BACHELOR’S THESIS

A study of technical indicators, their use in automated trading strategies and implications on individuals participating in financial markets

„Trading on Technical Indicators“

verfasst von / submitted by

Tobias Abraham Haider

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Bachelor of Science (BSc)

Wien, 2022 / Vienna, 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 033 521
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Informatik
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Ing. Dr.-Ing. Moritz Grosse-Wentrup

Acknowledgements

To my partner, Rafaela, who kept me motivated and nourished through this stressful time.
To my parents, especially my father, Engelbert, who taught me to be kind and trusting and supported all the decisions I made in life.
To my friends, Philipp and Florian, who always gave me a good reason to leave my desk and go outside.
To all Austrians who, through study grants, implicitly funded my studies and this thesis work.
To my manager, Oliver Mulzet, who gave me time off work when I needed it.
And most importantly, to all who contribute to peace in the world, allowing me to live a prosperous life without fear.

Thank you.

Abstract

The behavior of financial markets has been studied intensively by many economists and scientists. Most of the theories, rules, and metrics developed to value an asset are based on real-world observations such as the revenue and earnings of a certain company or the supply and demand for goods. Real-world data and metrics are very relevant and are therefore successfully used by governments, fund managers, and private investors to form sensible investment decisions, but they require human work and reporting to create the necessary data. This thesis work is focused on different kinds of metrics called technical or chart indicators. Primarily, they aim to describe previous price movements by quantifying concepts such as performance, risk, and deviations from means, but with the emergence of automatic trading engines, they are also employed with limited success to trade financial assets. Within the thesis, certain popular indicators are calculated, normalized, analyzed in terms of statistical properties, and ultimately tested for their value when building strategies for automated trading systems by simulating their performance using stock price data from various markets from around the world.

Contents

List of Figures

1 History of Technical Analysis

Technical analysis is the activity of analyzing the price history of an asset that is traded between individuals or institutions. Although the term, which is associated with computers, algorithms, and speculation in stock markets is rather new, the underlying activity has existed since ancient Babylon and earlier.

1.1 The Origin of Trade and Speculation

Many concepts, such as marketplaces, the profession of a trader, and business contracts, have their origin in ancient Babylonia.

> "The Akkadian term machiru, which initially had the abstract meaning 'price, market value' and 'commercial activity', acquired the concrete meaning 'marketplace' by the beginning of the Old Assyrian and Old Babylonian periods." [AWL10, p. 5]

Traders recorded prices of common commodities like barley, dates, sesame, and wool on clay tables and tried to predict changes in the commodity prices by analyzing the supply from merchants, demand from customers, and seasonal effects. Even astrology was used to predict the exact price of a commodity in the future [AWL10, p. 4ff].

1.2 Modern Trading of Financial Assets

Although the first publicly traded stock, the Dutch East India Company, was founded in 1602 in Amsterdam, modern financial trading really emerged at the beginning of the twentieth century in the United States, leading to a major market crash that ultimately resulted in the Great Depression of the 1930s [Ken17, p. 148]. After the economy recovered, trading activity picked up again and new strategies and indicators were developed [AWL10, p. 4ff]. In fact, most of the indicators discussed in the following chapters, such as Bollinger Bands and Relative Strength Index, were created between the Great Depression and the end of the twentieth century. Even though multiple millennia had passed, the methods of evaluating prices had not changed fundamentally. Prices were recorded on a regular basis on work sheets and analyzed in terms of their potential to move in a specific direction. Sophisticated indicators were developed that were initially calculated by hand and noted in columns next to the recorded prices, but soon were programmed into computers. Visualizations of changing prices in the form of printed charts were used to draw lines and detect special patterns believed to signal price movements. Many examples of worksheets and descriptions for the manual calculations can be found in books written by the creators of popular technical indicators [Bol02][Wil78][Cha97].

1.3 Electronic Communication and Automated Trading

The implementation of electronic communication in stock trading started in 1969, and alternative trading systems, which allow computers to execute trades, were established as late as in the 1990s at the NASDAQ stock exchange. Initially, automated trading had been heavily regulated in terms of the allowed size of trades and was only available to a few institutions, but regulations changed and numerous companies like Goldman Sachs, Morgan Stanley, Credit Suisse etc., started operating in financial markets using automated trading engines. From statistical arbitrage to automated market makers and private traders, the landscape of participants in the financial markets of today has evolved into a complex system where the impact of private investors placing individual orders seems to be shrinking [Zub11, p. 1ff].

2 Literature Research

Price prediction has been a controversial topic, splitting the academic community since the beginning of the 20th century [KH53, p. 34]. On the one hand, their use in financial trading is compared to astrology and refuted by the theory of random walks and proofs of different forms of market efficiency. On the other hand, they are praised by others for their ability to explain human behavior in financial markets and detect excessive movements of prices prone to mean reversion. In essence, most chart indicators seem to have no relevance when analyzed theoretically but are believed to have significant success when applied in practice.

2.1 Random Walk Theory

The random walk theory as well as the concept of market efficiency are commonly used to refute any claim that information from the past can be used to gain knowledge about future price movements. "[...] the theory of random walks says that the future path of the price level of a security is no more predictable than the path of a series of cumulated random numbers" [Fam65, p. 34]. Fama [Fam65], Alexander [Ale61], Kendall [KH53], among others, support the theory of random walks by showing that it holds when applied to US stock market data, British Industrial share prices, or commodities such as cotton or wheat. The theory originates from Louis Bachelier's doctoral thesis "Théorie de la Spéculation" [Bac00] [May11]. It assumes future prices, on average, to be approximately the same as current prices, and that changes occur randomly. The empirical tests of the theory sometimes mention exceptions where the null hypothesis (prices being random) in the statistical tests does not hold. However, they are ignored because of extraordinary events such as wars or inconsisted data. In total, proponents of the theory of random walks argue that changes in prices are random and future prices can not be predicted. As presented here, much research had been conducted on statistical analysis, but little research existed about "chartist theories that are popular in the financial world" [Fam65, p. 98]. However, new empirical research gives compelling evidence that many commonly used chart patterns, such as the head-and-shoulder pattern, occur completely randomly and have no significant influence on future prices. In fact, some authors even suggest that the occurrence of chart patterns might lead to opposite price movements than traders think, although this is still not relevant enough to be able to take advantage of this algorithmically. The only thing they confirm is an increase in trading activity due to the occurrences of these popular chart patterns [BOS12, p. 627].

2.2 Market Efficiency

Market efficiency is the theory that assumes all existing information is already priced into the value of an asset. It is another hypothesis that has implications similar to the theory of random walks, but uses another approach to explain them. It tries to refute any claim that past price history or even new financial data, such as revenue, earnings, etc., can be used with success to profit from future changes in price [Fam70, p. 383ff].

There are three different forms of market efficiency: strong, semi-strong, and weak. Although strong market efficiency does not make any exception to the statements from above, the other two forms do. The semi-strong form of market efficiency acknowledges the value of information not known to the public. Income statements, political decisions, and financial data in general are examples of such information that affect the price of financial assets and can, according to the theory, be used to estimate price changes if seen before the public. The weak form of market efficiency states that market efficiency is not achieved immediately, once the new information has been made public. Stock splits, for example, typically happen during successful periods of a company and lead to significant price changes. However, after some weeks, the information is fully processed by the market and does not affect the price of the stock anymore [Fam70, p. 405ff]. Applying any of the three forms to technical indicators and technical analysis in general, it would deny any value in using such indicators for trading because the information is publicly known and is based only on historical price data.

2.3 Reversion to the Mean

In contrast to the theory of market efficiency, reversion to the mean is a phenomenon that can be described as the readjustment of a price that has moved away from some hypothetical fair or efficient price. The efficient price can only be estimated and defines the price an asset would have, if all market participants would act perfectly rationally. If it is proven to be present in financial markets, it would invalidate the market efficiency and random walk hypotheses by demonstrating that publicly known information, such as price data, can be used to measure and take advantage of readjustments of asset prices. Studies, which try to show the existence of mean reversions, usually do not completely disregard the market efficiency and random walk hypotheses, but challenge whether they are valid to their full extent.

After their study of reversion to mean effects in US stock market data from the New York Stock Exchange (NYSE) between 1926 and 1985, Poterba and Summers concluded that although "[...] individual data sets do not consistently permit rejection of the random walk hypothesis at high significance levels, the various data sets together strengthen the case against its validity" [PS88, p. 53]. Malliaropulos and Priestley go further and state, concerning their investigations of Asian capital markets, that "[...] results suggest that excess returns exhibit mean reversion in a number of markets. Assuming that expected returns are constant, this evidence suggests that returns are predictable and, hence, can be exploited to earn abnormal profits" [MP99, p. 381]. Numerous studies with similar results have been conducted, showing the discrepancy between theoretically sound hypotheses and practical challenges to validate or refute them [BLL92] [BT85] [LM87].

2.4 Automated Trading Engines

Automated trading is becoming widely used by large institutions but also by individual investors in the stock market. As an example, according to a study conducted in 2011, the portion of trade volume that was automatically traded out of all trades on the DAX, an index that tracks the 30 largest publicly traded companies in Germany, was 52 % [TH11, p.31]. To clarify, this trading volume is not caused by speculative trades being executed by trading bots, but by various types of trading activity facilitated by algorithms to achieve better buying and selling prices, also called spread. They are successful because they take advantage of short-term deviations from a mean and help the discovery of an efficient price [TH11, p.23]. The application of real-time trading demonstrates an important use case for automated trading algorithms, but does not answer to what extent statistical data can be used to signal longer-term deviations from price efficiency and, therefore, estimate the potential for future price inclines or declines. There exist hedge funds, like the Renaissance Technologies Medallion Fund, that heavily rely on quantitative trading for generating their returns. However, the algorithms used are kept secret and only employees of the company are permitted to invest in the fund [Cor20, p. 256ff].

2.5 Machine Learning in Financial Trading

With the emergence of sophisticated machine learning models, most notably neural networks and support vector machines, in recent years, the interest in applying such models to financial trading is increasing rapidly. A survey conducted by the World Economic Forum in 2020 shows that 59 % of investment managers are currently using some kind of artificial intelligence for their operations. This number is likely to grow since the majority of surveyed managers anticipate the contribution of returns from AI-powered applications to their overall returns will be high or very high. In addition to this, the usage rates of artificial intelligence in different trading-related activities further demonstrate how widespread it has become in the field of finance [LR20, p. 100ff].

Recent scientific studies analyzing the capability of machine learning models for predicting price changes or future volatility in stock markets find that well-defined models are, although not consistently, able to create excess returns in many situations and markets. Among various methods, the application of support vector machines and neural networks appears to be particularly popular [CH18, p. 401] [HW12, p. 1397] [STDBR14, p. 3077]. One mentioned reason for the generation of excess returns is the capability of machine learning models to make more objective decisions than humans [CLXH20, p. 1]. Some results of trading strategies powered by deep neural networks suggest that they can be particularly useful in predicting sudden rises or falls in prices [ANHS16, p. 434].

Figure 2.1: Usage rates of AI in investment-related activities
[LR20, p. 101]

3 Software Project

As demonstrated in the Literature Research part, plenty of research on financial markets and its properties has been conducted and published by statisticians and economists. It is, however, harder to gain knowledge about actual algorithms used to prepare data for real-world applications of financial trading and implementations of working trading strategies. This is not to be underestimated, since the basis of any working trading strategy is mined and curated data. The process of preparing financial data is by far not discussed in as much detail as the actual models trained on them, although it is essential to the development of trading engines based on machine learning models.
The software developed during this bachelor project is a lightweight tool, which demonstrates the process of optimizing financial data for machine learning and data analysis tasks. Regarding the mathematical formulas, algorithms etc. defined in the following chapters, the intent was to implement them in the most comprehensible way, so that the algorithms could be reused and tuned when implementing more sophisticated applications in the future.

3.1 Chart Data Source

As a price data source, Yahoo Finance API was chosen [yah]. It can return historical price data for up to 10 years when provided with the ticker symbol of a publicly traded stock. A list of about 75000 stock tickers [tic] is used to iteratively query the API and retrieve price data, resulting in valid price history for about 36000 stocks.
Even though price data is monetized, the market for services providing historical price data has become competitive, reducing the cost of information about asset prices significantly. Changing the API would not lead to much programming work, because all popular API services provide similar endpoints and data formats.

3.2 Used Python Libraries

The core application is a low-level Python program, which does not depend on any frameworks. The only libraries used are NumPy and Pandas. The interactive Python worksheets implemented separately to demonstrate the algorithms used for the calculation of the indicators, examples of trading strategies, and simulations of their performance require more data science libraries, such as Matplotlib, Scipy, Scikit-learn, and PyTorch.

3.3 Configuration of the Program

A default configuration for the algorithms implemented is centrally stored and initialized with empirical values taken from books on financial indicators and online sources further explored in the chapter Technical Indicators. In this way, a user could quickly generate data sets from price data containing numerous indicators without in-depth knowledge of the detailed parameterization of individual indicators.

The only need for configuration is an API key. Without it, no data can be retrieved and processed.

3.4 Implemented Features

The developed Python tool provides the functionality for all steps of the price data processing for later to perform effective data analysis and efficient machine learning. The sequential steps of the process are:

- Download price for multiple assets using a list of tickers
- Persist the raw API response
- Extract relevant information (price data and metadata) from the raw API data
- Calculate numerous indicators for assets individually
- Normalize indicators and price data
- Store price data and indicators in a tabular format using various file types
- Sample price and indicator data from multiple assets
- Accumulate samples in large data sets and store them in various file formats

The examples of interactive Python worksheet sets show the drastically reduced effort for performing data science-related tasks when provided with preprocessed data. The results of data analysis implementations in interactive worksheets are documented in the chapters Simulations and Results.

