Profitability assessment and optimization for algorithmic trading based on the Long Short-Term Memory model in the cryptocurrency market

Vincent Heijstek
STUDENT NUMBER: 573753

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

dr. Yash Satsangi
dr. Grzegorz Chrupała

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
May 2021

Preface

Dear reader,

Thank you for taking the time to read this thesis about trading strategies in the cryptocurrency market, based on the Long Short-Term Memory algorithm. Firstly, and most importantly, I would like to thank my thesis coordinator dr. Yash Satsangi, for providing constructive feedback on the thesis drafts and taking a lot of time to help me out. I have found our discussions during the our online meetings very helpful. Furthermore, I would like to thank people close to me for their help with co-reading and their support.

I hope you enjoy reading this thesis. Please reach out for me if you want to discuss anything or hear more about it.

Best regards,

Vincent Heijstek

Profitability assessment and optimization for algorithmic trading based on the Long Short-Term Memory model in the cryptocurrency market

Vincent Heijstek

In this work, we investigate the profitability of trading strategies in the cryptocurrency market, based on the Long Short-Term Memory model. The prediction task has been researched extensively, but the relationship with regards to profitability remains unclear. This research uses price and volume data of three cryptocurrencies, and shows that improved model accuracy does improve profitability. However, this correlation is not that simple. Besides, this research investigated a number of methods for increasing the profitability of a trading strategy. We concluded that improving profitability is certainly possible, although dependent on certain conditions.

1. Introduction

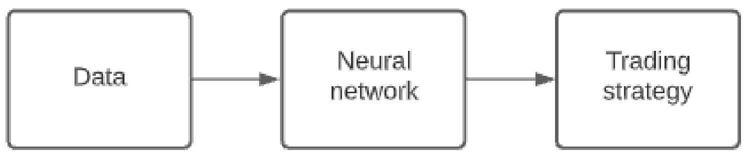
The goal of this research is to examine the profitability of trading strategies based on prediction models in the cryptocurrency market, and investigate methods to improve profitability.

The prediction of financial markets has been an object of studies for many decades. Numerous researchers, mathematicians, and computer scientists have tried to construct a prediction model, that could assist them in making trading decisions (Chang 2011). This quest for a working model has become even more popular in recent years, with the growing opportunities of complex algorithms and trading bots (Chang 2011; Park and Wang 2020). To illustrate, up to 80% of the overall trading volume nowadays is generated through algorithmic trading (Samuelsson 2021). Algorithmic trading is commonly defined as the use of computer algorithms to automatically make trading decisions (Hendershott, Jones, and Menkveld 2011). Hereby, the algorithm uses data to make a decision on whether to buy or sell a certain asset. In recent years, most academic interest on algorithmic trading has emerged from complex statistical models, like deep neural networks (Sebastião and Godinho 2021). What algorithmic trading models are, and how they predict price movements is explained later in the introduction.

But first, a brief overview of the structure of algorithmic trading with neural networks is provided. First, relevant data is extracted, and this data is fed to a neural network. A neural network can be used for regression tasks and classification tasks. In a regression task, the neural network will predict the price of a financial asset, whereas in a binary classification task, the model will predict whether the price of a financial asset will go up or down in the next period. According to Ican, Celik et al. (2017), only classification-based models are successful when using a neural network, for algorithmic trading. This is because regression-based prediction models with little forecasting error do not necessarily translate into profitable trading strategies (Leung, Daouk, and Chen

2000; Ican, Celik et al. 2017). After constructing a data set, a classification-based neural network model is built. After that, the price movement predictions (i.e. up or down) of the neural network are used as the buy and sell signals in an algorithmic trading strategy. In other words, buy an asset when the neural network predicts an upward price movement, and sell an asset when the model predicts an downward price movement. Figure 1 visualizes this structure and represents the main structure of this thesis. In the next paragraphs, these three elements will be discussed in more detail.

Figure 1
Decision architecture ensemble method



1.1 Data - The Cryptocurrency Market

Along the increasing popularity for algorithmic trading, a completely new financial market has gained traction in the last years: the cryptocurrency market, driven by digital currency Bitcoin (Fang et al. 2020). Bitcoin was founded in 2008 by Satoshi Nakamoto, and introduced a peer-to-peer payment system in which users did not have to rely on a trusted third-party, like a bank, for payment verification (Nakamoto 2008). Ever since, multiple other digital currencies have been created, that all rely on a similar technology, called blockchain technology. In the last ten years, the value of Bitcoin has skyrocketed from $0.0008 in 2009 to around $60,000 at the time of writing, with a market value of over 1 trillion US dollars (Edwards 2021; Coinmarketcap 2021).

The increase of interest in the cryptocurrency market is also visible in the academic domain, as shown by the considerable growth of publications related to cryptocurrency trading (Fang et al. 2020). Trading cryptocurrencies has a number of benefits compared to trading stocks and other financial assets, which could explain the increased interest in this domain. First, cryptocurrencies are highly volatile, which means that the prices fluctuate heavily. This provides traders with great money-earning opportunities, although it also increases the risk (Fang et al. 2020). Second, the cryptocurrency market is available 24 hours a day, 7 days a week, whereas the stock market is only available for a number of hours, 5 days a week. This provides more trading opportunities (Fang et al. 2020). Finally, the cryptocurrency market is a peer-to-peer market, and therefore do not involve financial institution intermediaries, which lowers transactions costs (Fang et al. 2020). Because of these benefits, the cryptocurrency market attracts numerous traders. Besides, the novelty of the market and the differences compared to regular financial markets, also seem to attract academics. Therefore, cryptocurrencies will be the research domain of this thesis, and data on cryptocurrencies will be fed to the neural network.

1.2 Neural Network - Algorithmic Trading using Neural Networks

The academic field on algorithmic trading focuses mostly on complex statistical models, like deep neural networks (Fang et al. 2020). Deep learning allows models, that are composed of multiple hidden processing layers, to learn representations of data with multiple levels of abstraction (LeCun, Bengio, and Hinton 2015). Deep neural networks pose a promising approach for extracting the patterns of financial assets, which are usually complex, nonlinear, and noisy (Hoseinzade and Haratizadeh 2019). Various deep learning algorithms have already been used successfully to predict patterns in the cryptocurrency market, such as Convolutional Neural Networks (CNN) (Jiang and Liang 2017), Bayesian Neural Networks (BNN) (Jang and Lee 2017; McNally, Roche, and Caton 2018), Artificial Neural Networks (ANN) (Nakano, Takahashi, and Takahashi 2018; de Souza et al. 2019; Ji, Kim, and Im 2019; Mallqui and Fernandes 2019), and the Long Short Term Memory (LSTM) model (Ji, Kim, and Im 2019; Lahmiri and Bekiros 2019; Smuts 2019; Chen, Li, and Sun 2020).

The LSTM model is currently the most-used model in research for this prediction task (Hu, Zhao, and Khushi 2021). LSTM models are variations of Recurrent Neural Networks (RNN), which are networks with a specialized architecture designed to deal with sequential (e.g. time-series) data. The main idea of these RNN's is that the state of the hidden layer is updated based on the current state, as well as its own previous state (Elman 1990). However, RNN's are difficult to train to capture long distance dependencies, and deals with the problem of vanishing or exploding gradients. The LSTM solves this issue, and therefore enables the hidden state to span over long sequences (Hochreiter and Schmidhuber 1997). Therefore, this model appears to be performing well in predicting nonlinear time series. Besides, the LSTM model is considered as the state-of-the-art model for this prediction task, since it has proven to yield the most accurate results (Hu, Zhao, and Khushi 2021).

Although these models show increasing ability to find a way in the complex and noisy data, the connection between the neural network and algorithmic trading (i.e. a trading strategy) seems limited. In the literature study of Sebastião and Godinho (2021), only 13 out of the 23 articles tested their forecasting algorithm on a trading strategy. Moreover, out of these 13 articles that tested profitability, only a few articles take relevant conditions like transaction costs into account. Besides, regular trading considerations like measures for risk management, and techniques for profit optimization, are unused. Therefore, the link between the accuracy of a neural network, and potential profitability that can be obtained when implementing the neural network in a trading strategy, is yet unclear in the cryptocurrency market.

