Bitcoin price movement direction prediction: A comparison between Decision Tree, SVM and LSTM

Henrique Oliveira

Dissertation written under the supervision of Dan Tran

Dissertation submitted in partial fulfilment of requirements for the MSc in Finance, at the Universidade Católica Portuguesa, January 2020.

Abstract

Title: Bitcoin price movement direction prediction: A comparison between Decision Tree, SVM and LSTM

Author: Henrique Martim Aguiar Sobral Oliveira

This thesis explores the ability of machine learning methods combined with the use of blockchain characteristics, a stable coin proxy and technical indicators to predict the price movement of Bitcoin correctly. The experimental work attempts to prove that machine learning models can outperform a Buy & Hold strategy with the selected features and compares the classification performance of Decision Tree, a Support Vector Machine and a Long Short-Term Memory Recurrent Neural Network. The results from the feature selection supported the findings of existing literature. Additionally, the results suggest that mining difficulty and blockchain characteristics that measure transaction activity can provide supplementary information about Bitcoin. In terms of the performance of the models, results showed that a Decision Tree model has the propensity to overfit due to its low complexity and the type of data. The results from the SVM showed that although it achieved the highest accuracy, it only managed to identify the overall trend and therefore it was not able to beat a Buy & Hold strategy. Even though the LSTM did not outperform the benchmarks, it was the model that showed the most promising results, and its performance would likely improve by additional hyperparameter tunning. Therefore, the inability to outperform the benchmarks was not conclusive. Finally, the fact that the Logistic regression model was able to outperform the Buy & Hold strategy in terms of returns and volatility leads us to conclude that machine learning methods might be effective in predicting bitcoin price movement with the selected features.

Keywords: Bitcoin; Price movement prediction; Machine learning; Blockchain; Technical Analysis; Stable coin.

Resumo

Título: Previsão da direção do movimento da Bitcoin: Comparação entre Árvores de Decisão, SVM e LSTM

Autor: Henrique Martim Aguiar Sobral Oliveira

Esta tese explora a capacidade dos métodos de machine learning em conjunto com o uso de características da blockchain, um proxy de stable coins e indicadores técnicos para prever corretamente o movimento do preço da Bitcoin. Esta tese tenta provar que modelos de machine learning podem superar uma estratégia Buy & Hold com as variáveis selecionadas e compara o desempenho de uma árvore de decisão, uma SVM e uma rede neural recorrente LSTM. Os resultados da seleção de variáveis apoiam a literatura existente. Além disso, os resultados também sugerem que a dificuldade de mineração e características do blockchain relacionadas com atividade transacional podem fornecer informações complementares sobre a Bitcoin. Em termos de desempenho dos modelos, os resultados mostraram que o modelo de árvore de decisão tem predisposição para sobre ajustar devido à sua baixa complexidade e ao tipo de dados. Os resultados do SVM mostraram que, embora tenha alcançado a maior precisão, apenas conseguiu identificar a tendência geral e, portanto, não foi capaz de superar a estratégia Buy & Hold. Mesmo que o LSTM não tenha superado os benchmarks, foi o modelo que mostrou os resultados mais promissores, e seu desempenho provavelmente melhoraria realizando ajuste adicional dos híper-parâmetros. Logo, a incapacidade de superar o benchmark não foi conclusivo. Finalmente, o facto de o modelo de regressão logística ter sido capaz de superar a estratégia Buy & Hold em termos de retornos e volatilidade, leva-nos a concluir que machine learning pode ser eficaz na previsão da Bitcoin com as variáveis selecionadas.

Palavras-chave: Bitcoin; Previsão do movimento do preço; Machine learning; Blockchain; Análise Técnica; Stable coin.

Acknowledgements

As humans, we need other humans to thrive. In the making of this dissertation, as well as in my academic career, there were people along the way that deserved recognition and acknowledgement for helping me accomplish my goals.

First of all, I have to thank my family, who always supported my choices and whose motivation was vital to complete this thesis—especially my mother, who provided me with a special kind of support in the form of food.

Second, I like to thank my friends and colleagues that provided me with much-needed breaks and intellectual knowledge. In times of pessimism and stress, Catarina Fatela was someone who helped me get through them with her advice and conversations. I also must thank Nelson Nunes for his time and patience.

Third, I would like to thank all the teachers and staff whom I had the pleasure to learn from. A special thanks to my supervisor, Dan Tran, who had the difficult task of teaching and guiding five students who had no previous contact with machine learning.

Finally, I want to dedicate this thesis to my grandfather, Fernando Oliveira, who passed away from COVID-19 in October and whose presence is truly missed.

Table of Contents

1. Introduction

1.1. Context and Motivation

Bitcoin entered the public discourse in 2009 as the world's first decentralized currency of the digital age. When it first appeared, Bitcoin's function was to provide a secure way for people to make digital transactions in anonymity and without the oversight of the traditional banking system. Market participants still disagree on Bitcoins' ability to serve as a currency. The risk derived from a lack of regulation, high energy consumption, slow processing capabilities, high transaction fees, and large price volatility still prevents it from being accepted by financial institutions and the general public alike.

Bitcoin has managed to keep its position as a point of reference in the cryptocurrency market despite the growing competition from the increasing number of new cryptocurrencies. This is not only due to being the first comer but also because of the high appreciation in value that is now associated with the digital currency. However, Bitcoin is also known for experiencing high volatile periods. These price fluctuations have not only plagued investors in recent years but also led some to consider Bitcoin as just a speculative asset (Yermack, 2015). Therefore, a need arises to construct a strategy that takes advantage of the continuous appreciation of Bitcoin but manages to have reduced volatility. Furthermore, due to the impact Bitcoin has in the cryptocurrency market, shown by the CAPM metrics (Appendix 1), correctly predicting the price movement of Bitcoin can inform us of the direction of the cryptocurrency market in general.

However, the source of Bitcoin's value is still debated. A lack of fundamental analysis based on economic and accounting information has prevented Bitcoin and other cryptocurrencies from being analyzed like other traditional financial assets, such as stocks or currencies (Detzel et al., 2020). This, in conjunction with the different manners investors, mostly individuals, perceive Bitcoin's fair market value, and the influence of speculators and momentum traders creates temporary shocks, which in turn makes "noise", resulting in the mispricing of Bitcoin and the failure of the efficient market hypothesis. Detzel et al. (2020) also pointed to the usefulness of technical indicators in predicting "hard to value" assets, such as Bitcoin.

In an attempt to model Bitcoin as a currency, researchers such as Cong et al. (2020), Bhambhwani et al. (2019), and Sockin & Xiong (2018) have highlighted the importance of user adoption and the computing power of the network as useful measures of supply and demand.

This thesis expands on these ideas of using the number of addresses and hashing power of the network and considers other relevant blockchain characteristics as complementary measures of supply and demand and network activity to access the viability of blockchain features to describe Bitcoin pricing. Additionally, following the research paper from Griffin & Shams (2020) which studied the impact of the issuance of the stable coin Tether in the 2017 boom, and due to the growing importance stable coins now have in the cryptocurrency world, Tether's market capitalization will also be included as a variable in order to predict the price movement and access if stable coins due in fact have an impact on Bitcoin.

To make accurate predictions of whether Bitcoin's price will increase or decrease, the thesis will use Machine Learning algorithms. Machine learning has been known to make accurate predictions in several fields, finance included, and improve on more traditional statistical methods. While conventional statistical methods impose structures to models, for example, a factor model that uses a linear regression method, machine learning methods allow the models to learn the specific data structure without impositions. However, there is still no consensus on which models yields the best results, especially in this case of price movement prediction, where correctly predicting periods with high absolute returns is considered more important. Therefore, in order to compare the performance of models with different complexities this thesis analyses the difference in results of a simple Decision Tree model, a Support Vector Machine Classifier, an older and more researched model that requires less computational power while still producing significant accuracy and a modified version of a recurrent neural network called Long Short-Term Memory, which has the ability to keep in memory past outcomes.

1.2. Goals and document structure

This dissertation's primary goal is to predict the price movement of Bitcoin to a degree that can outperform a standard Buy & Hold in terms of returns. With this purpose, hourly data related to blockchain characteristics, Tether issuance, and Bitcoin price was gathered from 23/02/2018 to 01/11/2020. In addition, by performing feature engineering on Bitcoin's price and volume information, technical indicators were added to help predict the price movement of a 24-hour period. This dissertation also covers three additional questions:

- Were the features chosen relevant in describing Bitcoin's price movement?
- Which of the chosen machine learning methods (Decision Tree, Support Vector Machine, and LSTM neural network) better understands the problem at hand?

- Can these models produce a strategy with improved risk-adjusted return metrics compared to a buy and hold?

Regarding the document structure, the next chapter discusses the main literature on Bitcoin, the developments of machine learning in price prediction, and the literature on the features considered. In chapter 3, an assessment of the data gathered is done as well as some statistical analysis of the features. Chapter 4 introduces the principles of machine learning and develops on the methodology used to label the data, select the most essential features as well as explaining the models and metrics used to evaluate their performance. Chapter 5 presents and analyses the obtained results in addition to answering the research questions. Finally, Chapter 7 concludes this report.

As a result, this paper focuses on the merging of features from technical analysis, bitcoin blockchain characteristics, and Tether's issuance to predict the direction of the price of Bitcoin and comparing the classification metrics and profitability from a relatively recent Recurrent Neural Network method (LSTM), a Support Vector Machine model and a Decision Tree.

2. Literature Review

2.1. Bitcoin in the financial world

Bitcoin has attracted the attention of researchers, investors, and the media for its innovative blockchain technology and its ability to perform as the world's first decentralized digital currency. Researchers have constructed economic models around Bitcoin as a currency in an attempt to shed light on the latter. Schilling & Uhlig (2019) provide a model of an endowment economy with two competing currencies (Dollar, Bitcoin) serving as a medium of exchange and monetary policy implications under different scenarios. They were able to provide conditions, under which no speculation in Bitcoin arises. Cong et al., (2020) contribute with a more generalized perspective on the adoption and valuation dynamics of cryptocurrencies and tokens by providing the first dynamic pricing model that takes into consideration the user-base externality and endogenous adoption. The paper draws on the previous work done by Athey et al., (2016) where they develop a model of user adoption and use of virtual currencies (such as Bitcoin), also focusing on the dynamics of adoption in the presence of frictions arising from exchange rate uncertainty, ranging from the risk of failure of the underlying technology of cryptocurrencies to the fact that the value of the technology is affected by the adoption of other users.

However, Yermack (2015) argues that Bitcoin is not yet a proper currency according to the criteria widely used by economists, in which a currency has to be able to function as a medium of exchange, a unit of account, and a store of value. He suggests that due to the excessive volatility, Bitcoin is more consistent with the behaviour of a speculative investment than of a currency. Furthermore, When Foley et al. (2019) analyzed the types of users of Bitcoin, they estimated that 46% of the transactions were from funds that originated from illegal activity. According to them, cryptocurrencies are transforming the black markets by enabling "black e-commerce".

Research has also found that cryptocurrencies have low exposures to traditional asset classes (stocks, currencies, and commodities) and that the returns of cryptocurrency can be predicted by two factors specific to its markets -momentum and investors' attention (Liu & Tsyvinski, 2020). Moreover, unlike standard financial assets, such as bonds and stocks, cryptocurrencies' have few, if any, publicly available fundamentals (Detzel et al., 2020). This lack of fundamental analysis based on economic and accounting information makes this asset "hard to value"

(Detzel et al., 2020). The results from Detzel et al. showed that Bitcoin and stocks with hard-to-value fundamentals are predictable by price-to-MA ratios, and simple real-time strategies based on this predictability significantly outperform the buy-and-hold strategy.

Literature also looks into the types of investors that participate in the bitcoin market. Jain et al. (2019), per example, show that, unlike equities and forex that are dominated by large institutions, the Bitcoin market is dominated by individuals. Furthermore, Anne H. Dyhrberg et al. (2018) found that Bitcoin is investible, particularly for retail-sized trades.

2.2. Machine Learning in Quantitative Finance

Machine learning algorithms have been widely applied to make accurate predictions in many areas, including finance. By learning from past instances, machine learning programs can be produced to make predictions based on training data.

In recent years, the interest in applying machine learning methods to trading strategies grown. Lo et al. (2000) used a non-parametric kernel regression to incorporate technical patterns of a large number of stocks, such as head and shoulders and found that technical indicators provide incremental information for investors raising the possibility that technical analysis can add value to the investment process. Dempster et al. (2001) compared four methods for foreign exchange trading (reinforcement learning, genetic algorithms, Markov chain linear programming, and simple heuristic). They concluded that a combination of technical indicators using genetic algorithms leads to better performance than using only individual indicators. Dempster & Leemans (2006) reached a similar conclusion using adaptive reinforcement learning. Creamer & Freund (2010) used a multi-stock automated trading system that relies on a layered structure consisting of a machine learning algorithm, an online learning utility, and a risk management overlay. The underlying alternating decision tree algorithm, which is implemented with Logitboost, was able to generate abnormal returns during the test period. Creamer (2012) further explores the application of Logitboost with different machine learning models, using high-frequency data of the Dow Jones EURO STOXX 50 Index Futures (FESX) and the DAX Futures (FDAX) for March 2009. The trading algorithm generated positive returns for the two major European index futures and outperformed a buy & hold strategy. Rechenthin (2014) proposed a framework involving creating a pool of thousands of base classifiers (SVMs, ANNs, decision trees), and interchanging between models depending on performance.

Among the several machine learning models, Support Vector Machines (SVM) and Neural Networks (NN) have been predominantly studied by researchers in price movement prediction.

Gorenc Novak & Velušček (2016) used daily high prices, in an attempt to limit the noise that can arise from using closing prices, for day ahead forecasting with the use of SVM, while Huang et al., (2005) found that SVMs can outperform more traditional statistical methods in forecasting weekly movement direction of NIKKEI 225 Index. More recent contributions to the study of recurrent networks include the work done by Nelson et al. (2017), in which they trained LSTM networks for a number of BOVESPA stocks, and reported accuracy metrics of 53-55% for the next direction price forecasts, the paper from Borovkova & Tsiamas (2019) which created an ensemble of LSTM neural networks for high-frequency stock market classification and the research project done by Fischer & Krauss (2018) in which they found that a trading strategy based on the predictions of LSTM was more profitable compared to random forests, deep networks, and logistic regression.

Although many papers show promising results of machine learning models' predictive ability and technical analysis trading rules in the cryptocurrency market, ANGHEL (2020) performed a reality check test and noticed that practices like data-snooping had been found to corrupt results. Therefore, to ensure the findings' integrity, this thesis will follow a protocol for research in quantitative finance (Arnott et al., 2018).

2.3. Technical Analysis

The use of technical analysis has been limited in the finance literature because of its lack of a reliable statistical or mathematical approach, its highly subjective nature, and its visual nature. Studies from the 60s and 70s have analyzed trading rules based on technical indicators and did not find them profitable((Achelis, 2000), (Fama, 1970)). However, Allen & Karjalainen (1999) found profitable trading rules using a genetic algorithm for the S&P 500 with daily prices, despite not being consistently better than a buy-and-hold strategy in the out-of-samples.