4 Technical Indicators

A technical indicator is a figure that provides information about the state of a price chart of some stock or traded asset in general. It can, if price data exists, be calculated at any point in time and accumulates information about preceding price movements into one single number.

4.1 Moving Average

The concept of mean has already been described by multiple Greek mathematicians, including Archytas and Hippasus [Hea21, p. 86]. The moving average is simply a new term for the windowed mean in some time series data and is meaningful in the analysis of various types of data, not only price data. For example, a graph showing the average temperature during a year in some place on Earth can be used to get clearer data on a change in climate over some time period [PVPnHGP^{+}15, p. 1681]. When applied to stock prices, the moving average is an indicator measuring the average price in the recent past for any point in time using a specified number of previous price points, also called interval or window length.

4.1.1 Proposed Application of the Moving Average

The moving average is seen as a smoothing tool that can be used to reduce noise in price data and identify trends in price movements. Trends are identified by plotting one or more moving averages and determining for one, the direction they are moving in (up or down or sideways), for another, whether they are above or below the current price, and lastly, how they compare to moving averages with different window sizes [Mur99, p. 197ff].

4.1.2 Moving Average Calculation

The calculation of the moving average is done by convolution of the time series price data with a rectangular window. The resulting operation can be defined as

$$ma[n] = \frac{1}{M} \sum_{k=0}^{M-1} price[n-k] \tag{4.1}$$

where M is the window length and *price* is usually the closing price time series data [AVO98, p. 19f].

4.1.3 Special Versions of the Moving Average Indicator

A criticism of the standard moving average is its slow reaction to recent price changes. For this reason, special versions of the moving average, which give recent prices a higher significance than previous prices, were developed [Mur99, p. 199ff].

Weighted Moving Average

The weighted moving average uses a different window function to calculate the mean. In this way, the mean can be adapted to put a stronger emphasis on some price points, for example, more recent price data [BB17, p. 30].

$$weighted\_ma[n] = \sum_{k=0}^{M-1} weights[k] * price[n-k] \tag{4.2}$$

where *weights* is the window function. The sum of the weights must be 1 in this case.

An example of the weighted moving average is the linear weighted moving average. It linearly increases the weights, so that more recent prices have a greater impact on the result.

$$weights[k] = 2 * \frac{M-k}{M*(M+1)} \tag{4.3}$$

The resulting weights are numbers from 1 to M in reverse order, divided by the Gaussian summation formula. Reversion is needed because of the convolution operation, which applies the weights in reverse order to the price data. [AJ08, p.36]
For the window with size M = 4, the resulting impulse response would be: [0.4, 0.3, 0.2, 0.1]

Exponential Moving Average

The exponential moving average can react even more quickly to large price moves, because it assigns exponentially larger weights to more recent prices.

$$ema[n] = price[n] * weight + ema[n-1] * (1 - weight) \tag{4.4}$$

$$weight = \frac{smoothing}{M+1} \tag{4.5}$$

where the parameter *smoothing* is an arbitrary scalar [Ko18, p.17].

Figure 4.1: The standard moving average

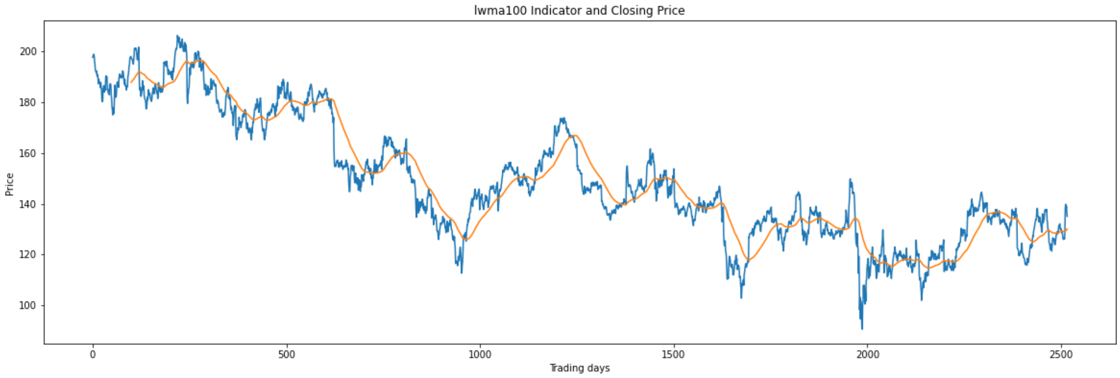


Figure 4.2: The linear weighted moving average

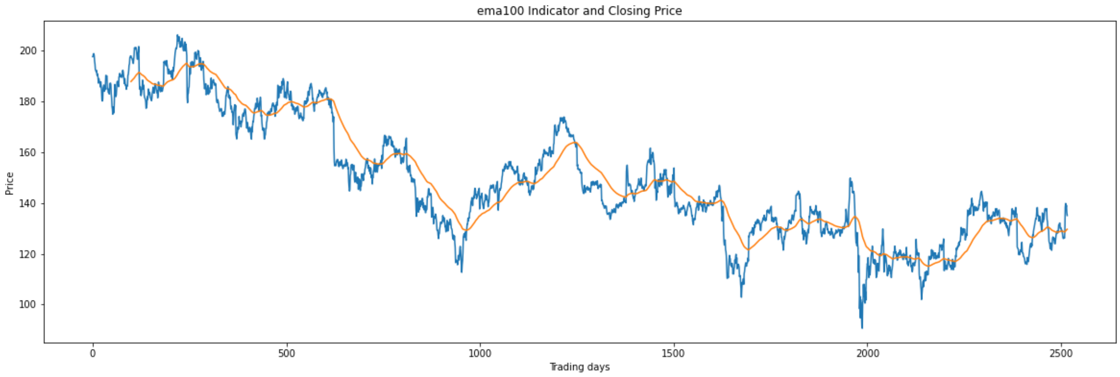


Figure 4.3: The exponential moving average

4.1.4 Example for the Moving Average

When plotted, the smoothing property becomes visible. The exponential and linear weighted versions move a bit closer around the price than the standard version, because higher weights are assigned to recent prices.

4.1.5 Normalization of the Moving Average

Moving averages are not normalized and, like the prices themselves, cannot be compared among different assets. To allow for this comparison, the indicator is shifted and scaled by the price using a feature transformation [Dub20, p.35ff].

$$ma\_normalized[n] = \frac{ma[n]}{price[n]} - 1 \tag{4.6}$$

As a result, the normalized moving average measures the difference of the moving average and the closing price by the magnitude of the closing price. A value of 0 expresses that the moving average is equal to the closing price, a deviation from the closing price is shown by a positive or negative value of the indicator.

4.2 Moving Average Convergence Divergence

The Moving Average Convergence Divergence indicator, short MACD, is a comparison between two exponential moving averages. The indicator consists of the MACD line itself and a smoothed MACD line, called signal line [Mur99, p. 253].

4.2.1 Proposed Application of the MACD Indicator

By explicitly calculating the difference of a short, or fast moving average and a longer, or slow moving average, one tries to better determine the trend of a stock. Gerald Appel, who first presented the indicator, also described trading rules that could be applied to automated trading systems. The described rules generally revolve around the crossing of two lines, which is believed to signal the beginning of a positive or negative movement in price [AD07, p. 13f].

4.2.2 MACD Calculation

First, two exponential averages are calculated as described in the section Exponential Moving Average. Second, the difference between the moving average is taken.

$$macd[n] = short\_ema[n] - long\_ema[n] \tag{4.7}$$

Last, the resulting MACD is smoothed using the formula for the Exponential Moving Average itself.

$$macd\_signal[n] = macd[n] * weight + macd\_signal[n-1] * (1 - weight) \tag{4.8}$$

4.2.3 Example for the MACD Indicator

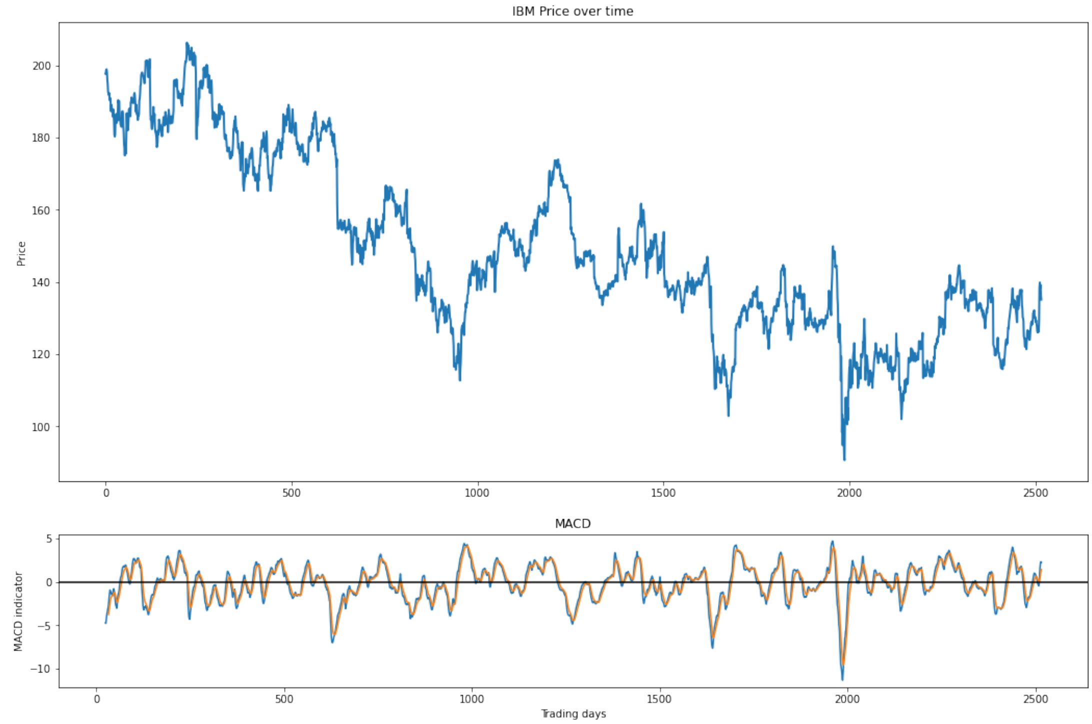


Figure 4.4: The MACD indicator

The indicator is above 0, if the price has moved up recently, and below otherwise. The orange smoothed MACD line is used to compare to the unsmoothed line to detect rapid changes in trend.

4.2.4 Normalization of the MACD

The MACD, as well as the MACD signal line, can be scaled using one of the moving averages. Here, the moving average with a longer time frame is chosen.

$$macd\_normalized[n] = \frac{macd[n]}{long\_moving\_average[n]} \tag{4.9}$$

4.3 Volatility

The volatility is the average variation in price, usually within one day, that can be measured in the previous price data.

4.3.1 Proposed Application of Volatility Indicator

Volatility is associated with risk and is even included in the pricing model of financial assets such as option contracts [Nat94, p. 290ff]. It is a core component of financial markets that is not treated in the same way by all market participants. In portfolio management, risk is avoided, but in other applications, it is also taken advantage of and results in the potential for higher reward when endorsed [Sha94, p. 49ff].

4.3.2 Calculation of Volatility

There exist multiple indices that try to measure volatility in different ways, but the simplest is the standard deviation [Ko18, p. 86].

$$volatility[n] = \sqrt{\frac{1}{M}\sum_{k=0}^{M-1}(price[n-k] - ma[n-k])^2} \qquad \text{[Gow08, p. 265]} \tag{4.10}$$

where $price[n] - ma[n]$ is the difference between the closing price and a moving average on a day.

4.3.3 Example of Volatility

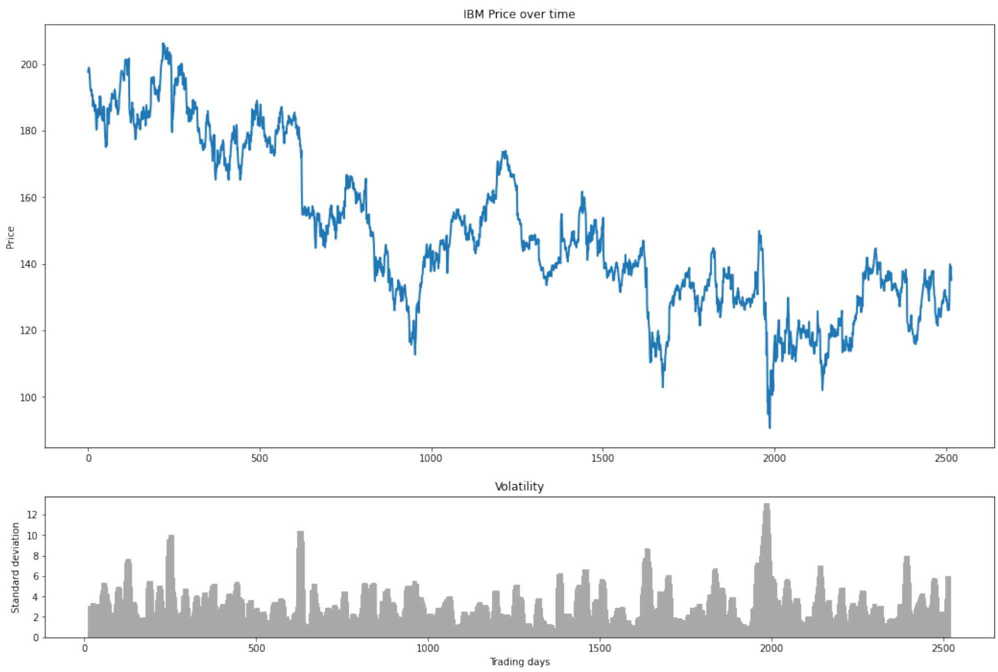


Figure 4.5: The standard deviation

Volatility shows extreme price movements. In this example, the highest volatility was recorded during the onset of the Covid-19 pandemic in early 2020.

