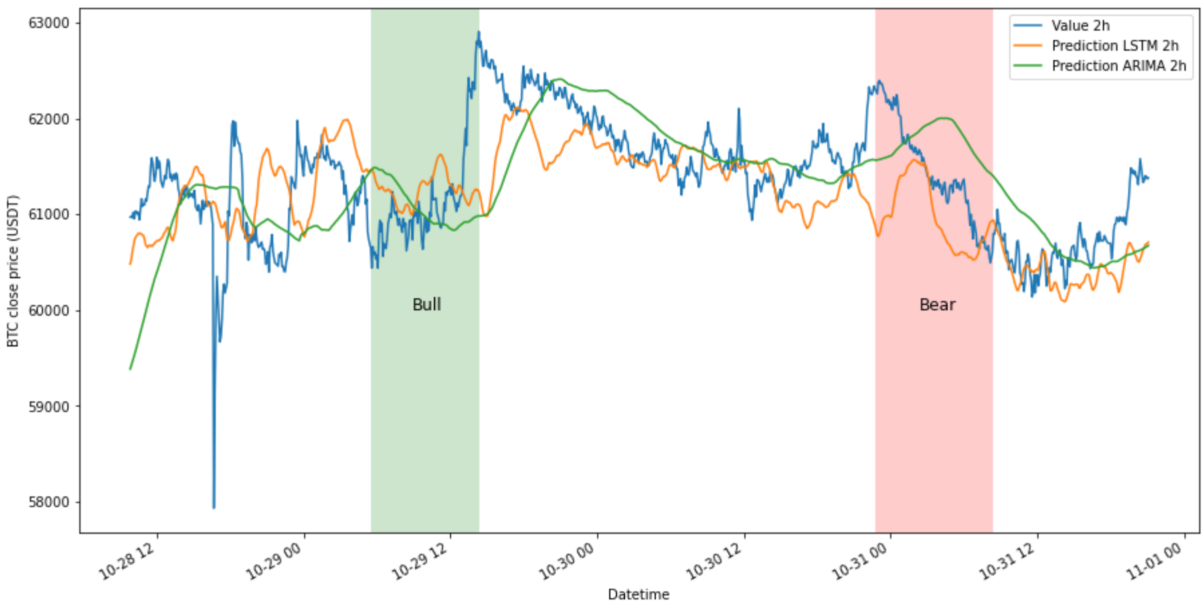
	Hyperparameter	***1h***	***2h***	***3h***
RF	max_depth	90	20	22
	n_estimators	55	14	10
	Validation result	0.67667	0.47867	0.458
XGB	max_depth	10	10	26
	n_estimators	80	30	10

	Validation result	0.608	0.359	0.36733
LR	C	10	1	6
	penalty	L2	L1	L1
	Validation result	0.627	0.41067	0.395
RNN	n_epochs	350	325	400
	batch_size	40	40	40
	optimizer	Adamax	Adagrad	SGD
	hidden_dim	40	140	140
	layer_dim	5	1	5
	dropout	0.1	0.1	0.2
	learning_rate	0.00025558	0.0001961	0.00021125
	weight_decay	2.49892836e-06	3.68185601e-07	1.52559081e-07
	Validation result	0.71849	0.54477	0.49637
LSTM	n_epochs	275	275	350
	batch_size	40	60	40
	optimizer	Adagrad	Adamax	Adagrad
	hidden_dim	100	140	100
	layer_dim	5	5	4
	dropout	0.1	0.2	0.2
	learning_rate	0.00032765	0.000036	1.14646199e-05
	weight_decay	5.21889279e-07	2.199196e-07	1.681352323e-06
	Validation result	0.71746	0.5544	0.47461
GRU	n_epochs	375	225	275
	batch_size	50	40	60
	optimizer	SGD	Adagrad	Adagrad
	hidden_dim	60	60	100
	layer_dim	6	6	5
	dropout	0.0	0.2	0.3
	learning_rate	2.12815675e-05	0.00015124	0.00016263
	weight_decay	3.63533828e-07	3.43294732e-06	4.57734953e-06
	Validation result	0.72746	0.54891	0.48508

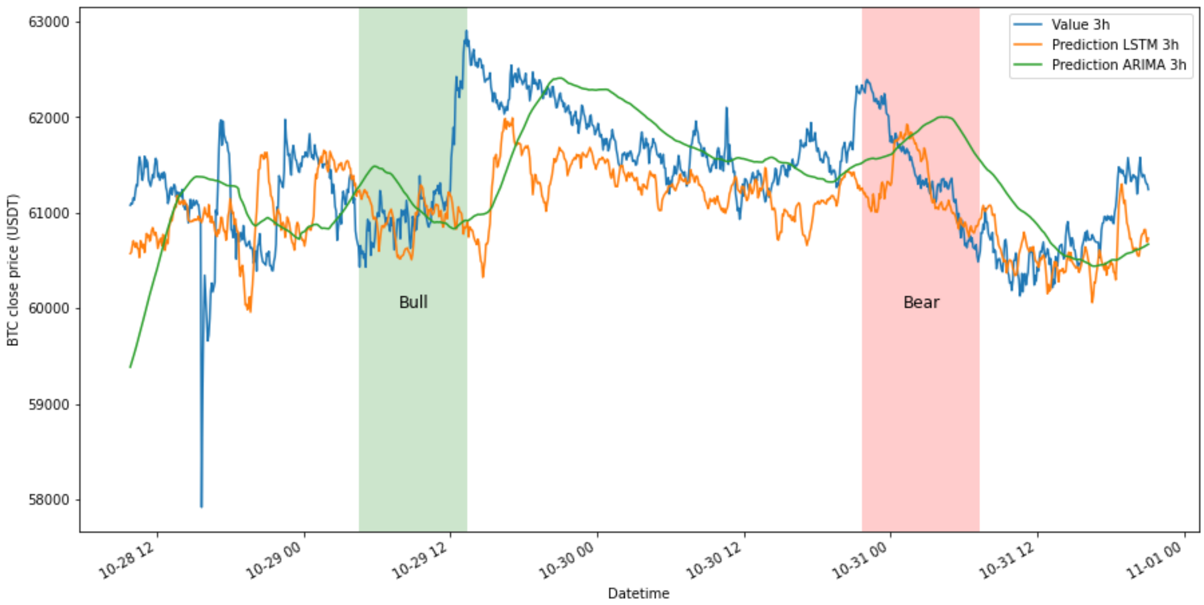
Appendix 25: Tuned Hyperparameter Regression models

	Hyperparameter	*1h*	*2h*	*3h*
Random Forest	max_depth	90	10	26
	n_estimators	55	28	22
	Validation result	0.13123	0.12336	0.15366
XGB	max_depth	20	10	14
	n_estimators	35	28	10
	Validation result	0.10691	0.10308	0.13779
RNN	n_epochs	125	200	350
	batch_size	30	50	60
	optimizer	Adagrad	Adam	Adamax
	hidden_dim	39	140	140
	layer_dim	1	1	1
	dropout	0.3	0.2	0.2
	learning_rate	4.36856644e-05	0.2	1.20835049e-05
	weight_decay	8.26508963e-06	1.000663e-07	9.31975619e-06
	Validation result	0.11939	0.074487	0.06693
LSTM	n_epochs	400	200	300
	batch_size	50	50	40
	optimizer	SGD	Adagrad	SGD
	hidden_dim	80	120	100
	layer_dim	1	2	1
	dropout	0.0	0.1	0.2
	learning_rate	0.00066	0.00012236	0.00010252
	weight_decay	4.93901468e-06	3.63424261e-07	4.76473615e-07
	Validation result	0.03875	0.05656	0.59596
GRU	n_epochs	375	225	350
	batch_size	60	60	50
	optimizer	Adamax	SGD	SGD
	hidden_dim	40	120	140
	layer_dim	1	2	1
	dropout	0.0	0.2	0.3
	learning_rate	1.5345354e-04	0.0056974	1.14987418e-05
	weight_decay	5.06782934e-06	2.67577814e-06	2.54986817e-06
	Validation result	0.05756	0.0733	0.05999