However, traders and High-Frequency algorithms have been known to use some degree of technical analysis to predict the price of securities. Technical analysis or technical trading strategies is a price analysis technique that tries to predict the future direction of security prices through the study of past price data. It considers only the price actions of the financial series, on the assumption that all relevant factors are expressed in the past prices. Technical analysts extensively use technical indicators, which are usually mathematical transformations of price or volume. A survey on US foreign exchange traders indicated that technical trading best characterizes 30% of traders adding that, economic fundamentals are perceived to be more important at longer horizons (Cheung & Chinn, 2001). Further investigation suggests that

technical analysis is seen as an instrument informing traders about non-fundamental price determinants, which is more important in the short-run and inflexible exchange rate markets, where the degree of volatility cannot be explained by fundamentals alone (Menkhoff & Taylor, 2007).

There are hundreds of possible predictors that can be derived from technical analysis. They provide two useful advantages. First, they require only historical data of the price of the security and the volume transacted and thus are easy to implement. Second, the vast number of predictors is appropriate for machine learning since it is a tool better suited to deal with multiple features.

2.4. Blockchain characteristics

Due to a lack of Bitcoin fundamentals, this dissertation also uses two proven blockchain characteristics as additional factors to predict the performance of Bitcoin, the computing power (hash rate) and network (number of users) suggested by the research done by Bhambhwani et al. (2019). They found these features to be useful as measures of supply and demand.

Blockchain is what is called a Distributed Ledger Technology. Its peer-to-peer architecture is what makes it truly independent of any single person. Instead of copied or transferred, the digital assets are recorded in a way that is available to everyone and is constantly updated by those who contribute to mining "blocks", which are blocks of data regarding Bitcoin transfer information.

Mining hash-rate is a key security metric. The more hashing (computing) power in the blockchain, the greater the security and overall resistance to an attack. Bitcoin's supply functions in accordance with its innovative proof of work protocol. It rewards random miners who contribute to the computational power of the blockchain. Although Bitcoin's exact hashing power is unknown, it is possible to estimate it from the number of blocks being mined and the current block difficulty. However, literature related to the connection of computing power and the price of Bitcoin is still lacking.

The number of users is also important as it shows the size of the cryptocurrency network and its future success as a currency (Sockin & Xiong, 2018). Analogous to established fiat currencies, an extensive network is indicative of greater adoption of the cryptocurrency, larger liquidity, and a higher propensity for developers to build applications targeted to bitcoin wallets, leading to an increase in usability of the currency (Bhambhwani et al., 2019).

2.5. Bitcoin manipulation and Tether

Looking into the causes of such high volatility papers have examined the shock of hacking and fraudulent activity, considered as side effects of this unsupervised market. The "hack" on Mt. Gox, a leading exchange that by 2013 was handling 70% of Bitcoin volume, resulted in its bankruptcy and the loss of approximately $450 million worth of Bitcoin. Gandal et al. (2018) argue that fraudulent trading on the Mt. Gox exchange led to a significant spike in Bitcoin prices in late 2013.[1] Griffin & Shams (2020) research, which is cited by two pending class-action lawsuits against Tether and Bitfinex, investigate whether Tether, a digital currency pegged to the US dollar, influenced Bitcoin during the 2017 boom. Their results are most consistent with a supply-driven approach, where the cryptocurrency exchange Bitfinex, a sister company of Tether, prints the stable currency regardless of the demand from cash investors. The additional Tether supply can then be exchanged for Bitcoins creating an artificial surge of demand and consequently inflating its price.

Tether is the world's most used stable coin, a category of tokens that seek to avoid price fluctuations. Although Bitcoin is the largest cryptocurrency by market capitalization, Tether has the highest volume traded, with around 54% higher daily volume traded[2] than Bitcoin. During the summer of 2018, around 80% of Bitcoin volume was traded in Tethers.[3] In countries like China, where crypto exchanges are banned, Tether and other stable coins can be used as a pathway for investors to acquire Bitcoin or other cryptocurrencies. Due to the importance Tether has in the cryptocurrency ecosystem and as a vehicle of possible Bitcoin manipulation, Tether's market capitalization will be considered as a feature and a proxy for all stable coins.

[1] In the second-biggest hack in Bitcoin history, on August 2, 2016, the Bitfinex exchange announced that $72 million had been stolen from investor accounts, leading Bitcoin to plummet 20% in value.
[2] Average volume traded in the last 3 months as of 19/10/2020. Source: https://finance.yahoo.com/u/yahoo-finance/watchlists/crypto-top-volume-24hr/
[3] https://coinlib.io/coin/USDT/Tether

3. Data and Setup

In order to overcome the fact that Bitcoin is a relatively recent asset, this thesis uses hourly data from 23/02/2018 to 01/11/2020 to increase the amount of data available to feed the machine learning models. Glassnode was used to retrieve data regarding the characteristics of Bitcoin's blockchain, as well as Tether's market cap and other bitcoin indicators available. Data regarding Bitcoin's hourly price and volume traded was gathered on the site cryptodatadownload, taking into account the recommendations of Alexander & Dakos (2020) research, which points out the perils of using flawed databases and compares popular databases used in cryptocurrencies research. The exchange chosen was Coinbase for being one of the leading cryptocurrency exchanges.

3.1. Train/Test data and training the models

To assure that the models can perform on unseen data, the dataset was divided into two parts, the training data and the test data. The training data is comprised of 80% of the data gathered, ranging from 23rd of February of 2018 till 19th April of 2020, and will be used to analyze the data, perform feature and model selection and define the type of data scaling the test data will be subjected to.

The Bitcoin/ US Dollar price, shown in figure 1, was extremely volatile in the training period. The descriptive statistics in appendix 2, show an annualized volatility of 73.85% and returns that reached a maximum of 18.19% and a minimum of -37.17%, the latter caused by the Covid-19 crash. Nevertheless, the training period has a balanced amount of Buy (50.35%) and sell (49.65%) signals.

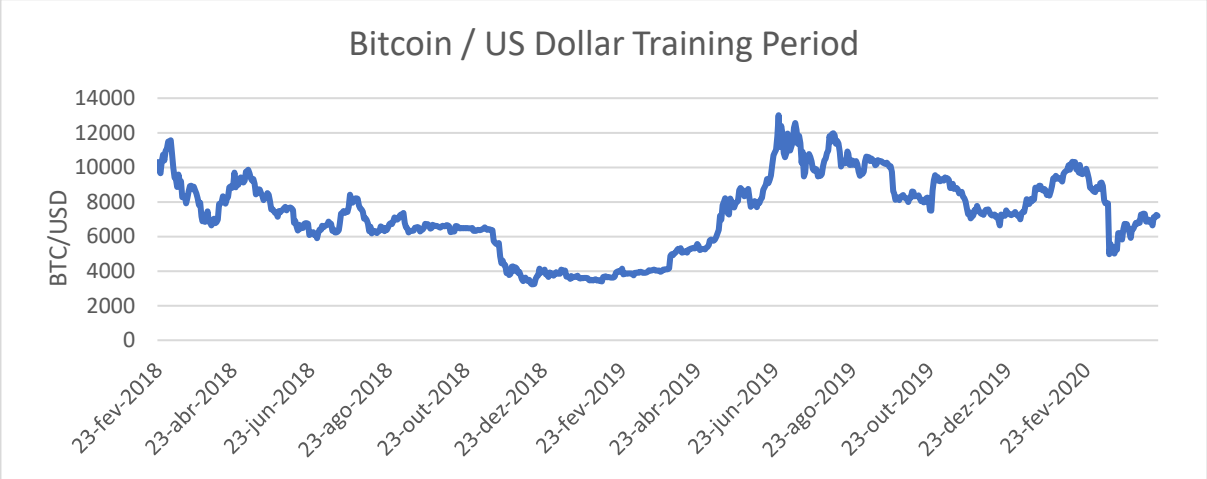


Figure 1 | BTC/USD exchange rate during the training period

The test data ranges from 20th of April of 2020 till 31st of October of 2020. Contrary to the training period, the test period is characterized by having experienced a single bullish trend which saw the price of Bitcoin rise around 96.93%.

For models to learn from the data gathered, one must first process the data which, most of the time, is not ready to be fed into the model. Data processing refers to a data mining technique that transforms raw data into an understandable and readable format. Therefore, I scaled the data by removing the mean and dividing it by the variance (Standard Scaler). Then I performed another scaling mechanism that transformed the data so that the values stayed in the range [-1,1] (Min Max Scaler)[4]. Missing values were not included as observations.

3.2. Feature description

Although the amount of data is important for machine learning, data quality, and its importance to describe the problem at hand is considered to play a bigger role in the model's performance. The choice of the features was motivated by the literature but also from Glassnode's available metrics which are considered relevant in the cryptocurrency world. Feature engineering was performed to arrive at the technical analysis features, which are all derived from candlestick patterns and volume (OHLCV). Features can be divided into four categories: OHLCV; Blockchain characteristics (Table 1); Stable Coin proxy (Tether Market Capitalization) and Technical Indicators (Table 2)[5].

Table 1 | Description of Blockchain characteristics.

Blockchain Characteristics	
Number of New Addresses	The number of unique addresses that appeared for the first time in a transaction of the native coin in the network.
Number of Active Addresses	The number of unique addresses that were active in the network either as a sender or receiver. Only addresses that were active in successful transactions are counted.
Mining Difficulty	The current estimated number of hashes required to mine a block. Bitcoin difficulty is often denoted as the

[4] For more information on StandardScaler and MinMaxScaler functions of Scikit-learn, see Pedregosa et al. (2011).
[5] Available in the library TA bukosabino/ta: Technical Analysis Library using Pandas and Numpy (github.com)

	relative difficulty with respect to the genesis block, which required approximately 2^{32} hashes. The values are denoted in raw hashes.
Market capitalization	The market capitalization (or network value) is defined as the product of the current supply by the current USD price.
Spent Output Profit Ratio (SOPR)	The Spent Output Profit Ratio (SOPR) is computed by dividing the realized value (in USD) divided by the value at creation (USD) of a spent output. Or simply: price sold / price paid. This metric was created by Renato Shirakashi.[6]
Mean Transaction Fees	The mean fee per transaction. Issued (minted) coins are not included.
Total Transaction Fees	The total amount of fees paid to miners. Issued (minted) coins are not included.
Mean Transfer Volume	The mean value of a transfer. Only successful transfers are counted.
Mean Hash Rate	The average estimated number of hashes per second produced by the miners in the network.
Number of Transactions per Second	The total amount of transactions per second. Only successful transactions are counted.
Mean Size of Transfers	The mean size of a transaction within the time period (in bytes).
Mean Value of Spent UTXOs	The mean number of coins in newly created UTXOs. The term UTXO refers to the amount of digital currency someone has left remaining after executing a cryptocurrency transaction.
Mean Value of Spent UTXOs	The mean number of coins in spent transaction outputs.
Sum of Mean Value of UTXOs	Sum of the mean value of UTXOs created and spent.

[6] For a detailed commentary see the post https://medium.com/unconfiscatable/introducing-sopr-spent-outputs-to-predict-bitcoin-lows-and-tops-ceb4536b3b9

Table 2 | Technical indicators.

Technical Indicators from the TA package	
Category	Technical indicators
Volume	• Money Flow Index (MFI) • Accumulation/Distribution Index (ADI) • On-Balance Volume (OBV) • Chaikin Money Flow (CMF) • Force Index (FI) • Ease of Movement (EoM, EMV) • Volume-price Trend (VPT) • Negative Volume Index (NVI) • Volume Weighted Average Price (VWAP)
Volatility	• Average True Range (ATR) • Bollinger Bands (BB) • Keltner Channel (KC) • Donchian Channel (DC) • Ulcer Index (UI)
Trend	• Simple Moving Average (SMA) • Exponential Moving Average (EMA) • Weighted Moving Average (WMA) • Moving Average Convergence Divergence (MACD) • Average Directional Movement Index (ADX) • Vortex Indicator (VI) • Trix (TRIX) • Mass Index (MI) • Commodity Channel Index (CCI) • Detrended Price Oscillator (DPO) • KST Oscillator (KST) • Ichimoku Kinkō Hyō (Ichimoku) • Parabolic Stop and Reverse (Parabolic SAR) • Schaff Trend Cycle (STC)
Momentum	• Relative Strength Index (RSI) • Stochastic RSI (SRSI)

	• True strength index (TSI) • Ultimate Oscillator (UO) • Stochastic Oscillator (SR) • Williams %R (WR) • Awesome Oscillator (AO) • Kaufman's Adaptive Moving Average (KAMA) • Rate of Change (ROC) • Percentage Price Oscillator (PPO) • Percentage Volume Oscillator (PVO)

Descriptive statistics can help provide basic information about the dataset variables and find potential relationships between variables. Although the tables in the Appendix (3,5,6,7,8) provide useful information with respect to the data gathered, it is also important to look at how the variables relate to each other. The correlation matrix in Appendix 9 is a simple but highly important way to describe the data. From the correlation matrix, we can observe that there is a diverse range of correlation coefficients. Not surprisingly, OHLC and Bitcoin's market capitalization variables experience high levels of correlation. This means that the addition of more than one of these variables will most likely not improve or add additional information to the model.

On the other hand, most of the features do not have a strong correlation with price except for Tether's market capitalization. Furthermore, it has a relatively high correlation with the mining metrics' features most likely because both the supply and the computing power have been increasing throughout time, as more and more people adopt cryptocurrencies. Technical indicators were excluded from this analysis because they are mostly derived from OHLCV.

3.3. Software

Data preparation and handling was conducted in Python 3.8, relying on the packages NumPy and pandas. The library TA was used to produce technical indicators. The deep learning LSTM networks are developed with Keras on top of Google TensorFlow. Moreover, the Sci-kit learn library was used to construct the SVM, Decision Tree, Logistic Regression, and classification metrics.

4. Methodology

4.1. Introduction to Machine Learning

Machine learning is a category of algorithms that enables the software to make accurate predictions by learning from data. The main premise of machine learning is to construct algorithms that can receive input data and use statistical analysis to predict an output. Today, machine learning can have many different applications, ranging from image recognition, medical diagnoses, self-driving cars, fraud detection, and many others, some of which are already incorporated into our daily lives. This thesis focuses, however, on supervised learning problems.

Supervised learning is characterized by having to learn a function that maps labelled input data to an output based on observations. The algorithm's goal is to adapt the function through the weight vector (function's parameters or coefficients) in order to determine the class labels for unseen observations correctly. An optimal scenario will allow the model to generalize from the training data to unseen instances without overfitting, a common peril in machine learning in which the model adapts to such an extent to the training data that negatively impacts the performance on unseen data.

