ZHAW Zurich University of Applied Sciences

School of Management and Law

Department of Banking, Finance, Insurance

Master of Science in Banking & Finance

A Machine Learning Approach to Stock Price Prediction

Sylvio Rhyner

Supervisor:

Dr. Marc Weibel

Zurich, June 15th 2023

Management Summary

Investing in the stock market is a complex task that relies heavily on the ability to predict future price movements. For decades, investors and financial analysts have sought ways to forecast these trends using a multitude of methodologies, ranging from fundamental to technical analysis.

This research implemented a Support Vector Regression (SVR) model, a powerful machine learning technique used for predicting real-valued outputs, to forecast the direction of Dow Jones Industrial Average (DJIA) stocks. The model used several input features, including six technical indicators, to predict future stock prices. The generated predictions were then used to produce trading signals that guided the construction and rebalancing of portfolios.

Three different portfolio strategies - weekly, monthly, and quarterly rebalancing - were considered, all of which were compared to a passive equal-weighted (EW) benchmark portfolio. Each strategy relied on the SVR model's trading signals for stock selection, thereby encapsulating a Machine Learning-driven approach to investment management.

The weekly rebalancing strategy emerged as the top performer across multiple metrics, including net cumulative return, annualized return, and the Sharpe ratio, thus outperforming the monthly, quarterly, and even the EW benchmark portfolio. This illustrates the potential benefits of frequent portfolio rebalancing in optimizing returns, despite the associated transaction costs. The results also show the influence of major global events, such as the Global Financial Crisis and the COVID-19 pandemic, as well as the rate of transaction costs on the portfolio performance.

Future research should be directed towards the optimization of time windows for technical indicators and the weights of selected stocks.

Statement of Authorship

“"I hereby declare that this thesis is my own work, that it has been created by me without the help of others, using only the sources referenced, and that I will not supply any copies of this thesis to any third parties without written permission by the head of this degree program."

Sylvio Rhyner

Table of contents

List of Tables

Table of Figures

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
APE	Absolute Percentage Error
API	Application Programming Interface
ATR	Average True Range
BB	Bollinger Bands
CNN	Convolutional Neural Network
CSI 300	China Securities Index
DJIA	Dow Jones Industrial Average
EMA	Exponential Moving Average
EMH	Efficient Market Hypothesis
EW	Equal Weighted
GAN	Generative Adversarial Network
ISE	Istanbul Stock Exchange
MACD	Moving Average Convergence Divergence
MAPE	Mean Absolute Percentage Error
MFI	Money Flow Index
ML	Machine Learning
NB	Naïve Bayes
OHLCV-Data	Open High Low Close Volume Data
rbf	Radial basis function
RMSE	Root Means Squared Error
RNN	Recurrent Neural Network
RSI	Relative Strength Index
SIFMA	Securities Industry and Financial Markets Association
SMA	Simple Moving Average
std	Standard Deviation
SVM	Support Vector Machine
SVR	Support Vector Regression
TA	Technical Analysis
TI	Technical Indicator
WMA	Weighted Moving Average

Introduction

Investing in the stock market is a complex task that relies heavily on the ability to predict future price movements. For decades, investors and financial analysts have sought ways to forecast these trends using a multitude of methodologies, ranging from fundamental to technical analysis. The latter approach, grounded in the belief that prices move in identifiable trends influenced by an array of factors, has gained considerable traction among practitioners. Among various forecasting techniques encompassed by technical analysis, the utilization of trading systems, defined by a set of rules generating trading signals based on certain parameter values, has been particularly prominent.

With the advent of artificial intelligence and machine learning, new frontiers have opened up in the realm of stock market prediction. The ability of machine learning algorithms to discern patterns in large, complex datasets lends itself naturally to financial forecasting. A technique that has garnered significant interest is the Support Vector Machine (SVM), a robust algorithm adept at both classification and regression tasks.

Research Question

The primary research question guiding this thesis is: **Can a Support Vector Regression model, trained on technical indicators, be utilized to implement a trading strategy that outperforms a benchmark?** To answer this question, three different strategies using different time spans for price prediction and rebalancing (weekly, monthly and quarterly) will be implemented and compared to a benchmark portfolio.

Objective and Significance

The objective of this thesis is to investigate the potential application of SVR in the financial market, with a specific focus on their capacity to predict stock price movements based on technical indicators. Given the potential of these predictive models, the implications of this research could be of significant value to investors, traders, and financial analysts. Better understanding of how machine learning algorithms can augment traditional technical analysis could revolutionize investment strategies, leading to more informed decision-making and more effective portfolio management.

The significance of this study lies in its potential to bridge the gap between traditional technical analysis and cutting-edge machine learning techniques. By providing empirical evidence on the feasibility and profitability of an SVR-based trading strategy, this research could significantly contribute to the ongoing discourse on the integration of machine learning in financial market analysis. Moreover, by comparing the performance of the proposed SVM-based strategy against a benchmark, this study seeks to provide a tangible measure of its viability in the real-world trading environment.

Structure of the Thesis

The thesis is organized as follows. Chapter One provides an introduction to the topic and states the scope of the thesis. Chapter Two provides an overview of relevant literature for the topics of stock price prediction, technical indicators and machine learning algorithms. Chapter Three gives information about the data and how it is processed, defines Support Vector Machines and Support Vector Regression and explains how the trained models were used for stock selection. Chapter Four presents the results and in Chapter Five conclusions are drawn.