Appendix 26: Bitcoin Close Price 2h True Value and LSTM and ARIMA Prediction



Appendix 27: Bitcoin Close Price 3h True Value and LSTM and ARIMA Prediction



Appendix 28: Confusion matrix Evaluation Logreg *1h* all

		Predicted label		
		0	1	2
True label	0	0	81	40
	1	1	571	167

	2	0	84	56

Appendix 29: Confusion matrix Evaluation GRU *1h* all

		0	1	2
	0	1	120	0
True label	1	7	703	29
	2	5	127	8

Appendix 30: Confusion matrix Evaluation Logreg *1h* bull

		0	1	2
	0	0	1	2
True label	1	0	26	40
	2	0	14	20

Appendix 31: Confusion matrix Evaluation GRU *1h* bull

		0	1	2
	0	0	3	0
True label	1	0	66	34
	2	0	34	0

Appendix 32: Confusion matrix Evaluation Logreg *1h* bear

		0	1	2
	0	0	34	0
True label	1	0	104	1
	2	0	1	0

Appendix 33: Confusion matrix Evaluation GRU *1h* bear

		0	1	2
	0	0	34	0
True label	1	0	105	4
	2	0	1	0

Appendix 34: Confusion matrix Evaluation Logreg *2h* all

		0	1	2
	0	2	178	35
True label	1	15	510	57
	2	4	159	40

Appendix 35: Confusion matrix Evaluation LSTM *2h* all

		0	1	2
	0	30	180	5
True label	1	54	482	46
	2	0	181	22

Appendix 36: Confusion matrix Evaluation Logreg *2h* bull

		0	1	2
	0	2	4	0
True label	1	9	48	0
	2	4	36	0

Appendix 37: Confusion matrix Evaluation LSTM *2h* bull

		0	1	2
	0	4	1	1
True label	1	1	12	44
	2	0	18	22

Appendix 38: Confusion matrix Evaluation Logreg *2h* bear

		0	1	2
	0	0	73	0
True label	1	0	66	0
	2	0	1	0

Appendix 39: Confusion matrix Evaluation LSTM *2h* bear

		0	1	2
	0	11	62	0
True label	1	0	66	0
	2	0	0	0

Appendix 40: Confusion matrix Evaluation Logreg *3h* all

		0	1	2
	0	103	109	68
True label	1	141	169	156
	2	38	93	123

Appendix 41: Confusion matrix Evaluation RNN *3h* all

0	1	2

	0	61	213	6
True label	1	73	347	46
	2	21	203	30

Appendix 42: Confusion matrix Evaluation Logreg *3h* bull

		0	1	2
	0	10	0	0
True label	1	28	1	18
	2	10	10	26

Appendix 43: Confusion matrix Evaluation RNN *3h* bull

		0	1	2
True label	0	10	0	0
	1	26	3	18
	2	0	20	26

Appendix 44: Confusion matrix Evaluation Logreg *3h* bear

		0	1	2
	0	24	37	14
True label	1	9	19	27
	2	1	5	4

Appendix 45: Confusion matrix Evaluation RNN *3h* bear

		0	1	2
True label	0	8	67	0
	1	0	55	0

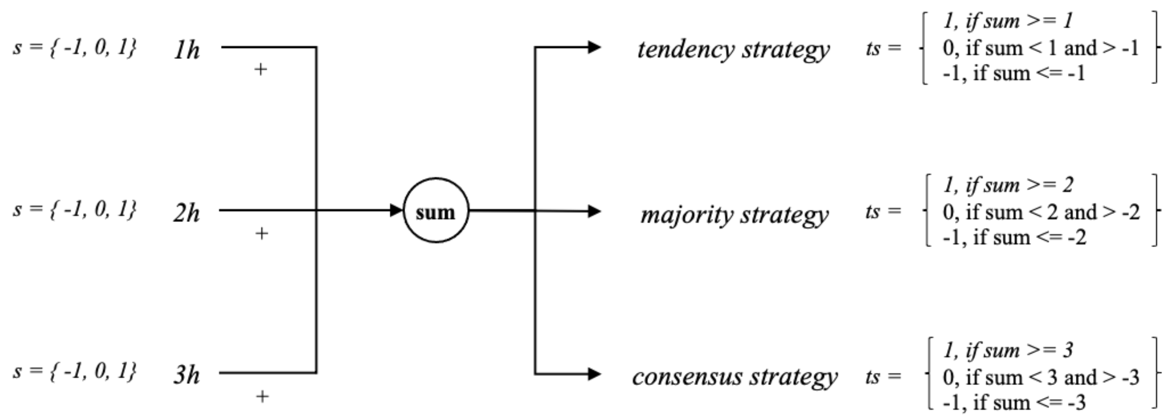
2	0	10	0

Appendix 46: Overview of reviewed paper for Study III: Trading Strategy.

Author (Year)	Strategy	Strategy Description	Ensemble Strategy
Atsalakis, et al. (2019)	BS	Starting with €100,000, all funds are either invested or devested	NO
Borges and Neves (2020)	BS	Starting with one unit of quote currency, all funds are either invested or devested	YES
Chevallier, Zhu and Zhang (2021)	LS	Using a position of 1 Bitcoin, profits/losses are calculated for each interval separately and accumulated later	NO
Chu, Chan and Zhang (2020)	BS	Starting with a long position of the cryptocurrency, all funds are either invested or devested	YES
de Souza, et al. (2019)	BS	Starting with a normalized position of 1, all funds are either invested or devested	NO
Dutta, Kumar and Basu (2020)	BS, LS	Starting with a normalized position of 1, for BS, all funds are either invested or devested; for LS, long and short positions are created, short positions are covered at the end of the day	NO
Garcia and Schweitzer (2015)	LS	Starting with $1, all funds are invested, and shorting is limited to the amount of capital held by trader and to one iteration	YES
Huang, Huang and Ni (2019)	LS	Using a position of 1 Bitcoin, profits/losses are calculated for each interval separately and accumulated later	NO
Ji, Kim and Im (2019)	BS	Starting with $10,000, all funds are either invested or devested	NO
Nakano, Takahashi and Takahashi (2018)	BS, LS	Starting with a normalized position of 1, for BS, all funds are either invested or devested; for LS, long and short positions are created, short positions are covered at the end of the day	NO

Sebastião and Godinho (2021)	BS	Starting with a long position of the cryptocurrency, all funds are either invested or devested	YES
Shintate and Pichl (2019)	BS	Starting with a normalized position of 1, all funds are either invested or devested	NO
Vo and Yost-Bremm (2018)	LS	Using $1,000, profits/losses are calculated for each trade separately and accumulated later	NO

Appendix 47: Derivation of tendency, majority, and consensus strategy.



Appendix 48: Equations of PV, ROI, SR, and MDD.

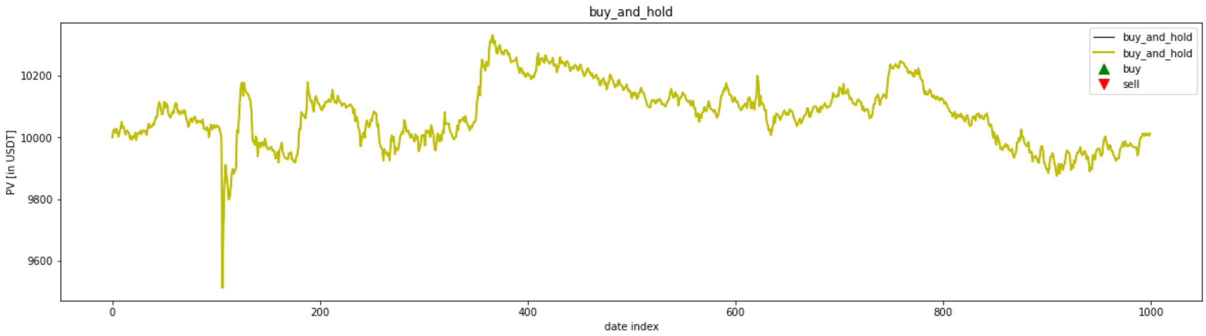
$$PV_i = PV_{i,n}$$

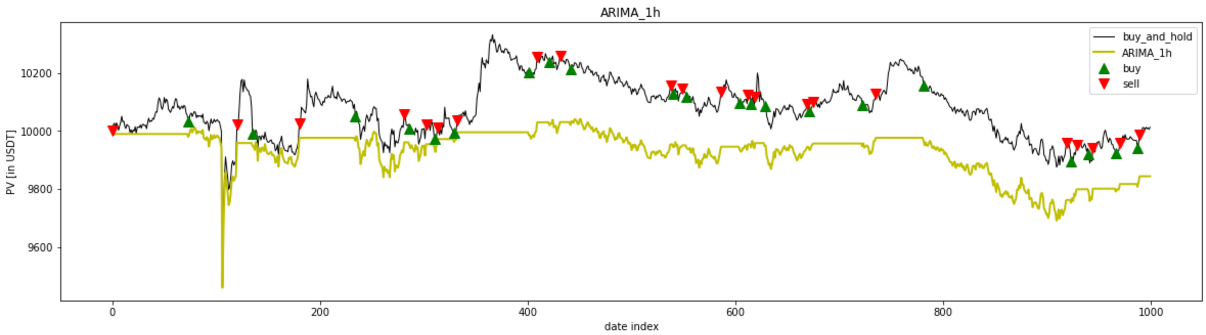
$$ROI_i = \frac{PV_{i,n} - PV_{i,0}}{PV_{i,0}} - 1$$

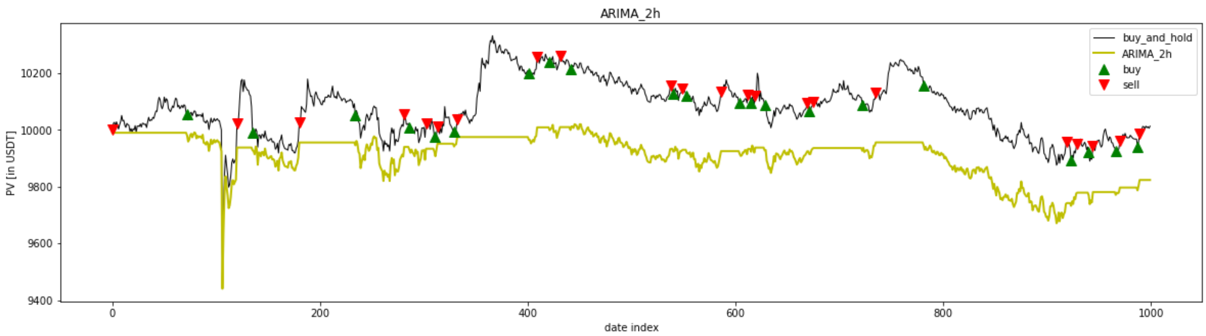
$$SR_i = \sqrt[(365\times24\times12)]{\frac{ROI_i - R_f}{\sigma_i}}$$