1.3 Trading Strategy - The Profitability of Trading Strategies

In order to trade with a prediction model, one needs a trading strategy. Trading can be defined as the "*act of buying and selling of [financial assets] with the intention of making a profit*" (Fang et al. 2020). When looking at a single asset, a trader wants to buy and sell at certain periods, to generate a greater return than when simpling holding the asset for a period of time. A trading strategy can be defined as "*a set of predefined rules to buy and sell on [financial] markets*" (Fang et al. 2020). As mentioned above, most recent algorithmic trading techniques rely on quantitative techniques, in order to identify the ideal buy and sell moments. To implement these buy and sell signals successfully, one relies on trading goals, risk controls, and trading or signaling rules, to find the right balance between profit and risk (Fang et al. 2020). An important note here is that, in order to identify a how to construct profitable trading strategies, one needs to have a model that performs well in both bullish, bearish, and stagnant market conditions. A bullish market condition exists when a market is on the rise, and most financial assets are increasing in value. On the other hand, a bearish market is marked by a decline of value of financial assets (Kramer 2021). In a stagnant market, the value of financial assets neither grows or shrinks. A number of trading rules can be applied, and these will be discussed in the following paragraphs.

First, trading rules used for providing buy or sell signals will be discussed, and various researchers have looked into the use of technical indicators for this purpose. Technical indicators are heuristics or mathematical calculations based on the price or volume of a certain financial asset, and can assist traders for deciding the right entry points (i.e. moment to buy a stock) and exit points (i.e. moment to sell a stock) (Chen 2021). For instance, Ellis and Parbery (2005) investigated trading based on the Simple Moving Average (SMA), Vezeris, Kyrgos, and Schinas (2018) looked into the algorithmic trading Moving Average Consolidation Divergence (MACD) technical indicator, and Chong and Ng (2008) researched the Relative Strength Index (RSI). How these technical indicators work and how these are calculated, will be discussed in more detail in the related work.

Second, there are trading rules with regards to risk management. According to Vezeris, Kyrgos, and Schinas (2018), traders in the stock market can increase profit and reduce risk by implementing take-profit and stop-loss configurations in their trading strategy. Take-profit configurations include setting a certain price above the current price of a stock, at which one sells their stock at a profit (Vezeris, Kyrgos, and Schinas 2018). A stop-loss conversely, includes setting a certain price below the current price, at which one will sell his stock at a loss, in order to prevent further losses Vezeris, Kyrgos, and Schinas (2018).

All in all, there are two main research problems. First, it is unclear how profitable algorithmic trading, based on neural networks, is in the cryptocurrency market. More specifically, it is unclear whether a better model in terms of accuracy always leads to a more profitable trading model. This is important to know when identifying profitable trading strategies. Second, methods for improving profitability, such as trading rules based on technical indicators and risk management, have only been tested in other financial markets. Hence, it is unclear whether these methods also increase profitability of algorithmic trading in the cryptocurrency market.

This thesis identifies how to construct profitable trading strategies in the cryptocurrency market. Since this market has a number of benefits compared to other financial markets, like the stock market, this could lead to valuable insights for traders. Besides, a better understanding of the relationship between model accuracy and profitability of a trading strategy addresses an important research gap.

The main research question that will be addressed in this research is:

- How can we identify profitable trading strategies based on the LSTM model in the cryptocurrency market?

To examine this main topic, three sub-questions will be explored for the cryptocurrency market:

- Does an increase in model accuracy of a LSTM model generate more profit for the related trading strategy, and is this constant across different market conditions?
- Does the addition of technical indicators into a trading strategy improve profitability, and is this constant across different market conditions?
- Do take-profit and stop-loss trading rules improve profitability of a trading strategy, and is this constant across different market conditions?

The rest of this paper is organized as follows. In section 2, the related work on predictive models and trading is discussed. Section 3 discusses the LSTM model, as well the trading simulations. Section 4 elaborates on the prediction task, and elaborates on the trading simulations under different market conditions. Section 5 discusses the results, and section 6 and 7 include the discussion and conclusion.

2. Related Work

In this chapter, a brief overview of prediction techniques used for financial market prediction will be given. Second, prediction models used in the cryptocurrency market will be described in more detail, as well as the relationship between the performance of a neural network, and the profitability of the related trading strategy. Finally, methods than can improve the profitability of a trading strategy will be explored.

2.1 A brief history of financial market prediction

The prediction of financial markets has been studied for many decades. In the early stages, financial economists dominated the research domain, who were skeptical about the prediction task. In 1970, Fama (1970) introduced the efficient-market hypothesis (EMH), which states that market prices (e.g. the price of a stock) reflect all information available. This means that every piece of information of a certain stock, is incorporated in the price. Therefore, is it impossible to purchase undervalued stocks, or sell stocks for inflated prices, since stocks always trade at their fair value. Based on this hypothesis, it is impossible to perform better than the overall market. An often linked theory is the random walk hypothesis, which states that stocks signal a random walk, expressing the random thus unpredictable movement of stock prices (Malkiel 1999). These financial economists hereby made a strong statement: beating and predicting the financial market is an impossible task.

However, beating financial markets can be very profitable, and thus several efforts were conducted to discard the EMH and beat the market. Broadly, two main categories of trading arose: Fundamental and technical analysis (Murphy 1991). The fundamental approach relies on examining what something is worth, based on the fundamental characteristics of an asset. Conversely, technical analysis relies on the study of market action. As described by Murphy (1991): "*The fundamentalist studies the cause of market movement, while the technician studies the effect.*". This effect is examined by reading the chart, and the use of a variety of technical indicators (Murphy 1991).

In the early 90s, a third kind of forecasting technique strategy gained traction: Quantitative trading (Fang et al. 2020). This type and uses complex statistical models to predict price movements (Fang et al. 2020). Kimoto et al. (1990) were one of these early researchers, and used a neural network to predict stocks of the Tokyo Stock Exchange.

From this point onwards, many variations of complex statistical models followed. These complex models were better able to extract the patterns of financial data (Hu, Zhao, and Khushi 2021).

All in all, the advance of computing and complex modeling seem to provide opportunities for successful financial market predictions.

2.2 Forecasting techniques, model performance and profitability

Looking at the cryptocurrency market, deep neural networks are the most used prediction algorithms, of which the different types of neural networks are discussed in the introduction (Sebastião and Godinho 2021). This section will examine these research papers in more detail:

First, almost all articles solely investigate Bitcoin prices, the largest cryptocurrency in terms of market cap valuation (Coinmarketcap 2021). Other cryptocurrencies that are relatively frequently researched are Ethereum, and Litecoin (Sebastião and Godinho 2021). Many cryptocurrencies are fewer than five years old, which could explain the limited focus on alternative cryptocurrencies.

Second, all reviewed papers use trading and price data as inputs for the prediction task. However, some articles also add other input variables: sentiment analysis (Kim et al. 2016; Smuts 2019), exogenous financial variables (Catania, Grassi, and Ravazzolo 2019; Mallqui and Fernandes 2019; Sun, Liu, and Sima 2020), blockchain features (Madan, Saluja, and Zhao 2015; McNally, Roche, and Caton 2018; Ji, Kim, and Im 2019; Mallqui and Fernandes 2019; Chen, Li, and Sun 2020), and technical analysis indicators (Żbikowski 2016; Nakano, Takahashi, and Takahashi 2018; Vo and Yost-Bremm 2020; Huang, Huang, and Ni 2019; Borges and Neves 2020).

Third, only a marginal number of reviewed papers use a trading strategy to evaluate the performance of the forecasting algorithm. More specifically, most researchers either do not evaluate on a trading strategy, or use a simplified trading simulation. Researchers acknowledge the simplicity of these strategies as well. Ji, Kim, and Im (2019) calls its trading strategy simple since it only buys and sells at the shown period, not taking transaction costs into consideration. Atsalakis et al. (2019) mention the fact that their buy-sell strategy does not take into account the methods to reduce return variance, used in real trading strategies. Additionally, Atsalakis et al. (2019) only assessed taking long positions, not looking at short positions (i.e. sell a security with plans to buy it later). Finally, Alessandretti et al. (2018) did not take intraday price fluctuations into account, which also simplifies the trading strategy applied. These limitations demand for a closer look at the possibilities of trading for testing these forecasting algorithms.