Literature Review

Stock market prediction is a classical problem in the intersection of finance and computer science. For this problem, the famous efficient market hypothesis (EMH) gives a pessimistic view and implies that financial market are efficient (Fama, 1965), which maintains that technical analysis or fundamental analysis (or any analysis) would not yield any consistent over-average profit to investors. However, many researchers disagree with EMH (Malkiel, 2003). Some studies are trying to measure the different efficiency levels for mature and emerging markets, while other studies are trying to build effective prediction models for stock markets. Machine learning algorithms and technical indicators have been used in numerous studies to analyze financial market trends and develop effective investment strategies.

Technical indicators provide valuable information about market trends and patterns, which can be used to train machine learning models and enhance their accuracy. The effort starts with fundamental analysis and technical analysis. Fundamental analysis evaluates the stock price based on its intrinsic value, i.e., fair value, while technical analysis only relies based on charts and trends. When it comes to technical indicators, according to Park and Irwin (2007), the number of studies that identify positive technical trading profits is far greater than the number of studies that find negative profits. Form a total of 95 modern studies, 56 studies find profitability (or predictability) of technical trading strategies, while 20 studies report negative results. The rest, 19 studies, indicate mixed results. Shynkevich et al. (2017) focus on the relationship between the forecast horizon and the time frame used to calculate technical indicators (TIs) in financial forecasting systems based on technical analysis (TA). The study aims to find the optimal combination of these parameters to maximize the performance of predictive systems in forecasting price movements. The research reveals a pattern where the highest prediction performance is achieved when the input window length of an indicator is approximately equal to the forecast horizon.

Machine learning (ML) is a type of data science that involves models that can learn from data and get better at their tasks over time. It started when scientists in the 1950s and 1960s wanted to teach computers to learn like humans. ML helps extract knowledge from data, which can be used to make predictions and find new information. It is helpful for tasks like processing images and voices, recognizing patterns, and solving complicated problems. ML methods have a distinct advantage over traditional statistics and econometrics methods. ML algorithms can handle large amounts of structured and unstructured data and quickly make decisions or predictions. This is because ML models don't rely on predetermined assumptions about equations, variable interactions, or statistical distributions of parameters. Instead, ML methods aim to accurately predict outcomes based on other variables. In their study Kumar et al. (2022) try to answer the question what different types of ML algorithms are used to predict stock market prices. In their review of 30 research papers, they find that the following ML techniques are used:

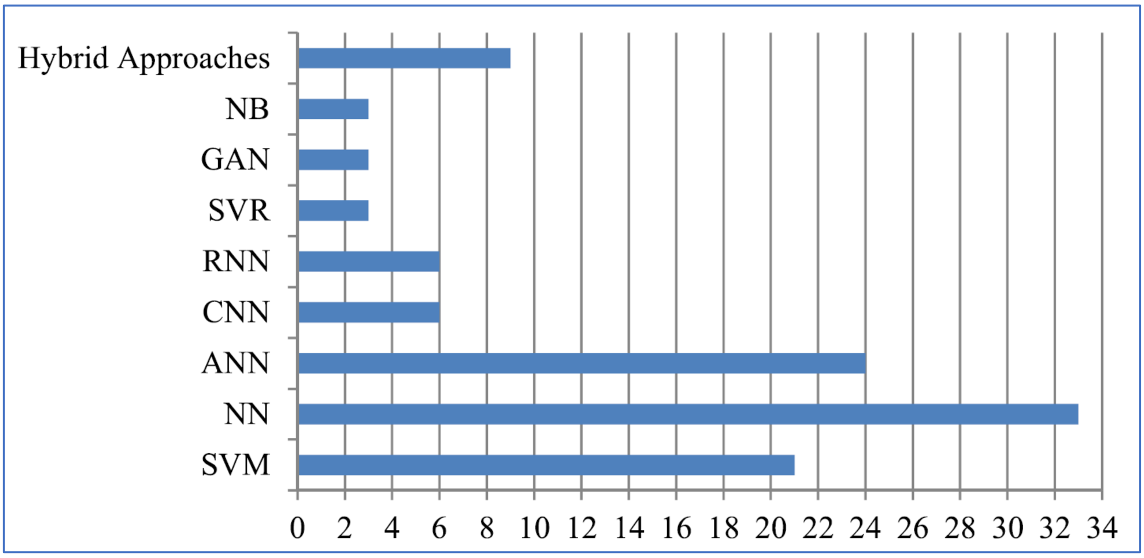

Figure 1: Most frequently used ML techniques according to Kumar et al (D. Kumar et al., 2022).

Naïve Bayes (NB) is classification algorithm that has been proven to be effective in practical applications such as text classification, medical diagnosis and system performance management (Domingos et al., 1997). It assigns the most likely class to an example described by its feature vector. It works well for classification tasks, despite the unrealistic assumption that features are independent given class (Rish et al., 2001). However, it requires a very large number of data to achieve good results (Jadhav & Channe, 2016).

Generative Adversarial Network (GAN), a novel framework, utilizes a zero-sum game concept to train two models (Ghahramani et al., 2014). In this adversarial setup, the generator acts as a cheater, generating data similar to real data, while the discriminator serves as the judge, distinguishing between real and generated data. The goal is to achieve a state where the discriminator cannot differentiate between the two types of data. At this convergence, the generator effectively captures the underlying data distributions.