How does the learning process, which updates the weights, occur? Each algorithm has a loss function, which may vary with the problem at hand, which measures the model's quality with respect to the prediction and the true label. The concept of loss functions is best understood by considering the example of a common loss function used in linear regression, the squared error loss (see Figure 2).

$$\mathcal{L}((\mathbf{x}, y), h) := (y - h(\mathbf{x}))^2.$$

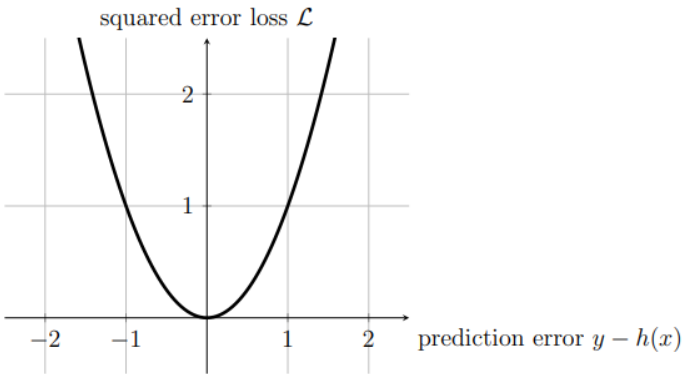


Figure 2 | A widely used choice for the loss function in regression problems (with label space Y = R) is the squared error loss L((x, y), h) := (y − h(x))2 . Note that in order to evaluate the loss function for a given hypothesis h, so that we can tell if h is any good, we need to know the feature x and the label y of the data point. From (Jung, 2018)

The objective of the algorithm is to achieve the lowest prediction error possible and to do that it performs an optimization algorithm called gradient descent. What gradient descent does is it attempts to reach the global minimum of the loss function by finding the values of the weight vector and moving in the opposite direction of the derivative of the cost function (gradient), exemplified in figure 3 and in equation 1.

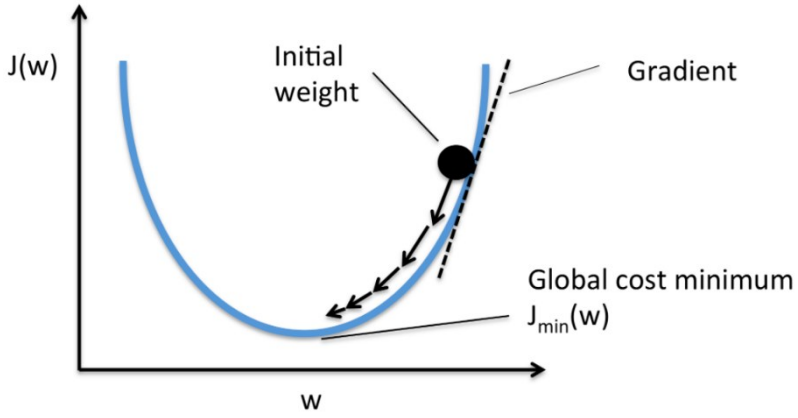
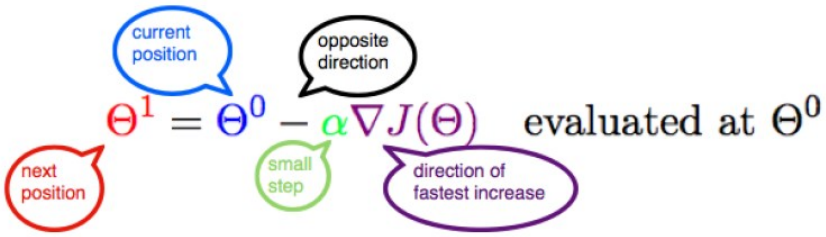


Figure 3 | Gradient Descend Visualization. Credit: rasbt.github.io

Equation 1 | Gradient descent. Θ is the weight vector and α is the learning rate, a parameter that decides the speed of the approximation to the minimum. From coursehero.com



4.2. Cross-Validation

Cross-Validation (CV) is an important step in machine learning for model validation and hyperparameter tuning. Its goal is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting and to give insight on how the model will generalize to an independent dataset. Furthermore, CV is particularly important in finding the best hyperparameters of the model. In contrast to the normal parameters (weights) which are learned by the model in the training phase, hyperparameters are parameters whose value is used to control the learning process, such as the previously mentioned learning rate. Therefore, instead of assigning values in a trial by error process, algorithms such as

Randomized Grid Search Cross-Validation, allows us to scan through ranges of possible values for hyperparameters.

Cross-validation is a useful process in cases where there is not a large dataset available. This is due to how CV functions. Instead of allocating a percentage of the data for validation purposes, which diminishes the amount of training data, CV splits the training set into different parts. There are different strategies with which we can divide the data, being one of the most common ones the K-Fold method, illustrated in figure 5.

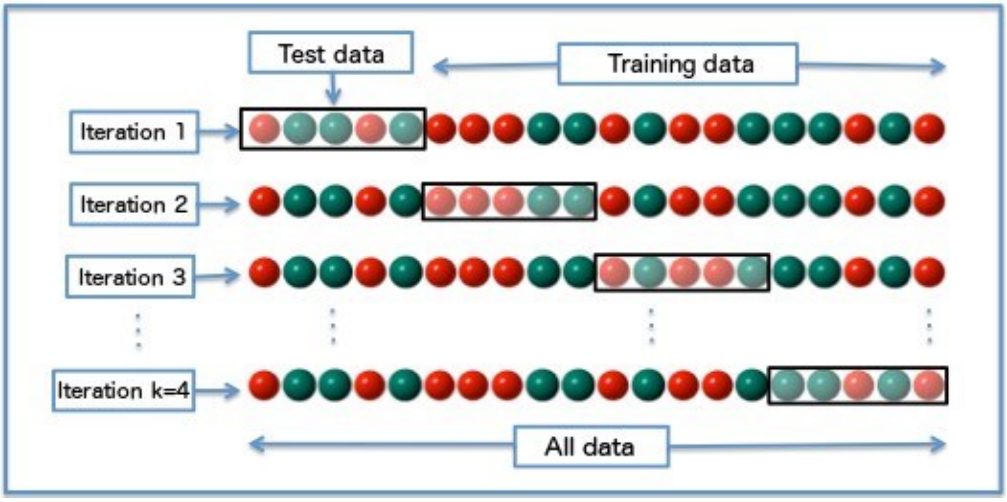


Figure 4 | Diagram of k-fold cross-validation with k=4. From towardsdatascience.com

For this thesis, a variation of the K-Fold method will be applied so that the model is trained on past data and only tested on future observations. This is done by slicing the train data with expanding windows, as in figure 4.

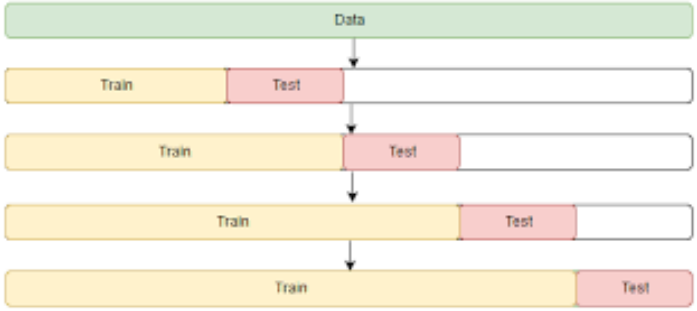


Figure 5 | Expanding window method. From stats.stackexchange.com

In order to find the best combination of hyperparameters Randomized Search CV, which performs a randomized search of combinations of hyperparameters, was used in 5 expanding windows.

4.3. Labelling

The thesis implements a Triple-Barrier Method, inspired by the book *Advances in Financial Machine Learning* by Peter Schwendner (2020), in order to classify the price movement in a range of 24 hours. The horizontal barriers simulate stop loss and take profit orders. If the take profit barrier is crossed, meaning the price increases by 5% compared to the initial value ($\mathrm{P}(0+t) > \mathrm{P}(0) * 1.05$), the label for the 24-hour period will automatically be set as Buy. If the opposite happens, meaning the price falls 5% compared to the initial value, the label will be set as Sell. The third barrier is a vertical one at the end of the period. This means that if those two conditions are not met, the classification of the period will be set as Buy if the return is positive and as Sell if it is negative. The LSTM model will be trained by analyzing the activity of the last 24 hours since it is a sequential model, while the rest of the models will be trained with the observation of the previous hour. The classification labels will be with respect to the price movement of the following 24 hours, as described.

4.4. Feature Selection

A total of 87 features were gathered as inputs for the machine learning models. However, not all of them are considered to be essential in the prediction task. Adding too many less significant or irrelevant features can lead to noise being created, which could lead to a hard to interpret model, increase in time complexity, and inaccurate or less reliable predictions. This problem is known as the curse of dimensionality. Hence, feature selection is a must component in machine learning to deal with this high-dimensional problem of datasets. For this research, the wrapper method of feature selection was used. It consists of evaluating all possible combinations of features against a performance measure. Since this process is based on a specific machine learning algorithm, SVM was chosen. The forward selection technique, one of the wrapper methods, was performed with an algorithm from the mlxtend package. The idea behind the forward selection is to start with a null model and then start fitting it with each feature, one at a time. The performance metric used was accuracy, and it was evaluated on expanding window cross-validation. For the sake of consistency, the features selected were utilized in all models.

4.5. Benchmark

To provide a benchmark for the performance of the models, the Logistic Regression classification model will be used. The selection of this method as a benchmark arises from the fact that it is relatively simple and easy to implement, does not need many hyperparameters tunning and is known to achieve good results in classification tasks.

4.5.1. Logistic Regression (LR)

Despite its name, Logistic Regression is used in classification problems. The name comes from the function in equation 3 used at the core of the method, the logistic function, also called the sigmoid function. The letter z represents the multiplication of the features with the respective weights that will be tuned in the learning process.

Through the hyperparameter tuning process explained in Section 4.2., the following parameters were chosen: The regularization strength C of the L2 regularization was set at 0.001; The Library for Large Linear Classification (liblinear) was used as the solver to minimize the cost function among the ones provided by Sklearn library; The maximum number of iterations for the solver to converge was limited to 150.

Equation 2 | Sigmoid Function.

$$g(z) = \frac{1}{1 + e^{-z}}$$

Logistic regression is a useful benchmark to the LSTM due to its simplicity and thus, it enables us to observe the value-added of a much more complex and computationally intensive network.

4.6. Decision Trees (DT)

Decision Trees are a non-parametric supervised learning method used in classification and regression tasks. Their goal is to create a model that predicts the value of a target variable by learning simple decision rules deduced from the input features. Figure 6 illustrates the decision tree architecture composed of nodes and branches.

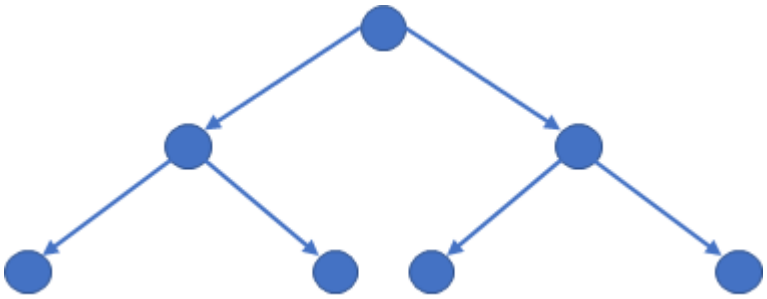


Figure 6 | Simple Decision Tree,

The function to measure the quality of the split in each node and the maximum depth of the tree was decided through cross-validation. The optimal choice for the function that delivers the best quality split was the Gini impurity and the maximum depth of the trees was set at 10.

4.7. Support vector machine (SVM)

A support vector machine (SVM) is a machine learning algorithm that analyses data for classification and regression analysis, but its predominant use lies in classification tasks. SVM is a supervised learning model that uses a linear model to implement non-linear class boundaries through some non-linear mapping of the input vectors x into the high-dimensional feature space. A linear model constructed in the new space can represent a non-linear decision boundary in the original space. The objective of SVM is to find a hyperplane (decision boundary) in N-dimensional space (N – number of features) and thus attempting to classify the observations. It achieves this by finding the hyperplane that has the maximum margin, the maximum distance between data points of both classes. For a linear separable case, a hyperplane separating the binary decision classes in a three-feature model can be represented as the following equation:

Equation 3: Hyperplane of 3-dimensional feature space.

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

Where y is the outcome, x_i is the feature values, and there are four weights w_i to be learned by the learning algorithm. In the equation, the weights w_i are parameters that determined the hyperplane. Figure 7 describes an example of a linear separable case.

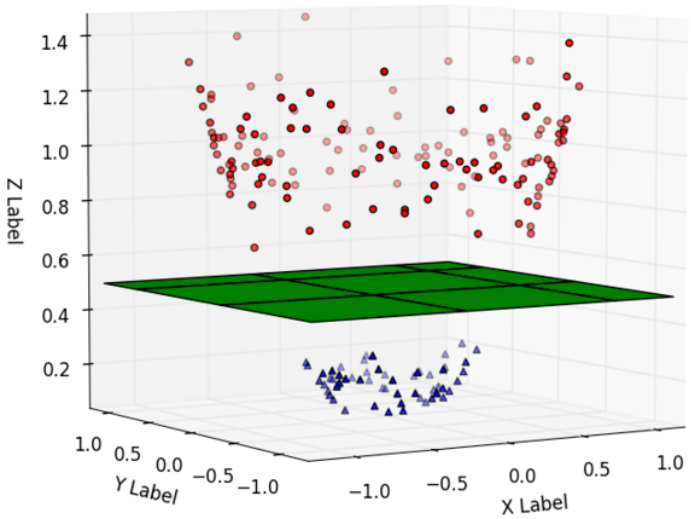


Figure 7 | Hyperplane in a 3-dimensional space | From commonlounge.com

However, in reality, datasets are almost never linearly separable in which we would get a 100% accuracy. The SVM addresses non-linearly separable cases by introducing soft margin and kernel tricks. Soft margin is a characteristic that tolerates misclassified vectors while balancing the trade-off between finding a line that maximizes the margin and minimizes the

misclassification. Soft margin can be tuned through the penalty term – "C" in Sklearn, where the bigger the C, the less penalty SVM gets when it makes a misclassification. Kernels are functions that generate inner products to construct machines with different types of non-linear decision surfaces in the input space. In Sklearn some already constructed kernels are available, such as polynomial, linear, sigmoid, radial basis function (RBF), and precomputed.

The loss function used to maximize the margin is the hinge loss function with a regularization parameter (λ) described in equation 5. The value of λ is inversely proportional to C. Figure 8 illustrates the loss function graphically.

Equation 4 | Hinge Loss Function.

$$L(w) = \sum_{i=1} \underbrace{max(0, 1 - y_i[w^T x_i + b])}_{\text{Loss function}} + \underbrace{\lambda ||w||_2^2}_{\text{regularization}}$$

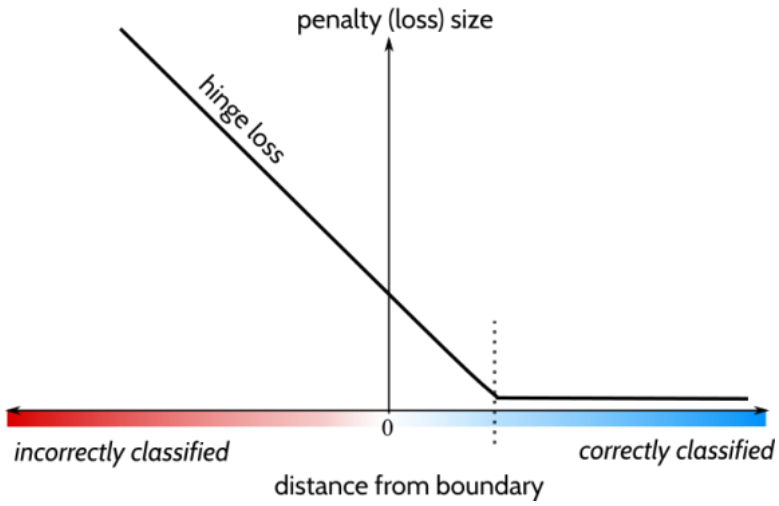


Figure 8 | Representation of the Hinge Loss Function |From stackexchange.com

Through hyperparameter tuning, performed with the cross-validation method the parameters C performs a regularization technique called L2 or Ridge, which penalizes the weights of the features according to the sum of the square of all feature weights, to prevent overfitting, was set to 10. The Ridge regularization has the advantage of not forcing any of the weights to be zero; however, it does not perform feature selection in contrast to the Lasso regularization. The kernel parameter is a crucial parameter since our dataset is not linearly separable and thus, it impacts the way the decision boundary is constructed. There is no specific rule to choose the kernel type and so considering the ones available from Sklearn, the RBF kernel was chosen as

it yielded better results. Lastly, the gamma parameter, which impacts the curvature of the kernel, was set to 0.001.