4.3.4 Normalization of Volatility

Like the moving average, the standard deviation is larger, when the price of an asset is higher. In order to be able to compare volatility among different assets, the standard deviation is put in relation to the price [Gow08, p. 655].

$$volatility\_normalized[n] = \frac{volatility[n]}{price[n]} \tag{4.11}$$

4.4 Bollinger Bands

Bollinger Bands, named after John Bollinger, are trend lines wrapped around the price chart in a way that the majority of price observations lie between the lower and upper band. They are constructed by combining the moving average indicator and a measure of volatility.

4.4.1 Proposed Application of Bollinger Bands

As the price of an asset usually stays within the defined Bollinger Bands, a move above the upper, or below the lower band is believed to signal an opportunity to enter or exit a position. John Bollinger mentions an example for a rule that can be implemented in an automated trading system using the relation of the current price to demonstrate the use of Bollinger Bands:

> "... when price closes outside the upper Bollinger Band and the 21-day Intraday Intensity(II) is negative, sell." [Bol02, p. 61]

4.4.2 Calculation of Bollinger Bands

With the two needed indicators, the moving average and volatility, defined, the bands are constructed as follows:

$$\begin{aligned} upper\_bollinger[n] &= ma[n] + 2 * volatility[n] \\ lower\_bollinger[n] &= ma[n] - 2 * volatility[n] \end{aligned} \tag{4.12}$$

For the moving average, the standard moving average is chosen. The measure of volatility is the standard deviation. By choosing a distance of 2 standard deviations from the mean, the resulting bands will create a channel where theoretically about 95 % of the price points lie within, assuming the price follows a normal probability distribution [Whe14, p. 34].

4.4.3 Example for Bollinger Bands

The price rarely touches the upper or lower line. In this example, many occasions can be found where the price moved back up after touching the lower line or down after touching the higher line.

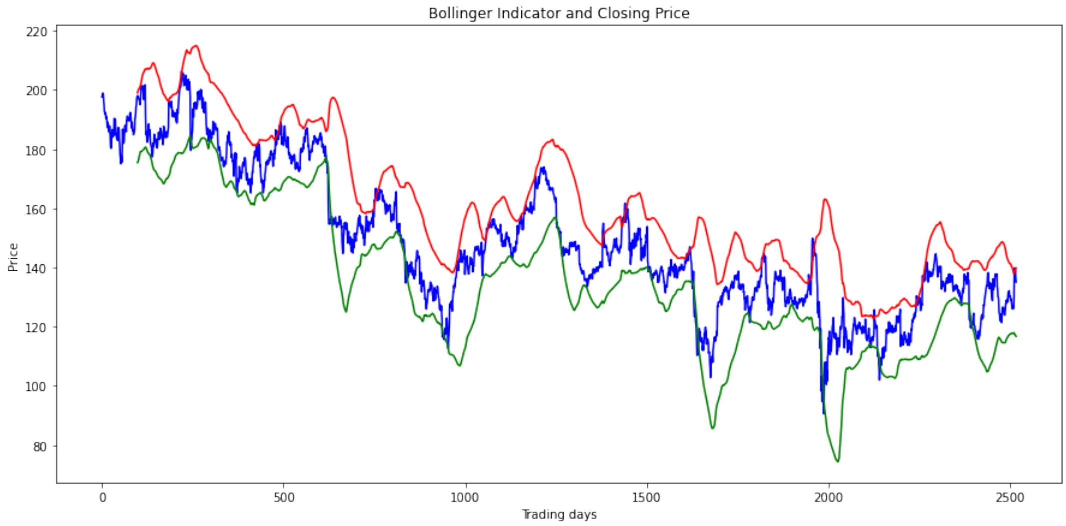


Figure 4.6: The lower and upper Bollinger Band

4.4.4 Normalization of Bollinger Bands

For the use of Bollinger Bands in automated trading systems, John Bollinger proposes a normalization of the Bollinger Bands called b%:

$$b\%[n] = \frac{price[n] - lower\_bollinger[n]}{upper\_bollinger[n] - lower\_bollinger[n]} \qquad \text{[Bol02, p. 60])} \qquad (4.13)$$

In the developed application, the same transformation is chosen but shifted so that it is centered around 0.

$$bollinger\_position[n] = 2 * b\%[n] - 1 \qquad (4.14)$$

4.5 Relative Strength Index

The indicators previously discussed are based on key statistics figures and could be used to analyze various types of time series data. The Relative Strength Indicator was developed by John Welles Wilder and has the properties of an oscillator (an indicator that oscillates between defined boundaries, often 0 and 100). It takes an interval of price data and compares the average upward movements to the average downward movements [Wil78, p. 64ff].

4.5.1 Proposed Application of the Relative Strength Indicator

The indicator is used as an auxiliary tool to detect the so-called "tops" and "bottoms" by quantifying the intensity with which the price moves in a certain direction. Commonly used barriers for the indicator signaling excessive price movements are:

1. rsi > 70: overbought (has moved up significantly)

2. rsi < 30: oversold (has moved down significantly) [Wil78, p. 63ff]

4.5.2 Calculation of the Relative Strength

The relative strength, which is the underlying value used to build the Relative Strength Oscillator, is calculated in multiple steps.

$$move[n] = price[n] - price[n-1] \tag{4.15}$$

$$mean\_up[n] = \sum_{k=0}^{M-1} max(move[n-k], 0) / \sum_{k=0}^{M-1} max(sign(move[n-k]), 0) \tag{4.16}$$

Note, that the expression is calculating the sum of all upward moves in a time interval and dividing it be the number of upward movements.

$$mean\_down[n] = \sum_{k=0}^{M-1} min(move[n-k], 0) / \sum_{k=0}^{M-1} min(sign(move[n-k]), 0) \tag{4.17}$$

The average downward movement is calculated similarly. The operation can be implemented more concisely using the functions np.diff, np.nanmean and np.where from the NumPy library [num].

As a result, the relative strength is the ratio between the average upward movements and the average downward movements.

$$relative\_strength[n] = \frac{mean\_up[n]}{mean\_down[n]} \tag{4.18}$$

4.5.3 Example for the Relative Strength Indicator

Traders generally watch for indicator movements below 30 or above 70. For better visualizations of such signals, vertical lines are plotted for reference.

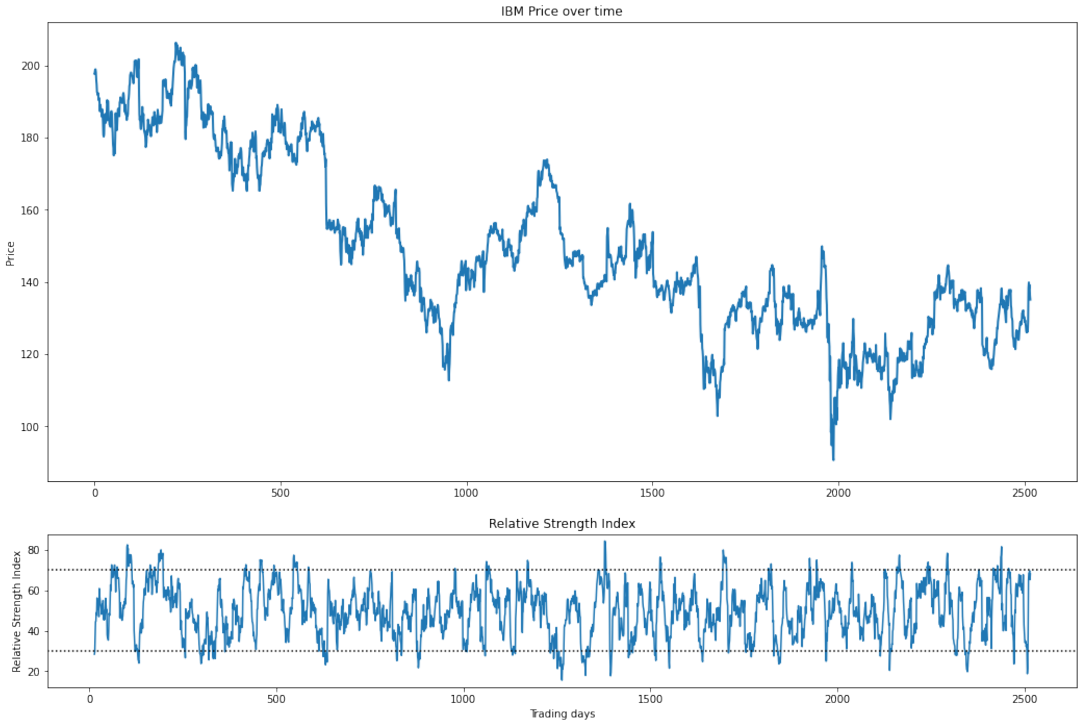


Figure 4.7: The Relative Strength Indicator

4.5.4 Normalization of the Relative Strength Index

The proposed normalization by J. Wilder is the actual Relative Strength Index. It is the inverse of the relative strength, which is shifted, such that it is bounded between 0 and 100 and therefore fits the definition of an oscillator.

$$rsi[n] = 100 - \frac{100}{1 + relative\_strength[n]} \qquad \text{[Wil78, p. 65]} \tag{4.19}$$

Again, this existing normalization is taken, but shifted and scaled, so that it is centered around 0 and lies between -1 and 1.

$$rsi\_normalized[n] = \frac{rsi[n]}{50} - 1 \tag{4.20}$$

4.6 Other Implemented Indicators

Other popular indicators that are used to build trading strategies are implemented in the application developed alongside this written thesis [PM16, p. 1575]. They try to capture similar properties about the price of an asset as previously described indicators, but use different methods aside from the mean and standard deviation. Some indicators, like the

Aaron Indicator or the Average Directional Movement Index also depend on multiple features of the daily price data such as the lowest or highest price of the day.

5 Experiments and Relation to Existing Indicators

During the implementation, some experimental indicators were designed with the idea of modifying other popular indicators or using new methods to improve the expressiveness of an indicator. Due to the existence of numerous indicators, similarities could be found between some experiments made and actual indicators.

5.1 Horizontal Channel

It has been observed that prices of assets like stocks behave differently around certain numbers in terms of how quickly they manage to rise above or fall below. For example, humans tend to place their orders at round numbers and multiples of 10, resulting in resistances and supports for the price [BHJ12, p. 413]. Such prices could be interpreted as horizontal lines, where prices bounce off once, or before they are reached [Sha08, p. 50]. Using this idea of horizontal lines, which are less likely to be breached, an indicator could be created that measures the price position between such borders.

5.1.1 Calculation of the Horizontal Channel

Although more sophisticated measures for a support or resistance could be chosen, the minimal and maximal values in a range are used here.

$$\begin{aligned} resistance[n] &= \max_{0 \leq k < M} price[n-k] \\ support[n] &= \min_{0 \leq k < M} price[n-k] \end{aligned} \tag{5.1}$$

where M is the interval chosen for this indicator.

The position of the price between the support and the resistance is calculated using the formula in section Relative Position of the Price.

5.1.2 Example of Horizontal Channels

As seen in the plot, the price is never moving above the upper line and below the lower line. This is the case because it instantly becomes the new highest/lowest price and, therefore, the value of the indicator.

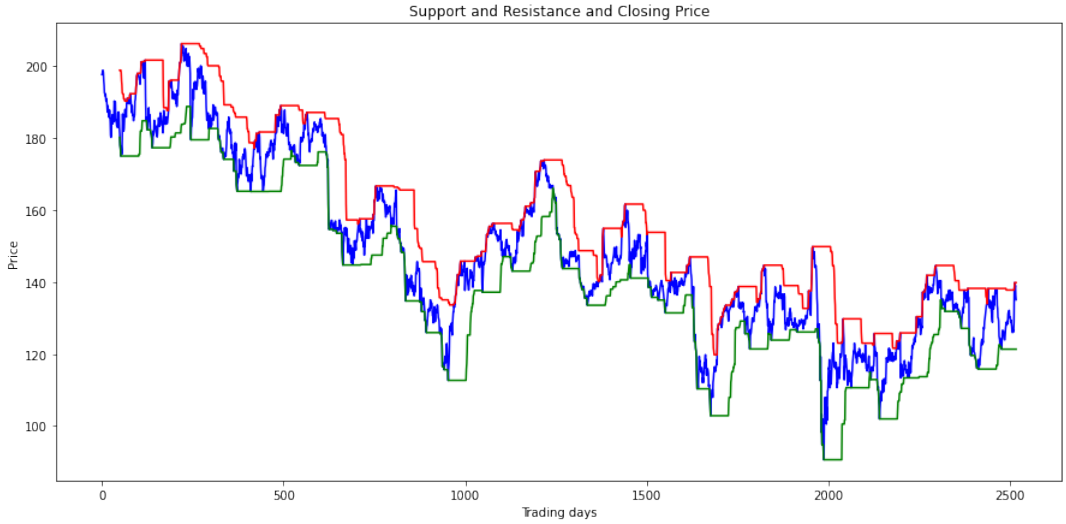


Figure 5.1: The recent support and resistance

5.2 Regression Channel

Another approach to creating a channel using linear support and resistance lines is regression. By placing a line above and below the regression line, a similar channel as the Horizontal Channel is created, but with a dynamically changing slope. The resulting indicator, though thought to be a new idea, is known as the Raff Channel Index [Raf91].