Leveraging this principle, GAN architectures have been designed specifically for predicting stock closing prices (K. Zhang et al., 2019).

RNN, or Recurrent Neural Network, is a type of deep learning architecture that offers several advantages. One key benefit of RNN is its ability to incorporate contextual information during the training phase (Zaremba et al., 2014). This makes it particularly useful for handling time series data and is well-suited for tasks like stock prediction. Since stock prices at a given moment often exhibit correlations with past trends, RNN can effectively capture these connections (Zhu, 2020).

A Convolutional Neural Network (CNN) is a type of deep learning architecture commonly used in computer vision tasks, introduced by LeCun et. al (1995). It is designed to automatically learn and extract meaningful features from images. CNNs are composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers use filters to perform convolutions across the input image, extracting spatial patterns and detecting features like edges and textures. Pooling layers downsample the feature maps, reducing the spatial dimensions and extracting the most important information (Hoseinzade & Haratizadeh, 2019). Finally, fully connected layers connect the extracted features to the output layer for classification or regression tasks.

Artificial neural networks (ANNs) are a popular and highly accurate soft computing technique utilized as forecasting models in various domains such as social, engineering, economic, business, finance, foreign exchange, and stock problems (Khashei et al., 2010). The widespread adoption of ANNs can be attributed to their numerous distinctive features that appeal to both researchers and industry professionals. ANNs are characterized by their data-driven nature and self-adaptive capabilities, requiring minimal prior assumptions. They excel as predictors, leveraging their capacity to derive generalized insights from learned data, enabling accurate inferences about the underlying aspects of a population (G. P. Zhang & Qi, 2005).

SVM and its regression counterpart, Support Vector Regression (SVR), have been extensively used for financial prediction tasks. Obthong et al. (2020) reviewed and compared the state-of-the-art of ML algorithms and techniques that have been used in finance, especially in stock price prediction. They find that Support Vector Machines and Support Vector Regression have many advantages, when applied to financial time-series, such as powerful prediction results (even in problems with high dimensions), suitability to handle multiple inputs and the ability to provide the optimal global solution. However, they mention disadvantages too, such as sensitivity to outliers and high sensitivity to parameter selection. Pan et al. (2016), study the impact of various data normalization methods on using support vector machine (SVM) and technical indicators to predict the price movement of stock index. The experimental results suggest that the prediction system based on SVM and technical indicators, should carefully choose an appropriate data normalization method so as to avoid its negative influence on prediction accuracy and the processing time on training. For the closing prices of TRY/USD and TRY/EUR Altan and Karasu (2019) obtained various models with different kernel scale values in SVM and the model that estimates financial time series with the highest accuracy is proposed. It is seen that the forecasting performance of the proposed SVM model for the financial time series data set is higher than the performance of other models. Choosing optimal hyperparameter values for support vector machines is an important step in SVM design. This is usually done by minimizing either an estimate of generalization error or some other related performance measure. In their paper, Duan et al. (2003) find for SVMs with L1 soft-margin formulation, none of the simple measures yields a performance uniformly as good as k-fold cross validation. For SVMs with L2 soft-margin formulation, the radius margin bound gives a very good prediction of optimal hyperparameter values.

Vapnik (2000) argues that SVR has an advantage over SVM in the case of predicting continuous variables such as stock prices. The reason is that SVR can manage both linear and non-linear cases using different kernel functions and provides a global optimum solution due to the use of convex optimization during training (Drucker et al., 1996). Moreover, studies like Tay and Cao (2001) have reported the superiority of SVR in financial forecasting over other machine learning algorithms.. The strength of SVR lies in its ability to generalize well, even with limited training data, which is particularly useful in the unpredictable world of stock market prediction. Due to these strong arguments in favor of its predictive power, SVR has been selected as the model-of-choice in this study.

Methodology

In this chapter, the methodology employed to investigate the research objective is presented. The following methodology will be applied to answer the stated research question.