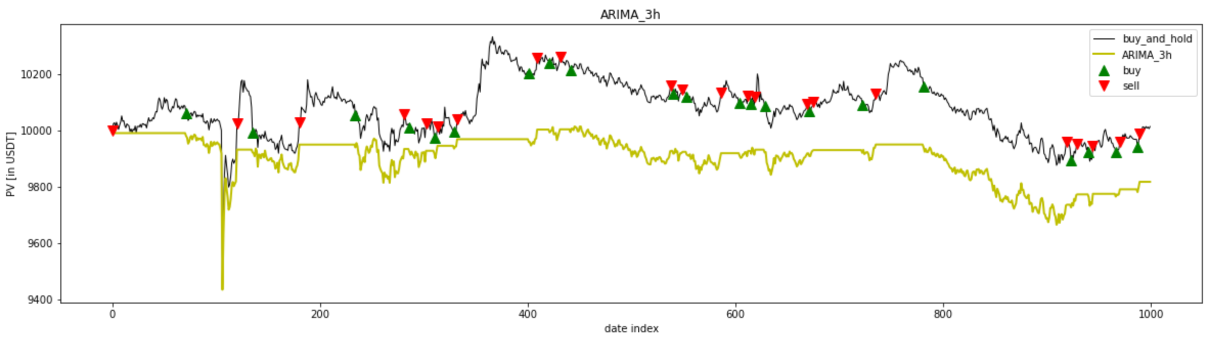
$$MDD_i = \max(\max(ROI_{i,t} - ROI_{i,T}))$$

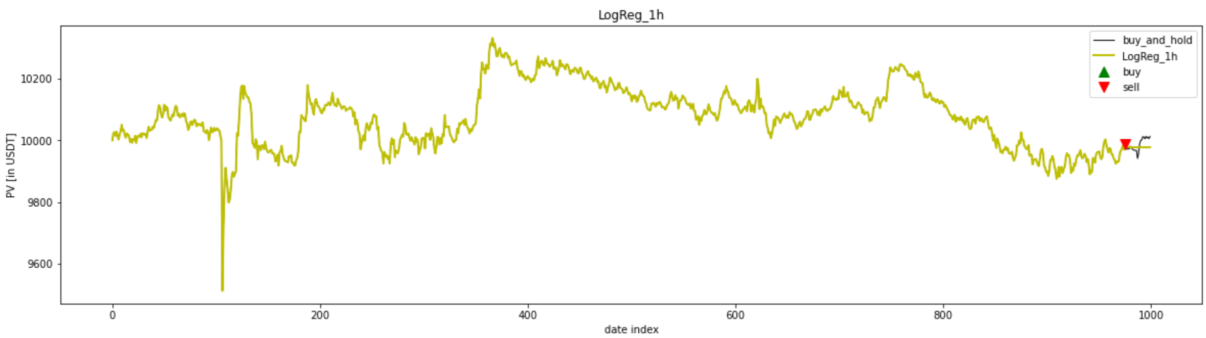
Appendix 49: PV development of trading strategies including buy and sell actions

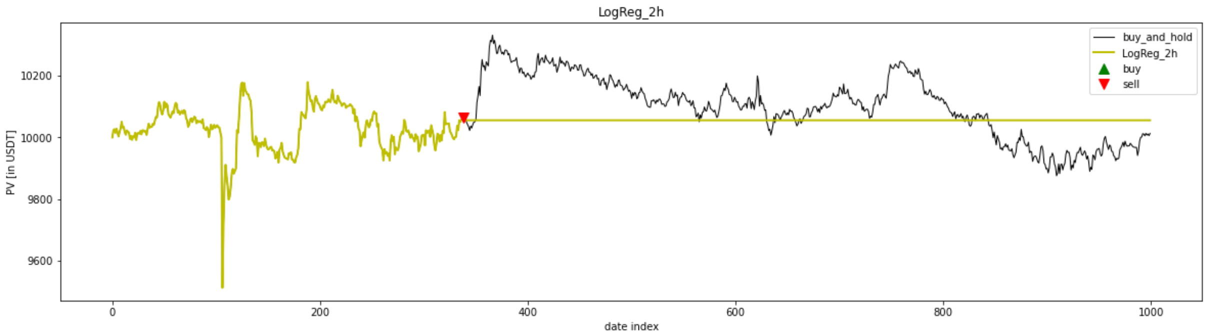

buy_and_hold
buy_and_hold
buy_and_hold
buy
sell
PV [in USDT]
10200
10000
9800
9600
0
200
400
600
800
1000
date index

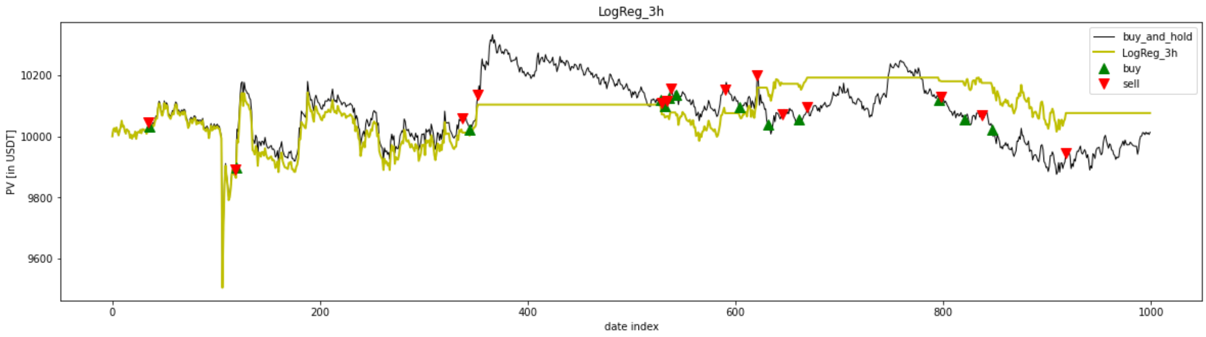

ARIMA_1h
buy_and_hold
ARIMA_1h
buy
sell
PV [in USDT]
10200
10000
9800
9600
0
200
400
600
800
1000
date index

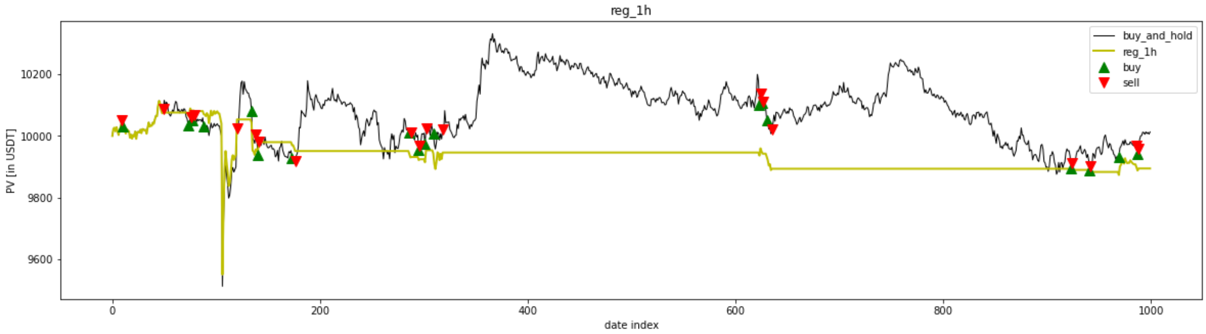

ARIMA_2h
buy_and_hold
ARIMA_2h
buy
sell
PV [in USDT]
10200
10000
9800
9600
9400
0
200
400
600
800
1000
date index

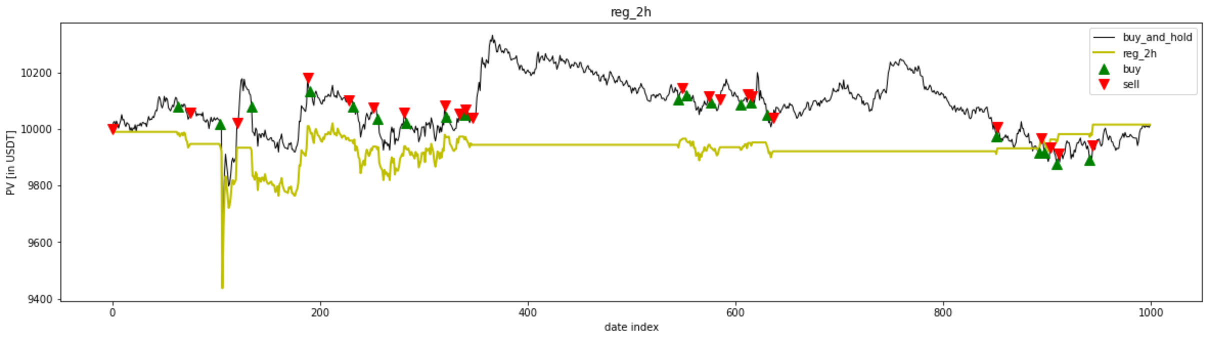

ARIMA_3h
buy_and_hold
ARIMA_3h
buy
sell
PV [in USDT]
10200
10000
9800
9600
9400
0
200
400
600
800
1000
date index

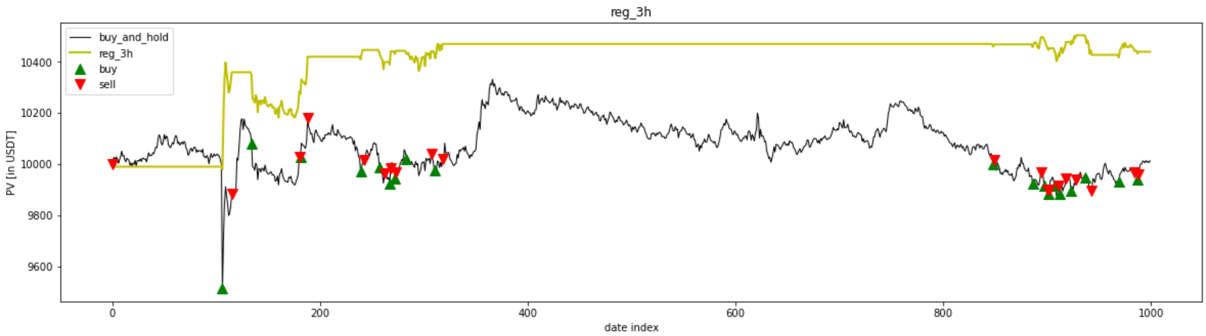

LogReg_1h
buy_and_hold
LogReg_1h
buy
sell
PV [in USDT]
10200
10000
9800
9600
0
200
400
600
800
1000
date index