5.2.1 Calculation of Regression Channels

For the calculation, the least squares formula is applied to slices of the price data.

$$A^T A x = A^T b \qquad \text{[Str06, p. 181]} \tag{5.2}$$

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ l & l+1 & \cdots & n \end{bmatrix} \begin{bmatrix} 1 & l \\ 1 & l+1 \\ \vdots & \vdots \\ 1 & n \end{bmatrix} x[n] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ l & l+1 & \cdots & n \end{bmatrix} \begin{bmatrix} price[l] \\ price[l+1] \\ \vdots \\ price[n] \end{bmatrix} \tag{5.3}$$

where n is the point in time for which the channel is constructed and $l = n - M$. M is the interval for which the regression is calculated.

This operation requires 3-dimensional matrix operations, most importantly the application of the Einstein Summation to multidimensional matrices as defined in the NumPy library in the function np.einsum [num].

When decomposing the result x into its two parameters $initial\_value$ and $slope$, the regression line is defined by.

$$regression\_line[n] = initial\_value[n] + slope[n] * n \tag{5.4}$$

The method used to establish the distance where the lower and upper lines are placed is the same as for the Horizontal Channel experiment. The maximal difference between the price and the regression line is chosen.

$$max\_difference[n] = \max_{0 \leq k \leq M} |price[n-k] - regression\_line[n-k]| \tag{5.5}$$

Finally, the resistances according to the linear regression can be created for the whole time series.

$$\begin{aligned} resistance[n] &= regression\_line[n] + max\_difference[n] \\ support[n] &= regression\_line[n] - max\_difference[n] \end{aligned} \tag{5.6}$$

Again, the position of the price within the lines is defined by the formula described in the section Relative Position of the Price.

5.2.2 Example for Regression Channels

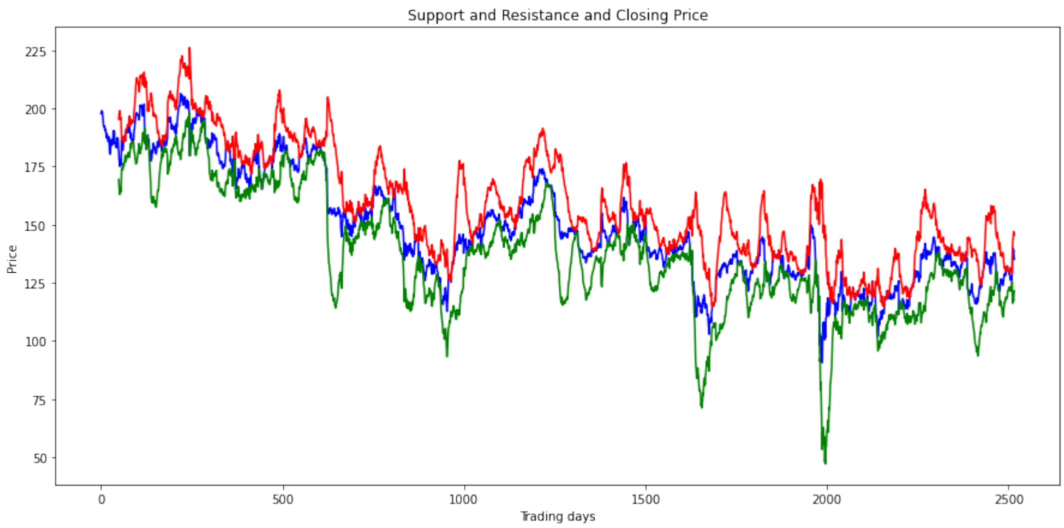


Figure 5.2: The regression support and resistance lines

The visualization of the indicator is not intuitive, because every single point is calculated from its own linear regression trend line. It moves up and down quickly depending on the slope evaluated from the previous data points and the maximal distance of previous prices from the trend.

6 Feature Transformation

As seen in the description of Technical Indicators, normalization techniques can be applied to make the comparison between multiple assets even possible. Further transformations can be used to extract details from existing indicators.

6.1 Absolute Value Transformation

An indicator that was shifted to a neutral position at the value 0 can be transformed by taking the absolute value to show how far the indicator is from the neutral value and lose the rest of the information [AJ08, p. 28].

6.1.1 Calculation of the Absolute Value Transformation

$$indicator\_absolute[n] = |indicator[n]| \tag{6.1}$$

where *indicator* is a normalized indicator centered around 0.

6.2 Threshold Transformation

Trading rules that can be found in [Wil78], [Bol02] and other textbooks define thresholds at which some state such as "overbought" or "oversold" is entered or a position in a stock should be opened or closed. To provide the possibility to verify some of these rules, a threshold transformation, also called winsorizing, could be applied [Dub20, p. 45].

6.2.1 Calculation of the Threshold Transformation

$$indicator\_threshold[n] = \begin{cases} 1 & \text{if } indicator[n] \geq threshold \\ -1 & \text{if } indicator[n] \leq -threshold \\ 0 & \text{else} \end{cases} \tag{6.2}$$

where *threshold* is a positive value which has to be surpassed in order to get any signal and *indicator* is a normalized indicator centered around 0.

6.3 Logistic Function Transformation

The threshold transformation filters out small and potentially insignificant values of the indicator by mapping all values to 1, -1, or 0, losing much of the detailed information. A squashing function can be applied to create a similar result as the threshold transformation, but without the loss.

6.3.1 Calculation of the Logistic Function Transformation

$$indicator\_logistic[n] = \frac{sign(indicator[n])}{1 + base^{-|indicator|+inflection\_point}} \quad \text{[Dub20, p. 62]} \tag{6.3}$$

where $base$ and $inflection\_point$ are constants of the sigmoid function that define the steepness and center point of the function and $sign$ maps a value to -1, 0 or 1 based on its sign.

The inflection point, which is the point where the function is the steepest, is chosen like the threshold in the section Threshold Transformation. This way, all indicator values, which do not surpass a threshold value, are moved closer to zero and values above the threshold closer to their sign (-1 or 1).

6.4 Relative Position of the Price

Indicators, such as Bollinger Bands, can be interpreted as a range or an upper and lower barrier. When analyzing another indicator or value in relation to an indicator consisting of an upper and lower boundary, it could be of relevance to quantify the position of this value within this range.

6.4.1 Calculation of the Relative Position

$$relative\_position[n] = 2 * \frac{position[n] - lower\_barrier[n]}{upper\_barrier[n] - lower\_barrier[n]} - 1 \tag{6.4}$$

As mentioned before, this function is just a modification of the b% formula [Bol02, p. 60].

6.5 Trend

In some applications, it is useful to numerically describe the relation of two indicators. A trend, for example, is believed to be positive, meaning upwards if a faster moving average is higher than a slower moving average, and negative, if the other way around.

6.5.1 Calculation of the Trend

This can be calculated by taking the sign of a difference between two moving averages:

$$ma\_trend[n] = sign(short\_moving\_average[n] - long\_moving\_average[n]) \tag{6.5}$$

6.6 Indicator Crossings

Analyzing the behavior of chart indicators plotted as lines is a major component of technical analysis. Crossings in indicator lines seem to be especially interesting for analysts, because they mark fundamental changes in the behavior of price. For example, a crossing of two moving averages is seen as a change in trend and therefore a signal to speculators to adapt their positions [Cha97, p. 17].

6.6.1 Calculation of Indicator Crossings

To calculate crossings between two lines, the trend is calculated like in the section Trend:

$$trend[n] = sign(fast\_signal[n] - slow\_signal[n]) \tag{6.6}$$

The changes in trend are then interpreted as crossings.

$$crossing[n] = \frac{trend[n] - trend[n-1]}{2} \tag{6.7}$$

This way, a move of the faster line above the slower line causes the crossing to be 1 and -1 in the other case. If there is no change in trend, the crossing value is 0. In the application, a crossing signal is not returning to 0 immediately on the next day, but gradually decreases over time. To facilitate this behavior, the crossing signal is convolved with a finite impulse response filter.

$$crossings\_convolved[n] = \sum_{k=0}^{M-1} \frac{M-k}{M} * crossing[n-k] \tag{6.8}$$

where M is the length of the filter.

7 Data Sets

The main result of the developed tool is multiple types of data sets. For one, the price downloaded from an external API is combined with the calculated indicators and stored in a more compact format. For another, data sets can be created by taking samples from price data from multiple assets.

7.1 Individual Chart Data Sets

The chart data is ultimately persisted in a table with all the indicators calculated for each time point. The first rows (corresponding to trading days) are empty, because the indicators are only valid after a specified number of price points exist. The index is the same as the number of trading days that have passed.

index	volatility10	volatility20	$\cdots$	current_price
$\vdots$	$\vdots$	$\vdots$		$\vdots$
98	4.7283568503268	4.09081827750218	$\cdots$	198.040145874023
99	4.87906677994747	4.35908599108775	$\cdots$	197.963668823242
100	4.93092401293914	4.58423118838439	$\cdots$	197.351821899414
101	4.83551348746046	4.75760206389719	$\cdots$	197.112808227539
$\vdots$	$\vdots$	$\vdots$		$\vdots$

Table 7.1: A section of downloaded and processed chart data

7.2 Sample Data Set

The sampled data set is the union of samples taken from the data from the charts of all assets or a subset of the stored charts, such as the assets traded on the main stock exchange of a country.

7.2.1 Sampling Process

In order to gather valid samples from each asset, only time points where all indicators are defined are used in the sampling process. Additionally, the length of price data is considered such that assets with a short trading history are not represented with as many samples as an asset with a long trading history.

$$number\_of\_samples[k] = \frac{samples\_per\_year}{253} * chart\_length[k] \tag{7.1}$$

where $samples\_per\_year$ is a freely chosen parameter that defines the number of samples taken from a year of data from a chart k and $chart\_length$ is the length of these daily chart data. 253 is the average number of trading days per year in the United States [tra].

To be able to perform any learning tasks, the future price and the volatility is evaluated by shifting those data columns in time and including them in the samples.

$$future\_price[n] = current\_price[n + interval] \tag{7.2}$$

where $interval$ is the time shift for creating the future price. When normalizing the future price, it is divided by the current price.

$$future\_price\_normalized[n] = \frac{future\_price[n]}{current\_price[n]} \tag{7.3}$$

7.2.2 Result of Sampling from Chart Data

The result is a random array of samples from all charts provided in a list. No metadata about the asset itself, like name, stock exchange, country etc. are included in the samples. This is done to allow a model to only base its results on numerical indicator data. The index is again the number of trading days that have passed since the first recorded price in the downloaded data. The current price is normalized by the first recorded price. They are included in the data set, but are only used for trading strategy simulations.

index	volatility10	volatility20	⋯	future_price	⋯
2387	0.012026124883559	0.026801340744153	⋯	1.12209149504059	⋯
1689	0.027831882191039	0.044977499391506	⋯	1.11824076175864	⋯
1457	0.040846902146251	0.052946291616804	⋯	0.9776555365111	⋯
2369	0.01473200188551	0.033139523170386	⋯	1.14062617431811	⋯
1648	0.059189745558863	0.098958678005976	⋯	0.852520215664347	⋯
⋮	⋮	⋮		⋮	

Table 7.2: Random normalized samples taken from various stocks in the world

7.3 Probability Distribution of Future Price

As just previously defined, the future price in the data sets is normalized by the current price to express the price change based on the price in the past. Because of this normalization and general principles of prices, the future price can not go below 0, but can, if the price increases quickly, go far beyond 2 (the price doubled during the specified time period). This asymmetry becomes visible when plotting the distribution of future prices in a histogram.

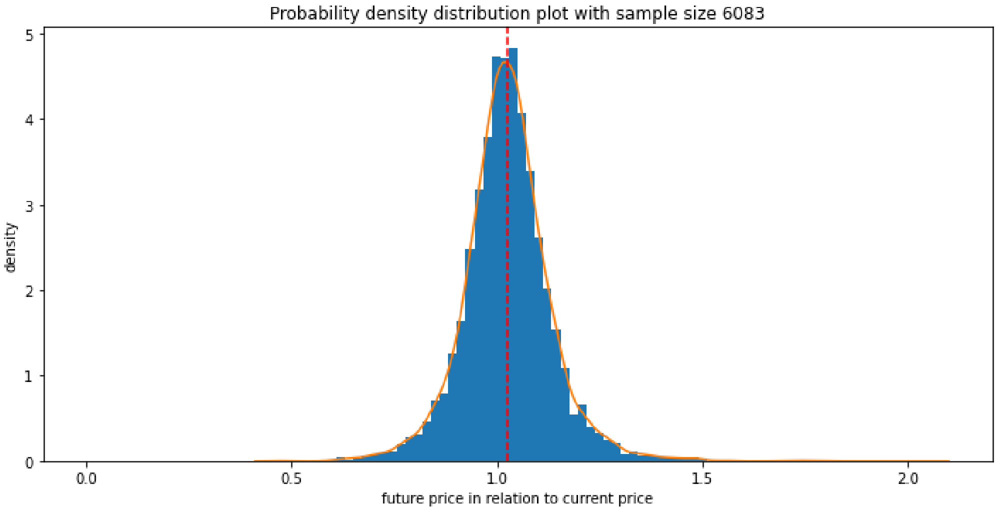


Figure 7.1: The probability distribution of the future price

The relation of prices, as calculated here, is assumed to follow a log-normal distribution [Kri06, p. 247]. This assumption is a key part of the Black-Shore-Merton model, a theoretical foundation for pricing option contracts [BS73, p. 639] [Gow08, p. 912].