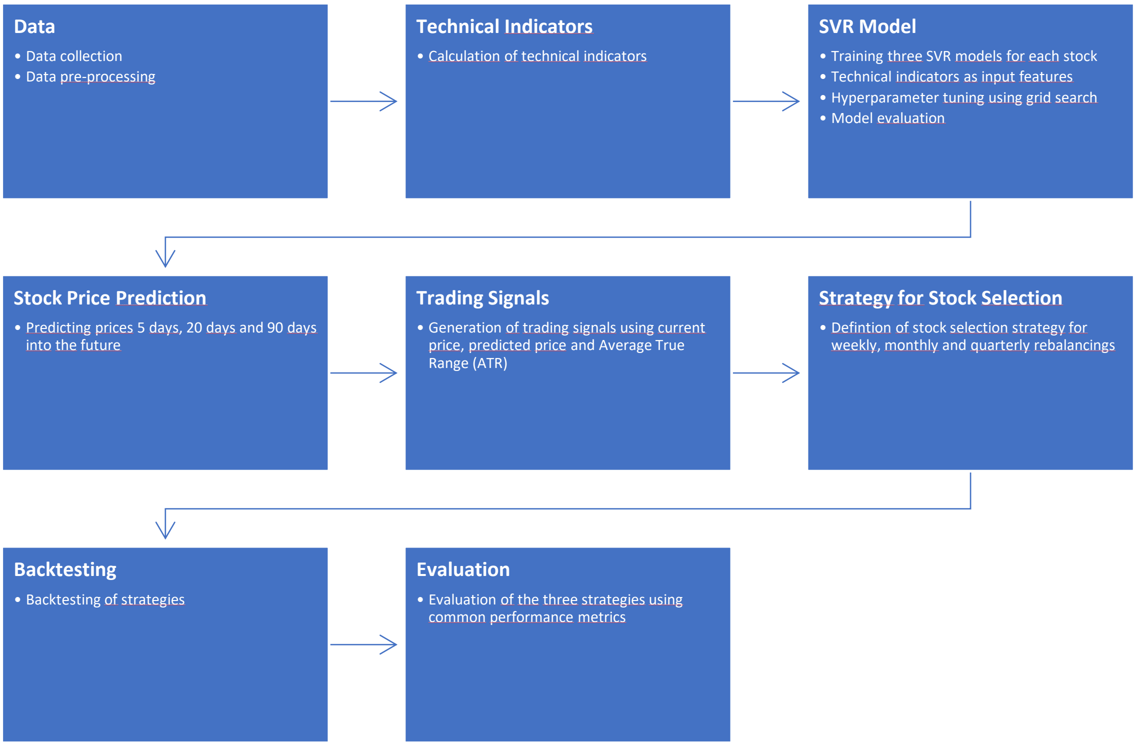

Figure 2: Overview of Methodology

The collected data will be pre-processed to be suitable for a Support Vector Regression model. For each of the 27 stocks the six technical indicators will be calculated at each point in time, based on a rolling time window. These indicators will be used as input features for the Support Vector Regression model. Since the aim of this study is to compare a weekly, monthly and quarterly rebalanced stock selection approach to the equal weighted benchmark portfolio, for each stock three different models will be trained. The input features will be the same for each of the three models, but the target variable will be the close price of the stock 5 days, 20 days or 90 days into the future. Each of the models will be subject to hyperparameter tuning using a grid search approach. This will results in 81 tuned models (3 time shifts × 27 stocks). Each model will then be used to predict the closing prices of the stocks. Using the predicted prices and a 2 ATR threshold trading signals will be generated. The Buy, Sell and Hold Signals will be used for stock selection of within each strategy. The strategies will then be backtested on the dataset and evaluated. The individual steps of the process will be explained in the following subchapters.

Data

The predictive model developed herein targets future price fluctuations of the constituents forming part of the Dow Jones Industrial Average (DJIA) stock market index. The DJIA comprises 30 significant companies listed on U.S. stock exchanges (*Dow Jones Industrial Average®*, n.d.). For this thesis's purposes, the focus is confined to those companies within the DJIA that have maintained a trading history from January 01, 2002 to April 14, 2023, to eliminate the possibility of missing data. The selection of DJIA constituents is strategic, given their historically low correlation (S. Jones & Kincaid, 2014), a characteristic that is anticipated to yield more intriguing outcomes as compared to stocks with high correlation. The dataset was retrieved from Refinitiv via their Python-based API.

The implementation of machine learning techniques necessitates pre-processing to transition raw time series data into a format that is more suitable for machine learning purposes. The following pre-processing steps were employed in this regard: First, the dataset was examined for missing values.

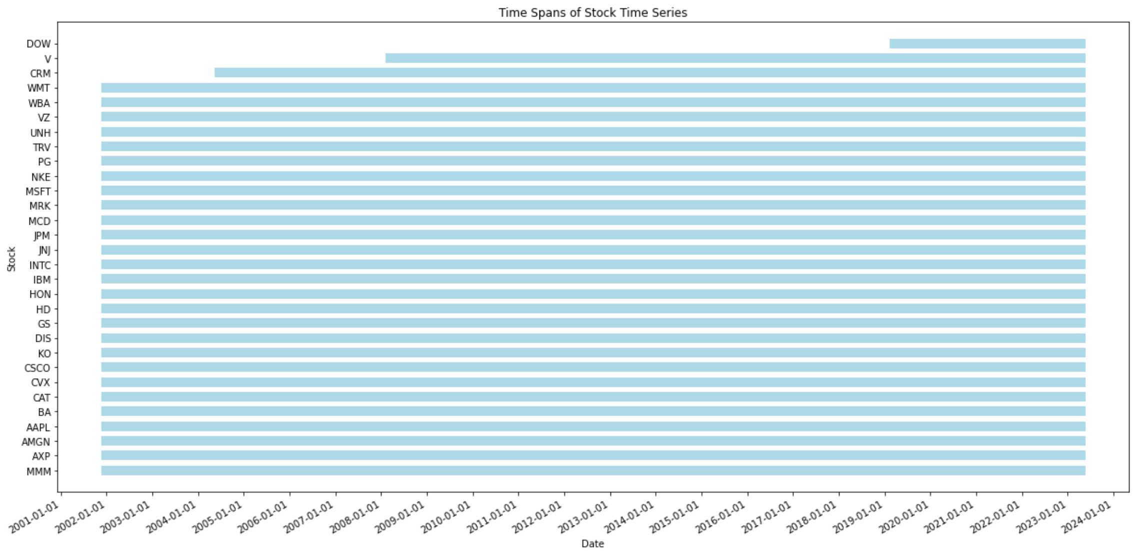

Figure 3: Timespan of Stock Time Series for each stock in DJIA

Since for the three stocks CRM, DOW and V a larger parts of historical price data are missing, they were excluded completely from the data set. The remaining 27 stocks were checked for any missing data but were found to be complete. For each stock the following data was collected:

- Opening price (Open)
- Highest price during the day (High)
- Lowest price during the day (Low)
- Closing price (Close)
- Adjusted closing price, accounting for dividends and stock splits (Adj. Close)
- Volume of shares traded during the day (Volume)

Figure 4 clearly shows the differences of scale between the closing prices of certain stocks in the data set.