2.3 Profit optimalization methods

Since the focus on profit maximization seems limited in research on prediction tasks for cryptocurrency prices, this section will dive into trading techniques to improve profitability. As discussed in the introduction, there are two types of trading rules that are common in trading strategies. On the one hand, people use technical indicators for generating buy and sell signals, and on the other hand, take-profit and stop-loss configurations are oftenly used as a risk management technique. This section will discuss both concepts in more detail. Regarding the technical indicators, the following indicators are discussed: the Simple Moving Average (SMA), Moving Average Convergence/Divergence (MACD), and the Relative Strength Index (RSI).

The SMA is a lagging indicator, which plots the average asset prices over time, based on a predefined horizon (e.g. 10 previous days). The direction of crossing of the current price with the SMA, provides buy and sell signals: Buy when the current price crosses the SMA from below, sell when the current price crosses the SMA from above (Ellis and Parbery 2005). The SMA can be calculated using the formula below, where A_n indicates the price of an asset at period n, and n indicates the number of total periods.

$$SMA = \frac{A_1 + A_2 + ... + A_n}{n} \quad , \tag{1}$$

The MACD is a trend-following momentum indicator, and is based on two different lines. The first line is called the MACD line, and calculates the difference between two exponential moving averages (EMA), namely the fast EMA minus the slow EMA. The EMA is a type of a SMA that puts a greater weight on the most recent data points. The fast EMA is usually calculated using the closing prices of the 12 previous days, whereas the slow moving EMA is usually calculated using the 26 previous days. This line is the blue line in figure 2. Besides, there is the signal line, which is calculated by taking the 9-day EMA of the values of the MACD line. The signal line is the yellow line in figure 2. It is regarded as a bullish, and therefore buy signal, when the MACD line crosses the signal line from below. Conversely, it is regarded as a bearish sign when the MACD line crosses the signal line from above (Vezeris, Kyrgos, and Schinas 2018). The formulas for calculating the MACD line, the signal line, and the EMA are provided below. A_t indicates the price on an asset at period t, S indicates a smoothing factor, which is usually equal to 2, n indicates the number of days used for calculating the EMA. EMA_{t-1} refers to the value of yesterday's EMA. Consequently, $EMA_{12}(A_n)$ refers to the fast EMA, and $EMA_{26}(A_n)$ refers to the slow EMA.

$$EMA_n = (A_t * (\frac{S}{1+n})) + EMA_{At-1} * (1 - (\frac{S}{1+n})) \quad , \tag{2}$$

$$MACDline = EMA_{12}(A_n) - EMA_{26}(A_n) \quad , \tag{3}$$

$$Signalline = EMA_9(MACDline) \quad , \tag{4}$$

The RSI is a momentum indicator that measures the magnitude of recent price changes, in order to evaluate whether the price of an asset is overbought or oversold. The value of the RSI can range between 0 and 100, and an asset is considered overbought when its RSI is above 70, while it is oversold when the RSI is below 30. When the RSI is below 30, a buy signal is given, whereas an RSI is above 70 results in a sell signal (Chong and Ng 2008). Chong and Ng (2008) research showed that both the MACD and RSI can outperform the buy-and-hold strategy. However, the usefulness of these trading rules based on technical indicators has not been tested in the cryptocurrency market at the time of writing. The necessary formulas for calculating the RSI are shown below. $AvgU_n$ refers to the average of all up moves in the last n price bars, whereas $AvgD_n$ refers to the average of all down moves in the last n price bars.

$$RSI = \frac{100 - 100}{1 + RS} \quad , \tag{5}$$

$$RS = \frac{AvgU_n}{AvgD_n} \quad , \tag{6}$$

Next to having trading rules for buy or sell signals, traders usually also incorporate trading rules for risk management. These include rules to systematically take profit and avert losses. state that the addition of take-profit and stop-loss to a trading strategy could considerably improve profitability. A take-profit rule sets a certain price target above the current price, and automatically sells the asset if that price target has been reached, allowing to cash in profits gained. A stop-loss rule sets a price target below the current price, and automatically sells the asset at that price in order prevent further losses (Vezeris, Kyrgos, and Schinas 2018). These trading rules for risk management has also not been tested in the cryptocurrency market at the time of writing.

From section 2.1, it became clear that a wide variety of research has been done on the price movement prediction of financial assets. Section 2.2 evaluated price prediction methods for cryptocurrencies in more detail, and stated that the examination on profitability of prediction models is limited. Section 2.3 looked into trading and the possibility of improving the profitability in trading, and stated that these profit optimization techniques have not been tested in the cryptocurrency market yet. In the next section, the methods of this research will be described. First, a number of variations of the LSTM are constructed, and the buy and sell signals that these models provide will be used in a trading simulation. How these trading simulations work, will also be described. These results will be used to answer sub-question 1. Second, a number of trading rules, extracted from the technical indicators discussed above, will be used to create an ensemble model, to answer sub question 2. Finally, take-profit and stop-loss trading rules will be added in the trading simulation, in order to answer sub question 3. All in all, this should provide insight in how to design profitable trading strategies for forecasting algorithms in the cryptocurrency market, answering the main research question.

3. Methods

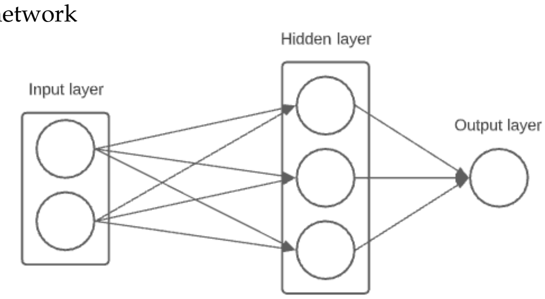
In this section, the methods used in this thesis will be discussed. First, the prediction model to predict price movements is described. A LSTM model will be used for this purpose, since the related work showed that this is currently the most popular algorithm for this prediction task. Second, the trading simulation is described, used to get an insight into the profitability of the signals provided by the LSTM model. Third, technical indicators are added to the trading strategy, to investigate how how these affect profitability. Finally, take-profit and stop-loss are added to the trading simulation to explain their effectiveness for increasing profitability.

3.1 Long Short-Term Memory model

As mentioned in the introduction, the LSTM model is a variant of the Recurrent Neural Network (RNN), which is consequently a variant or type of a neural network. This chapter will provide more background about the architecture of neural networks in general, the RNN and the finally the LSTM.

Neural networks allow models to have multiple layers, which enables these models to learn complex data. As shown in figure 2, the architecture of neural networks is usually based on three layers: the input layer, hidden layer, and output layer. In the example of figure 2, there is one hidden layer, that is composed of three dimensions. Each dimension contains certain weights and a bias, and by training the model with the training data, these weights can be optimized in order to find the optimal weights. Optimal weights are weights that can predict the outcome (i.e. output layer), with the minimal error (LeCun, Bengio, and Hinton 2015).

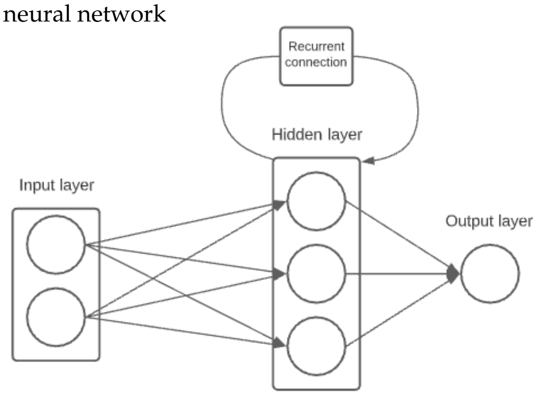
Figure 2
Architecture neural network



RNN are a specialized architecture designed to deal with sequential data, and is therefore useful for time-series data (Elman 1990). These neural networks have a hidden layer that is updated based on the current state, but also on its previous state, which is known as recurrence. The architecture of this is visualized in figure 3. The recurrent connection consists of hidden-to-hidden weights, that are shared across time-steps. Therefore, the model can learn from prior inputs, and is therefore successful in predicting time-steps (Elman 1990; LeCun, Bengio, and Hinton 2015).