When taking the natural logarithm of the future price, the asymmetry disappears, resulting in a normal distribution [ANHS16, p. 427].

7.4 Neutral Future Price Distribution

Since the performance of any trading strategy is a result of the product of changes caused by the executed trades, a completely neutral data set, which does not imply prices going up or down, should have the following property:

$$1 = \prod_{k=0}^{N-1} \frac{future\_price[k]}{current\_price[k]} \tag{7.4}$$

where k is the index in a set of random samples and N the number of samples.
By taking the logarithm and applying the product rule, it is possible to check this condition using a sum instead of a product [AJ08, p. 121].

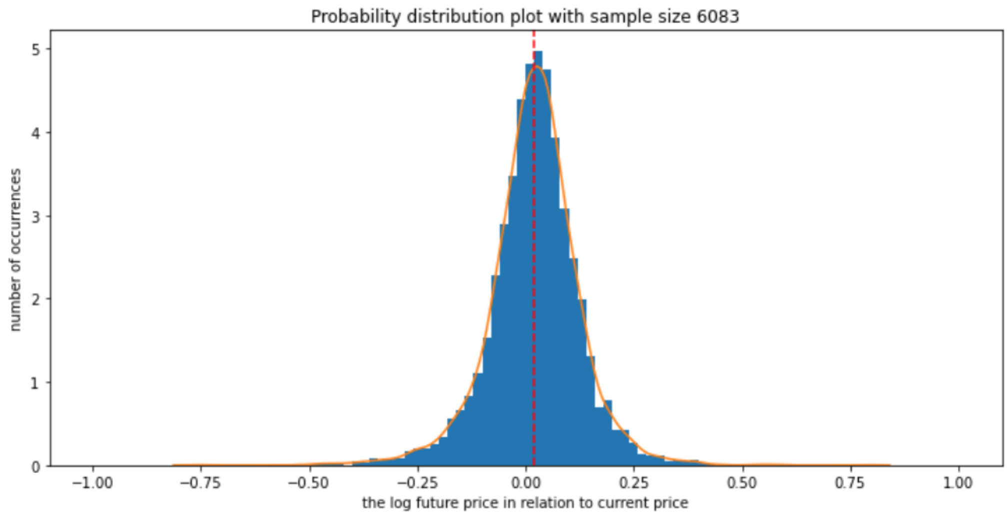
$$\log 1 = \log \prod_{k=0}^{N-1} \frac{future\_price[k]}{current\_price[k]} \tag{7.5}$$

$$0 = \sum_{k=0}^{N-1} \log \frac{future\_price[k]}{current\_price[k]} \tag{7.6}$$

Using this condition, a bias in terms of future prices that implies prices going up or down can be checked by verifying whether the sum of the logarithm of all price changes is close to zero.

7.5 Bias in Data Sets

Although the sampling process is random and large amounts of data are processed, it is practically impossible to create unbiased data sets from price charts. A bias becomes notable, when calculating the average log future price, for example of samples taken from all Nasdaq100 companies, a set of large enterprises with shares traded on the New York Stock Exchange, between 2012 and 2022.



`original number of samples: 6083, log mean: 0.020312618754336854, std: 0.10397078369075115`

Figure 7.2: The probability distribution of the log future price

7.6 Bias Reduction in Data Sets

The existence of such bias is not directly harmful to a learning method or statistical analysis, but should be taken into account when interpreting the result. Publicly traded stock, especially from companies based in the United States, have a long history of delivering positive returns for investors that explain this slight positive bias. However, this is no guarantee of a similar behavior of US stocks in the future.

7.6.1 Bias Reduction Through Shift of the Future Price

A solution for moving the mean of the distribution could be shifting all log future price values by the difference from the desired mean.

$$future\_price\_shifted[n] = \exp(\log future\_price[n] - mean(\log future\_price)) \tag{7.7}$$

where *mean* takes the arithmetic mean of all future prices.

This solution, while moving the log mean exactly to 0 resulting in a perfectly neutral data set, manipulates every single sample in the data set by artificially changing the future price.

7.6.2 Bias Reduction Through Deletion of Samples

Another approach to shift the mean of a feature is to use a subset of samples that fits the desired probability distribution function [Dub20, p. 21]. For this purpose, a rule is defined which assigns to every sample a probability of being kept in the data set.

$$\mathcal{N}(\mu_1, \sigma^2) * keep\_probability(x) = \mathcal{N}(\mu_2, \sigma^2) \tag{7.8}$$

The standard deviation is chosen to be the same for the initial and the desired normal distribution.

$$\frac{1}{\sigma * \sqrt{2 * \pi}} \exp -\frac{(x-\mu_1)^2}{2\sigma} * keep\_probability(x) = \frac{1}{\sigma * \sqrt{2 * \pi}} \exp -\frac{(x-\mu_2)^2}{2\sigma} \tag{7.9}$$

$$\exp -\frac{(x-\mu_1)^2}{2\sigma} * keep\_probability(x) = \exp -\frac{(x-\mu_2)^2}{2\sigma} \tag{7.10}$$

$$keep\_probability(x) = \exp \frac{(x-\mu_1)^2 - (x-\mu_2)^2}{2\sigma} \tag{7.11}$$

Note that the original distribution is log normal. Because of this, the logarithm is taken before taking out samples in order for the resulting distribution of future prices to be unchanged except for the shifted mean.

7.6.3 Decreased Bias in Larger Data Sets

The first two methods described are both not ideal, because they significantly manipulate the data set and destroy the consistency of the samples within the data set. Since financial data is abundant and relatively inexpensive to source, it is a sensible decision to gather more price data from a more representative set of assets in the world. In this way, the bias in the future price is likely to decrease.

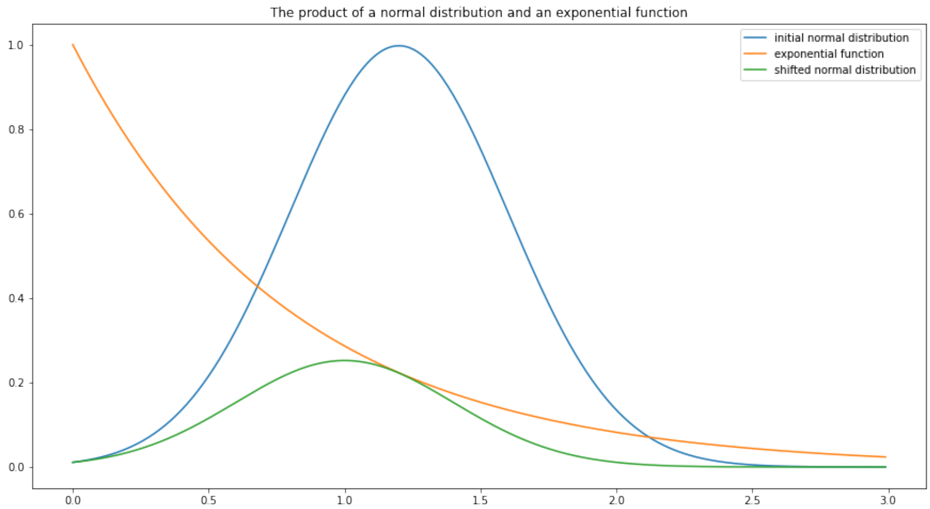
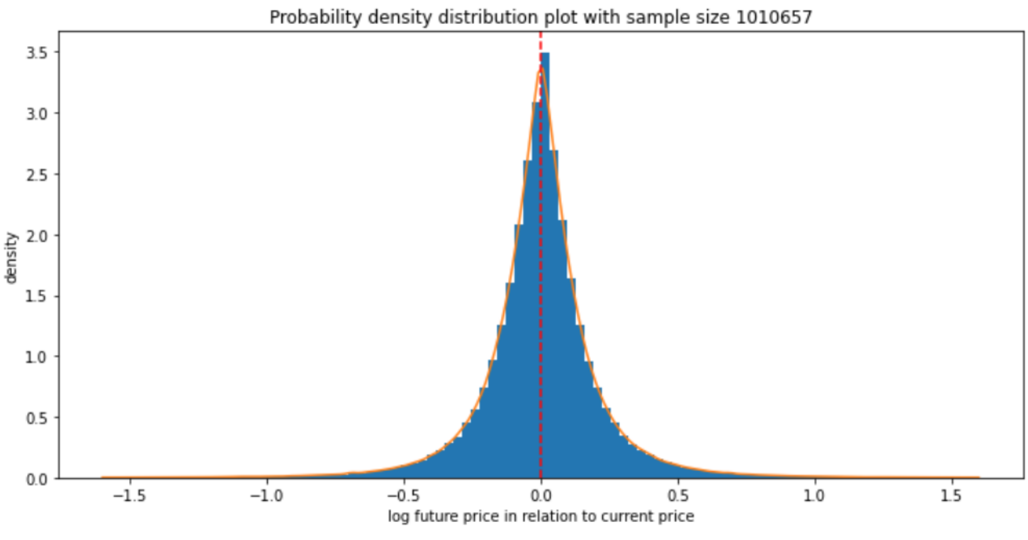


Figure 7.3: A visualization of the systematic deletion of samples




```
number of samples: 1010657, average: 0.00021682861026033756, std: 0.24134662350471184
```

Figure 7.4: The probability distribution of a large and unbiased data set

8 Simulations

The simulation of trading rules based on price movements in the past is a way to better analyze the decisions generated by the trading strategy. The goal is to create an environment that is as close as possible to reality and, therefore, allows evaluating the performance a strategy would have had during certain time periods.

8.1 Simulation Algorithm

Data: $stock\_data$, $trading\_rule$, $rebalance\_interval \geq 1$, $trading\_cost \geq 0$, $leverage \geq 0$, $weights$
Result: $total, volume, leverage$
$price\_changes \leftarrow get\_price\_changes(stock\_data)$;
$positions \leftarrow trading\_rule(stock\_data) * weights$;
$cash \leftarrow \{1, 0, 0, 0, ...\}$;
$portfolio[0] \leftarrow \{0, 0, 0, 0, ...\}$;
$total \leftarrow \{1, 0, 0, 0, ...\}$;
$volume \leftarrow \{0, 0, 0, 0, ...\}$;
$n \leftarrow 1$;
while $1 \leq n \leq length(stock\_data)$ **do**
  $portfolio[n] \leftarrow portfolio[n-1] * (1 + leverage * price\_changes[n])$;
  $cash[n] \leftarrow cash[n-1]$;
  $total[n] \leftarrow sum(portfolio[n]) + cash[n]$;
  if $n \mod rebalance\_interval = 0$ **then**
    $portfolio[n] \leftarrow total[n] * positions[n]$;
    $cash[n] \leftarrow total[n] - sum(portfolio[n])$;
    $volume[n] \leftarrow sum(abs(portfolio[n-1] - portfolio[n]))$;
    $cash[n] \leftarrow cash[n] - volume[n] * trading\_cost$;
    $total[n] \leftarrow sum(portfolio[n]) + cash[n]$;
  end
  $n \leftarrow n + 1$;
end
$volume \leftarrow sum(volume)$;
$leverage \leftarrow mean(positions) * leverage$;

Algorithm 1: The performance simulation for a trading rule

8.2 Experimental Trading Rules

A trading rule is a set of conditions from which actions, such as buying or selling, are derived. Conditions can be arbitrarily defined, also just in the form of sentences, but need to be transformed so that an algorithm can interpret them. Trading engines which try to mimic such strategies require the implementation of fuzzy logic and are hard to validate and understand. To reduce the complexity of the analysis, a trading rule in the context of this thesis is a function that deterministically assigns a theoretical position in a stock based on the chart indicators of the stock at a certain point in time. A position of 1 means that all capital is invested in this one stock and a position of zero means that no capital is invested.

The rules shown are inspired by ideas found in books about technical indicators, but simplified to allow automatic calculation of positions. They are not advertised as profitable strategies, but should be seen as examples demonstrating the mechanics of simple strategies created by traders.

$$position(indicators) = x \tag{8.1}$$

where *position* is the trading rule function, assigning the desired position x based on a number of *indicators*, holding information about specific technical indicators.

Here, some examples for the constructions of trading rules are given. Note, that short selling is explicitly avoided in all the trading strategies by applying a *max* function that does not allow the position to go below 0.

8.2.1 Buy and Hold Trading Rule

The reference trading rule is the buy and hold strategy. It is the simplest existing trading rule and is not based on any technical indicator, because it always assigns the value 1 to the position.

$$position(indicators) = 1 \tag{8.2}$$

8.2.2 Trend Trading Rule

The trend follower is a direct application of the feature transformation defined in the section Trend. The rule tries to be involved only in a stock when it is trending upwards.

$$position(ma\_trend) = max(ma\_trend, 0) \tag{8.3}$$

where ma_trend shows whether a stock is trending upwards (1) or downwards (-1).

8.2.3 Relative Strength Trading Rule

The relative strength strategy tries to enter a position only if the RSI signals that the stock is undervalued. This happens when the threshold of 30 is breached.

$$position(rsi\_threshold) = max(-rsi\_threshold, 0) \tag{8.4}$$

where $rsi\_threshold$ shows whether the RSI indicator is above 70 (1) or below 30 (-1).