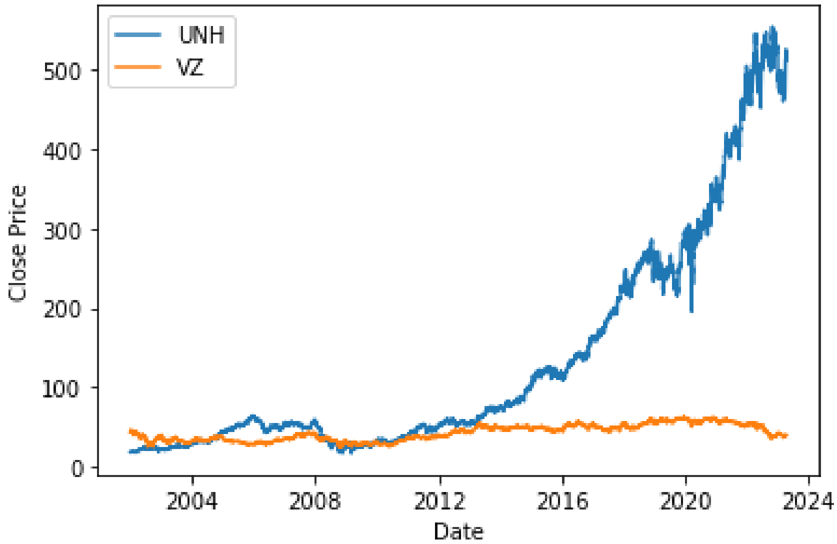

Figure 4: Closing Prices for Stocks UNH and VZ

To avoid giving variables with bigger values to much weight, they need to be transformed for better comparability. In this study, the data was standardized using z-score standardization

$$x'' = \frac{x - mean}{\sigma}$$

leading to a mean of 0 and standard deviation of 1.

Technical Indicators

Technical indicators are complex calculations that traders and investors use to predict future price movements and make informed decisions about buying and selling stocks. They are based on past performance data such as price and volume, and the field of technical analysis has a rich history, numerous controversies, and varying levels of success. The origins of technical analysis can be traced back to the 18th century in Japan, where rice traders used what would later become known as candlestick charts. However, it wasn't until the late 19th and early 20th century when Charles Dow, co-founder of Dow Jones & Company, set down the basic principles of technical analysis (Edwards et al., 2018). Many indicators we use today were developed in the mid-20th century. The Relative Strength Index (RSI), for example, was introduced by J. Welles Wilder Jr. in his 1978 book "New Concepts in Technical Trading Systems." The Moving Average Convergence Divergence (MACD) was developed by Gerald Appel in the late 1970s. In the 1980s, John Bollinger introduced Bollinger Bands. All these indicators are still widely used in today's trading environment.

In this thesis, six technical indicators are calculated and used to form input features for the Support Vector Regression model. Each indicator facilitates the inclusion of additional information derived from a stock price in a different way. For each stock, technical indicators are calculated for every trading day from raw time series data which may include open, close, high and low stock prices and trading volume. The following six technical indicators are computed in Python using the TA-Lib Library.

Exponential Moving Average

An Exponential Moving Average (EMA) is a type of Weighted Moving Average (WMA) which assigns a weighting factor to each value in the data series according to its age. Like WMA, in EMA the most recent data gets the greatest weight, and each data value gets a smaller weight as we go back chronologically. But unlike WMA, in EMA the weighting for each older data point decreases exponentially, so its' never reaching zero. The EMA for a time series Y can be calculated recursively as

$$S_1 = Y_1,$$

$$\text{for } t > 1, S_t = \alpha \times Y_r + (1 - \alpha) \times S_{r-1},$$

where Y_t is the value at time period t, S_t is the value of EMA at time period t, and α represents the degree of weighting decrease, a constant smoothing factor between 0 and 1. Commonly, α is calculated using the formula

$$\alpha = \frac{2}{(n+1)}$$

By giving more weight to recent data, the EMA reacts more significantly to recent price changes than a simple moving average (SMA). This can be beneficial in volatile markets. Due to its weighting scheme, the EMA reduces the lag that typically occurs with the SMA, providing a more accurate and timely indication of trend changes. Once the initial EMA is calculated, subsequent EMA calculations are straightforward. Also, EMAs are easy to plot and interpret on a chart, which makes them user-friendly for traders and analysts. On the other hand, while the EMA is more responsive to recent changes, this sensitivity can also lead to false signals, causing traders to enter or exit trades prematurely. The EMA performs well in markets with consistent trends but can perform poorly in range-bound or choppy markets. The weighting scheme of the EMA may overemphasize recent data at the expense of older, but still potentially relevant, information. In this study two EMAs are calculated and used as input features for the SVR Mode: a short term 5-day EMA and a longer 20-day EMA.