RNN's seem to be the right type of model for this research, since we are predicting time-series data. However, RNN's have difficulties when trying to capture long distance dependencies. This is because the gradients of a RNN tend to become very small (i.e. vanishing gradients), or very large (i.e. exploding gradients). Gradients are used for updating the weights that are present in the hidden layers, and when these vanish or

Figure 3
Architecture recurrent neural network



explode, the model is not able to learn properly anymore (LeCun, Bengio, and Hinton 2015). The solution for this issue has been invented by (Hochreiter and Schmidhuber 1997), and is called the LSTM model.

The LSTM model poses a solution by focusing on two types of memory: The long- and short-term memory. The weights that are present in the hidden layers are a sort of long-term memory, whereas the evolving hidden state works as a short-term memory. These two memory states enables the hidden state of the network to span over long sequences, and relieves the issue of exploding or vanishing gradients. The main idea behind the LSTM architecture is that it has a number of gates, that either let information pass or block the information (Hochreiter and Schmidhuber 1997). This improves the recurrent connection seen in figure 4, and is therefore considered as state-of-the-art for prediction data related to financial assets over time (Hu, Zhao, and Khushi 2021).

In this thesis, the LSTM model will be trained using cryptocurrency data. The architecture of the LSTM neural network, as well as the network configurations, will be explained in section 4.

3.2 Trading simulation

The LSTM model will predict whether the price of a cryptocurrency will go up or down in the following day. These predictions are interpreted as buy and sell signals. In our trading simulation, the starting balance will be 1000 fictive dollars. When the forecasting model provides a buy-signal, the selected cryptocurrency is bought with all available dollars, at the current price. Conversely, when the forecasting model provides a sell-signal, all available assets are sold for the price at that moment. For each buying transaction, a 0.1% transaction fee is deducted from the total amount of dollars. This is the same transaction fee as Binance uses, one of the major cryptocurrency exchange at the moment (Binance 2021).

All in all, this buy-and-sell strategy is constructed to provide a trading experience, as would be observed when putting the trading strategy in practice for a period of time. Therefore, it provides a realistic insight into the possible returns that could obtained over a period of time. This strategy is similar to the simplified trading strategies used in the articles reviewed in the related literature that included transaction costs. In the next two sections, extensions of the simplified trading strategy will be described, by the addition of an ensemble method and a risk management strategy.

3.3 Technical indicators: Ensemble method

As mentioned in the related work, technical indicators can assist traders to identify current market conditions. In this research, technical indicators are used to assist the trading strategy based on the LSTM model. This is done by using trading rules, that can be generated from the technical indicators. These trading rules are the same to the ones discussed in related work: SMA, MACD, and RSI. For the SMA, a number of different preceding days have been tested for its usefulness. After a process of trial-and-error, 25-day SMA proved to be the most valuable indicator. According to Sebastião and Godinho (2021), ensemble methods have been proven to be successful in literature, and this part of this research will test if an ensemble with technical indicators improves the profitability of a forecasting algorithm in the cryptocurrency market. The following trading rules are incorporated into the trading strategy:

- If $SMA_{t-1} > close_{t-1}$ and $SMA_t < close_t$: Buy signal =+1
- Else if $SMA_{t-1} < close_{t-1}$ and $SMA_t > close_t$: Sell signal =+1
- Else: Neutral signal

- If $MACD_{t-1} < 0$ and $MACD_t > 0$: Buy signal =+1
- Else if $MACD_{t-1} > 0$ and $MACD_t < 0$: Sell signal =+1
- Else: Neutral signal

- If $RSI_t < 30$: Buy signal =+1
- Else if $RSI_t > 70$: Sell signal =+1
- Else: Neutral signal

- If buy signal >= 1: Buy
- Else if sell signal >= 1: Sell
- Else: Neutral

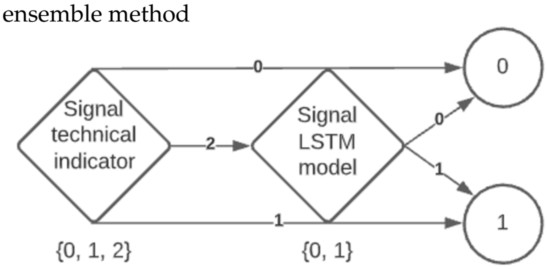
Second, the ensemble method is constructed. Ensemble learning is the process by which multiple models are generated and strategically combined to solve a particular problem (Polikar 2012). For this task, it works as follows: if one of the technical indicators gives a buy or sell signal, this signal will overrule the signal of the LSTM model. The reason for overruling is that the LSTM gives a buy or sell signal at every given period (i.e. 0 or 1), whereas the technical indicators can also give a neutral signal (i.e. 0, 1, and 2). As shown in table 1, the technical indicators provide a lot less buy and sell signals than the LSTM model. To get a better insight into the usefulness of these technical indicators, these indicators get more power in deciding whether to buy or sell.

Table 1
Buy-sell signals of the LSTM model and the technical indicators for Bitcoin (sample period 1)

Average buy-and-sell signals Cryptocurrency	LSTM	Technical Indicators		
		SMA25	MACD	RSI
Buy-signals	153	15	27	9
Sell-signals	151	15	24	9

All in all, this part of this thesis generates an extension to the LSTM model by the use of technical indicators. The ensemble method hereby expands the simplified buy-sell trading strategy, since technical indicators have the tendency to sell an asset if it is overbought, and buy if it is oversold. Therefore, these indicators can be seen as a method for risk management, and thereby address the limitation described in previous research (Atsalakis et al. 2019). Figure 1 describes the decision architecture of the ensemble method.

Figure 4
Decision architecture ensemble method



3.4 Trading simulation with take-profit and stop-loss

Finally, take-profit and stop-loss configurations are added to the simple trading simulation. The addition of these configurations address two limitations found in the relation work. First, take-profit and stop-loss configurations are known as methods to reduce variance and thereby risk, since risk is averted by the stop-loss and take-profit ensures profit is taken along the way. This is is used as a measure for risk management, and addresses the limitation posed by (Atsalakis et al. 2019). Second, these configurations take intraday price fluctuations into account, by checking the highest and lowest price of the day, addressing the limitation mentioned by (Alessandretti et al. 2018). For the first trading simulation, the take-profit percentage is set at 10%, whereas the stop-loss percentage is set at 3%, as suggested by Teixeira and De Oliveira (2010). The added trading rules for this simulation are as follows:

At the moment of buying assets (t):

- Take-profit trigger = 1.1 * $close_t$
- Stop-loss trigger= 0.97 * $close_t$

Each following moment ($t + 1$ ... t_n):

- If take-profit < $high_t$: Sell at take-profit price
- If stop-loss > low_t: Sell at stop-loss price

Here, $high_t$ refers to the highest price of day t, and low_t refers to the lowest price of day t. Besides the simulation with a 10% take-profit and 3% stop-loss, this thesis will also investigate whether different take-profit and stop-loss percentages leads to a higher profit. This is because the percentages suggested by Teixeira and De Oliveira (2010) were tested on the stock market, and Liu and Serletis (2019) stated that the cryptocurrency market is more volatile than the stock market. Therefore, this experiment will widen the take-profit and stop-loss boundaries, to see whether that better suits the cryptocurrency market. All possible combinations between a 3 and 20 percent take-profit and stop-loss will be examined, with steps of one percent.

4. Experimental setup

This section describes the procedure of the experiments. It starts with describing the dataset, the data pre-processing, and the data partitioning. After that, the LSTM model configurations are described, and the method for performing time-series forecasting. Finally, the evaluation metrics and the used soft- and hardware will be discussed.

4.1 Dataset

The input data that is used for this research is extracted from cryptocurrency exchange Binance. This is done through an Application Programming Interface (API), and the data is organized in Python. Any cryptocurrency price data that is on the Binance exchange, can be generated with this API. Data can be extracted in hourly, daily, weekly and monthly format. Since most research is focused on the daily price data, this thesis extracts daily data (Sebastião and Godinho 2021). This daily data includes the open, high, low, and close price of each day, as well as the trading volume. The variables are summarized in table 2.