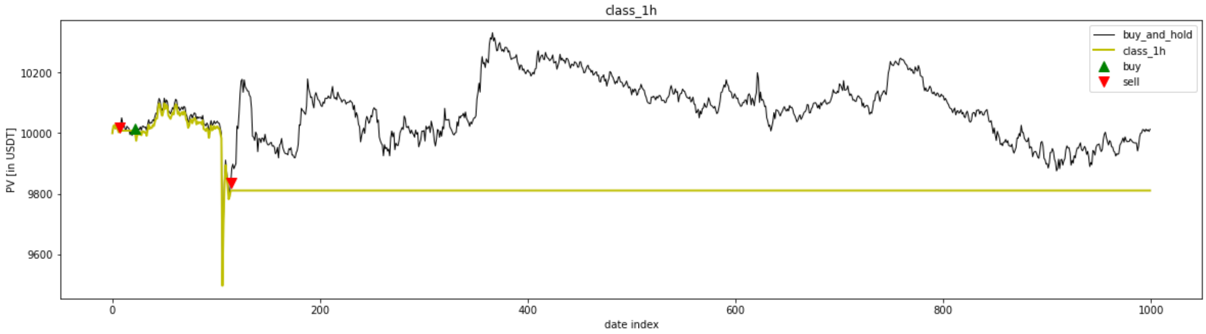

LogReg_2h
PV [in USDT]
date index
buy_and_hold
LogReg_2h
buy
sell

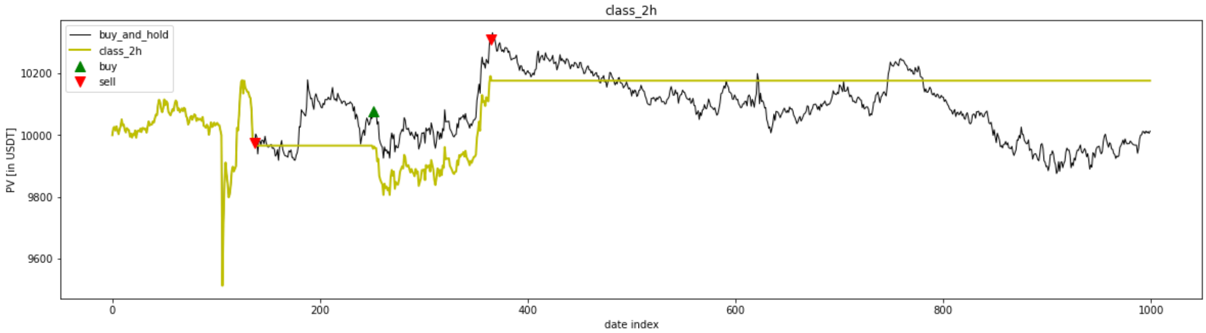

LogReg_3h
PV [in USDT]
date index
buy_and_hold
LogReg_3h
buy
sell

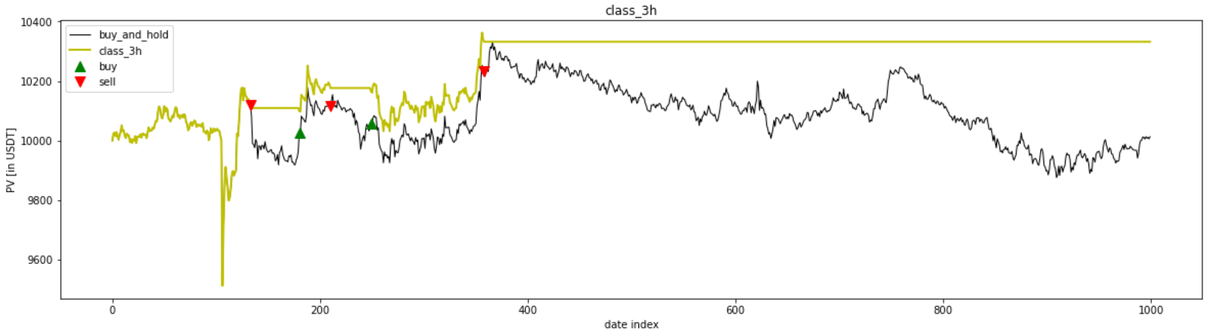

reg_1h
PV [in USDT]
date index
buy_and_hold
reg_1h
buy
sell

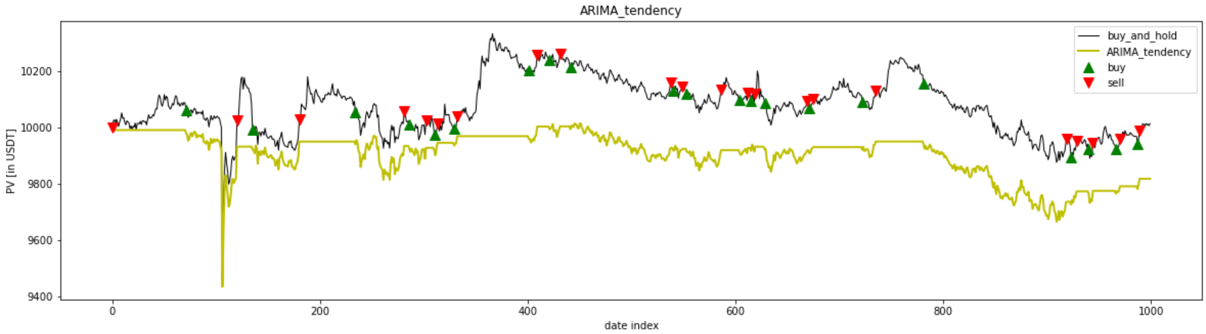

reg_2h
PV [in USDT]
date index
buy_and_hold
reg_2h
buy
sell

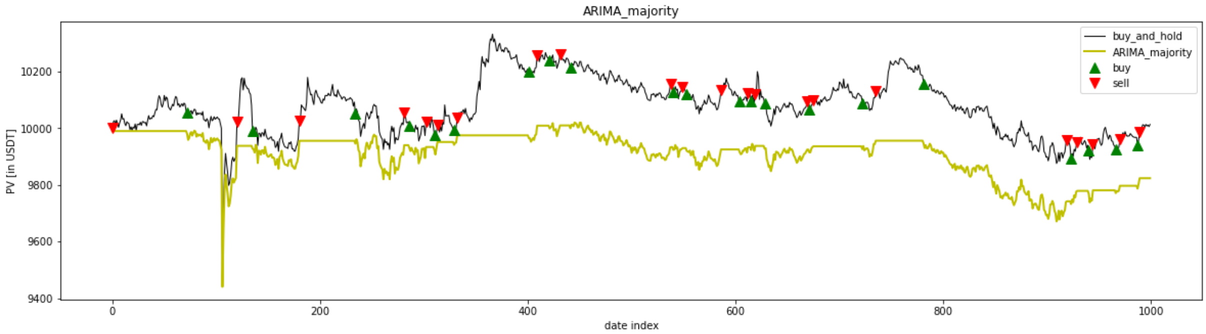

reg_3h
PV [in USDT]
date index
buy_and_hold
reg_3h
buy
sell

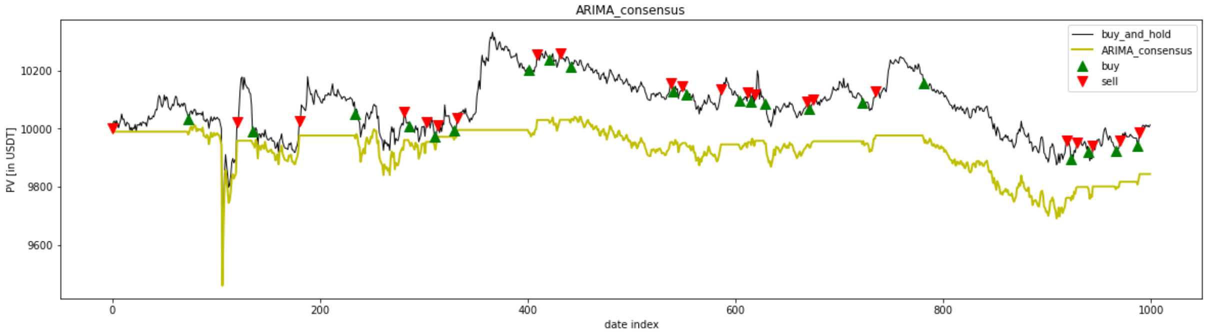

class_1h
buy_and_hold
class_1h
buy
sell
PV [in USDT]
date index
9600
9800
10000
10200
0
200
400
600
800
1000