Many studies have applied EMA to predict stock prices. For instance, Brock, Lakonishok, and LeBaron (1992) found that technical trading rules based on moving averages and trading range break methods yielded significant positive returns in the Dow Jones Industrial Average index from the 1897 to 1986 period. However, they also noted that transaction costs could erode these returns. Other studies like Neely, Weller, and Dittmar (1997) used a genetic algorithm to find optimal parameter values for moving average rules in foreign exchange markets. They found that these rules can be profitable, though the profit was smaller than that reported in other studies.

Moving Average Convergence Divergence

Moving Average Convergence Divergence (MACD) is used in financial analysis to identify potential changes in trend and momentum in the price of an asset. The MACD is calculated by subtracting a n period EMA from m period EMA, with a j-period EMA used as a signal line to generate buy and sell signals. In this study, the default values 26, 12 and 9 for the signal line were used, according to Murphy (1999, p. 253).

MACD can help identify the direction of a trend. When the MACD line crosses above the signal line, it generates a bullish signal (upward trend), indicating that it may be an optimal time to buy. Conversely, when the MACD line crosses below the signal line, it generates a bearish signal (downward trend), indicating that it may be a good time to sell. The MACD can also help identify potential price reversals. If the price of a security is substantially diverging from the MACD value, it could signal that the security is overbought or oversold and may soon. The MACD can help traders understand the momentum and the duration of a particular trend. A higher MACD value indicates stronger upward momentum, while a lower MACD value indicates stronger downward momentum.

Like all moving averages, the MACD is inherently a lagging indicator. While this can help confirm trends and generate trading signals, it can also generate false or late signals. This can cause traders to enter or exit trades late and lose potential profits. MACD tends to be more effective in volatile markets that move within a range. In a ranging market, the MACD may generate many false signals. MACD, like all technical analysis tools, relies heavily on historical data.

MACD has been widely used in studies aiming to predict stock prices. For instance, in a study by Chong and Ng (2008), MACD was used as a short-term trading rule and provided profitable results over the examined periods. Anghel (2015) found that traders who employ the MACD as a technical analysis investment approach in the stock market may occasionally achieve abnormal returns that are adjusted for both cost and risk

Bollinger Bands

Bollinger Bands are a volatility indicator that uses a moving average and standard deviation to show the range of price movement around a central point. Bollinger Bands are plotted two standard deviations above and below a simple moving average (Bollinger, 1992). The time frame for the calculations is such that it is descriptive of the intermediate-term trend. The distance between the upper and lower bands indicates the level of volatility.

$$\sigma = \sqrt{\frac{\sum_{j=1}^{N}\left(X_j - \bar{X}\right)^2}{N}}$$

$$\bar{X} = \frac{\sum_{j=1}^{N} X_j}{N}$$

Upper Band = $\bar{X} + 2\sigma$

Middle Band = $\bar{X}$

Lower Band = $\bar{X} - 2\sigma$

When the price is continually touching the upper Bollinger Band, the market is considered overbought, suggesting a possible price decrease in the future. Conversely, when the price is continually touching the lower Bollinger Band, the market is considered oversold, suggesting a possible price increase in the future. The width of the Bollinger Bands can be an indicator of market volatility. If the bands are significantly apart, the market is more volatile. If they are closer together, the market is less volatile. Bollinger Bands can also signal potential price reversals. For instance, when prices move outside the Bollinger Bands, a continuation of the current trend is implied. A move that originates at one band tends to go all the way to the other band. This observation is useful when projecting price targets. In this study, a 20-day average is used as a middle line, as suggested by Murphy (1999, p. 209)

Like all indicators that use moving averages, Bollinger Bands are lagging indicators and can generate signals late. This can cause traders to enter or exit trades at suboptimal points. Bollinger Bands can sometimes produce false signals, particularly in sideways markets where price often bounces between the upper and lower bands. This can lead to false assumptions of price reversals. While Bollinger Bands can indicate potential oversold or overbought conditions, they do not provide any indication of the future direction of the price. Therefore, they are typically used in conjunction with other indicators.

Bollinger Bands have been used in several studies to predict stock prices. For example, a paper by Chen et al. (2018) explored the profitability of technical analysis by using Bollinger Bands strategies on the CSI 300 stock index futures. They found that Bollinger Bands could generate profitable trading signals under certain market conditions.

Fibonacci Retracements

Fibonacci retracements are a set of horizontal lines used to identify potential areas of support or resistance based on key levels derived from the Fibonacci sequence. Fibonacci retracement is used by many technical traders to identify strategic places for transactions to be placed, target prices or stop losses. The Fibonacci sequence of numbers is as follows: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, and so on. The key Fibonacci ratios are 0%, 23.6%, 38.2%, 50%, 61.8% and 100% (Gaucan et al., 2011).

The 1% Fibonacci ratio is calculated as follows:

$$F_{100\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{-0} = 1$$

The key Fibonacci ratio of 0.618% - also referred to as "the golden ratio" or "the golden mean" is found by dividing any number in the sequence by the number that immediately follows it. For example: 8/13 is approximately 0.6154, and 55/89 is approximately 0.6180.

$$F_{61.8\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{-1} \approx 0.6180$$

The 0.382 ratio is found by dividing any number in the sequence by the number that is found two places to the right. For example: 34/89 is approximately 0.3820.

$$F_{38.2\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{-2} \approx 0.381966$$

The 0.236 ratio is found by dividing any number in the sequence by the number that is three places to the right. For example: 55/233 is approximately 0.2361.

$$F_{23.6\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{-3} \approx 0.236068$$

The 0 ratio is:

$$F_{0\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{-\infty} = 0$$

Fibonacci retracements can help traders identify potential reversal points in the market. Traders watch these levels for signs of a bounce or break, which can suggest the continuation or reversal of a trend. Fibonacci retracements can be applied to any timeframe or trading instrument, making them a flexible tool for traders. Traders often use Fibonacci retracements to set stop-loss orders or target prices, which can help with risk management. In this study, the Fibonacci retracements were calculated on a rolling 20-day window.