8.2.4 Price Position Trading Rule

The first described trading rules are binary rules that assign values 0 or 1 to all positions. The following rule tries to gradually increase and decrease the position in a stock, based on the position of the price within the Bollinger Bands, regression lines or supports and resistances.

$$position(price\_position) = clip(1 - price\_position, 0, 1) \tag{8.5}$$

where $price\_position$ is the position of the price between an upper and lower value like its described in the section Relative Position of the Price and $clip$ ensures that the resulting position to be between 0 and 1.

8.2.5 Exponential Moving Average Trading Rule

This strategy tries to profit from reversions to means that are, in this case, expected to occur when the price is below the exponential moving average.

$$position(ema) = max(sign(ema), 0) \tag{8.6}$$

where ema is the normalized exponential moving average indicator.

8.2.6 Drawdown Reduction Trading Rule

A risk averse strategy could be built using the position of the price within a range. The position is reduced, if the price moves away from the middle of a range, limiting the exposure of the strategy to rapid price movements.

$$position(price\_position) = max(1 - abs(price\_position), 0) \tag{8.7}$$

where $price\_position$ is the position of the price between an upper and lower value like its described in the section Relative Position of the Price.

8.2.7 Classification Models

With the availability of large amounts of price data and normalized indicators, it is possible to train machine learning models such as neural networks and support vector machines. A trained classifier itself is a trading rule that directly assigns a position in a stock based on a number of given indicators.

$$position(indicators) = clip(classifier(indicators), 0, 1) \tag{8.8}$$

8.3 Portfolio Weights in Simulations

The implemented algorithm can deal with weighted portfolios. It multiplies all desired positions (usually between 0 and 1) by weights, resulting in larger simulated trades for certain stocks and smaller trades for other stocks. The weighting of individual components of a portfolio is an important topic in finance and especially fund management, but adds a bias to the portfolio. In the simulation shown here, all strategies are tested with equal weighted portfolios. Every stock has therefore the same potential to receive a portion of the theoretical capital from the portfolio.

8.4 Leverage Parameter in Simulations

The leverage is the sum of all open positions or trades at a given time compared to available funds. Therefore, a leverage of 1 means that all available funds are invested and a leverage value of 0 means that no funds are invested. This definition is slightly different from the common description of the term in finance, which is the ratio between borrowed and own capital on a balance sheet [BH09, p- 93]. Some strategies take advantage of very little of the available capital and therefore have small leverage. To allow comparison of different strategies with some reference, such as the buy and hold strategy, the leverage can be artificially increased with a parameter in the simulation.

8.5 Simulated Trading Volume

During the simulation, many theoretical trades are initiated. This is done explicitly by changing the position in the components of the portfolio. The sum of all the position changes is the volume of trades caused by the strategy. It is relevant to take this into consideration, because trading volume is in reality punished with costs.

8.6 Trading Costs Parameter in Simulations

Financial trading is costly and cost plays a key role in the development of any trading engine. Costs are incurred in different ways, making it difficult to simulate them. Instead of combinations of fixed fees per transactions and variable fees based on transaction volumes, the simulation is only incorporating the variable part. All volume caused by rebalancing and trading of the portfolio is punished with a cost factor [BVF+21].

Realistic Trading Commissions on Exchanges

As a source for realistic costs, the commissions charged by a major provider of a platform usable for trading bots are used [int]. It can be seen that the total transaction fees can be expected to lie between 0% and 1% of the volume. The fee per traded share of companies based in the USA that is incurred by the platform, ranges from 0.005 to 0.0005 USD and strongly favors active traders with large volume.

8.7 Rebalance Interval Parameter

Considering the trading costs, especially fixed trading costs, it is not always realistic to rebalance a portfolio on a daily basis. With the rebalance interval parameter, a custom interval can be chosen. An interval of 5 days would, for example, in reality represent a weekly rebalancing of the portfolio.

9 Results

The result of the simulation is a time series data that reflects the development of the portfolio value for a certain trading strategy and time period. It is by no means a guarantee for similar performance when applied to the future. When plotted together with a reference strategy like the buy-and-hold strategy, portfolio changes during certain time periods like market downturns or growth phases can be analyzed, giving more intuitive information about the usefulness of the trading strategy.

9.1 Observed Leverage in Simulations

All trading strategies, as defined in the chapter Experimental Trading Rules have a leverage below 1 when applied to stock price data. Low leverage is an advantage because it means that the strategy does not use all available funds and is stricter when choosing positions. It opens the possibility to increase the positions taken or to apply multiple trading strategies in parallel without the need for more funds. For fair comparisons with the buy-and-hold strategy, leverage is artificially adjusted in some examples.

9.2 High Simulated Volume

It is notable for many trading strategies that they require the execution of many trades and as a result cause a lot of trade volume over time. Extreme examples are strategies based on indicators with short calculation intervals. The indicator values change frequently, and so do the decisions made by the strategy.

9.3 Theoretical Performance in Simulations

The ideal performance for a trading strategy is achieved when the trading engine can decide at every time step how large the position in a stock is and trades are not associated with any costs. The results have to be viewed with caution but show whether certain strategies were successfully choosing better positions than a reference strategy.

9.3.1 Performance of the Trend Trading Rule

The trend strategy often manages to perform similarly to the buy and hold strategy in terms of the total return, just with less required capital. This is the case because it sells positions once they move downward but then buys them back once they have moved up

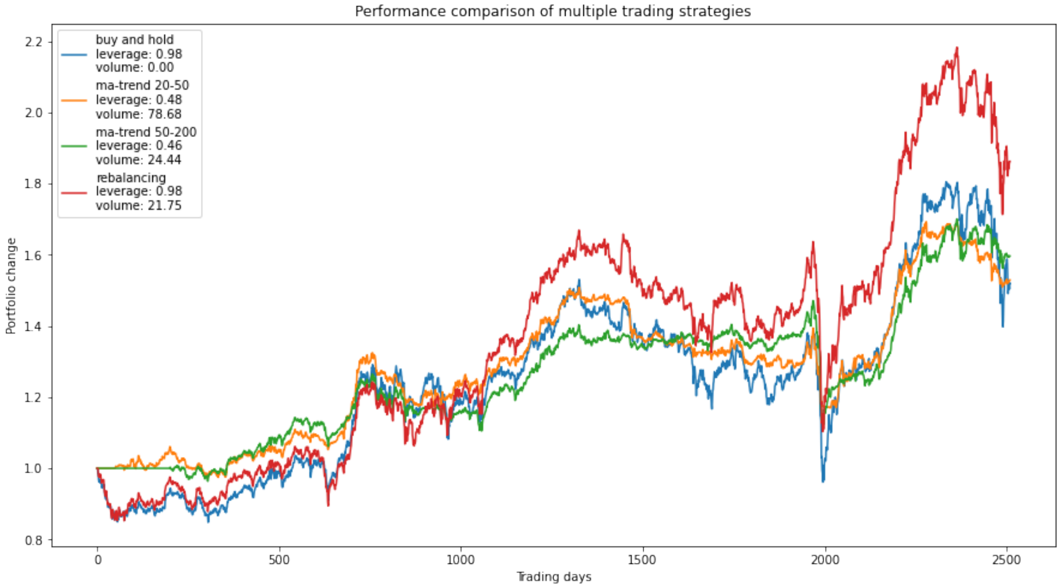


Figure 9.1: Trend strategies applied to a set of Finnish stocks

again. The strategy certainly is not consistent, because prices trend in different ways that often cannot be captured by one measure of trend.

9.3.2 Performance of the Relative Strength Trading Rule

On its own, the trading rules that use the relative strength indicator perform poorly. As defined in the context of this thesis, they buy stocks that have moved down quickly and are highly volatile. Interestingly, in the Austrian stock market, an anomaly can be observed in which the strategy based on the relative strength indicator with an interval of 14 days managed to consistently select undervalued stocks for 8 years (between 2012 and 2020).

9.3.3 Performance of the Price Position Trading Rule

The experimental technical indicators, which were developed using concepts from Bollinger Bands, support and resistances and regression, yielded in theory good results in many markets. Especially during time periods with little price movements, they manage to create better returns than the buy and hold strategy. A large downside is the bad performance during volatile events. As a falling price causes an increase of the position taken, the strategies aim to hold particularly large positions in crashing stocks. The opposite is true for rapidly rising stocks, where the strategies are reducing positions rapidly.

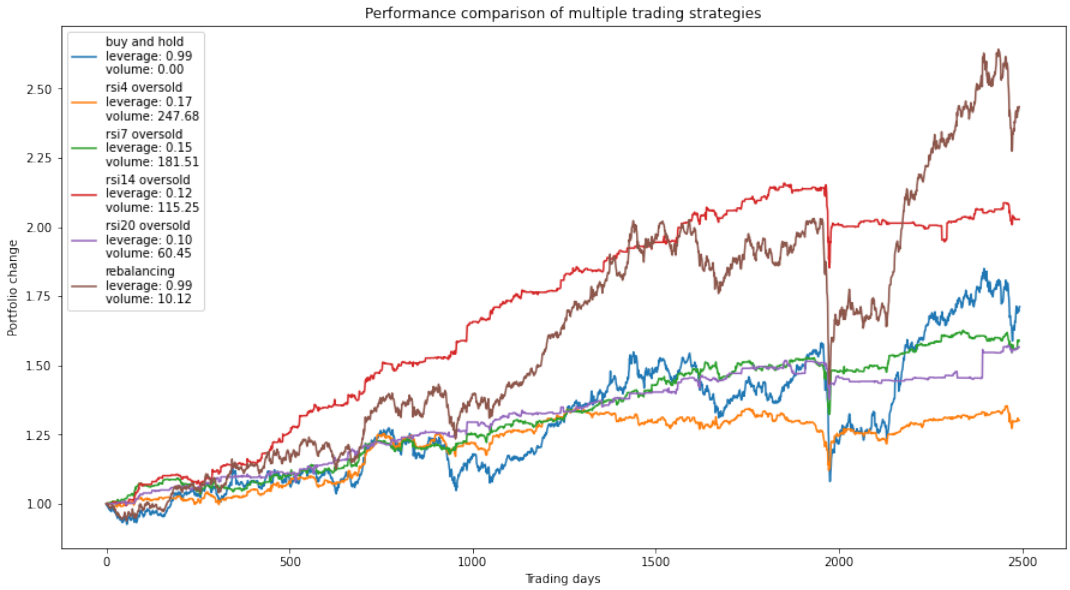


Figure 9.2: RSI threshold strategies applied to a set of Austrian stocks

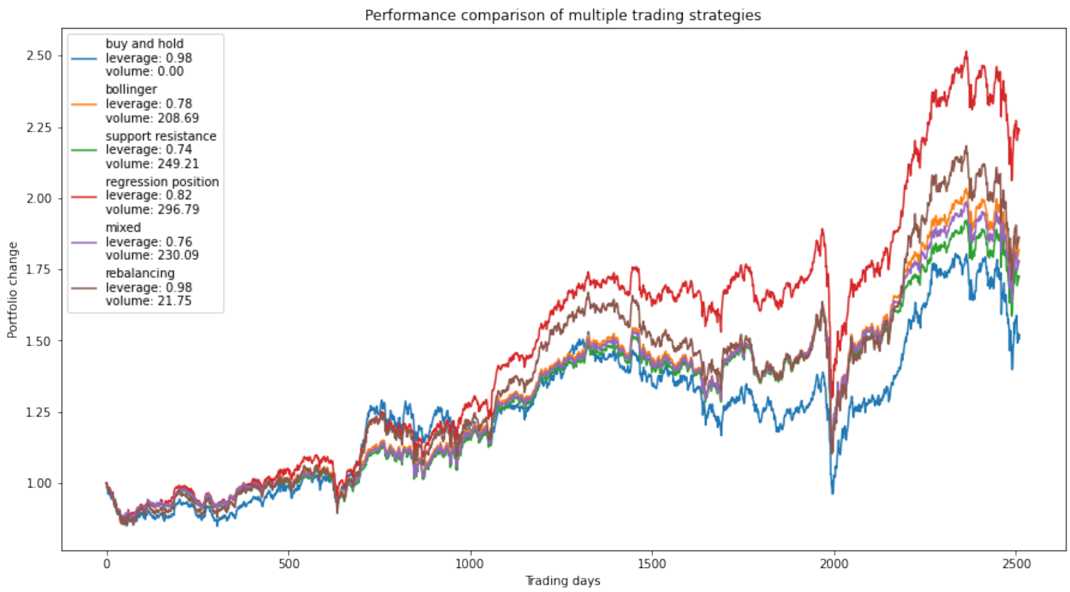


Figure 9.3: Price position strategies applied to Finnish stocks

9.3.4 Performance of the Exponential Moving Average Trading Rule

Since the exponential moving average strategy is very similar to the strategies developed around the price position, the results are similar. Notable is the exceedingly large total transaction volume caused by this strategy. The reason is the quick movements below and above the price that results in frequent buy and sell signals.

9.3.5 Performance of the Drawdown Reduction Trading Rule

Strategies which hold larger positions of stocks that are staying in certain price ranges like Bollinger bands manage to have less extreme movements in quick market downturns. They showcase the potential of using technical indicators to select stocks from a basket that fit the strategy of an individual, in this case, an investor trying to avoid risk.