Table 2
Variables in dataset

Variable	Definition
$open_t$	The opening price on day t
$high_t$	The highest price on day t
low_t	The lowest price on day t
$close_t$	The closing price on day t
$volume_t$	The total volume on day t

As mentioned in related work, most research in cryptocurrency markets is focused on predicting price movements for Bitcoin, and only a marginal part of the researchers include other cryptocurrencies, like Ethereum and Litecoin. In this thesis, Bitcoin data will be extracted, as well the data of two other major cryptocurrencies: Ethereum and Litecoin. The addition of these cryptocurrencies improve generalizability and help to test the models in different market conditions. The sample period of Bitcoin and Ethereum is from 14-07-2017 till 11-02-2021, which constitutes 1309 trading days. The sample period of Litecoin is from 13-12-2017 till 11-02-2021, which constitutes 1225 days.

4.2 Pre-processing and normalization

The ground-truth labels used for prediction are derived from the input data. As mentioned in the introduction, one could use either regression or classification for this prediction task, and since the purpose of this algorithm is to create buy and sell signals and generate profit, a binary target value is preferred. For classification, the ground-truth labels are extracted from the data by the following formulae:

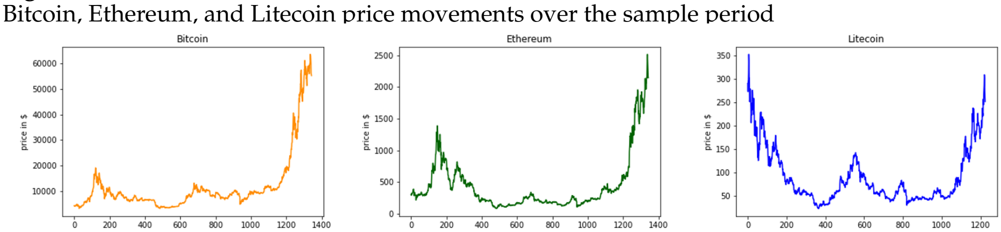
- If $(close_{t+1} \geq close_t)$, label = 1
- If $(close_{t+1} < close_t)$, label = 0

The input data for the model is normalized, in order to transform every feature to the same scale. Min-max normalization has been used to distribute the original scores on a range of 0 to 1.

4.3 Data Partitioning

In order to get familiar with the data and the researched cryptocurrencies, this section will consist of a Exploratory Data Analysis, after which the data partitioning will be substantiated. Figure 5 plots the Bitcoin, Ethereum, and Litecoin price in U.S. Dollars over the sample period, mentioned above. As shown, all cryptocurrencies made significant moves over the last months. The correlation between the price movements in the cryptocurrency market is significant, according to (Aslanidis, Bariviera, and Martínez-Ibañez 2019).

Figure 5
Bitcoin, Ethereum, and Litecoin price movements over the sample period



Usually in machine learning applications with a rolling window approach, a three-sub-samples logic is used (Sebastião and Godinho 2021). In this approach, the first sample is used for training the model. The second, validation sample is used for the optimization of the hyperparameters of the model, in order to find out what configuration works best on the data. In the final, test sample, a untouched sample of data is used for testing the performance of the model (Sebastião and Godinho 2021). The first part of the thesis however, aims to identify the relationship between model accuracy and profitability. In order to test this properly, multiple models are desired, with varying accuracy and profitability. Therefore, this approach for optimizing hyperparameters is not used in this research. Optimizing model performance for the prediction task has been researched extensively, and would therefore have little academic relevance. Hence, this research uses a two sub-samples approach. Similar to Sebastião and Godinho (2021), the first 75% of the data is used for training the model, and the final 25% for testing. However, as figure 6 shows, this approach only allows us to test our models in bullish markets.

This is not desirable, because this thesis aims to generalize its results over different market conditions. Therefore, also a sample that tests in a bearish or stagnant market is desired. Hence, this research also uses a second sample period, in which the first 50% of the data is used for training, and the following 25% of the data as the test set, as shown in figure 6. This will enable us to investigate our findings in bullish, stable, and bearish market conditions.

Figure 6
Bitcoin, Ethereum, and Litecoin data partitioning 1

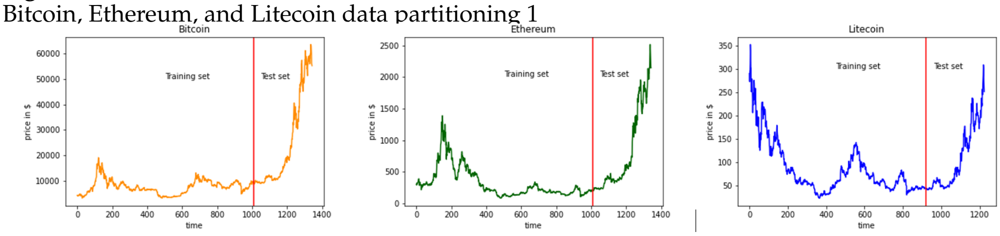
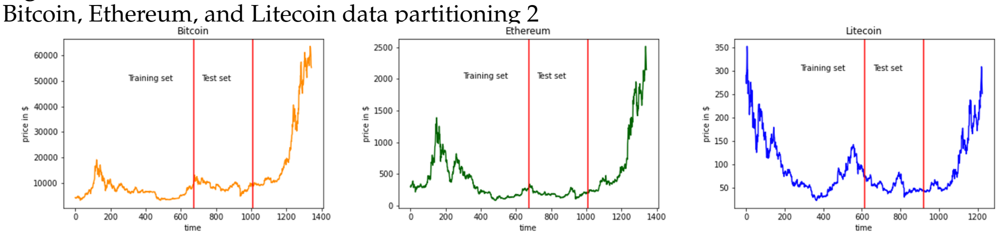


Figure 7
Bitcoin, Ethereum, and Litecoin data partitioning 2



4.4 Time-series methodology

Since this is a time-series forecasting, we need to use an approach that works for this type of forecasting. Sebastião and Godinho (2021) state that the window rolling approach is the most common approach for this prediction task. In the window rolling approach, each following day is predicted by using m preceding days as input variables. After testing a number of different values for m, a value of $m = 25$ is chosen as the window rolling period. To sum up, for predicting the price movement at day t, the 25 preceding days are used. Here, each day consists of the opening price, $open_t$, highest price, $high_t$, lowest price, low_t, closing price, $close_t$, and the volume $volume_t$ of that day.

4.5 Model configurations and hyperparameters

In order to generate multiple LSTM models, a number of hyperparameters settings will be used. The different configurations are shown in table 2. Through trial-and-error, 500 epochs has shown to yield the best results in terms of validation accuracy, together with a layer size of 64 and the Adam optimizer with a 0.001 learning rate. For other variables, different combinations are used to construct different LSTM models.

First, there are three different combinations for the hidden layer. The first option is having no hidden layers, which are expected to predict the data poorly. The other two options have either one or two hidden layers, which are expected a better job in predicting the data.

Second, there are three different combinations regarding dropout. Dropout is a method to avoid overfitting that randomly omits input or hidden layers during training. This prevents complex co-adaptations in which a feature detector is only helpful in the context of several other specific feature detectors (Hinton et al. 2012). If dropout is equal to zero, there is no dropout used, whereas if dropout is equal to 0.2, 20% of the features are randomly omitted. Finally, is dropout is equal to 0.5, 50% of the features are randomly removed.

Third, there are three different combinations regarding the activation functions used in our LSTM models. If the activation function is equal to 'none', no activation function is used. There are two other activation functions used in our models. The first activation function is the rectified linear activation function, ReLU, which is a linear function that will output the input directly if it is positive. Otherwise, it will output zero, as shown in formula 7. The second activation function is the hyperbolic tangent, tanh, which maps the input values to values between -1 and 1, by using the formula 8. In both formulas, the value z is the value that is outputted by the LSTM model.

$$ReLU(z) = max(z, 0) \quad , \tag{7}$$

$$tanh(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}} \quad , \tag{8}$$

Finally, a sigmoid activation function is used to map the results of the LSTM model to a binary target: 0 or 1. The sigmoid function is defined by the following formula. Here, z is again the value that is provided by the LSTM model.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad , \tag{9}$$

Eventually, 54 different LSTM models are created through different combinations of hyperparameters. These are summarized in table 3.