4.8. LSTM network (Long short-term memory)

Recurrent Neural Networks (RNN) is a type of feedforward neural network that has internal memory. The word recurrent describes how it performs the same function for every input of data while considering the output of the previous computation, exemplified in Figure 9. LSTM networks are a modified version of an RNN, designed to be better at storing and accessing information than standard RNNs and are capable of overcoming inherent problems of RNNs, i.e., vanishing and exploding gradients. LSTMs are Sequence Models, as they can be applied to any type of sequential data. They were first introduced by Hochreiter & Schmidhuber (1997) but later popularized in their use of solving tasks such as language modelling at the character and word level, image caption, speech (Graves et al., 2013), and handwriting recognition (Graves & Schmidhuber, 2009) and time series prediction.

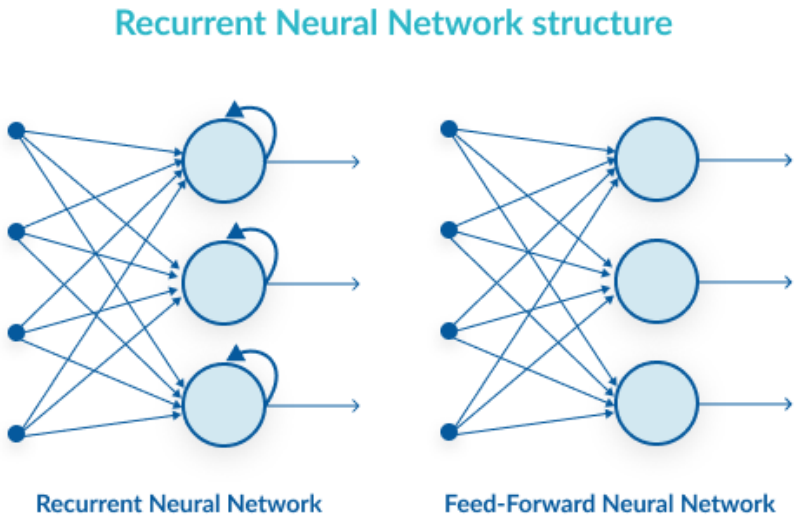


Figure 9 | Difference of network connections between RNN and Feed-Forward NN.

Figure 10 demonstrates the basic RNN prediction architecture. The input vector sequence $x = (x_1, \ldots, x_T)$ is passed through weighted connections to a stack of N recurrently connected hidden layers to compute initially the hidden vector sequences $h^n = (h^n_1, \ldots, h^n_T)$ and then the output vector sequence $y = (y_1, \ldots, y_T)$. The output vector of y_t is used to parameterise a predictive distribution $\Pr(x_{t+1}|y_t)$ over the possible next inputs x_{t+1}.

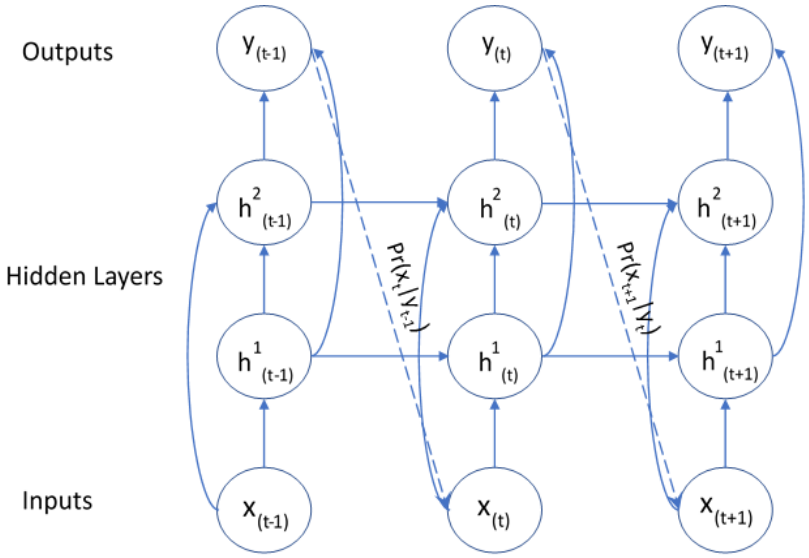


Figure 11 | Deep recurrent neural network prediction architecture. The circles represent network layers, the solid lines represent weighted connections and the dashed arrows represent predictions | From (Graves, 2014)

LSTM networks are also composed of an input layer, one or more hidden layers, and an output layer. The number of neurons in the input layer will be the same as the number of explanatory variables. The number of neurons in the output layer depends on the problem the network is trying to solve. The difference between the RNN and the LSTM lies in the hidden layer(s) consisting of so-called memory cells. Each memory cell has three "gates" that adjust the cell state s_t, the network's "short-term memory" which takes as its input both the state of the previous step and the output of the current step. They are called the forget gate (f_t), the input gate (i_t), and the output gate (o_t). The structure of the LSTM memory cell is shown in figure 11.

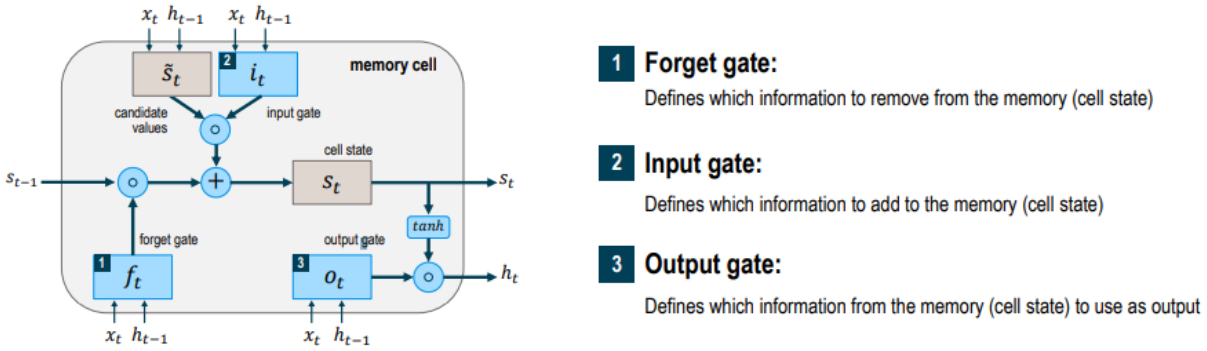


Figure 10 | LSTM cell with corresponding gate's explanation.

At every timestep t, each of the gates receives as input x_t plus the output h_{t-1} of the cells at the precious timestep t-1. The gates are essentially sigmoid activation functions that determine whether information should enter the cell – 1, or not – 0. In the training process, the network learns the optimal weights for each of the gates and consequently, how much of the information should be retained.

Due to the high number of possible parameters and specifications, some of the parameters were chosen based on research and what is traditionally used while others were chosen with a Randomized Search, a cross-validation method employed with an expanding window method.

The pre-selected parameters were the binary cross-entropy loss function, a frequently used loss function for binary classifications, the number of LSTM layers and dense layers, the rectified linear activation (ReLU) function for the LSTM layers, instead of the usual Tanh, and the sigmoid activation function for the final dense layer. Additionally, a recurrent drop rate was introduced in the first layer (a regularization parameter which drops a fraction of the units for the linear transformation of the recurrent state preventing overfitting and better generalization) Since the classification task is a binary one, the output of the final dense layer will be of one neuron, with the usual 0.5 threshold for deciding which class it belongs. Finally, the Adam optimization algorithm, an extension of the stochastic gradient descent, was used to optimize the weights.

The parameters chosen to be selected through cross-validation were the number of neurons on each layer, the drop rate of the first LSTM layer and the first dense layer, and lastly the learning rate.

The trained LSTM network is hence as follows:

- Input layer with 35 features and 24 timesteps.
- LSTM layer with 20 neurons, a recurrent drop rate of 0.1, ReLU as activation function, and a drop out layer of 0.2.
- LSTM layer with 50 neurons and ReLU as the activation function.
- Dense layer with 10 neurons and ReLU activation.
- Dense layer with one output with sigmoid activation.
- Adam optimizer with a learning rate of 0.0001 and decay of 1e-6.

For training the network, the number of epochs was set to 100 with the addition of the early stopping method to prevent overfitting. Once the validation loss does not decrease for five patience periods/epochs, the training is stopped, and the weights of the network with the lowest validation loss are stored with the help of Model Checkpoint.

4.9. Evaluating Machine Learning Models

The evaluation metrics used to evaluate the machine learning models depend on the type of problem at hand and the data available. In this case, the models will be accessed according to classification metrics. As observed in section 3.1., the weights of the classes are similar in the training data, and therefore the standard metrics for classification can be used to have a full understanding of the predictions. Table 3 represents the structure of a confusion matrix. This data visualisation method allows us to intuitively understand the performance of the models in terms of accurate predictions and the type of misclassification error the models might face.

Table 3 | Confusion Matrix – example

		Prediction Values	
		Sell (0)	Buy (1)
Actual Values	Sell (0)	True Negative (TN)	False Positive (FN)
	Buy (1)	False Negative (FN)	True Positive (TP)

4.9.1. Accuracy

Accuracy is the standard metric for evaluating classification problems. It accounts for the fraction of correct predictions over the total number of predictions. Accuracy can be misleading in imbalanced datasets. However, that is not the case in this research. Therefore, accuracy will be the main metric used.

Equation 5 | Accuracy formula.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

4.9.2. Precision and Recall

Precision measures the proportion of positive classes that were correctly classified. On the other hand, recall is the percentage of correctly identified positives in the total number of actual members of the positive class. These metrics are essential in evaluating the model's performance and informs us of the type of misclassification error the model is performing.

Equation 6 | Recall formula | TP is the number of true positives and FN is the number of false negatives.

$$Recall = \frac{TP}{TP + FN}$$

Equation 7 | Precision formula | TP is the number of true positives and FP is the number of false positives.

$$Precision = \frac{TP}{TP + FP}$$

4.9.3. F1-Score

Also known as balanced F-score, the F1 score is another performance measure in the statistical analysis of binary classification. It is the harmonic mean of precision and recall. The benefit of using this metric is that it gives a more concise understanding of the model's performance concerning the precision and recall metrics.

Equation 8 | F1-Score formula.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

4.10. Performance of Trading Strategy

While classification metrics are important for validation purposes and for deciding which models best predict bitcoin's price movement on the test data, they lack the impact returns have on the profitability of the model. For example, a model can have greater accuracy than another but have the worst profitability because it failed to predict days with high volatility while the other correctly predicted them. In order to understand how the models might fare in the real world, the results of the models will be backtested on unseen data with profit and loss metrics, risk metrics and combined risk and reward metrics, and descriptive analysis on the distribution of returns. The returns will be subjected to transaction and slippage costs. These costs should always be included because they affect the profitability of trading strategies. The average transaction fees paid to miners amounted to an average of 2 basis points in the period analyzed. Including slippage costs, it was decided to set these costs at 5 basis points.

5. Results & Discussion

5.1. Feature selection

The outcome of the feature selection process was a reduced number of variables that best described the problem at hand. From the OHLCV metrics, we were left with Open, Close and Volume, which are useful metrics that can establish patterns with some of the technical indicators.

Out of the blockchain characteristics, the most explanatory variables according to the selection process were mining difficulty, mean transaction fees, total transaction fees, number of active addresses, number of new addresses, number of transactions per second, mean transfer volume, the market capitalization of Tether, mean value of created UTXO. Against expectations, the mean hash rate was not included as a feature, and instead, mining difficulty was considered to describe the network from a supply perspective better. The number of new and active addresses as well as the supply of the stable coin Tether, continue to have an impact on bitcoin's price, which supports the current findings of existing literature. However, the majority of blockchain metrics were related to transaction activity. This points to the importance of transaction metrics, such as fees, volume, and UTXO, in influencing bitcoin price. Due to the blockchain's inability to manage large transaction activity, which causes fees to increase, these types of metrics may be more useful than expected in predicting bitcoin's behaviour and deserve to be further studied by future research.

In terms of the technical features, only the momentum indicators were excluded altogether from the final selection. Of the 35 final features, 23 of them were technical indicators. The volume indicators chosen were the Accumulation/Distribution index, On-balance Volume, Chaikin Money Flow, Force Index Money Flow Index, and Ease of Movement. Out of the volatility indicators, only the Bollinger High Band Indicator, the Keltner Channel High Band, and Donchian Channel Low Band were selected. The other 14 indicators belong to the trend category. Most of them constitute different variants of moving averages such as MACD (includes the signal line and the difference between the MACD line and signal line), SMA (includes two types of simple moving averages), EMA (exponential moving average), and Trix, which shows the percentage rate of change of a triple exponentially smoothed moving average. With respect to the others, two are derived from the KST Oscillator (Signal line and the difference from the KST line and signal line) and the rest from the Ichimoku Indicator, which

included the Conversion line, Base line, Leading Span A and B, and the squared values of Leading Span B.

5.2. Classification Metrics

As mentioned before, accuracy will be the main metric to evaluate the models as it influences the most the profitability of the buy/sell strategy. In this metric, the models were able to reach similar values found in the literature. The classification metrics are shown in table 4.

Table 4 | Test metrics of the models and benchmarks.

	Test set metrics			
	DT	SVM	LSTM	LR
Accuracy (%)	48,91	59,43	57,09	58,02
Precision (%)	61,32	59,46	63,71	61,08
Recall (%)	37,68	99,42	65,16	80,65
F1-Score (%)	46,68	74,42	64,43	69,52

Both the LSTM and SVM were able to outperform the DT by a considerable margin, 818 bp and 1052 bp respectively. However, these values reflect more the inability of a simple DT to predict the price movement of Bitcoin than that of the predictive superiority of LSTM and SVM models. With its 48.91% accuracy, the DT was no better than a random choice. The reason behind this can be traced back to the type of data fed into the model and the algorithm behind decision trees. Decision trees tend to perform better when they are faced with repetitive scenarios, evidenced in categorical data. Historical data, mostly in a continuous form, makes it hard for decision trees to generalize well to future observed data because a lot of the information from the variables is lost in each node division. This led to the model overfitting in the training data and achieving bad results in the test data. One way to improve DT, so that it is able to arrive at a higher than 50% accuracy and reduce overfitting, would be to transform the features into categorical data and switch to a tree-ensemble method, called Random Forest, that combines many decision trees trained on different versions of the dataset.