9.3.6 Performance of Classifiers

Neural networks and support vector machines perform exceptionally well in the simulations. The main reason for this is the fact that they are fitted to all downloaded data and make fairly good decisions when classifying samples drawn from the same data set again. Although this is the case, the rudimentary classifiers developed here are not able to predict strong market downturns. An example is added for completeness, but would need to be validated using completely unrelated price data, preferably from different time periods.

9.4 Limitations of the Theoretical Strategy Performance

As mentioned previously, the theoretical performance can not be achieved in real trading environments. The main limitations are costs involved in trading and the practical challenges of rebalancing portfolios frequently.

9.4.1 Simulation of Transaction Costs

The main limitation for all the trading performances shown previously is the transaction costs on exchanges. Especially for individuals with small amounts of available funds, they represent a large portion of every trade. For almost any situation and trading strategy, all excess profits are lost when the transaction cost factor is moved from 0% to 0.5%.

9.4.2 Simulation of Less Frequent Rebalancing

Choosing larger rebalancing intervals also significantly reduces the success of most of the strategies presented. They are not designed to generate long-term trading decisions and therefore perform poorly when not rebalanced daily.

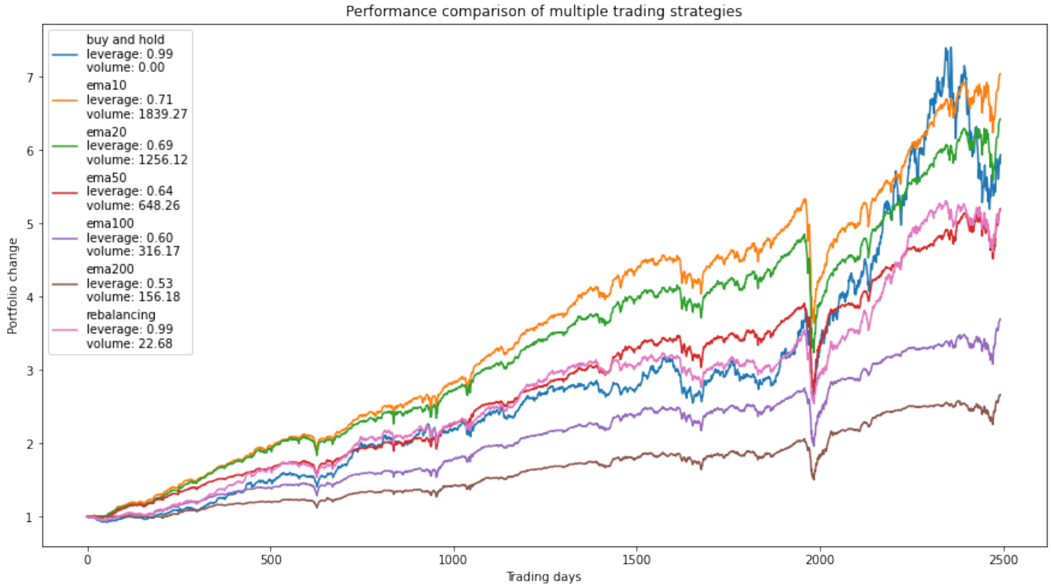


Figure 9.4: EMA strategies applied to Danish stocks

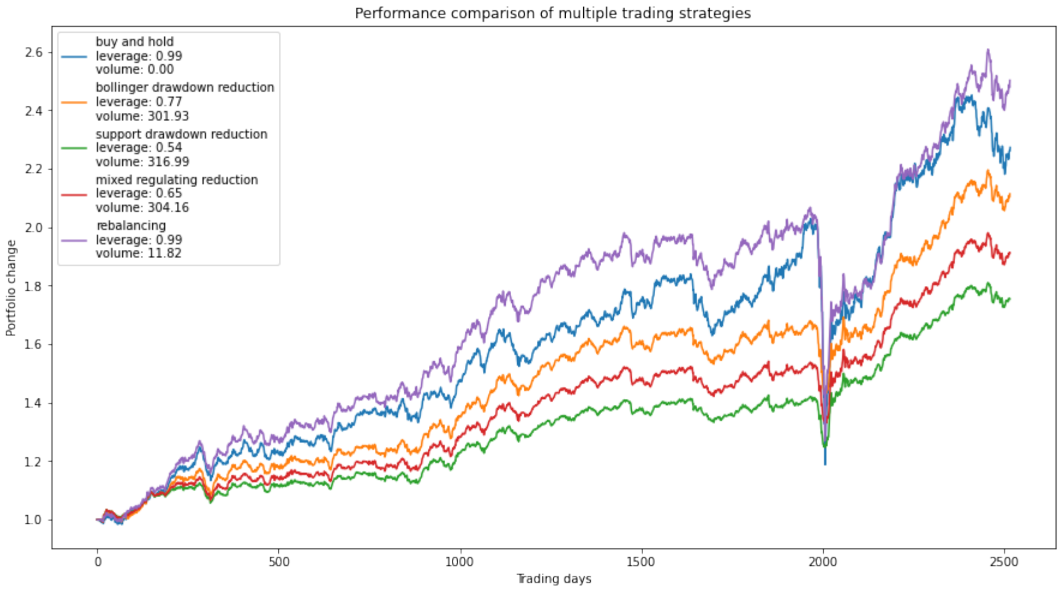


Figure 9.5: Drawdown reduction strategies applied to New Zealand stocks

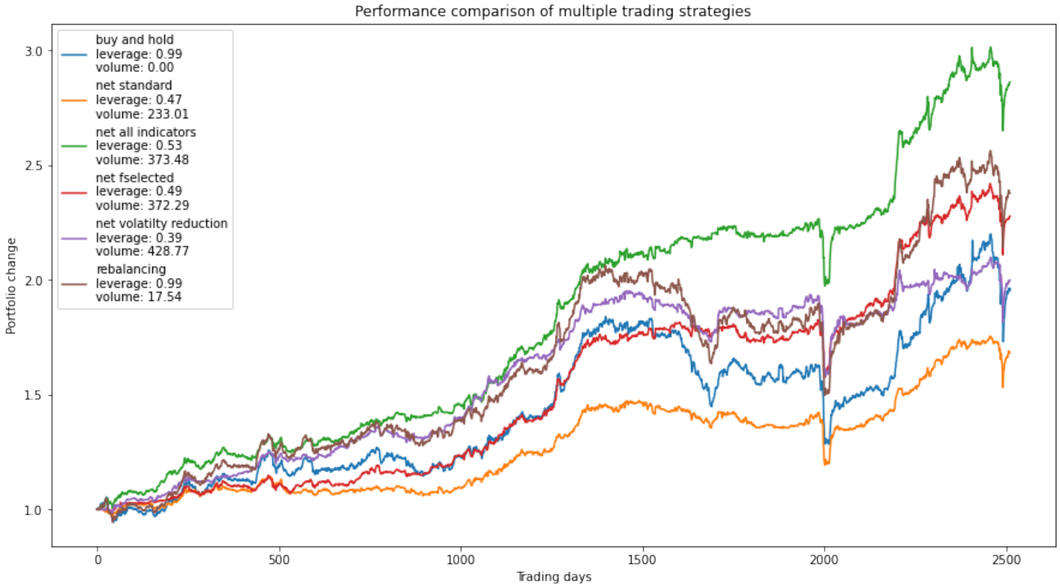


Figure 9.6: Neural net classifiers applied to Lithuanian stocks

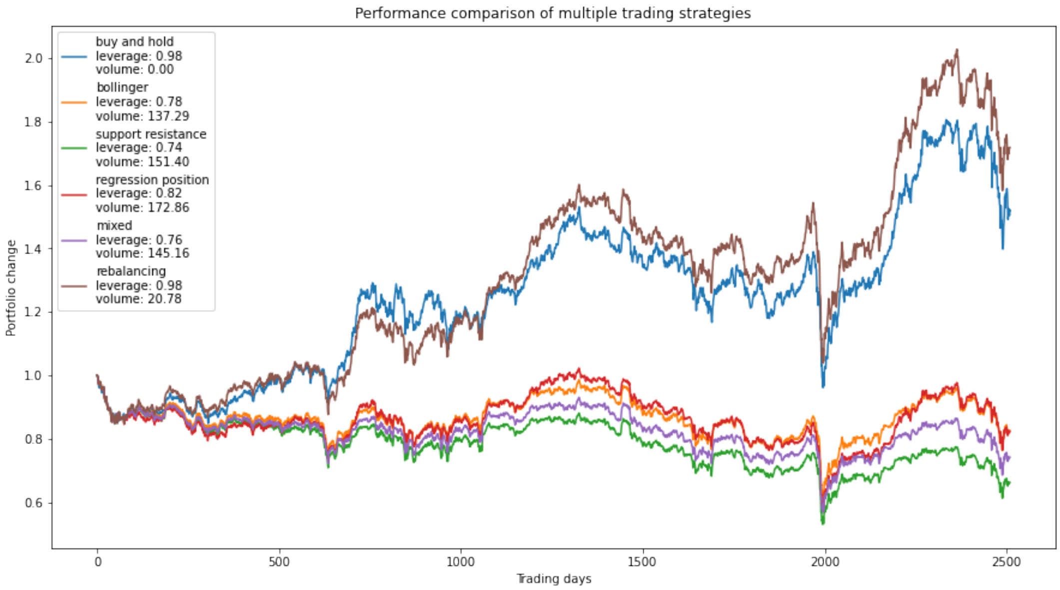


Figure 9.7: Price position strategies with a transaction cost factor of 0.5%

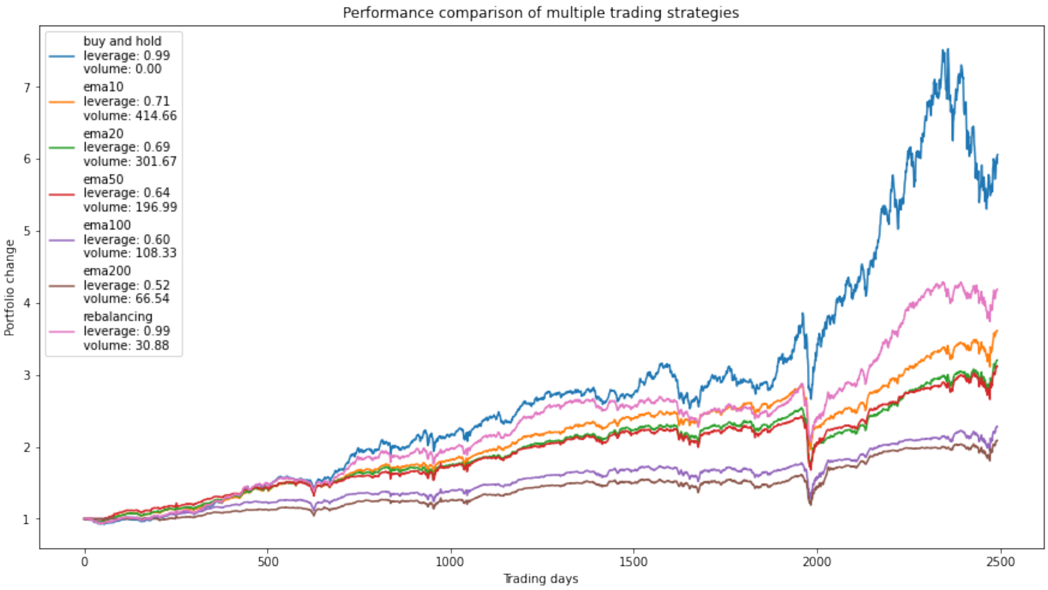


Figure 9.8: EMA strategies with less frequent rebalancing

10 Reflection

This thesis dealt with topics in the field of finance, with much research already being done. The distinction from other similar work is its strong focus on the algorithms needed for creating usable data for various applications dealing with financial data. It is a summary of all main process indicator calculations that can be applied to large sets of price data.

10.1 Reflection of the Outcomes

Many parts of the developed software could be used in further work by transforming it into more specialized software. This could be done by adding components enabling real-time trading, dynamic visualizations, user interactions, etc.

10.1.1 Reflection of Used Indicator Formulas and Algorithms

All algorithms and formulas defined in the chapter Technical Indicators are implemented int the developed application and can be applied to price data. Here, the purpose was the creation of data sets and performing data analysis, but the same algorithms are needed for real-time trading applications based on technical indicators.

10.1.2 Reflection of Implemented Data Curation Methods

A particularly strong emphasis was put on data curation. All indicators are normalized such that they can directly be used for training machine learning models or perform data analysis in general. The resulting data sets are cleaned up to be free of data inconsistencies and, therefore, can be loaded and processed with almost no precautions.

10.1.3 Reflection of the Trading Simulation

Similar research deals mainly with statistical tests and other theoretical methods to validate the hypotheses. In this work, a completely different approach was chosen for demonstrating the results. Trading rules are simulated and results are visualized with the goal of better comprehensibility. It is not usable for the validation of any observations, but it tries to give valuable insights into the underlying processes and mechanics of trading engines.

10.2 Improvements

With the positive outcomes in mind, there is still room for significant improvements that should be taken into account when doing similar work in the future.

10.2.1 Improvements for the Computational Performance

The algorithms for calculating the indicators are implemented without any performance optimizations. Basic adaptations such as the incorporation of multiprocessing and just-in-time compilation would significantly speed up the execution of mathematical operations involved in the formulas.

10.2.2 Improvements for Further Supported Data Sources

The tool is very static in terms of supported data sources. For real use, it should allow the input of price data from different data sources, such as files and multiple APIs. It is also not capable of continuously updating stored data sets by adding real-time price data. This feature is essential for trading engines and any reporting services based on technical indicators.