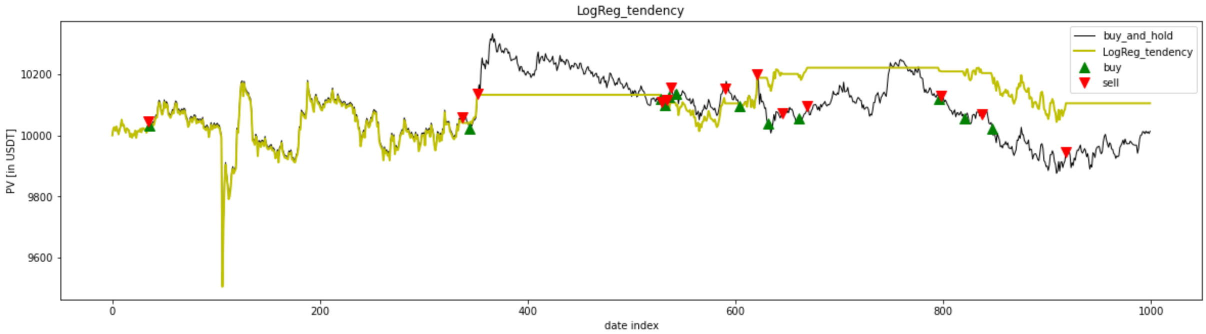

class_2h
buy_and_hold
class_2h
buy
sell
PV [in USDT]
date index
9600
9800
10000
10200
0
200
400
600
800
1000

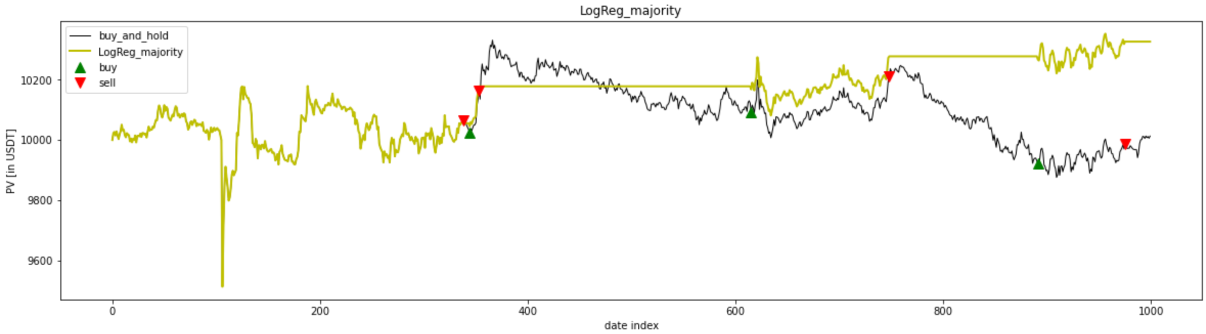

class_3h
buy_and_hold
class_3h
buy
sell
PV [in USDT]
date index
9600
9800
10000
10200
10400
0
200
400
600
800
1000

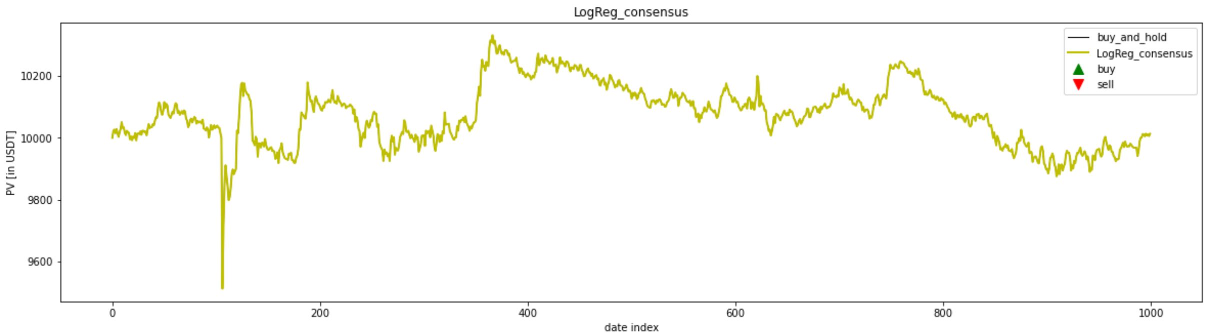

ARIMA_tendency
buy_and_hold
ARIMA_tendency
buy
sell
PV [in USDT]
date index
9400
9600
9800
10000
10200
0
200
400
600
800
1000

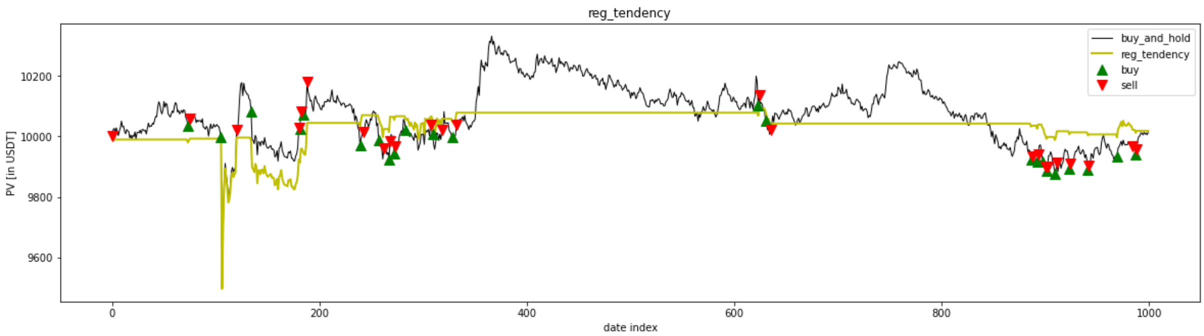

ARIMA_majority
buy_and_hold
ARIMA_majority
buy
sell
PV [in USDT]
date index
9400
9600
9800
10000
10200
0
200
400
600
800
1000