The placement of Fibonacci retracement levels can be subjective and depends on the trader's interpretation of significant price points. Different traders may identify different high and low points, resulting in different retracement levels. Like all technical analysis tools, Fibonacci retracements can generate false signals. The price may not always react as expected at a Fibonacci level, leading to potential losses. Fibonacci retracements work best when combined with other technical analysis tools or indicators. Depending on them as a standalone method can lead to inaccurate predictions.

In the study "Magic of Fibonacci sequence in prediction of stock behavior" by R Kumar (2014), a new model for predicting the retracement of financial trends was proposed. This model focuses on the analysis of the falling wave, referred to as wave 2, in relation to the preceding rising wave, referred to as wave 1. According to the research findings, it appears that the retracement of wave 2 with respect to wave 1 can be predicted more accurately by the proposed model. This potentially offers a valuable tool for investors, who often struggle to accurately predict the bottoms or peaks of financial trends. Such mispredictions can lead to financial losses, making the need for reliable predictive tools crucial.

Relative Strength Index

In technical analysis, there are two major problems that occur when constructing a momentum line using price differences. One problem is the erratic movement caused by sharp changes in the values being dropped off. A sharp advance or decline n days ago, in the case of a n day momentum line, possibly causes sudden shifts in the momentum line event if the current price shows little change. The second problem is that there is the need for a constant range for comparison purposes. The Relative Strength Index (RSI) solves both issues by providing soothing and a constant vertical range of 0 to 100 (Murphy, 1999, p.240)

$$RSI = 100 - \frac{100}{1 + RS}$$

$$RS = \frac{Average\ of\ x\ days'up\ closes}{Average\ of\ x\ days'down\ closes}$$

The shorter the observed time period, the more sensitive the oscillator becomes and the wider its amplitude gets. The RSI works best, when its fluctuations reach the upper and lower extremes (Murphy, 1999, p. 241).

The RSI can help identify overbought (typically an RSI above 70) and oversold (typically an RSI below 30) conditions. These might suggest that a price correction is due, or that the security's price is set to rebound, respectively. The RSI can show divergences from the price trend which can signal potential reversals. For instance, if a security's price reaches a new high, but the RSI does not reach a new high, this divergence could be an indication that the trend is losing strength and may reverse. These are variations of divergences which can also predict reversals. A failure swing occurs when RSI enters overbought or oversold territory, pulls back from it, then crosses back through the 70 or 30 levels again, failing to reach the previous extreme. In this study the last 14 days closing prices are used for the calculation, according to Wilder (1978, p. 63)

Like all technical indicators, the RSI can produce false signals. For example, the RSI may indicate that a security is overbought, but the price may continue to rise. RSI can be too sensitive to market noise and may suggest trend changes too often in a volatile market. While RSI can provide valuable insight, it's most effective when used in conjunction with other technical analysis tools and indicators.

Regarding the application of RSI in predicting stock prices, Chong and Ng (2008) found in their study titled "Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30" that using the RSI rule could generate significant positive return. However, they also emphasized the importance of transaction costs in the practical application of such rules.

Stochastic Oscillator

The stochastic oscillator relies on the notion that during price hikes, closing prices usually gravitate towards the top end of the price spectrum. On the other hand, when prices are dropping, the closing price is generally situated near the range's lower end. This stochastic method employs two lines: the %K line and the %D line. Out of the two, the %D line holds greater significance and is responsible for producing key signals The primary objective is to establish the position of the most recent closing price in comparison to the price range over a selected time period.(Murphy, 1999). The formula is

$$\%K = 100 \, \frac{C - L_n}{H_n - L_n}$$

Where C is the latest close, L_n is the lowest low for the last n periods, and H_n is the highest high for the same n periods. The formula measures on a percentage basis of 0 to 100 where the closing price is in relation to the price range for the selected period n.

The Stochastic Oscillator can generate signals for overbought (above 80) and oversold (below 20) market conditions. These can be valuable signals for potential market reversals. When the price of a security makes a new high or low that isn't confirmed by the Stochastic Oscillator, it can indicate a potential price reversal. The Stochastic Oscillator can be used to confirm the strength of trends. During a strong uptrend, the Stochastic Oscillator tends to remain in the overbought area, while during a strong downtrend, it may remain in the oversold area. In this study the lookback period is 14 days and the %D is a 3 day moving average, according to Murphy (1999, p. 247).

Like many indicators, the Stochastic Oscillator can produce false signals. For example, it might signal that a market is overbought when the upward trend is strong, and prices are likely to continue up. The Stochastic Oscillator doesn't provide signals for entry and exit points on its own. Therefore, it is best used in conjunction with other technical indicators and charting tools. In volatile markets, the Stochastic Oscillator can give signals too early or even false signals.