10.2.3 Improvements for the User Interface

The only interface available for the implemented data set tool is a command line interface. It was satisfactory for personal use during work, but should be exchanged with a modern visual interface like a website in a more sophisticated application.

10.2.4 Improvements for the Application of Statistical Methods

The main limitation of this work is the lack of statistical analysis of the generated results and the lack of the appropriate statistical methods necessary to design usable classification models, robust trading rules, etc. Furthermore, no validation was performed for the trained neural network and support vector machines. Because of this, no clear statement can be made about the performed experiments or the expressiveness of any trading rule used in the simulations.

10.3 Future Work

The biggest value for future work could be created by providing implementations to the public. The resulting projects would be helpful in promoting financial education in the field of personal finance. Existing applications and services charge large fees or are not available to the public at all.

10.3.1 Publication of Data Sets

Numerous data sets were created during the thesis work. They could be provided online to allow other researchers to mitigate the costs involved in working with financial data. The main problem is the need to reload them on a regular basis to keep them up-to-date.

10.3.2 Portfolio Structuring Service

Another valuable contribution to the public would be a portfolio management service that allows people to create personalized portfolios based on their personal interests, goals, and tolerance to risk. In essence, it would provide information usually hidden by robo-advisors and help persons with little available funds to achieve comparable results without the need of paying for such a service. Such services could incorporate trading ideas based on technical indicators, but would run into legal problems as soon as users lose money applying them.

10.3.3 Technical Indicator Package

The implemented indicator algorithms are already a collection of functionalities that could be optimized for performance, made more dynamic, and packaged into a package for other developers to use. Only some similar packages exist and are hard to use and incorporate into custom applications.

11 Conclusion

Although the work mainly deals with basic concepts of trading and technical analysis, the simulations ultimately give insights on modern financial markets. There exists a large discrepancy between the theoretical potential of simple trading strategies and their application in realistic environments. Three major components crucial for success of a trading engine, but also a humans trading financial assets are: High-quality data, sound trading strategies, and inexpensive access to markets. They seem to be linked in different ways to the wealth of an individual, and therefore could increase inequality among market participants.

11.1 Access to Financial Data

In various fields of science, data is widely available and free. This is not the case for financial data. Depending on the quality of the services, they incur fees or require payments for subscriptions. Basic services that provide raw price data are relatively inexpensive, but stock screeners and other advanced tools are not [gur].

11.2 Implications of Simulations on the Real Use of Trading Strategies

Institutions and corporations have the resources to develop more sophisticated and robust trading strategies. The examples given in previous chapters illustrate how simple strategies could work, but do not come close to trading engines applied in today's markets.

11.3 Operation of Trading Engines

The operation costs of a trading engine, but also the costs associated with investments, significantly impact the success of the individual.

11.3.1 Inequality Cause by Fixed Trading Costs

Fixed costs such as fixed transaction costs and other fees particularly hurt low-net-worth investors. The simulations showed that even profitable strategies were no longer profitable, once transaction fees of only 0.5% are applied. For an individual wanting to apply such an automated strategy and trade 100 USD repeatedly, the total fee per trade is required to be significantly less than 0.50 USD. This is in almost any case unrealistic, leading to the individual losing money even when applying a good strategy.

Operational costs of deploying a trading bot are also fixed and reduce the profitability of the strategy. For individuals with a large total portfolio value, these costs are insignificant.

11.3.2 Discrepancies in Incurred Variable Trading Fees

Additionally, variable costs associated with trading decrease significantly with the volume traded, as described in the section Realistic Trading Commissions on Exchanges In this particular example, the cheapest option which requires huge amount of trading volume results in fees being only 1/7 of the fees incurred for almost any normal investor.

12 Summary

Having analyzed the basic mechanics of trading algorithms and common technical indicators, some implications can be derived for regular market participants. However, it is important to note that almost anything learned in this work is irrelevant to the majority of people who invest in stock markets. This is the case because only a fraction of households actively trade financial assets and even less base their decisions on technical indicators.

Technical indicators are a source of information that can help to make objective investment decisions and detect certain price behaviors, like deviations from means, trending movements, etc. The normalization of indicators significantly reduces the effort for applying them in data science or machine learning related tasks and trading engines.

Although some strategies based on technical indicators yielded impressive results in the simulations, the validation of the results is challenging and requires unrelated data sets. This is particularly the case for machine learning models trained on historical indicator data.

The simulations showed that it is hard to build effective automated trading strategies and even more demanding to apply them in real environments. The main reasons for this are transaction costs, inconsistent strategies only successful under certain market conditions and difficulties regarding the operation of frequent and automated trade execution. Because of this, it is unlikely that a trader is successful in building a new trading bot with the methods applied in this work, even if the underlying strategy used is profitable in theory.

Participants in modern markets should be aware of the existence of algorithmic trading engines. They represent a very sophisticated opponent that takes advantage of anomalies in markets when detected. Automated trading engines are widely applied in practice, but benefit primarily wealthy investors and institutions who manage to overcome the difficulties associated with developing and operating them.

Bibliography

[AD07] Gerald Appel and Edward Dobson. *Understanding MACD*. Traders Press, 2007.

[AJ08] Hui-Hui Dai Alan Jeffrey. *Handbook of Mathematical Formulas and Integrals*. Elsevier, fourth edition, 2008.

[Ale61] Sidney S. Alexander. Price movements in speculative markets: Trends or random walks. *Industrial Management Review (pre-1986)*, 2(2):7, 05 1961. Copyright - Copyright Sloan Management Review Association, Alfred P. Sloan School of Management May 1961; Last updated - 2021-09-12.

[ANHS16] Andrés Arévalo, Jaime Niño, German Hernández, and Javier Sandoval. High-frequency trading strategy based on deep neural networks. In De-Shuang Huang, Kyungsook Han, and Abir Hussain, editors, *Intelligent Computing Methodologies*, pages 424–436, Cham, 2016. Springer International Publishing.

[AVO98] Ronald W. Schafer with John R. Buck Alan V. Oppenhein. *Discrete-Time Signal Processing*. Prentice-Hall, second edition, 1998.

[AWL10] Jasmina Hasanhodzic Andrew W. Lo. *The Evolution of Technical Analysis*. John Wiley & Sons, 2010.

[Bac00] Louis Bachelier. Théorie de la spéculation. *Annales scientifiques de l'Ecole Normale Supérieure*, 3(17):21–86, 1900.

[BB17] Peter Bruce and Andrew Bruce. *Practical Statistics for Data Scientists*. O'Reilley Media, first edition, 2017.

[BH09] Eugene Brigham and Joel Houston. *Fundamentals of financial management*. Cengage Learning, twelth edition, 2009.

[BHJ12] Utpal Bhattacharya, Craig W. Holden, and Stacey Jacobsen. Penny wise, dollar foolish: Buy-sell imbalances on and around round numbers. *Management Science*, 58(2):413–431, 2012.

[BLL92] William Brock, Josef Lakonishok, and Blake LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of Finance*, 47(5):1731–1764, 1992.

[Bol02] John Bollinger. *Bollinger on Bollinger Bands*. McGraw-Hill, 2002.

[BOS12] Jennifer C. Bender, Carol L. Osler, and David Simon. Noise Trading and Illusory Correlations in US Equity Markets*. *Review of Finance*, 17(2):625–652, 02 2012.

[BS73] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, May 1973.

[BT85] Werner F. M. De Bondt and Richard Thaler. Does the stock market overreact? *The Journal of Finance*, 40(3):793–805, 1985.

[BVF+21] Patrizia Beraldi, Antonio Violi, Massimiliano Ferrara, Claudio Ciancio, and Bruno Antonio Pansera. Dealing with complex transaction costs in portfolio management. *Annals of Operations Research*, 299(1):7–22, Apr 2021.

[CH18] Yingjun Chen and Yijie Hao. Integrating principle component analysis and weighted support vector machine for stock trading signals prediction. *Neurocomputing*, 321:381–402, 2018.

[Cha97] Tushar S. Chande. *Beyond Technical Analysis: How to Develop and Implement a Winning System*. John Wiley & Sons, 1997.

[CLXH20] Yanjun Chen, Kun Liu, Yuantao Xie, and Mingyu Hu. Financial trading strategy system based on machine learning. *Mathematical Problems in Engineering*, 2020:3589198, Jul 2020.

[Cor20] Bradford Cornell. Medallion fund: The ultimate counterexample? *The Journal of Portfolio Management*, 46(4):156–159, 2020.

[Dub20] Pablo Duboue. *The Art of Feature Engineering*. Camebridge University Press, first edition, 2020.

[Fam65] Eugene F. Fama. The behaviour of stock-market prices. *The Journal of Business*, 38(1):34–105, 1965.

[Fam70] Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417, 1970.

[Gow08] Timothy Gowers. *The Princeton Compation to Mathematics*. Princeton University Press, 2008.

[gur] Guru focus platform. `https://www.gurufocus.com/`. Accessed: 2022-06-19.

[Hea21] Sir Thomas Heath. *A History of Greek Mathematics*. Clarendon Press, 1921.

[HW12] Xiao-feng Hui and Ya-jun Wu. Research on simple moving average trading system based on svm. In *2012 International Conference on Management Science Engineering 19th Annual Conference Proceedings*, pages 1393–1397, 2012.

[int] Iteractive brokers pro fees. `https://www.interactivebrokers.com/en/pricing/commissions-stocks.php/`. Accessed: 2022-06-19.

[Ken17] James C. Kennedy. *A Concise History of the Netherlands*. Cambridge Concise Histories. Cambridge University Press, 2017.

[KH53] M. G. Kendall and A. Bradford Hill. The analysis of economic time-series-part i: Prices. *Journal of the Royal Statistical Society. Series A (General)*, 116(1):11–34, 1953.

[Ko18] Chiu Yu Ko. *Applied Financial Economics*. Chiu Yu Ko, 2018.

[Kri06] Kalimuthu Krishnamoorthy. *Handbook of Statistical Distributions with Applications*. Chapman & Hall, 2006.

[LM87] Andrew Lo and A. Craig MacKinlay. Stock market prices do not follow random walks: Evidence from a simple specification test. Technical report, February 1987.

[LR20] Bryan Zheng Zhang Jesse McWaters Lukas Ryll, Mary Emma Barton. Transforming paradigms – a global ai in financial services survey. Technical report, World Economic Forum, January 2020.

[May11] D. May. Theory of speculation. `https://www.investmenttheory.org/uploads/3/4/8/2/34825752/emhbachelier.pdf`, 2011. Accessed: 2022-05-22.

[MP99] Dimitrios Malliaropulos and Richard Priestley. Mean reversion in southeast asian stock markets. *Journal of Empirical Finance*, 6(4):355–384, 1999.

[Mur99] John. J. Murphy. *Technical Analysis of the Financial Markets*. New York Institute of Finance, 1999.

[Nat94] Sheldon Natenberg. *Option Volatility & Pricing*. McGraw-Hill, first edition, 1994.

[num] Numpy documentation. `https://numpy.org/doc/stable/reference/index.html`. Accessed: 2022-05-22.

[PM16] Pravin Srinath Praktik Mulay, Nishant Poojary. Automated trading system - a survey. *International Research Journal of Engineering and Technology*, 03(08):1572–1578, 2016.

[PS88] James M. Poterba and Lawrence H. Summers. Mean reversion in stock prices: Evidence and implications. *Journal of Financial Economics*, 22(1):27–59, 1988.

[PVPnHGP+15] M. Pulido-Velazquez, S. Peña Haro, A. García-Prats, A. F. Mocholi-Almudever, L. Henriquez-Dole, H. Macian-Sorribes, and A. Lopez-Nicolas. Integrated assessment of the impact of climate and land use changes on groundwater quantity and quality in the mancha oriental system (spain). *Hydrology and Earth System Sciences*, 19(4):1677–1693, 2015.

[Raf91] Gilbert Raff. Trading the regression channel. *Stocks & Commodities*, 9(10):403–408, 1991.

[Sha94] William F. Sharpe. *The Sharpe Ratio*. The Journal of Portfolio Management, 1994.

[Sha08] Brian Shannon. *Technical Analysis Using Multiple Timeframes*. LifeVest Publishing, 2008.

[STDBR14] Mininder Sethi, Philip Treleaven, and Sebastian Del Bano Rollin. Beating the s amp;p 500 index — a successful neural network approach. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 3074–3077, 2014.

[Str06] Gilbert Strang. *Linear algebra and its applications*. Thomson, Brooks/-Cole, Belmont, CA, fourth edition, 2006.

[TH11] Ryan Riordan Terrence Hendershott. Algorithmic trading and information *, June 2011.

[tic] Yahoo finance ticker symbols. `http://investexcel.net/all-yahoo-finance-stock-tickers/`. Accessed: 2022-05-29.

[tra] Nyse trading days per year. `https://www.nyse.com/publicdocs/Trading_Days.pdf`. Accessed: 2022-05-21.

[Whe14] Charles J. Wheelan. *Naked Statistics: Stripping the Dread from the Data*. W. W. Norton, 2014.

[Wil78] J. Welles Wilder. *New Concept in Technical Trading Systems*. Hunter Publishing Company, 1978.

[yah] Yahoo finance api. `https://www.yahoofinanceapi.com/`. Accessed: 2022-05-22.

[Zub11] Paul Zubulake. *The high frequency game changer: how automated trading strategies have revolutionized the markets*. John Wiley & Sons, 2011.