ARIMA_consensus
buy_and_hold
ARIMA_consensus
buy
sell
PV [in USDT]
date index

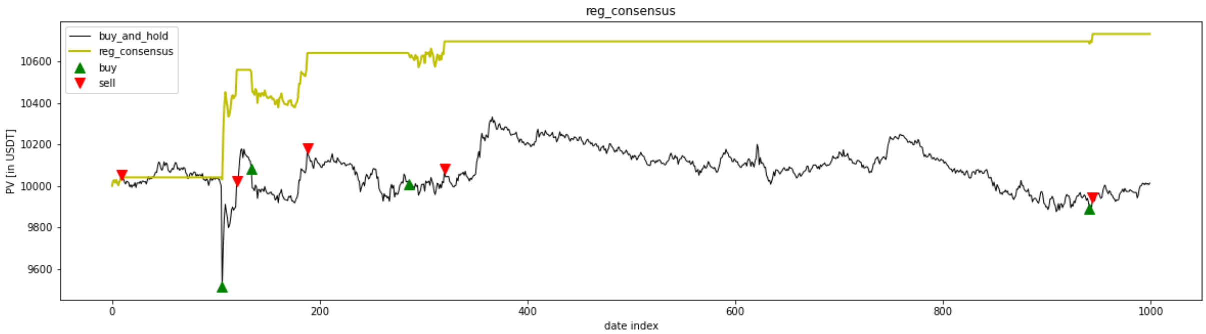

LogReg_tendency
buy_and_hold
LogReg_tendency
buy
sell
PV [in USDT]
date index

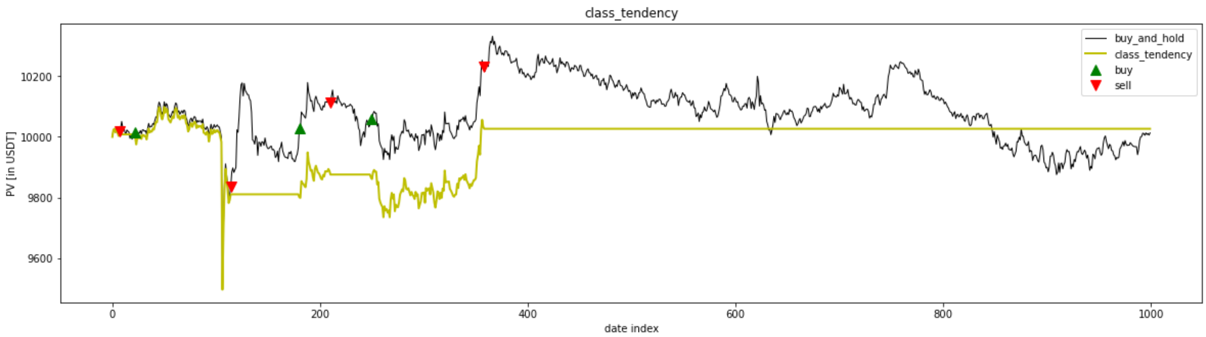

LogReg_majority
buy_and_hold
LogReg_majority
buy
sell
PV [in USDT]
date index


LogReg_consensus
buy_and_hold
LogReg_consensus
buy
sell
PV [in USDT]
date index

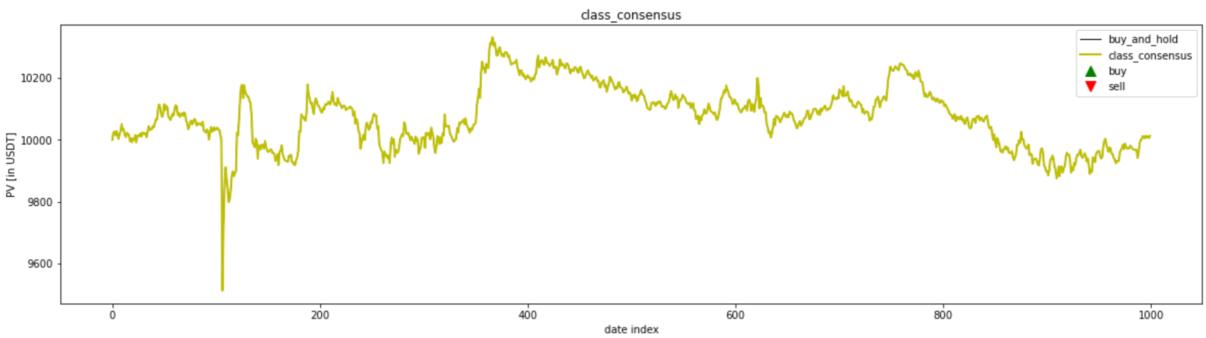

reg_tendency
buy_and_hold
reg_tendency
buy
sell
PV [in USDT]
date index

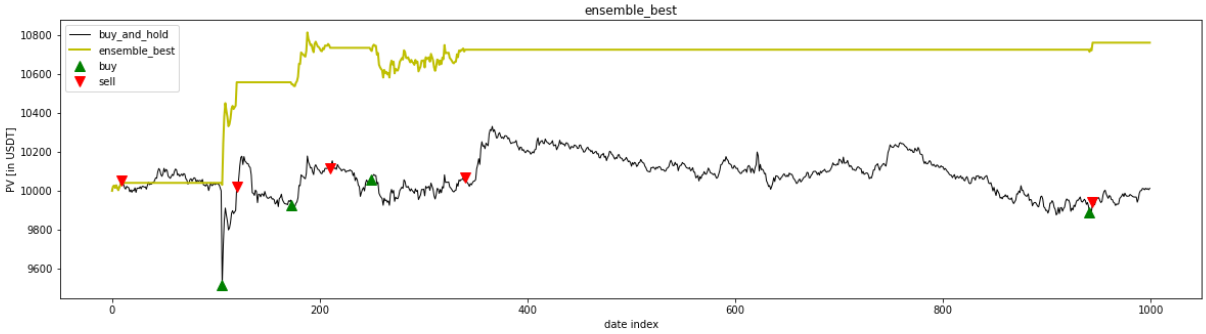

reg_majority
buy_and_hold
reg_majority
buy
sell
PV [in USDT]
date index


reg_consensus
buy_and_hold
reg_consensus
buy
sell
PV [in USDT]
date index


class_tendency
buy_and_hold
class_tendency
buy
sell
PV [in USDT]
date index


class_majority
buy_and_hold
class_majority
buy
sell
PV [in USDT]
date index


class_consensus
buy_and_hold
class_consensus
buy
sell
PV [in USDT]
date index


ensemble_best
buy_and_hold
ensemble_best
buy
sell
PV [in USDT]
date index
10800
10600
10400
10200
10000
9800
9600
0
200
400
600
800
1000

References

Abbad, Jumah, Bashar Fardousi, and Muneer Abbad. 2014. "Advantages of Using Technical Analysis to Predict Future Prices on the Amman Stock Exchange." *International Journal of Business and Management* 1-16.

Abraham, Jethin, Daniel Higdon, John Nelson, and Juan Ibarra. 2018. "Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis." *SMU Data Science Review: Vol. 1 : No. 3 , Article 1* 22.

Abramovich, Felix, and Claudia Angelini. 2006. "Testing in mixed-effects FANOVA models." *Journal of Statistical Planning and Inference* 136(12): 4326-4348.

Aharona, David Yechia, and Mahmoud Qadan. 2019. "Bitcoin and the day-of-the-week effect." *Finance Research Letters 31* 415-424.

Ahmed, Walid M.A., and Mustafa AL Mafrachi. 2021. "Do higher-order realized moments matter for cryptocurrency returns?" *International Review of Economics and Finance 72* 483-499.

Ahn, Yongkil, and Dongyeon Kim. 2021. "Emotional trading in the cryptocurrency market." *Finance Research Letters 42* 101912.

Ante, Lennart. 2021. "How Elon Musk's Twitter activity moves cryptocurrency markets." *BRL Working Paper Series No. 16* 13.

Antonakak, Despoina, Paraskevi Fragopoulou, and Sotiris Ioannidis. 2021. "A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks." *Expert Systems with Applications. Elsevier BV Volume 164* 25.

Aroussi, Ran. 2021. *yfinance 0.1.67.* Accessed December 07, 2021. https://pypi.org/project/yfinance/.

Atsalakis, George S., Ioanna G. Atsalaki, Fotios Pasiouras, and Constantin Zopounidis. 2019. "Bitcoin price forecasting with neuro-fuzzy techniques." *European Journal of Operational Research 276* 770-780.

Bali, Turan G., and Peng Lin. 2006. "Is there a risk-return trade-fff? Evidence from high-frequency data." *Journal of Applied Econometrics 21* 1169-1198.

Barclay, Michael J., and Terrence Hendershott. 2003. "Price Discovery and Trading After Hours." *The Review of Finanical Studies* 1041-1073.

Behrendt, Simon, and Alexander Schmid. 2018. "The Twitter myth revisited: Intraday investor sentiment, Twitter activity and individual-level stock return volatility." *Journal of Banking and Finance Volume 96* 355–367.

Bergstra, James, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. "Algorithms for hyper-parameter optimization." *Proceedings of the 24th International Conference on Neural Information Processing.* Granada, Spain.

Bernal, Armando, Sam Fok, and Rohit Pidaparthi. 2012. "Market Time Series Prediction with Recurrent Neural Networks."

Binance. 2021. *API Documentation.* December 03. Accessed December 07, 2021. https://binance-docs.github.io/apidocs/futures/en/#change-log.

Binance. 2021. *Binance.* December 7. Accessed December 7, 2021. https://www.binance.com/en/fee/schedule.

Blockchain.com. 2021. *Blockchain.com Wallets.* December 12. Accessed December 13, 2021. https://www.blockchain.com/charts/my-wallet-n-users.

Bohte, Rossini. 2019. "Comparing the Forecasting of Cryptocurrencies by Bayesian Time-Varying Volatility Models." *Journal of Risk and Financial.*

Boldt, Martin, and Anton Borg. 2020. "Using VADER sentiment and SVM for predicting customer response sentiment." *Expert Systems with Applications 162* 11.

Borges, Tomé Almeida, and Rui Ferreira Neves. 2020. “Ensemble ofmachine learning algorithms for cryptocurrency investment with different data resampling methods.” *Applied Soft Computing Journal* 90.

Bouri, Elie, Peter Molnár, Georges Azzi, David Roubaud, and Lars Ivar Hagfors. 2017. "On the hedge and safe haven properties of Bitcoin: Is it really more than a diversifier?" *Finance Research Letters* 20: 192-198.

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 5–32.

Breiman, Leo. 1996. "Bagging Predictors." *Machine Learning* 24: 123-140.

Brownlee, Jason. 2021. *Machine Learning Mastery.* Februar 17. Accessed December 4, 2021. https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/.

Bugár, Gyöngyi, and Márta Somogyvári. 2020. "Bitcoin: Digital Illusion or a Currency of the Future?" *Financial and Economic Review, Vol. 19 Issue 1* 132–153.

Carvalho, Diogo V, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. "Machine Learning Interpretability: A Survey on Methods and Metrics." *Electronics 2019, 8, 832* 34.

Chan, Ernest P. 2021. *Quantitive Trading, How to Build Your Own Algorithmic Trading Business.* New Jersey: John Wiley & Sons, Inc., Hoboken.

Cheah, Eng-Tuck, and John Fry. 2015. "Speculative bubbles in Bitcoin markets? An empirical investigation." *Economics Letters* 130: 32-36.

Chen, Zheshi, Chunhong Li, and Wenjun Sun. 2020. “Bitcoin price prediction using machine learning: An approach to sample dimension engineering.” *Journal of Computational and Applied Mathematics* 365.

Chevallier, Julien, Bangzhu Zhu, and Lyuyuan Zhang. 2021. "Forecasting Inflection Points: Hybrid Methods with Multiscale Machine Learning Algorithms." *Computational Economics* 537-575.

Chong, Eunsuk, Chulwoo Han, and Frank Chongwoo Park. 2017. "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies." *Expert Systems with Applications* 83: 187-205.

Chu, Jeffrey, Stephen Chan, and Yuanyuan Zhang. 2020. "High frequency momentum trading with cryptocurrencies." *Research in International Business and Finance 52* 101176.

Chu, Xiaojun, Chongfeng Wu, and Jianying Qiu. 2015. "A nonlinear Granger causality test between stock returns and investor sentiment for Chinese stock market: a wavelet-based approach." *Applied Economics, 48:21* 1915-1924.

Chung, Hyejung, and Kyung-shik Shin. 2018. "Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction." *Sustainability* 10(10): 3765-3783.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." *NIPS 2014.*

Claesen, Marc, Jaak Simm, and Dusan Popovic. 2014. *Optunity.* Accessed December 14, 2021. https://optunity.readthedocs.io/en/latest/user/solvers/TPE.html.

Clower, Eric. 2021. *APTECH.* October 4. Accessed October 20, 2021. https://www.aptech.com/blog/introduction-to-granger-causality/.

Cocco, Luisanna, Roberto Tonelli, and Michele Marchesi. 2021. "Predictions of bitcoin prices through machine learning based frameworks." *PeerJ Computer Science* 7:e413.

Cohen, Jerome Bernard, Edward D Zinbarg, and Arthur Zeikel. 2014. *Investment analysis and portfolio management^.* Homewood: R.D. Irwin.

Coinbound. 2021. *Coinbound.* Accessed October 25, 2021. https://coinbound.io/best-crypto-influencers-on-twitter/.

CoinMarketCap. 2021. *CoinMarketCap.* December 07. Accessed December 07, 2021. https://coinmarketcap.com/de/.

Cointaxlist. 2021. *Cointaxlist.* 12 14. Accessed 12 14, 2021. https://cointaxlist.com/blog/is-portugal-really-a-tax-haven-for-crypto.