Table 3
Model configurations for the LSTM model

Hyperparameters	**Configurations**
Number of neurons per layer	64
Number of hidden layers	0, 1, 2
Dropout	0, 0.2, 0.5
Activation function	None, Tanh, ReLU
Batch size	48, 96
Number of epochs	500
Optimizer	Adam
Learning rate	0.001

4.6 Evaluation metrics

First, the forecasting model performance is evaluated by accuracy. As mentioned before, this is the most suitable method for evaluating forecasting models with the purpose of optimizing profit. The formula for accuracy is provided below.

$$Accuracy = \frac{CorrectPredictions}{TotalPredictions} \quad , \tag{10}$$

Second, the profitability of the three different trading strategies, are all evaluated by simulating the trading experience over the sample period. The three different simulations, being the simplified simulation, ensemble method, and method with take-profit and stop-loss, were discussed in section 3.

Third, the Pearson correlation coefficient is used to evaluate the relationship between accuracy and profitability. This coefficient calculates the covariance of two variables, divided by the product of their standard deviations (Baron-Epel, Dushenat, and Friedman 2001). The null-hypothesis here is that there is no correlation between forecasting accuracy and trading profitability. This null-hypothesis will be tested for the simple trading strategy.

Fourth, an independent t-test is conducted in order to investigate whether the ensemble method and the method with profit-taking significantly differ from the simplified trading strategy. Here, the null-hypothesis is that there is no difference between the mentioned strategies.

4.7 Software used

The programming language used is Python 3.6. Using Anaconda Navigator, the processing took place in the JupyterLab (version 2.2.6). The LSTM models have been constructed using Keras (Chollet et al. 2018). Multiple additional libraries and packages have been used during this research:

- Pandas (McKinney et al. 2011)
- NumPy (McKinney 2012)
- Scikit learn (Pedregosa et al. 2011)
- Keras (Chollet et al. 2018)

Everything has been hosted on a HP Envy 13-ad1xx, with a 1.60GHz Intel i5 Core with 8GB RAM.

5. Results

In this section, the results of the experiments conducted will be described. To sum up, three different trading strategies were tested. The first strategy is based on the LSTM model, which buys when the LSTM model predicts an upward price movement, and vice versa. This will be called the simple buy-sell strategy. The second trading strategy is also based on the LSTM model, but here the technical indicators are used as an extra price movement prediction approach. The third strategy is similarly based on the LSTM model, but here take-profit and stop-loss configurations are used in the trading strategy. First, the results of the test accuracy and the profit of the simple buy-sell simulation will be described. These results provide an insight into the relationship between model accuracy and trading profitability. After that, the results of the trading strategy with technical indicators will be described, as well as the trading strategy with take-profit and stop-loss extensions. These results will be compared to the simple buy-sell strategy in order to get an insight into the change in profitability.

5.1 Results simple strategy

After creating 54 different LSTM models for Bitcoin, Ethereum, and Litecoin, for two different sample periods, we can evaluate the test accuracy for of these models. Besides, we can also evaluate the profitability of trading strategies using these buy and sell signals. Table 4 summarizes the average findings of the 54 different LSTM models for both sample periods. In the first sample period, Bitcoin, Ethereum and Litecoin are tested in a bearish or stagnant period of time. This is because the buy-hold strategy, in which an asset in bought at the beginning of the sample period, and sold at the end of the sample period, is close to or below the initial amount of 1000 dollars. In the second sample period, Bitcoin, Ethereum and Litecoin are tested in a bullish period of time, since the buy-hold strategy exceeds the initial amount of 1000 dollars.

Results indicate that for sample period 1, the simple buy-sell strategy outperformed the buy-and-hold strategy for Bitcoin and Litecoin. For Ethereum, however, simple buy-sell strategy did not outperform the buy-and-hold strategy. In the second, bullish, sample period, none of the buy-sell strategies outperformed the buy-hold strategy.

The last two columns of table 4 show the Pearson correlation correlation coefficient on the correlation between test accuracy and the profitability of the buy-sell strategy, and the respective p-values. As shown, the correlation in the first sample period was significantly different from zero for Bitcoin and Litecoin, but not for Ethereum. However, in sample period 2, the correlation between the test accuracy and the profit of the buy-sell strategy was significant for all cryptocurrencies.

Table 4
Simple strategy. All values indicate the mean of the 54 samples, whereas all values in brackets indicate the standard deviation.

Pair	Test accuracy	Buy-hold	Buy-sell	Correlation	p-value
Sample period 1					
BTC	0.519 (0.022)	933	**956 (295)**	**0.329**	0.015
ETH	0.484 (0.021)	969	666 (220)	0.154	0.265
LTC	0.516 (0.027)	664	**814 (462)**	**0.384**	0.004
Sample period 2					
BTC	0.481 (0.028)	5897	1989 (874)	**0.744**	**<0.001**
ETH	0.501 (0.031)	9363	2748 (1054)	**0.639**	**<0.001**
LTC	0.491 (0.030)	5984	2765 (1278)	**0.586**	**<0.001**

When identifying a profitable trading strategy, one would mostly be interested in the best performing models. If we focus on the best 10 models in terms of test accuracy, we would get the results as shown in table 5. On average, the best 10 models in terms of test accuracy all provide significant profits with the buy-sell strategy, except for Ethereum in the first sample period. Looking at the best model in terms of test accuracy (e.g. test accuracy (max) & buy-sell (max)), each cryptocurrency outperformed both the buy-sell and buy-hold strategy, except for Bitcoin and Ethereum in sample period 2.

Table 5
Simple strategy (Best 10 models in terms of test accuracy)

Pair	Test accuracy (mean)	Test accuracy (max)	Buy-hold	Buy-sell (mean)	Buy-sell (max)
Sample period 1					
BTC	0.551	0.567	933	1167	1698
ETH	0.515	0.544	969	751	1429
LTC	0.552	0.575	664	1223	**2823**
Sample period 2					
BTC	0.522	0.564	5897	3064	5444
ETH	0.547	0.574	9363	4200	5116
LTC	0.530	0.553	5984	4067	**6718**

5.2 Ensemble profit optimization technique

In this section, the results of the ensemble trading strategy is displayed. Here, we will use an independent t-test to test whether the difference in profitability between the simple strategy and the ensemble strategy is significantly different from zero.

For both sample period 1 and sample period 2, it can be observed that the ensemble method's test accuracy is higher than the test accuracy of the LSTM model, as shown in table 6. This shows that technical indicators can improve erroneous buy-and-sell signals of the LSTM model. However, this increase in accuracy, does not automatically cause an increase in profitability. For the first sample period, the ensemble method significantly outperforms the simple buy-sell strategy, for both Bitcoin and Litecoin but not for Ethereum. For the second sample period, only the ensemble profit of Litecoin is higher than the simple buy-sell strategy.

Table 6
Ensemble profit. All values indicate the mean of the 54 samples, whereas all values in brackets indicate the standard deviation.

Pair	Test accuracy	Ensemble accuracy	Buy-hold	Buy-sell	Buy-sell ensemble	p-value
Sample period 1						
BTC	0.519 (0.022)	0.533 (0.018)	933	956 (295)	**1110 (383)**	0.01265
ETH	0.484 (0.021)	0.519 (0.020)	969	666 (220)	629 (270)	0.4391
LTC	0.516 (0.027)	0.528 (0.023)	664	814 (462)	**1058 (547)**	0.0140
Sample period 2						
BTC	0.481 (0.028)	0.524 (0.015)	5897	1989 (874)	1613 (417)	0.005
ETH	0.501 (0.031)	0.542 (0.019)	9363	2748 (1054)	1946 (419)	<0.001
LTC	0.491 (0.030)	0.520 (0.024)	5984	2765 (1278)	**3433 (1452)**	0.012

5.3 Take-profit and stop-loss optimization technique

In this final section, the profitability of trading strategies with take-profit and stop-loss is evaluated. As shown in table 7, the take-profit and stop-loss configurations as suggested by Teixeira and De Oliveira (2010) (e.g. 10% tp and 3% sl), significantly outperform the profit obtained with the simple buy-sell strategy, for all three cryptocurrencies in sample period 1. However, in the second sample period, the strategy with take-profit and stop-loss configurations significantly underperforms the simple strategy. Finally, the last column on table 7 indicates that significantly more profit could be obtained when finding the optimal take-profit and stop-loss configurations.