In relation to the LR in terms of accuracy, the SVM achieved the highest value while the LSTM underperformed. However, looking deeper into the recall and precision metrics, as well as the confusion matrices, help us understand why. In figure 12, we can interpret that recall had such high values for the SVM and LR models because both models heavily predicted buy signals. The SVM predicted almost entirely buy signals while the LR was able to have a better

performance in predicting sell signals, having correctly predicted more sell signals than SVM, by a margin of 973 bp, but only correctly forecasting 25% of the sell signals. The fact that the test period was characterized by an impressive bullish period, with 59.35% of the test data achieving positive returns, in conjunction with the fact that the LR and SVM mostly predicted buy signals, leads us to conclude that these two models were better at identifying the price trend than the shifts in price movements and therefore experienced a better accuracy than the LSTM model.

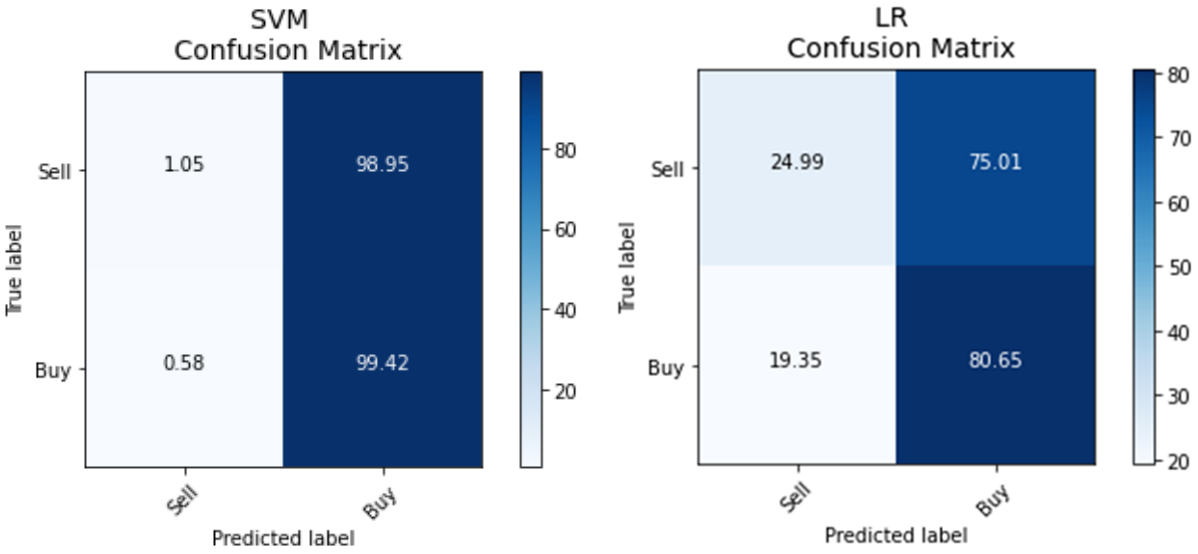


Figure 12 | Confusion Matrices of SVM and LR. The values represent the percentage of predictions in relation to each True label.

On the other hand, figure 13 shows us that, while the LSTM missed more buy signals, it was able to predict more sell signals correctly. In contrast with LR, which only correctly forecasted 25% of the sell observations, the LSTM was able to foresee 45.16% of the sell signals with a decrease in accuracy of only 93 bps. Furthermore, it was able to have a more balance prediction having a similar percentage of Buy signals as the test data (60.89%). Therefore, although it had a smaller accuracy than the other two models, we can conclude that the LSTM was better at understanding the problem and spotting the periods which had a decrease in price. The reason behind this may be due to the differences in the complexity of the models or insufficient parameter tuning. In the case of the SVM, the model was tuned for the more important parameters (C, γ, and kernel) and found middle values for C and γ. However, the type of kernel (RBF) was chosen from those available in the package Sklearn, which may lack compared to a more customized kernel. The same can be said for the LR model, which for the sake of simplicity was only tuned for the regularization strength (C) and the maximum number of iterations taken for the solver to converge. Another possible explanation is that for the sake of maximizing accuracy, the SVM and LR created more generalized decision rules in the training

period by penalizing variables that pointed to more subtle changes in price and not penalizing variables that tracked the overall trend. That decision might have been impacted by the

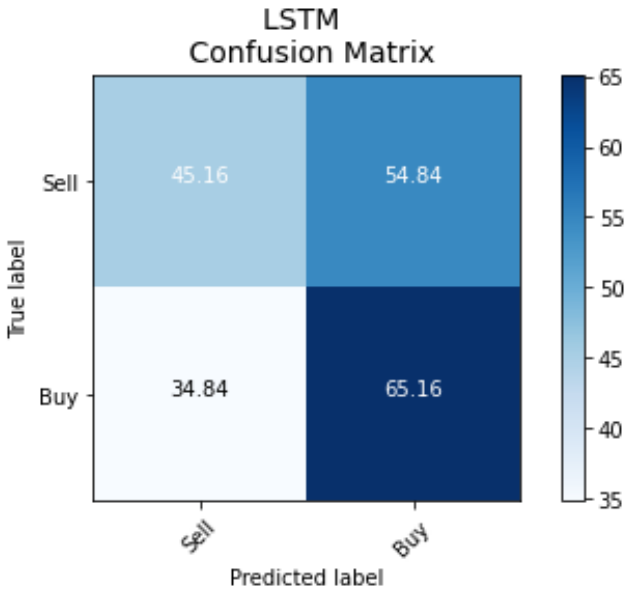


Figure 13 | Confusion matrix of test results from LSTM.. The values represent the percentage of predictions in relation to each True label.

predominance of technical indicators, especially due to the high number of trend indicators.

5.3. Profitability Metrics

5.3.1. Models vs Benchmarks in a Buy & Sell strategy

The difference in the profitability of the LSTM and the SVM shows us the importance of backtesting the results from trading algorithms, illustrated in table 5. Despite having a difference in the accuracy of 237 bps, the LSTM was able to achieve very similar returns and volatility compared to the SVM model, as well as a slightly lower downside deviation which resulted in both achieving a Sharpe ratio of 2,60 and a somewhat higher Sortino Ratio for the LSTM.

Table 5 | Profitability metrics of the trading signals forecasted by the machine learning models. The P-value relates to the significance of the average daily returns being different from zero. All models, except for the DT, produce mean daily returns significant at a 10% confidence level and the mean daily returns from LR were significant at a 5% confidence level.

Trading performance of the models and benchmarks

	LSTM	SVM	DT	LR	B&H
Mean Daily Return (%)	0,39	0,39	0,02	0,43	0,40
P-value	*0,059*	*0,058*	*0,917*	*0,039*	*0,058*
Annualized Return (%)	143,48	143,91	7,95	155,99	144,21
Standard Error	0,002	0,002	0,002	0,002	0,002
Standard Deviation (%)	2,89	2,89	2,92	2,88	2,89
Annualized Volatility (%)	55,26	55,27	55,74	55,08	55,24
Kurtosis	5,46	5,38	5,35	5,64	5,35
Skewness	0,17	0,28	0,09	-0,09	0,31
Minimum (%)	-12,00	-11,79	-12,00	-13,86	-11,74
Quartile 1 (%)	-0,93	-0,75	-1,14	-0,74	-0,77
Median (%)	0,40	0,47	0,04	0,47	0,46
Quartile 3 (%)	1,59	1,59	1,48	1,59	1,59
Maximum (%)	13,86	13,86	13,86	11,95	13,86
5 % VaR (%)	-4,27	-4,27	-4,35	-4,27	-4,03
5% CVaR (%)	-6,48	-6,39	-7,14	-6,34	-6,29
Profit orders (%)	57,51	60,62	50,26	62,18	60,10
Loss orders (%)	42,49	39,38	49,74	37,82	39,90
Mean Return of Profit Orders (%)	2,04	1,94	1,96	1,91	1,95
Mean Return of Loss Orders (%)	-1,83	-1,98	-1,94	-2,01	-1,95
Max Drawdown (%)	-25,53	-19,35	-0,44	-0,15	-18,07
Sharpe Ratio	2,60	2,60	0,14	2,83	2,61
Downside Deviation (%)	39,88	40,60	41,88	41,81	39,94
Sortino Ratio	3,60	3,54	0,19	3,73	3,61

As expected from the classification results, we can observe in figure 14 that the returns of SVM followed a similar path as Bitcoin, except for the period between 03/05 – 14/05 that saw two profitable sell transactions and one unprofitable, with the latter bringing the cumulative returns back to being smaller than a Buy and Hold strategy.

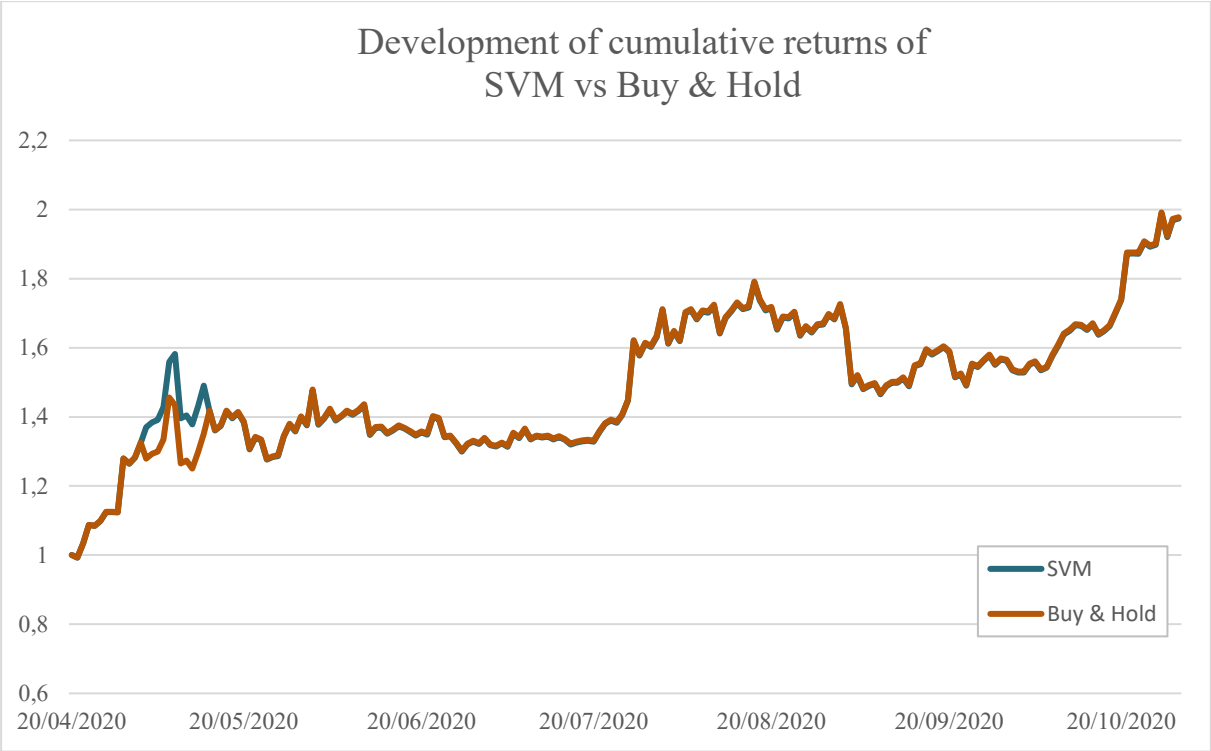


Figure 14 | Cumulative returns of SVM Buy & Sell in comparison to a Buy & Hold strategy.

Although all models had promising Sharpe and Sortino ratios, except for the DT, the only model which was able to outperform a Buy & Hold strategy was the benchmark model LR. The LR had the lowest volatility and highest returns of all strategies that can be explained by looking at figure 15 and the previous classification results. The higher predominance of buy signals, similar to the SVM, led it to correctly predict the majority of the days with the highest positive returns which influenced the overall appreciation of Bitcoin. On the other hand, opposite to the SVM which performed poorly in predicting sell signals, the LR was able to profit in a Buy & Sell strategy from higher predictability of sell signals.

The reason LR was able to achieve a higher number of profit orders in comparison to the accuracy results can be explained by the timing of the transactions. While the accuracy metric was calculated with predictions in different hours of the day, the predictions acted upon were from the hour before the start of the day.

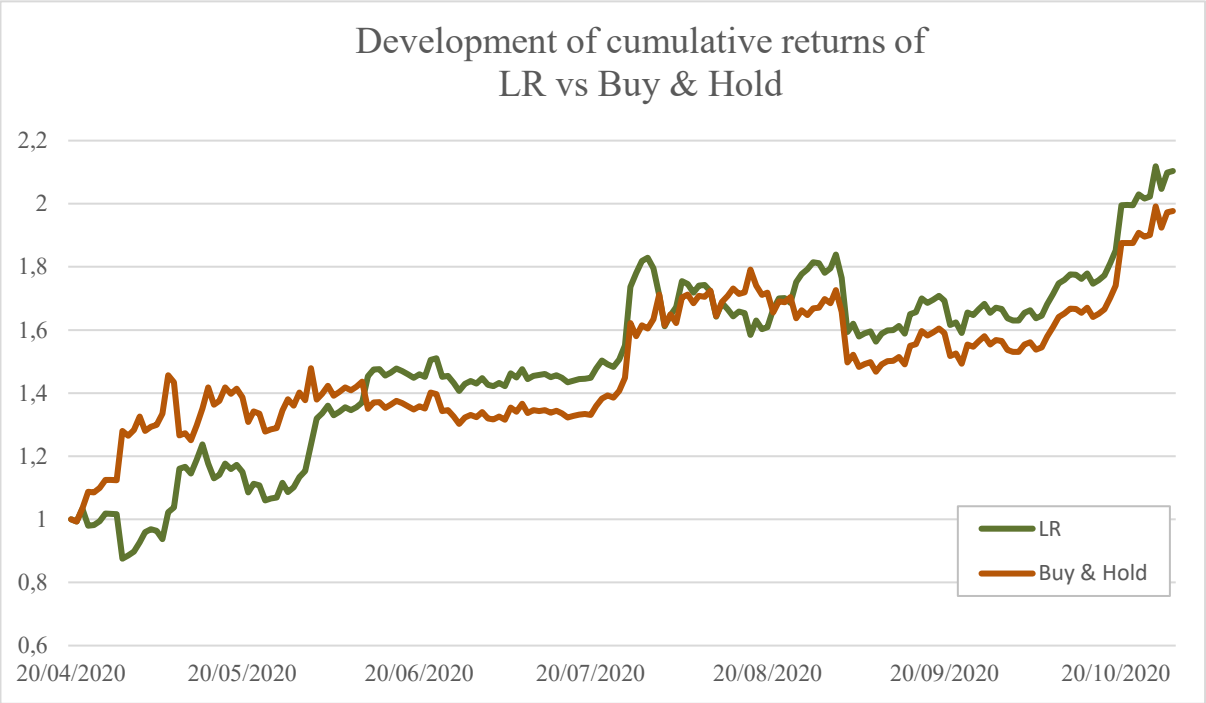


Figure 15 | Cumulative returns of LR Buy & Sell in comparison to a Buy & Hold strategy.

However, not all LR profitability metrics are superior, especially when comparing to the LSTM. Despite having lower CVaR and VaR compared to LSTM, the presence of negative skewness means that LR has a higher propensity to experience extreme negative events. Furthermore, there is a notable difference in terms of the mean returns of profit and loss orders and the cumulative returns of LSTM. Figure 16 allows us to observe the contrasting performance of both strategies further. The LSTM outperformed all models in the first half of the test period and was able to reach a maximum cumulative return of 242%. While the second half of the test period was not as profitable as the first, it still managed to have higher cumulative returns, except for the last few days.