Comment_Picker. 2021. *Twitter ID & Follower Count.* Accessed October 10, 2021. https://commentpicker.com/twitter-id.php.

Cybenko, George. 1989. "Approximation by superpositions of a sigmoidal function." *Mathematics of Control, Signals and Systems* 2: 303–314.

Dastgir, S., E. Demir, G. Downing, G. Gozgor, and C. K. Lau. 2019. "The causal relationship between Bitcoin attention and Bitcoin returns: Evidence from the Copula-based Granger causality test." *Finance Research Letters Volume 28* 160-164.

de Souza, Matheus José Silva, et al. 2019. "Can artificial intelligence enhance the Bitcoin bonanza." *The Journal of Finance and Data Science 5,* 83-98.

Dickey, David A., and Wayne A. Fuller. 1981. "Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root." *Econometrica Vol. 49, No. 4* 1057-1072.

Diks, Cees, and Valenty Pancheko. 2006. "A new statistic and practical guidelines for nonparametric Granger causality testing." *Journal of Economic Dynamics & Control 30* 1647–1669.

Dorsey, Jack. 2006. *Twitter.* March 21. Accessed October 20, 2021. https://twitter.com/jack/status/20.

Dr. Rajab, Muhanad, and Feda Hassan Jahjah. 2020. "Impact of Twitter Sentiment Related to Bitcoin on Stock Price Returns." *University of Baghdad Engineering Journal 26 (6)* 60-71.

Dutta, Aniruddha, Saket Kumar, and Meheli Basu. 2020. "A Gated Recurrent Unit Approach to Bitcoin Price Prediction." *Journal of Risk and Financial Management* 13(2):23.

Eisner, Ben, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. "emoji2vec: Learning Emoji Representations from their Description." *4th*

International Workshop on Natural Language Processing for Social Media at EMNLP 2016 7.

Elbagir, Shihab, and Jing Yang. 2019. "Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment." *Proceedings of the International MultiConference of Engineers and Computer Scientists 2019* 5.

Elsaraiti, Meftah, and Adel Merabet. 2021. "A Comparative Analysis of the ARIMA and LSTM Predictive." *Energies* 14: 6782-6798.

Fang, Fan, Carmine Ventrea, Michail Basiosb, Hoiliong Kong, Leslie Kanthan, David Martinez-Rego, Fan Wu, and Li Lingbo. 2020. "Cryptocurrency Trading: A Comprehensive Survey." *arXiv preprint arXiv:2003.11352.*

Farrar, Donald E., and Robert R. Glauber. 1967. "Multicollinearity in Regression Analysis: The Problem Revisited." *The Review of Economics and Statistics* 49(1): 92-107.

FIA. 2021. *Commodities.* September 24. Accessed October 21, 2021. https://www.fia.org/commodities.

Finance, Yahoo. 2021. *Top Cryptos by Volume.* November 11. Accessed November 11, 2021. https://ca.finance.yahoo.com/u/yahoo-finance/watchlists/crypto-top-volume-24hr.

Fischer, Thomas Günter, Christopher Krauss, and Alexander Deinert. 2019. "Statistical Arbitrage in Cryptocurrency Markets." *Journal of Risk and Financial* 12(1), 31.

Garcia, David, and Frank Schweitzer. 2015. "Social signals and algorithmic trading of Bitcoin." *Royal Society open science 2.9.*

Gensis, Cloud. 2021. *Pricing.* December 12. Accessed December 12, 2021. https://www.genesiscloud.com/pricing.

Gil-Alana, Luis Alberiko, Emmanuel Joel Aikins Abakah, and María Fátima Romero Rojo. 2020. "Cryptocurrencies and stock market indices. Are they related?" *Research in*

INternational Business and Finance 51 (2020) 101063 11. doi:https://doi.org/10.1016/j.ribaf.2019.101063.

Granger, C. W. J. 1969. "Investigating Causal Relations by Econometric Models and Cross-spectral Methods." *Econometrica Vol. 37, No. 3* 424-438.

Greaves, Alex, and Benjamin Au. 2015. "Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin." *Computer Science.*

Gupta, Nitin, Shashank Mujumdar, Hima Patel, Satoshi Masuda, Naveen Panwar, Sambaran Bandyopadhyay, Sameep Mehta, et al. 2021. "Data Quality for Machine Learning Tasks." *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21).* Singapore: ACM. 2.

Hannun, Awni, Guo Chuan, and van der Maaten Laurens. 2021. "Measuring Data Leakage in Machine-Learning Models with Fisher Information." *Conference on Uncertainty in Artificial Intelligence.*

Heimerl, F., Lohmann S., S. Lange, and Ertl T. 2014. "Word Cloud Explorer: Text Analytics Based on Word Clouds." *47th Hawaii International Conference on System Sciences.* 1833-1842.

Henckaerts, Roel, Marie-Pier Côté, Katrien Antonio, and Roel Verbelen. 2021. "Boosting Insights in Insurance Tariff Plans with Tree-Based Machine Learning Methods." *North American Actuarial Journal* (25:2): 255-285.

Hochreiter, Sepp. 1997. "Long Short-term Memory." *Neural Computation* 9(8): 1735-1780.

Hochreiter, Sepp. 1998. "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions." *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems* 6(2): 107-116.

Hoi, Steven C.H., Doyen Sahoo, Jing Lu, and Peilin Zhao. 2021. "Online learning: A comprehensive survey." *Neurocomputing* 459: 249-289.

Huang, Jing-Zhi, William Huang, and Jun Ni. 2019. "Predicting bitcoin returns using high-dimensional technical indicators." *The Journal of Finance and Data Science 5* 140-155.

Hutto, C.J., and Eric Gilbert. 2014. "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text." *Eighth International AAAI Conference on Weblogs and Social Media* 216-225.

Ibrahim, Ahmed, Rasha Kashef, Menglu Li, Esteban Valencia, and Eric Huang. 2020. "Bitcoin Network Mechanics: Forecasting the BTC Closing Price Using Vector Auto-Regression Models Based on Endogenous and Exogenous." *Journal of Risk and Financial Management* 13(9), 189.

Jacobs, AM., B. Herrmann, G. Lauer, J. Lüdtke, and S. Schroeder. 2020. "Sentiment Analysis of Children and Youth Literature: Is There a Pollyanna Effect?" *Frontiers in Psychology. Frontiers Media SA.* 8.

Jacobs, Arthur M. 2019. "Sentiment Analysis for Words and Fiction Characters From the Perspective of Computational (Neuro-)Poetics." *Frontiers in Robotics and AI. Frontiers Media SA.* 13.

Jain, Achin, and Vanita Jain. 2019. "Sentiment classification of twitter data belonging to renewable energy using machine learning." *Journal of Information and Optimization Sciences (Vol. 40, Issue 2)* 521-533.

Ji, Suhwan, Jongmin Kim, and Hyeonseung Im. 2019. “A Comparative Study of Bitcoin Price Prediction Using Deep Learning.” *Mathematics* 7(10): 898-918.

John, Alun, Samuel Shen, and Tom Wilson. 2021. *Future of Money.* September 24. Accessed December 07, 2021. https://www.reuters.com/world/china/china-central-bank-vows-crackdown-cryptocurrency-trading-2021-09-24/.

Khoo, Christopher, and Sathik Basha Johnkhan. 2018. "Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons." *Journal of Information Science, Volume 44 (4): 22 - Aug 1, 2018.*

Kim, J., H. Wimmer, Liu Kim, and S. Kim. 2021. "A Streaming Data Collection and Analysis for Cryptocurrency Price Prediction using LSTM." *IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD)* 45-52.

Kraaijeveld, Oliver, and Johannes De Smedt. 2020. "The predictive power of public Twitter sentiment for forecasting." *Journal of International Financial Markets, Institutions and Money Volume 65* 22.

Kristjanpoller, Werner, and Marcel C. Minutolo. 2018. "A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis." *Expert Systems With Applications 109* 1-11.

Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling.* New York: Springer-Verlag.

Kunesh, Andrew. 2020. *Traject.* February 25. Accessed November 25, 2021. https://fanbooster.com/blog/social-media-post-lengths/.

Lahmiri, Salim, and Stelios Bekiros. 2021. "Deep Learning Forecasting in Cryptocurrency High-Frequency Trading." *Cognitive Computation* 13: 485–487.

Li, Susan. 2020. *towardsdatascience.* December 23. Accessed November 26, 2021. https://towardsdatascience.com/a-quick-introduction-on-granger-causality-testing-for-time-series-analysis-7113dc9420d2.

Lim, Bryan, and Stefan Zohren. 2020. “Time Series Forecasting With Deep Learning: A Survey.” *Philosophical Transactions A.*

Liu, Yang, Jian-Wu Bi, and Zhi-Ping Fan. 2017. "Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms." *Expert Systems with Applications (Vol. 80)* 323–339.

Liu, Zhen, Ruoyu Wang, Ming Tao, and Xianfa Cai. 2015. "A class-oriented feature selection approach for multi-class imbalanced network traffic datasets based on local and global metrics fusion." *Neurocomputing* 168: 365–381.

Lu, Haitian, Bingzhong Wang, Qing Wu, and Jing Ye. 2020. "Fintech and the Future of Financial Service: A Literature Review and Research Agenda." *China Accounting and Finance Review* 107-136.

Mallqui, Dennys C.A., and Ricardo A.S. Fernandes. 2019. "Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques." *Applied Soft Computing Journal* 75: 596–606.

Matsuo, Yutaka. 2003. "Prediction, Forecasting, and Chance Discovery." In *Advanced Information Processing*, by Y. Ohsawa and P. McBurney, 30-43. Berlin, Heidelberg: Springer.