Table 7
Take-profit and stop-loss profit. All values indicate the mean of the 54 samples, whereas all values in brackets indicate the standard deviation. *** Indicates that the take-profit/stop-loss profit is significantly (<0.001) different from the simple strategy profit.

Pair	Test accuracy	Buy-hold	Buy-sell	Buy-sell 10% take-profit 3% stop-loss	Buy-sell Optimized take-profit and stop-loss
Sample period 1					
BTC	0.519 (0.022)	933	956 (295)	1228 (258)***	1730
ETH	0.484 (0.021)	969	666 (220)	904 (217)***	1915
LTC	0.516 (0.027)	664	814 (462)	1071 (350)***	2281
Sample period 2					
BTC	0.481 (0.028)	5897	1989 (874)	**1427 (380)*** **	2475
ETH	0.501 (0.031)	9363	2748 (1054)	**1730 (393)*** **	3416
LTC	0.491 (0.030)	5984	2765 (1278)	**1459 (448)*** **	**6933**

Table 8 describes the optimal take-profit and stop-loss configurations in more detail. For the first sample period, the optimal configurations are quite close to the optimal configurations suggested by Teixeira and De Oliveira (2010), being a 10% take-profit and a 3% stop-loss. For the second sample period however, the take-profit and stop-loss percentages are very different from the initial configurations. The optimal take-profit percentage is around 15%, whereas the optimal stop-loss percentage is around 9%.

Table 8
Sample period 1 - Optimal configurations for take-profit and stop-loss

Pair	Take-profit (mean)	Take-profit (median)	Stop-loss (mean)	Stop-loss(median)
Sample period 1				
BTC	10.39	11	5.30	3
ETH	9.74	10.5	3.94	3
LTC	12.85	13	4.51	5
Sample period 1				
BTC	12.76	14	8.48	4.50
ETH	15.76	16	10.83	10
LTC	16.24	17	6.02	4

6. Discussion

The main goal of this thesis is to investigate how to identify profitable trading strategies that can be applied to prediction models in the cryptocurrency market. This thesis is focused on the relationship between model accuracy and profitability, and explores techniques to improve profitability. These techniques have been researched in other financial markets, like the stock market, but not yet in the cryptocurrency market. By discussing the sub-questions posed for this research, we will get a better insight into the main research question for this research: *How can we identify profitable trading strategies that work with prediction models in the cryptocurrency market?* Besides, the limitations, implications, and directions for future research will be discussed.

The first sub-question focused on the relationship between model accuracy and profitability. The results show that for five of the six samples, there is a correlation between test accuracy and profit obtained via the corresponding buy-sell strategy. This correlation is considerably stronger in sample period 2, which was the bullish sample period. A reason for this is that in bullish sample periods, considerable profits can be obtained with successfull trades, which will have a big effect on the profitability. Furthermore, the ensemble method section shows that ensemble methods increase model accuracy for all cryptocurrencies in each sample period. However, the increase in accuracy did not always lead to a higher profitability. The profitability only increased for two cryptocurrencies in the first sample period, and one cryptocurrency in the second sample period. For the other three samples, the profitability was lower, although the accuracy was higher. This could be due to the fundamental characteristics of technical indicators, which are inclined to sell if the market is overbought, and buy if the market in oversold. This could have led to selling a profitable position too early. All in all, the results indicate that better models in terms of accuracy are expected to yield better results for a trading strategy. However, a good predictive model does not necessarily guarantee a profitable trading strategy.

The second sub-question focused on the effect of adding technical indicators as an ensemble method to the simple trading strategy. For all trading pairs in both sample periods, the ensemble accuracy was considerably higher than the regular test accuracy. Increased accuracy means more successful predictions of price movevements, and therefore one would expect the profitability to increase, but this was not the case. The addition of technical indicators only contributed to a higher profitability for Bitcoin and Litecoin in the first sample period, and for Litecoin in the second sample period. For the other samples observed, it did contribute negatively to the profitability. Hence, the addition of technical indicators may be profitable in a stagnant market condition (i.e. Bitcoin and Litecoin in sample period 1). For a bearish market condition (i.e. Ethereum in sample period 1), it seems like the technical indicators have little effect on the profitability. However, more samples would be needed to verify this statement. In the bullish market condition, we have found contradictory results, and drawing a conclusion here would be difficult. All in all, it is clear that technical indicators improve the accuracy of a prediction model. However, it remains unclear whether the addition of technical indicators positively or negatively affects the profitability of a trading strategy. This finding is different from the findings in the stock market, which suggested that technical indicators improve the profitability of a trading strategy (Vezeris, Kyrgos, and Schinas 2018; Chong and Ng 2008; Ellis and Parbery 2005). Further research on the addition of technical indicators in a bullish, stagnant, or bearish market condition, could provide a better understanding in this relationship for the cryptocurrency market.

The last sub-question evaluated the profitability of the trading strategy with added take-profit and stop-loss configurations. First, these results showed that a 10% take-profit and 3% stop-loss can significantly increase the profitability in a bearish or stagnant market. On the other hand, these configurations significantly decrease the profitability in a bullish market condition. Besides, the take-profit and stop-loss configurations lowered the standard deviation for both sample periods, which indicates that the variability of profits and thereby the risk of the trading strategy is reduced. The optimal take-profit and stop-loss percentages suggest that significantly more profit can be obtained when finding the optimal percentages. In a bullish sample period, these optimal percentages were on average very high, which indicates that it is better to hold on to your positions and not sell too early. In a bearish or stagnant period, these optimal percentages were relatively close to the percentages suggested by Teixeira and De Oliveira (2010). All in all, take-profit and stop-loss configurations can significantly increase the profitability of a trading strategy, especially in a bearish or stagnant market. These increases in profitability can be maximized by finding the optimal percentages for these mechanisms.

There are a number of limitations in this research. First, this research only investigated the LSTM model as the prediction model, even though several other prediction methods have been used over time. This limits the generalizability of our results to the LSTM model only. Although this model is considered as state-of-the-art for the prediction task regarding accuracy, it remains unclear whether this model would also be state-of-the-art for profitability. Second, this research only compared its results to a buy-hold strategy, instead of comparing it to other baselines as well. The addition of other baselines would provide more clarity in the performance compared to other methods. Third, this research did not used a three-sub-samples approach, which is normally used for the optimization of prediction models for the financial market. In order to actually design a profitable trading strategy, one is expected to use this approach. However, the two-sub sample method satisfied for the purpose of identifying profitable trading approaches. Finally, this research limited itself to investigating three different cryptocurrencies: Bitcoin, Ethereum, and Litecoin. Although there is a significant correlation between cryptocurrencies, investigating more cryptocurrencies could expose certain differences in performance (Aslanidis, Bariviera, and Martínez-Ibañez 2019)

This research provides various directions for future research. First, the limitations described above provide directions for increasing the generalizability of this thesis. Second, this research is purely focused on the daily data. In order to investigate the profitability over a longer time frame, it would be interesting to investigate whether hourly or weekly data can be used to improve the trading profitability. Third, the addition of technical indicators and take-profit and stop-loss configurations could be investigated in a separate study, in order to get a more in-depth insight into the value of these trading mechanisms. This has been done already for the stock market, but this research shows that these trading mechanisms work differently in the cryptocurrency market. Besides, this research shows how the results of these trading mechanisms vary in different market conditions. This has not been researched by Vezeris, Kyrgos, and Schinas (2018), nor Chong and Ng (2008), nor Teixeira and De Oliveira (2010), nor Ellis and Parbery (2005). The exploratory scope of this thesis on those subjects hopes to motivate future research to investigate this in more detail.

Finally, this research provides a number of academic and practical implications. The academic implications are mainly described in the paragraph above about the recommendations for future research. Especially the fact that most of the results in this thesis differ in different market conditions, should be taken into account in future research. This is also relevant when putting a LSTM model with added trading rules into practice: The profitability of certain trading rules heavily depend on the current market condition.