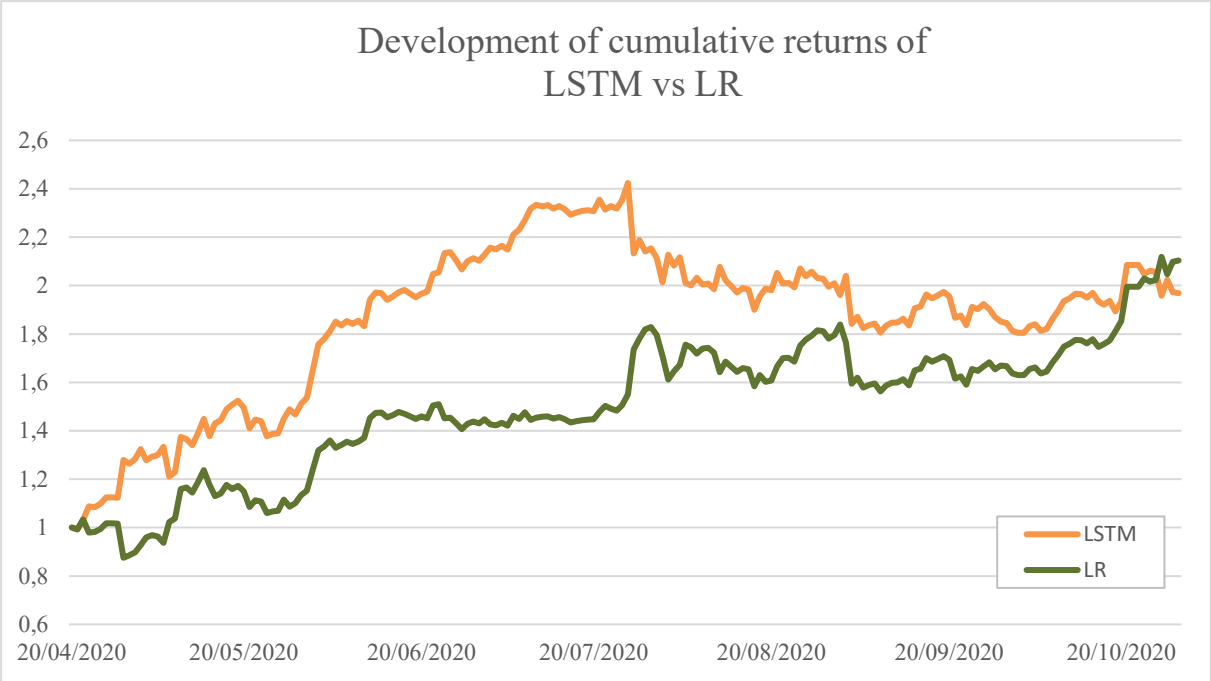


Figure 16 | Comparison of cumulative returns of LSTM and LR models in a Buy & Sell strategy.

In order to understand what caused the decline in the performance of the LSTM, it might be helpful to look at the signals and performance of a buy only and sell only strategy. While this may seem a biased approach, given the high increase in the price of bitcoin in the test period, it is reasonable to think that an investor might consider a buy only strategy in this period especially due to the circumstances that affected the markets prior to the test period, the Covid-19 related crash in March 2020. The SVM and DT models were not included in this analysis because the former lacked a balance prediction of Buy and Sell signals, and the latter presented predictions that were not reliable due to overfitting. Moreover, the sell only strategy was not performed for the LR because of insufficient predictions of sell signals.

5.3.2. LSTM and LR in a Buy Only strategy

The fact that the LSTM had a more balanced prediction of buy and sell signals raises the possibility that the model was better at identifying days that had a stronger impact on returns.

Looking at the results from table 6, we can see an improvement in the risk-adjusted return metrics of both models when using a buy only strategy, surpassing even the buy and hold values. Although the LSTM had smaller annualized returns than the benchmark, it had far better results in terms of risk metrics. It achieved lower volatility and downside deviation and the 5% VaR and the 5% CVaR improved substantially from a Buy & Sell strategy which points to a lower risk strategy and that the worst trades were mainly from sell signals.

Table 6 | Profitability metrics of the trading signals forecasted by the machine learning models in a Buy Only strategy. The P-value relates to the significance of the average daily returns being different from zero. Both models arrived at significant mean daily returns at a 1% confidence level.

Trading performance of buy only strategy			
	LSTM	LR	B&H
Mean Daily Return (%)	0,39	0,42	0,40
P-value	*0,007*	*0,010*	*0,058*
Annualized Return (%)	143,85	152,59	144,21
Standard Error	0,001	0,002	0,002
Standard Deviation (%)	2,03	2,27	2,89
Annualized Volatility (%)	38,87	43,35	55,24
Kurtosis	13,80	6,59	5,35
Skewness	1,30	0,56	0,31
Minimum (%)	-9,72	-9,67	-11,74
Quartile 1 (%)	0,00	0,00	-0,77
Median (%)	0,00	0,00	0,46
Quartile 3 (%)	0,78	1,20	1,59
Maximum (%)	13,86	11,95	13,86
5 % VaR (%)	-1,85	-3,40	-4,03
5% CVaR (%)	-3,81	-4,98	-6,29
Profit orders (%)	64,46	66,43	60,10
Loss orders (%)	35,54	33,57	39,90
Mean Return of Profit Orders (%)	1,82	1,83	1,95
Mean Return of Loss Orders (%)	-1,54	-1,94	-1,95
Max Drawdown (%)	-11,46	-14,98	-18,07
Sharpe Ratio	3,70	3,52	2,61
Downside Deviation (%)	24,19	29,81	39,94
Sortino Ratio	5,95	5,12	3,61

Overall, these results coupled with the fact the LSTM had fewer buy signals and profitable transactions, support the hypothesis that the LSTM was better at predicting days with high positive returns and might be a better model in periods where outliers are not as impactful as the one observed in the test period.

Figure 17 also gives us insight into the inner workings of the LSTM. In the periods after the occurrence of outliers of absolute returns, the LSTM network would predict an opposite movement for the future periods. This is why the period after 28th of June and 21st of October experienced no sell signals. The same happened after the sharp decrease in price on the 3rd of September, in which the strategy seems only to have buy signals. The reason behind these predictions might be due to the ability of the network to keep in memory past events. During

the training process, the network probably learned that a correction would take place after a sharp increase or decrease of the price. This characteristic may have proved to be extremely profitable in a setting like the one experienced in the training period, however, given the overall appreciation of Bitcoin in the test period, those corrections either did not happen or did not have the expected outcome.

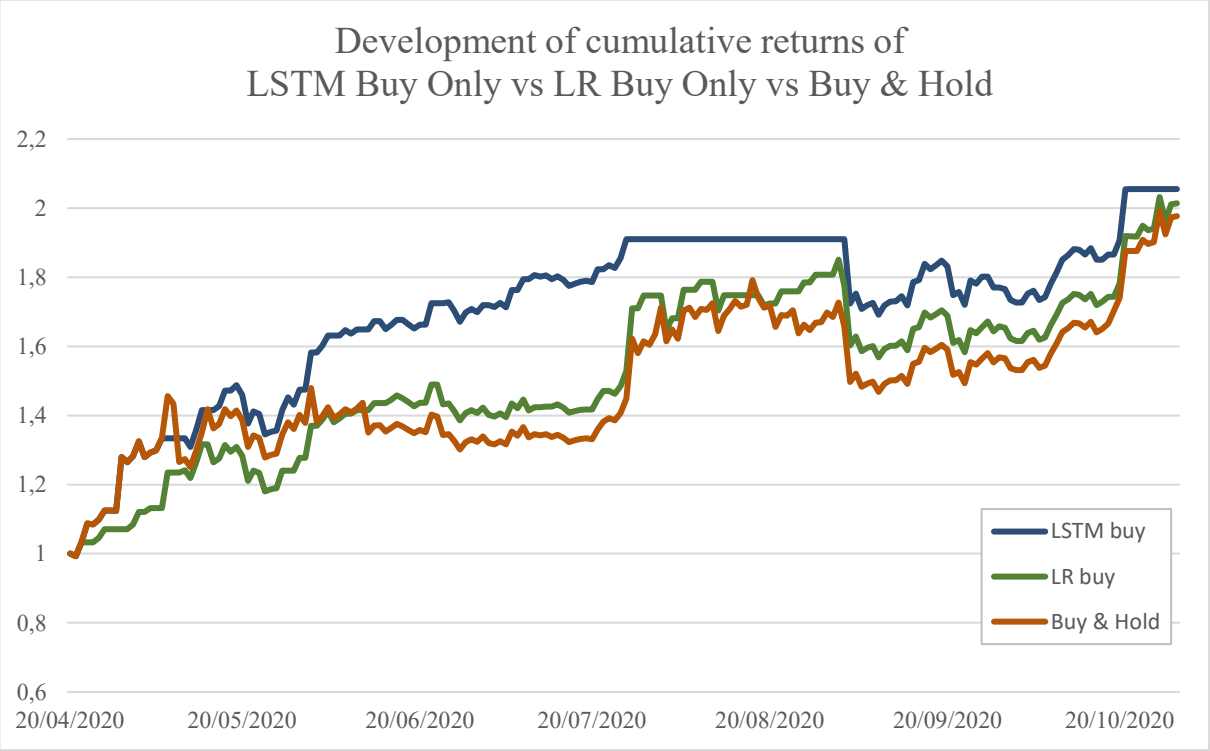


Figure 17 | Comparison of cumulative returns between LSTM and LR in a Buy Only strategy and a Buy & Hold strategy.

This limitation of the network could be improved by further hyperparameter tuning, especially concerning the recurrent drop out of the layers (a regularization parameter that controls the amount of information taken into account from past outputs) and the number of layers, allowing for a higher network complexity. Although, the tunning of the recommend hyperparameters might not guarantee an improvement of accuracy due to the danger of the network to overfit to the training data, a collateral effect of a more complex model, the low level of complexity and number of hyperparameters tunned in the current model raises the possibility that future development of the LSTM could achieve better results.

5.3.3. LSTM in a Sell Only strategy

In terms of the sell only strategy, the results were not as conclusive as those from the buy only strategy due to a lower number of sell predictions, evidence by the first and third quartile figures in appendix 14. Although the annualized returns were negative and the strategy had more loss

than profit orders, looking at figure 18, we can see that the performance was heavily affected by the worst trades, for example on 27/07. Furthermore, since it had fewer trading days (72 days), the 5% CVaR, which accounted for the ten worst trades, had a big impact on performance.

As evidenced in the previous results, periods of predominant buy signals came after a sharp decrease in price, making these outliers having a clear impact on future predictions.

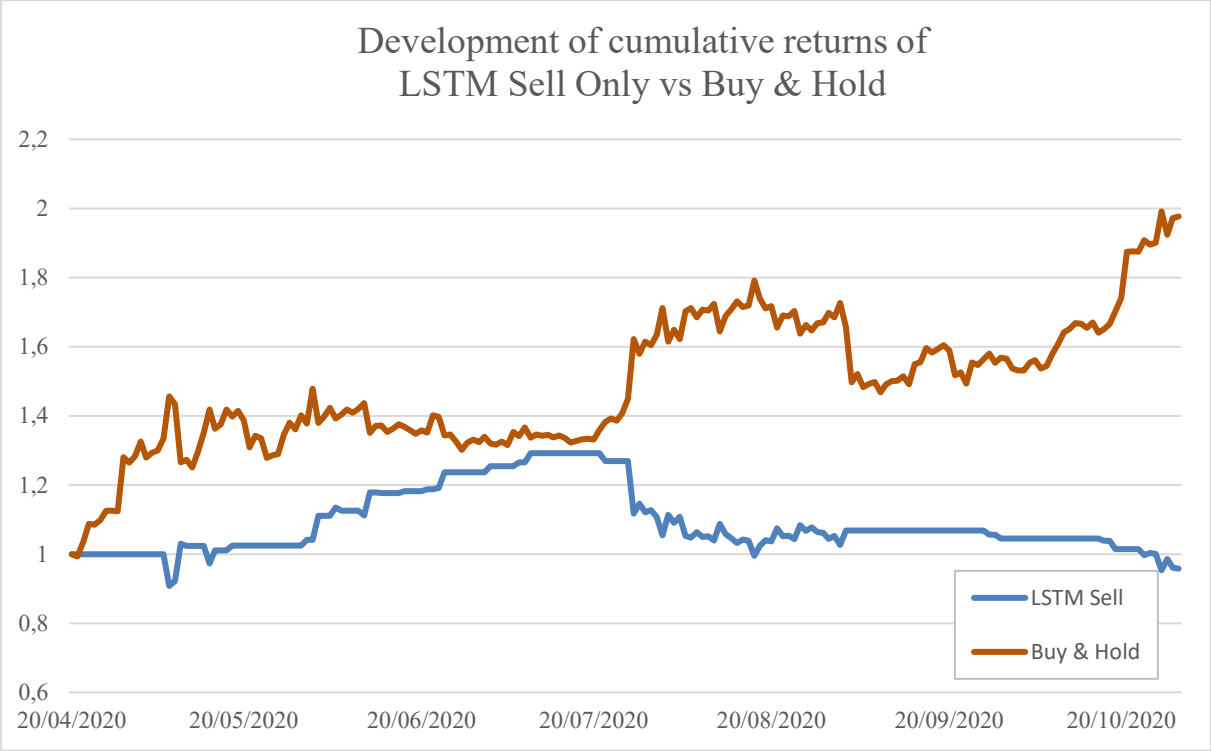


Figure 18 | Comparison of cumulative returns between an LSTM Sell Only strategy and a Buy & Hold strategy.

5.4. Outliers

The impact of outliers and its inconsistent prediction were not exclusive to the LSTM and was present in all strategies. It is important to note that most of the big swings in price were not something that the models were built to predict since it was a result of news or related bitcoin procedures. The 11.74% decrease in price on 9th of May was due to a schedule halving process which is a part of the blockchain program that continues to decrease the reward miners receive for mining blocks. The sharp increase in price on 28th of June and 21st of October was due to news of the integration of transactions with bitcoin by Paypal, which would mean an increase in cryptocurrency adoption. The raiding of the biggest exchange in South Korea by the police and the offloading of bitcoins by miners were the reasons behind the decrease of 9.67% on 3rd of September.

Implementation of stop-loss limits and other hedging tools like options could further improve the performance of all models by limiting the exposure of these extreme cases. On the other hand, adding features of sentiment analysis, that would track news and overall sentiment on popular online forums that follow cryptocurrencies might allow the models to incorporate better the magnitude and relevance of the outliers by detecting if the move was a shift in trend or not. This would be particularly beneficial for the LSTM since the current model appears always to expect a correction after the occurrence of outliers.

6. Conclusion

This dissertation has proposed a novel way to understand and predict the price movement of Bitcoin through the use of variables that measure the activity of its blockchain technology, the stable coin market, and the addition of technical indicators, as a replacement for the lack of traditional fundamental information from the inherent nature of this relatively new asset. Furthermore, this thesis uses machine learning models of different complexities to attempt to beat a Buy & Hold strategy.