McCabe, Michael. 2019. *intelligencefusion.* June. Accessed November 15, 2021. https://www.intelligencefusion.co.uk/insights/resources/article/top-30-most-followed-news-accounts-on-twitter/.

Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. 2014. "Sentiment analysis algorithms and applications: A survey." *Ain Shmas Engineering Journal Volume 5, Issue 4* 1093-1113.

Mitra, A. 2020. "Sentiment Analysis Using Machine Learning Approaches (Lexicon based on movie review dataset)." *Journal of Ubiquitous Computing and Communication Technologies (Vol. 2, Issue 3)* 145–152.

Mordor Intelligence. 2020. *Algorithmic Trading Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026).* Online: Mordor Intelligence.

Mudassir, Mohammed, Shada Bennbaia, Devrim Unal, and Mohammad Hammoudeh. 2020. “Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach.” *Neural Computing and Applications.*

Munim, Ziaul Haque, Mohammad Hassan Shakil, and Ilan Alon. 2019. “Next-Day Bitcoin Price Forecast.” *Journal of Risk and Financial* 12(2.

Murphy, J. J. 1999. "Technical analysis of the financial markets: A comprehensive guide to trading methods and applications." *Penguin.*

Nakano, Masafumi, Akihiko Takahashi, and Soichiro Takahashi. 2018. “Bitcoin technical trading with artificial neural network.” *Physica A* 510: 587–609.

Neufeld, Dorothy. 2021. *visualcapitalist.* January 27. Accessed November 25, 2021. https://www.visualcapitalist.com/the-50-most-visited-websites-in-the-world/.

Niu, Tong, Jianzhou Wang, Haiyan Lu, Wendong Yang, and Pei Du. 2020. "Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting." *Expert Systems With Applications* 148.

Ossinger, Joanna, and Jim Silver. 2021. *Bloomberg.* June 14. Accessed Oktober 17, 2021. https://www.bloomberg.com/news/articles/2021-06-13/musk-says-tesla-sold-about-10-of-bitcoin-holdings#:~:text=Elon%20Musk%20said%20Tesla%20Inc,use%20about%2050%25%20clean%20energy.&text=Musk%20has%20whipsawed%20Bitcoin%20and,in%20the%20past%20few%20months.

Pano, Toni, and Rasha Kashef. 2020. "A Complete VADER-Based Sentiment Analysis of Bitcoin (BTC) Tweets during the Era of COVID-19." *In Big Data and Cognitive Computing (Vol. 4, Issue 4, p. 33)* 17.

Pant, Dibakar Raj, Prasanga Neupane, Anuj Poudel, Anup Kumar Pokhrel, and Bishnu Kumar Lama. 2018. "Recurrent Neural Network Based Bitcoin Price Prediction by Twitter

Sentiment Analysis." *IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* 6.

Patel, Mohil Maheshkumar, Sudeep Tanwar, Rajesh Gupta, and Neeraj Kumar. 2020. "A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions." *Journal of Information Security and Applications 55 (2020) 102583* 12.

Petukhina, Alla A., Raphael C. G. Reule, and Wolfgang Karl Härdle. 2021. "Rise of the machines? Intraday high-frequency trading patterns of cryptocurrencies." *The European Journal of Finance* 8-30.

Phaladisailoed, Therasak, and Thanisa Numnonda. 2018. "Machine Learning Models Comparison for Bitcoin Price Prediction." *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE).* Bali, Indonesia.

Raschka, Sebastian. 2015. *Python Machine Learning.* Birmingham: Packt Publishing Ltd.

Saxena, Anshul, and T.R. Sukumar. 2018. "Predicting bitcoin price using lstm And Compare its predictability with arima model." *International Journal of Pure and Applied Mathematics* 119(17): 2591-2600.

Schafer, Cullen. 1993. "Technical Note: Selecting a Classification Method by Cross-Validation." *Machine Learning* 13: 135-143.

Sebastião, Helder, and Pedro Godinho. 2021. "Forecasting and trading cryptocurrencies with machine learning under changing market." *Financial Innovation* 7:3.

Shintate, Takuya, and Lukáš Pichl. 2019. "Trend Prediction Classification for High Frequency Bitcoin Time Series with Deep Learning." *Journal of Financial Risk and Financial Management* 1-15.

Silver, Caleb. 2020. *Investopedia.* December 24. Accessed October 21, 2021. https://www.investopedia.com/insights/worlds-top-economies/.

Sovbetov, Yhlas. 2018. "Factors Influencing Cryptocurrency Prices: Evidence from Bitcoin, Ethereum, Dash, Litcoin, and Monero." *Journal of Economics and Financial Analysis* 2(2): 1-27.

Srishthi, Vashishtha, and Susan Seba. 2020. "Fuzzy Interpretation of Word Polarity Scores for Unsupervised Sentiment Analysis." *11th International Conference on Computing, Communication and Networking Technologies (ICCCNT).* Kharagpur, India: IEEE.

Statista. 2021. "Biggest cryptocurrency exchanges based on 24h volume in the world on November 22, 2021." *Statista.* November 22. Accessed December 7, 2021. https://www.statista.com/statistics/864738/leading-cryptocurrency-exchanges-traders/.

Staudemeyer, Ralf C., and Eric Rohstein Morris. 2019. “Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks.”

Stenqvist, Evita, and Jacob Lönnö. 2017. *Predicting Bitcoin price fluctuation with Twitter sentiment analysis.* Stockholm, Sweden.

Swarnkar, Naman. 2020. *quantinsti.* May 01. Accessed October 12, 2021. https://blog.quantinsti.com/vader-sentiment/.

Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. "Lexicon-Based Methods for Sentiment Analysis." *Computational Linguistics. MIT Press - Journals* 267–307.

Tao, Ran, Chi-Wei Su, Yidong Xiao, Ke Dai, and Fahad Khalid. 2021. “Robo advisors, algorithmic trading and investment management: Wonders of fourth industrial revolution in financial markets.” *Technological Forecasting & Social Change 163* 6.

Timmins, Fiona, and Catherine Mccabe. 2005. “How to conduct an effective.” *art & science* 20(11): 41-47.

Tsai, Chih-Fong, and Yu-Chieh Hsiao. 2010. "Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches." *Decision Support Systems* 50: 258–269.

Turner, Zane, Kevin Labille, and Susan Gauch. 2021. "Lexicon-based sentiment analysis for stock movement prediction." *Journal of Construction Materials. Institute of Construction Materials* 12.

Twitter. 2021. *Fiscal Year 2020 Annual Report.* Annual Report, Twitter, Inc. Accessed November 20, 2021. https://s22.q4cdn.com/826641620/files/doc_financials/2020/ar/FiscalYR2020_Twitter_Annual_Report.pdf.

Twitter. 2021. *Twitter Developer.* Accessed December 14 , 2021. https://developer.twitter.com/en/docs/twitter-api/tweets/lookup/migrate/standard-to-twitter-api-v2.

Twitter. 2021a. *Developer Platform.* Accessed November 29, 2021. https://developer.twitter.com/en/docs/twitter-api.

Twitter. 2021b. *Developer Platform.* Accessed December 2, 2021. https://developer.twitter.com/en/docs/twitter-api/tweets/search/integrate/build-a-query.

Twitter. 2021c. *Developer Platform.* December 12. Accessed December 2021, 2021. https://developer.twitter.com/en/pricing/search-30day.

Twitter. 2021d. *Developer Platform.* Accessed November 25, 2021. https://developer.twitter.com/en/docs/counting-characters.

Twitter. 2021e. *Developer Platform.* Accessed November 25, 2021. https://developer.twitter.com/en/docs/twitter-api/rate-limits.

Valencia, Franco, Alfonso Gómez-Espinosa, and Benjamín Valdés-Aguirre. 2019. "Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning." *Entropy* 12.

Vo, Au, and Christopher Yost-Bremm. 2018. “A High-Frequency Algorithmic Trading Strategy for.” *Journal of Computer Information Systems* 60(6): 555-568.

Wall Street Journal. 2021. *Wall Street Journal.* Accessed 12 14, 2021. https://www.wsj.com/market-data/quotes/TSLA/historical-prices.

Walther, Thomas, Klein, Tony, & Bouri, Elie. 2019. "Exogenous drivers of Bitcoin and Cryptocurrency volatility–A mixed data sampling approach to forecasting." *Journal of International Financial Markets.*

Wang, Tai-Yue, and Huei-Min Chiang. 2007. “Fuzzy support vector machine for multi-class text categorization.” *Information Processing and Management* 43: 914–929.

Wei, Wang. 2016. *Achieving Inclusive Growth in China Through Vertical Specialization.* Elsevier Science & Technology.

Williams, Sarah C., and Anita K. Foster. 2011. “Promise Fulfilled? An EBSCO Discovery Service.” *Journal of Web Librarianship* 5(3): 179-198.

Woebbeking, Fabian. 2021. "Cryptocurrency volatility markets." *Digital Finance* 273–298.

Wollams, Ben. 2021. *The Drum.* 09 06. Accessed November 25, 2021. https://www.thedrum.com/opinion/2021/09/06/why-twitter-making-comeback-one-the-most-popular-social-channels.

Xiaolei, Sun, Liu Mingxi, and Sima Zeqian. 2020. "A novel cryptocurrency price trend forecasting model based on LightGBM." *Finance Research Letters* 32.

Zhou, Liyi, Kaihua Qin, Christof Ferreira Torres, Duc V Le, and Arthur Gervais. 2021. "High-Frequency Trading on Decentralized On-Chain Exchanges." *IEEE Symposium on Security and Privacy (SP)* 18.