7. Conclusion

In this work, the profitability of trading strategies in the cryptocurrency market has been researched. This research showed that in most cases, improved model accuracy results in an increase of profitability. However, this is dependent on different market conditions, and trading rules that were applied. Besides, this research investigated a number of methods for increasing the profitability of a trading strategy. Ensemble methods with technical indicators increased the profitability in a stagnant market period, but decreased profit in a bearish market period. A similar conclusion was drawn for the take-profit and stop-loss configurations. Furthermore, these configurations decreased the standard considerably, which is beneficial from a risk management perspective. In conclusion, predicting price movements is a difficult task, and the same could be said for analyzing profitability. This research has found numerous starting points that can be used by traders or academics for further research.

References

Alessandretti, Laura, Abeer ElBahrawy, Luca Maria Aiello, and Andrea Baronchelli. 2018. Anticipating cryptocurrency prices using machine learning. *Complexity*, 2018.

Aslanidis, Nektarios, Aurelio F Bariviera, and Oscar Martínez-Ibañez. 2019. An analysis of cryptocurrencies conditional cross correlations. *Finance Research Letters*, 31:130–137.

Atsalakis, George S, Ioanna G Atsalaki, Fotios Pasiouras, and Constantin Zopounidis. 2019. Bitcoin price forecasting with neuro-fuzzy techniques. *European Journal of Operational Research*, 276(2):770–780.

Baron-Epel, Orna, Marina Dushenat, and Nurit Friedman. 2001. Evaluation of the consumer model: relationship between patients' expectations, perceptions and satisfaction with care. *International Journal for Quality in Health Care*, 13(4):317–323.

Binance. 2021. Fee schedule.

Borges, Tomé Almeida and Rui Ferreira Neves. 2020. Ensemble of machine learning algorithms for cryptocurrency investment with different data resampling methods. *Applied Soft Computing*, 90:106187.

Catania, Leopoldo, Stefano Grassi, and Francesco Ravazzolo. 2019. Forecasting cryptocurrencies under model and parameter instability. *International Journal of Forecasting*, 35(2):485–501.

Chang, Tsung-Sheng. 2011. A comparative study of artificial neural networks, and decision trees for digital game content stocks price prediction. *Expert systems with applications*, 38(12):14846–14851.

Chen, James. 2021. Technical indicator.

Chen, Zheshi, Chunhong Li, and Wenjun Sun. 2020. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, 365:112395.

Chollet, François et al. 2018. Keras: The python deep learning library. *Astrophysics Source Code Library*, pages ascl–1806.

Chong, Terence Tai-Leung and Wing-Kam Ng. 2008. Technical analysis and the london stock exchange: testing the macd and rsi rules using the ft30. *Applied Economics Letters*, 15(14):1111–1114.

Coinmarketcap. 2021. Todays cryptocurrency prices by market cap.

Edwards, John. 2021. Bitcoin price history.

Ellis, Craig A and Simon A Parbery. 2005. Is smarter better? a comparison of adaptive, and simple moving average trading strategies. *Research in International Business and Finance*, 19(3):399–411.

Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Fama, Eugene F. 1970. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417.

Fang, Fan, Carmine Ventre, Michail Basios, Hoiliong Kong, Leslie Kanthan, Lingbo Li, David Martinez-Regoband, and Fan Wu. 2020. Cryptocurrency trading: a comprehensive survey. *arXiv preprint arXiv:2003.11352*.

Hendershott, Terrence, Charles M Jones, and Albert J Menkveld. 2011. Does algorithmic trading improve liquidity? *The Journal of finance*, 66(1):1–33.

Hinton, Geoffrey E, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hoseinzade, Ehsan and Saman Haratizadeh. 2019. Cnnpred: Cnn-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129:273–285.

Hu, Zexin, Yiqi Zhao, and Matloob Khushi. 2021. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1):9.

Huang, Jing-Zhi, William Huang, and Jun Ni. 2019. Predicting bitcoin returns using high-dimensional technical indicators. *The Journal of Finance and Data Science*, 5(3):140–155.

Ican, Ozgur, Taha Bugra Celik, et al. 2017. Stock market prediction performance of neural networks: A literature review. *International Journal of Economics and Finance*, 9(11):100–108.

Jang, Huisu and Jaewook Lee. 2017. An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *Ieee Access*, 6:5427–5437.

Ji, Suhwan, Jongmin Kim, and Hyeonseung Im. 2019. A comparative study of bitcoin price prediction using deep learning. *Mathematics*, 7(10):898.

Jiang, Zhengyao and Jinjun Liang. 2017. Cryptocurrency portfolio management with deep reinforcement learning. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 905–913, IEEE.

Kim, Young Bin, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, and Chang Hun Kim. 2016. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PloS one*, 11(8):e0161197.

Kimoto, Takashi, Kazuo Asakawa, Morio Yoda, and Masakazu Takeoka. 1990. Stock market prediction system with modular neural networks. In *1990 IJCNN international joint conference on neural networks*, pages 1–6, IEEE.

Kramer, Leslie. 2021. An overview of bull and bear markets.

Lahmiri, Salim and Stelios Bekiros. 2019. Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals*, 118:35–40.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

Leung, Mark T, Hazem Daouk, and An-Sing Chen. 2000. Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of forecasting*, 16(2):173–190.

Liu, Jinan and Apostolos Serletis. 2019. Volatility in the cryptocurrency market. *Open Economies Review*, 30(4):779–811.

Madan, Isaac, Shaurya Saluja, and Aojia Zhao. 2015. Automated bitcoin trading via machine learning algorithms. *URL: http://cs229. stanford. edu/proj2014/Isaac% 20Madan*, 20.

Malkiel, Burton Gordon. 1999. *A random walk down Wall Street: including a life-cycle guide to personal investing*. WW Norton & Company.

Mallqui, Dennys CA and Ricardo AS Fernandes. 2019. Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques. *Applied Soft Computing*, 75:596–606.

McKinney, Wes. 2012. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.".

McKinney, Wes et al. 2011. pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, 14(9):1–9.

McNally, Sean, Jason Roche, and Simon Caton. 2018. Predicting the price of bitcoin using machine learning. In *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)*, pages 339–343, IEEE.

Murphy, John J. 1991. *Intermarket technical analysis: trading strategies for the global stock, bond, commodity, and currency markets*, volume 6. John Wiley & Sons.

Nakamoto, Satoshi. 2008. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4.

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

Park, Andreas and Jinhua Wang. 2020. Did trading bots resurrect the capm? *Available at SSRN 3515635*.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Polikar, Robi. 2012. Ensemble learning. In *Ensemble machine learning*. Springer, pages 1–34.

Samuelsson. 2021. What percentage of trading is algorithmic?

Sebastião, Helder and Pedro Godinho. 2021. Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation*, 7(1):1–30.

Smuts, Nico. 2019. What drives cryptocurrency prices? an investigation of google trends and telegram sentiment. *ACM SIGMETRICS Performance Evaluation Review*, 46(3):131–134.

de Souza, Matheus José Silva, Fahad W Almudhaf, Bruno Miranda Henrique, Ana Beatriz Silveira Negredo, Danilo Guimarães Franco Ramos, Vinicius Amorim Sobreiro, and Herbert Kimura. 2019. Can artificial intelligence enhance the bitcoin bonanza. *The Journal of Finance and Data Science*, 5(2):83–98.

Sun, Xiaolei, Mingxi Liu, and Zeqian Sima. 2020. A novel cryptocurrency price trend forecasting model based on lightgbm. *Finance Research Letters*, 32:101084.

Teixeira, Lamartine Almeida and Adriano Lorena Inacio De Oliveira. 2010. A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert systems with applications*, 37(10):6885–6890.

Vezeris, Dimitrios, Themistoklis Kyrgos, and Christos Schinas. 2018. Take profit and stop loss trading strategies comparison in combination with an macd trading system. *Journal of Risk and Financial Management*, 11(3):56.

Vo, Au and Christopher Yost-Bremm. 2020. A high-frequency algorithmic trading strategy for cryptocurrency. *Journal of Computer Information Systems*, 60(6):555–568.

Żbikowski, Kamil. 2016. Application of machine learning algorithms for bitcoin automated trading. In *Machine Intelligence and Big Data in Industry*. Springer, pages 161–168.