In terms of the features' relevance, results of feature selection and profitability of the models show promising results in all types of variables considered. Developing on existing literature that points to the usefulness of blockchain metrics, such as the number of addresses and hash-rate, as proxies of supply and demand, this thesis found that features related to the number of addresses are still relevant but found that mining difficulty might provide more explanatory power than the current hash-rate of the blockchain. Furthermore, a predominance of features related to transaction activity suggests that these types of variables can provide additional information on blockchain activity, not only in terms of demand but also of supply since transaction fees are paid to the miners who process transactions. The use of Tether's market capitalization, as a proxy for stable coins, also proved to be relevant, in accordance with recent literature. Additional inclusion of features related to stable coins could increase the predictive power of the models. The use of technical features, mainly of the Trend category, allowed the models to identify better the overall trend experienced in the test period, especially for the SVM and LR models.

Regarding the models' ability to understand the problem at hand, the results were somewhat on par with the different complexities of the models. The Decision Tree proved to be far too simple of a model to generalize well into unseen data. Decision trees tend to perform better when faced with repetitive scenarios and data in a categorical form, which was not the case. An ensemble method that groups several Decision Trees trained on random subsets of the data, called Random Forest, would probably improve the accuracy and prevent it from overfitting.

With respect to the SVM, its accuracy results pointed to the superiority of this model compared to the others. However, looking at the other classification metrics, we can see that the reason behind it was because it predicted a majority of buy signals. These results convey the model's inability to detect slight changes from the overall trend that would produce higher returns in a Buy & Sell strategy, compared to the Buy & Hold.

Contrary to Fischer & Krauss (2018) findings, the LSTM based trading strategy was not able to outperform the one from the Logistic Regression in terms of returns. While LSTM did manage to achieve similar returns to the SVM, despite lower accuracy, it was also not able to outperform a Buy & Hold Strategy in terms of returns and volatility. However, it did experience far better performance in the first half of the test period than all models and benchmarks. By examining the cumulative returns of a Buy Only and Sell Only strategy, we can see that the performance of the LSTM deteriorated after sharp movements in Bitcoin's price. Due to the LSTM neural network's characteristic to consider past events in future predictions, the model most likely learned that a correction in the price would take place in the periods following those events and predicted a stream of signals oppositive to the direction of the outlier. This trait and the overall profitability of the LSTM could be improved by further hyperparameter tunning. Therefore, this model's inability to outperform the Logistic Regression benchmark was not conclusive.

In conclusion, in a Buy & Sell strategy, the only model that was able to outperform the Buy & Hold strategy was the benchmark Logistic Regression. By correctly identifying the trend, similar to the SVM, but making critical profitable sell trades it managed to have better return and volatility metrics, proving that combining machine learning methods with the selected features can provide additional returns and lower volatility. Additionally, in a Buy Only strategy, the better performing model in terms of risk-adjusted returns was the LSTM. The results point to the fact that the LSTM was better at identifying days with significant returns.

Notwithstanding, this study faced a few limitations. First, due to reduce data availability, the number of observations could have been more significant, which would increase the ability of the models to learn from the data to an extent. Second, the expanding window method used in the cross-validation might not have been the most efficient and a rolling window method could have provided better hyperparameters. Finally, profitability could have been improved if a third labelling class were added in which no decision would be taken if absolute returns were smaller than transaction costs.

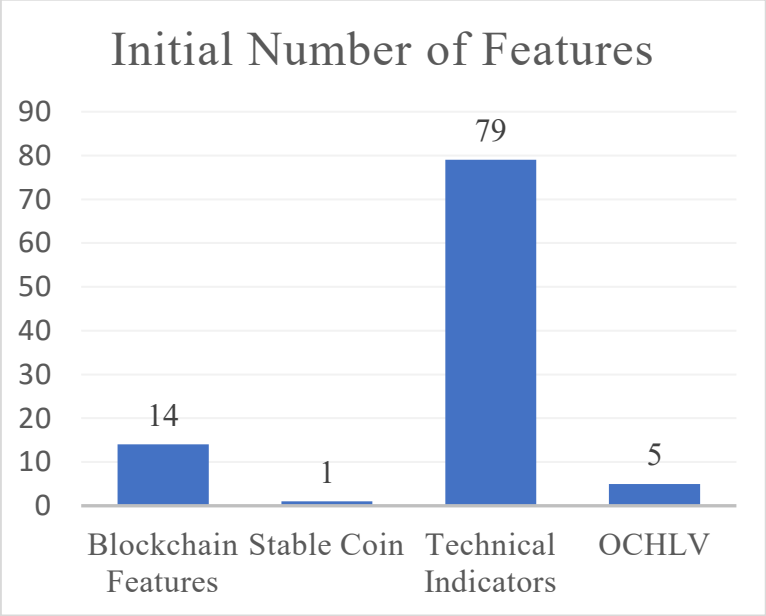
Lastly, in future research, despite further hyperparameter tuning, the addition of Sentiment Analysis could also improve the model's performance, especially for the LSTM, by better understanding the trend in the periods after significant price changes. Furthermore, given the promising results found, further blockchain characteristics, as well as variables related to the stable coin market, could improve the prediction of the price movement of Bitcoin.

7. Appendix

Appendix 1 | CAPM metrics from Bitcoin. The P-value for the alpha was not significant and the P-value for the beta was significant at a 1% confidence level. The market returns were retrieved from Crescent Crypto Market Index (CCMIX).

CAPM results of Bitcoin

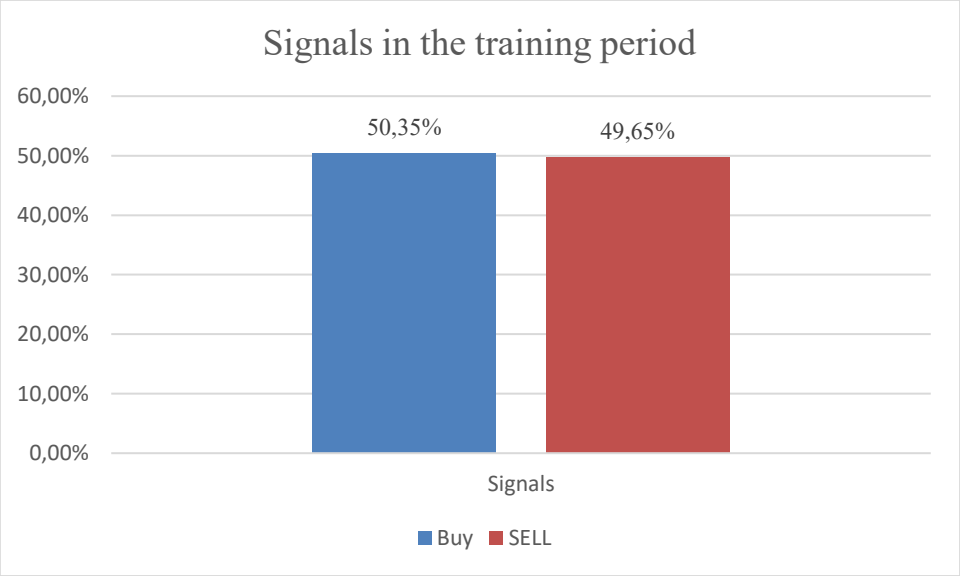
	BTC
Alpha (%)	0,029
P-value	0,419
Beta	0,957
P-value	0,000
R^2	0,963
Observations	193



Appendix 2 | Distribution of the number of features.

Descriptive Statistics	
Mean Daily Return (%)	0,03
Annualized Return (%)	11,48
Standard Error	0,00
Standard Deviation (%)	3,87
Annualized Volatility (%)	73,85
Kurtosis	13,10
Skewness	-0,78
Minimum (%)	-37,17
Quartile 1 (%)	-1,56
Median (%)	0,00
Quartile 3 (%)	1,51
Maximum (%)	18,19
Count	786

Appendix 3 | Descriptive Statistics of Bitcoin daily returns in the training period.



Appendix 4 | Distribution of signals in the training period.

Descriptive Statistics

	Open	High	Low	Close	Volume USD
Count	23590,0	23590,0	23590,0	23590,0	23590,0
Mean	7856,4	7890,3	7819,4	7856,6	4124493,0
Standard Deviation	2374,4	2385,4	2361,7	2374,7	5479482,3
Min	3139,8	3153,6	3130,0	3139,8	0
25%	6400,0	6410,0	6385,8	6400,0	1274587,6
50%	7982,2	8024,8	7940,0	7982,2	2443500,7
75%	9561,2	9599,7	9515,3	9561,2	4815678,9
Max	13948,4	14098,9	13861,5	13948,4	102621245,2

Appendix 5 | OHLCV Descriptive Statistics.

Blockchain characteristics descriptive statistics.

Descriptive Statistics

	Nº Active Addresses	Total Transaction Fees	Mean Transaction Fees	SOPR	Market Capitalization	Mining Difficulty	Mean Hash Rate
Count	23590	23590	23590	23548	23540	23540	23540
Mean	41168	2,0471	0,00018	1,000	1,407E+11	4,35E+22	7,54E+19
Standard Deviation	16622	2,0977	0,00019	0,034	4,414E+10	2,10E+22	4,91E+19
Min	0	0	0,00000	0,428	5,573E+10	1,29E+22	2,79E+18
25%	29544	0,7925	0,00008	0,994	1,118E+11	2,60E+22	3,84E+19
50%	38810	1,2708	0,00012	1,000	1,410E+11	3,41E+22	6,20E+19
75%	50834	2,4296	0,00021	1,005	1,736E+11	6,35E+22	1,02E+20
Max	134284	29,6000	0,00657	3,084	2,573E+11	8,59E+22	5,07E+20

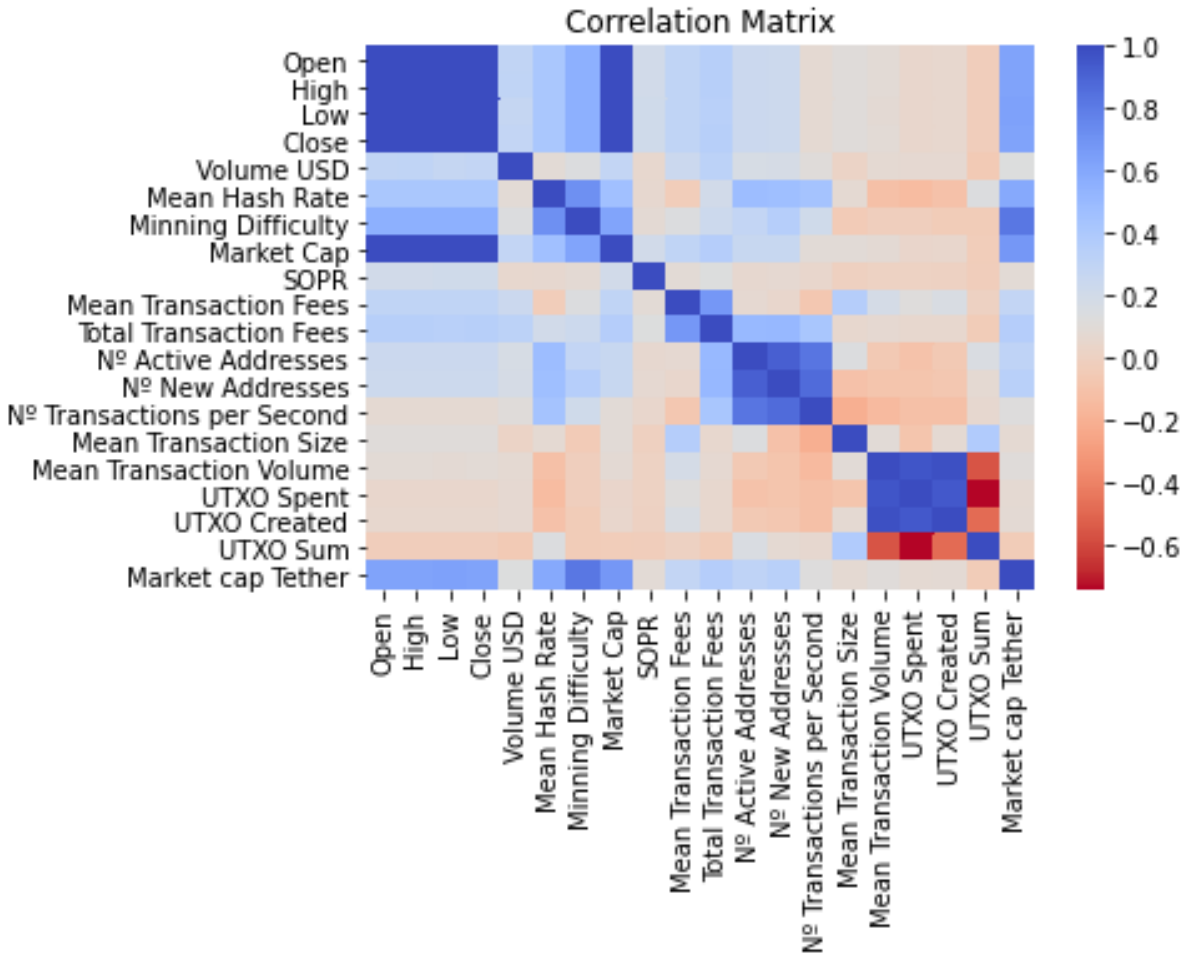
Blockchain characteristics descriptive statistics.

Descriptive Statistics

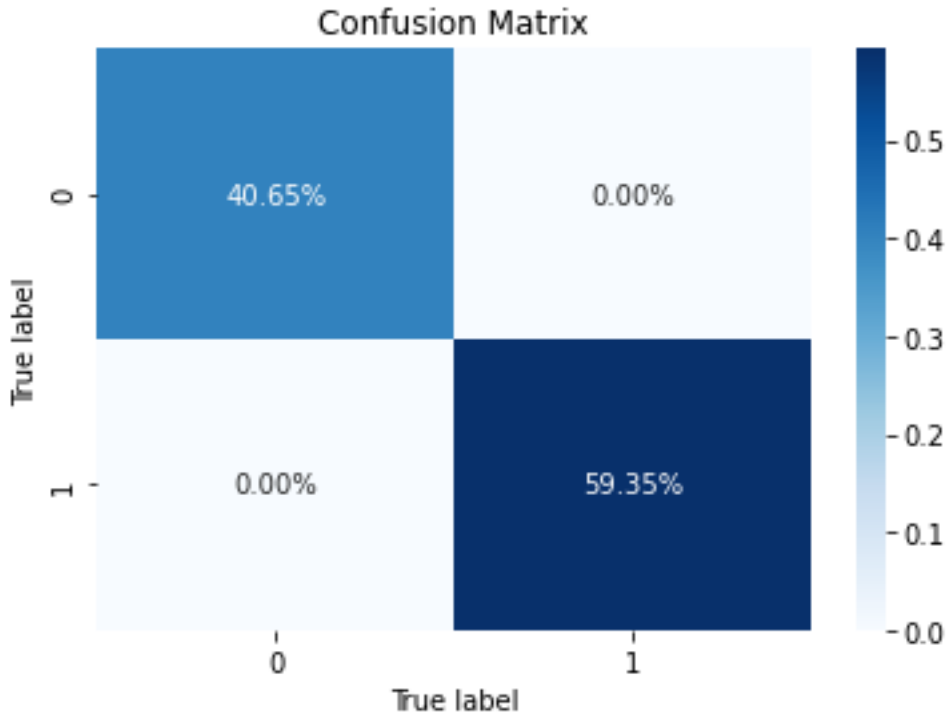
	Sum of UTXOs	Spent UTXOs	Created UTXOs	Mean Transaction Volume	Mean Transaction Size	Nº Transactions per Second	Nº New Addresses
Count	23590	23590	23590	23590	23540	23590	23590
Mean	-0,249	1,770	2,019	4,762	541,787	3,318	15045,012
Standard Deviation	0,831	1,749	2,270	5,150	163,604	1,390	6582,892
Min	-26,859	0,000	0,000	0,000	300,531	0,000	0,000
25%	-0,404	1,003	1,086	2,555	457,291	2,300	10413,000
50%	-0,162	1,393	1,548	3,658	502,818	3,177	13945,000
75%	0,026	2,015	2,286	5,421	571,322	4,177	18721,000
Max	13,025	61,096	84,757	182,342	5377,949	12,837	55576,000

Appendix 8 | Tether Market Capitalization Descriptive Statistics

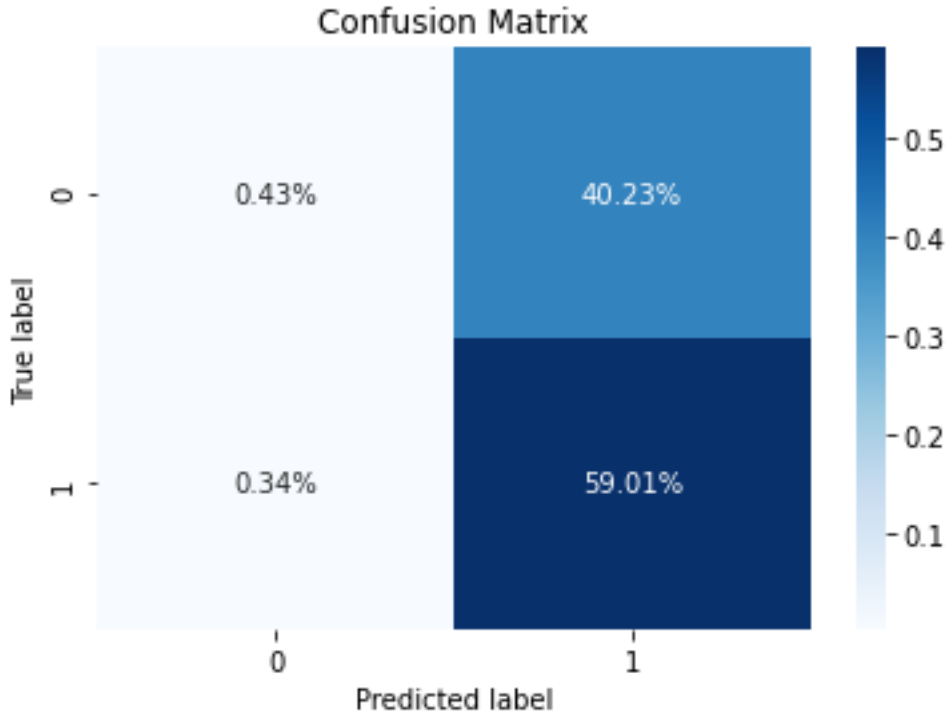
Descriptive Statistics	
	Tether Market Capitalization
Count	23590
Mean	4688924195
Standard Deviation	3567915419
Min	0
25%	2291049633
50%	3577922339
75%	4649151904
Max	16662662845



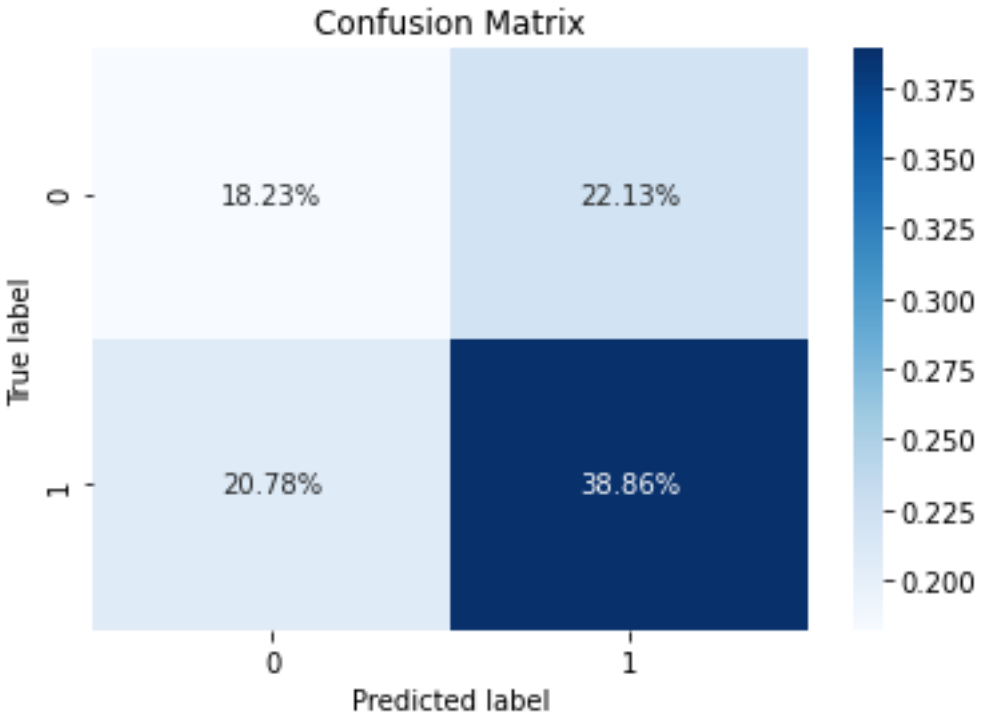
Appendix 9 | Correlation Matrix of all Features, except for Technical Indicators



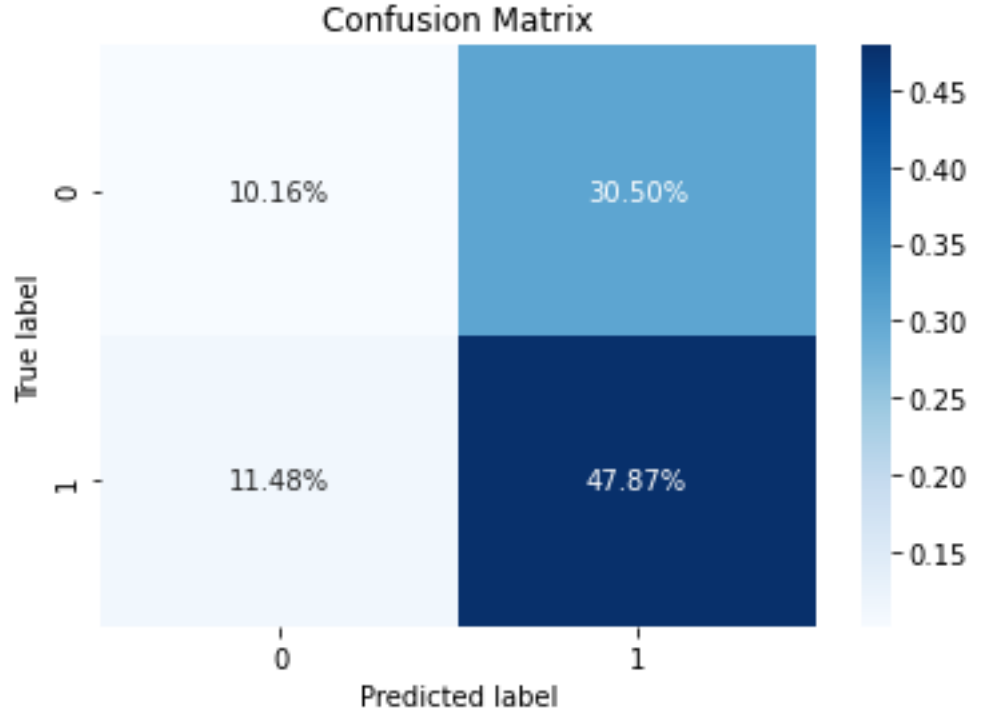
Appendix 10 | Confusion Matrix in a perfect scenario, with values corresponding to the percentage of total predictions.



Appendix 11 | Confusion matrix of the SVM model with values corresponding to the percentage of total predictions.



Appendix 12 | Confusion Matrix of the LSTM model, with values corresponding to the percentage of total predictions.



Appendix 13 | Confusion Matrix in the LR model, with values corresponding to the percentage of total predictions.

Appendix 14 | Profitability metrics of the trading signals forecasted by the LSTM in a Sell Only strategy. The P-value relates to the significance of the average daily returns being different from zero. The P-value shows that the returns were not significantly different from zero

Trading performance of sell only strategy

	LSTM	B&H
Mean Daily Return (%)	0,00	0,40
P-value	*0,995*	*0,058*
Annualized Return (%)	-0,37	144,21
Standard Error	0,001	0,002
Standard Deviation (%)	2,06	2,89
Annualized Volatility (%)	39,28	55,24
Kurtosis	13,75	5,35
Skewness	-0,21	0,31
Minimum (%)	-12,00	-11,74
Quartile 1 (%)	0,00	-0,77
Median (%)	0,00	0,46
Quartile 3 (%)	0,00	1,59
Maximum (%)	11,74	13,86
5 % VaR (%)	-2,47	-4,03
5% CVaR (%)	-5,25	-6,29
Profit orders (%)	45,83	60,10
Loss orders (%)	54,17	39,90
Mean Return of Profit Orders (%)	2,55	1,95
Mean Return of Loss Orders (%)	-2,16	-1,95
Max Drawdown (%)	-26,24	-18,07
Sharpe Ratio	-0,01	2,61
Downside Deviation (%)	46,58	39,94
Sortino Ratio	-0,01	3,61

8. References

Achelis, S. B. (2000). Technical Analysis from A to Z. *Search*.

Alexander, C., & Dakos, M. (2020). A critical investigation of cryptocurrency data and analysis. *Quantitative Finance*. https://doi.org/10.1080/14697688.2019.1641347

Allen, F., & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*. https://doi.org/10.1016/S0304-405X(98)00052-X

ANGHEL, D. G. (2020). A reality check on trading rule performance in the cryptocurrency market: Machine learning vs. technical analysis. *Finance Research Letters*. https://doi.org/10.1016/j.frl.2020.101655

Arnott, R. D., Harvey, C. R., & Markowitz, H. (2018). A Backtesting Protocol in the Era of Machine Learning. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3275654

Athey, S., Parashkevov, I., Sarukkai, V., & Xia, J. (2016). Bitcoin Pricing, Adoption, and Usage: Theory and Evidence. *Stanford University Graduate School of Business Research Paper*.

Bhambhwani, S., Delikouras, S., & Korniotis, G. M. (2019). Do Fundamentals Drive Cryptocurrency Prices? *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3342842

Borovkova, S., & Tsiamas, I. (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. *Journal of Forecasting*. https://doi.org/10.1002/for.2585

Cheung, Y. W., & Chinn, M. D. (2001). Currency traders and exchange rate dynamics: A survey of the US market. *Journal of International Money and Finance*. https://doi.org/10.1016/S0261-5606(01)00002-X

Cong, L. W., Li, Y., & Wang, N. (2020). Tokenomics: Dynamic Adoption and Valuation. *The Review of Financial Studies*. https://doi.org/10.1093/rfs/hhaa089

Creamer, G. (2012). Model calibration and automated trading agent for Euro futures. *Quantitative Finance*. https://doi.org/10.1080/14697688.2012.664921

Creamer, G., & Freund, Y. (2010). Automated trading with boosting and expert weighting. *Quantitative Finance*. https://doi.org/10.1080/14697680903104113

Dempster, M. A. H., & Leemans, V. (2006). An automated FX trading system using adaptive reinforcement learning. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2005.10.012

Dempster, M. A. H., Payne, T. W., Romahi, Y., & Thompson, G. W. P. (2001). Computational learning techniques for intraday FX trading using popular technical indicators. *IEEE Transactions on Neural Networks*. https://doi.org/10.1109/72.935088

Detzel, A., Liu, H., Strauss, J., Zhou, G., & Zhu, Y. (2020). Learning and predictability via technical analysis: Evidence from bitcoin and stocks with hard-to-value fundamentals. *Financial Management*. https://doi.org/10.1111/fima.12310

Dyhrberg, A. H., Foley, S., & Svec, J. (2018). How investible is Bitcoin? Analyzing the liquidity and transaction costs of Bitcoin markets. *Economics Letters*. https://doi.org/10.1016/j.econlet.2018.07.032

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*. https://doi.org/10.2307/2325486

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2017.11.054

Foley, S., Karlsen, J. R., & Putnins, T. J. (2019). Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies? In *Review of Financial Studies*. https://doi.org/10.1093/rfs/hhz015

Gandal, N., Hamrick, J. T., Moore, T., & Oberman, T. (2018). Price manipulation in the Bitcoin ecosystem. *Journal of Monetary Economics*. https://doi.org/10.1016/j.jmoneco.2017.12.004

Gorenc Novak, M., & Velušček, D. (2016). Prediction of stock price movement based on daily high prices. *Quantitative Finance*. https://doi.org/10.1080/14697688.2015.1070960

Graves, A. (2014). *Generating Sequences With Recurrent Neural Networks*. https://arxiv.org/pdf/1308.0850.pdf

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. https://doi.org/10.1109/ICASSP.2013.6638947

Graves, A., & Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*.

Griffin, J. M., & Shams, A. (2020). Is Bitcoin Really Untethered? *Journal of Finance*. https://doi.org/10.1111/jofi.12903

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers and Operations Research*. https://doi.org/10.1016/j.cor.2004.03.016

Jain, P. K., McInish, T. H., & Miller, J. L. (2019). Insights from bitcoin trading. *Financial Management*. https://doi.org/10.1111/fima.12299

Jung, A. (2018). Machine Learning: Basic Principles. In *arXiv*.

Liu, Y., & Tsyvinski, A. (2020). Risks and Returns of Cryptocurrency. *The Review of Financial Studies*. https://doi.org/10.1093/rfs/hhaa113

Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *Journal of Finance*. https://doi.org/10.1111/0022-1082.00265

Menkhoff, L., & Taylor, M. P. (2007). The obstinate passion of foreign exchange professionals: Technical analysis. *Journal of Economic Literature*. https://doi.org/10.1257/jel.45.4.936

Nelson, D. M. Q., Pereira, A. C. M., & De Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2017.7966019

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*.

Rechenthin, M. D. (2014). Machine-learning classification techniques for the analysis and

prediction of high-frequency stock direction. *Iowa Research Online*.

Schilling, L., & Uhlig, H. (2019). Some simple bitcoin economics. *Journal of Monetary Economics*. https://doi.org/10.1016/j.jmoneco.2019.07.002

Schwendner, P. (2020). Advances in Financial Machine Learning. *Quantitative Finance*. https://doi.org/10.1080/14697688.2019.1703030

Sockin, M., & Xiong, W. (2018). A Model of Cryptocurrencies. In *Working Paper*.

Yermack, D. (2015). Is Bitcoin a Real Currency? An Economic Appraisal. In *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*. https://doi.org/10.1016/B978-0-12-802117-0.00002-3