Regarding the application of the Stochastic Oscillator in predicting stock prices, a study by Guresen, Kayakutlu, and Daim (2011) titled "Using artificial neural network models in stock market index prediction" applied Stochastic Oscillators among other indicators to forecast the ISE (Istanbul Stock Exchange) National-100 Index values. They found that models incorporating Stochastic Oscillators, along with other technical analysis tools, had good predictive abilities.

The time window for the calculation of technical indicators will not be optimized in this study. The study thereby follows the rational of Murphy (1999, p. 221): “Some argue that optimization helps their trading results and other that it doesn’t. (..). The decision to optimize is a personal one. Most evidence, however, suggests that optimization is not the Holy Grail some think it to be.” Therefore, the time window(s) for each technical indicator will be as described before and in the following table:

Indicator	**Time Window**
MACD	12, 26, 9
EMA_5	5
EMA_20	20
Bollinger Bands	20
RSI	14
Stochastic Oscillator	14, 3, 3
Fibonacci Retracement	20

Table 1: Overview of Technical Indicators and their time windows

Support Vector Machines

Machine Learning algorithms are mainly divided into four categories: Supervised learning, Unsupervised learning, Semi-supervised learning, and Reinforcement learning (Mohammed et al., 2016), as shown in figure 5.

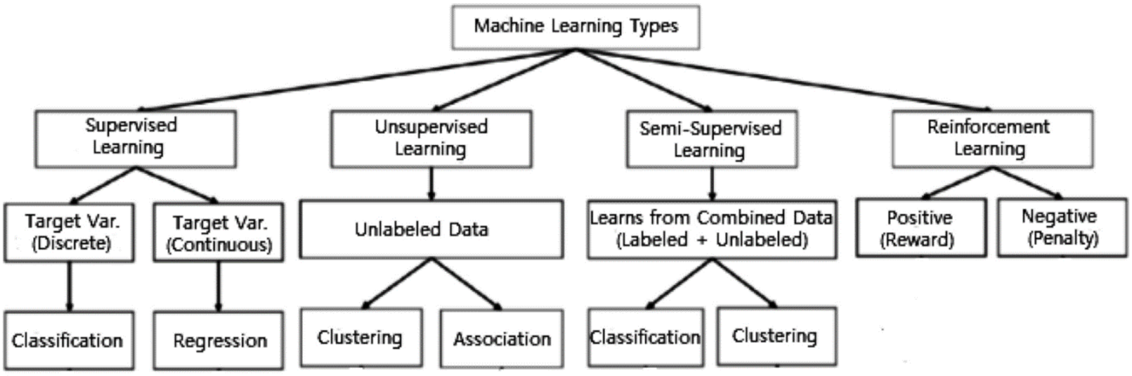

Figure 5: Various types of machine learning techniques (Sarker, 2021)

Supervised learning: Supervised learning is a task within machine learning that revolves around the learning of a function that connects an input to an output, given a set of input-output sample pairs (Han et al., 2022). By utilizing labeled training data and a series of training examples, the function is inferred. This method of learning is employed when there are specific goals to be achieved with a predefined set of inputs - a task-oriented approach (Sarker et al., 2020). Two widely recognized supervised tasks include classification, which differentiates data, and regression, which fits the data. For example, supervised learning can be used to predict the class label or sentiment of a text, such as a tweet or product review - an application known as text classification.

Unsupervised Learning: Unsupervised learning is a process that examines unlabeled datasets without human intervention - a data-driven process (Han et al., 2022). It's popularly employed to derive generative features, discern meaningful patterns and structures, groupings in outputs, and for exploration. Common tasks in unsupervised learning include clustering, density estimation, feature learning, dimensionality reduction, finding association rules, anomaly detection, among others.

Semi-supervised Learning: Semi-supervised learning is a fusion of supervised and unsupervised methods, operating on both labeled and unlabeled data (Sarker, 2021). Therefore, it lies between learning "with supervision" and learning "without supervision".

In many real-world scenarios, labeled data may be scarce, while unlabeled data are plentiful - here, semi-supervised learning proves useful (Mohammed et al., 2016). The primary objective of a semi-supervised learning model is to offer a more accurate prediction than what could be achieved using only the labeled data from the model. Examples of application areas include machine translation, fraud detection, data labeling, and text classification.

Reinforcement Learning: Reinforcement learning is a machine learning paradigm that empowers software agents and machines to determine the best course of action within a specific context or environment, thus enhancing performance - an environment-focused approach (Kaelbling et al., 1996). This learning type hinges on the concept of rewards and penalties, with the end goal being to use insights from environmental interactions to make decisions that either maximize rewards or minimize risks (Mohammed et al., 2016). It's a powerful technique for training AI models, aiding in the enhancement of automation or the optimization of the performance of complex systems like robotics, autonomous driving tasks, and supply chain logistics. However, it's typically not favored for addressing simple or direct problems.

Support Vector Machines (SVMs) are part of the supervised learning models and are a powerful and popular machine learning algorithm used for both classification and regression tasks. SVMs were first proposed by Vladimir Vapnik and Alexey Chervonenkis in 1963, and later extended in the 1990s by Corinna Cortes and Vladimir Vapnik. At a high level, SVMs are based on the idea of finding the hyperplane that best separates the data into different classes. This hyperplane is chosen such that it maximizes the margin between the two closest points from different classes, which are called support vectors. Following Schölkopf and Smola (2001) one of the fundamental problems of learning theory is the following: suppose we are given two classes of objects. We are then faced with a new object, that must be assigned to one of the two classes. This problem can be formalized as follows:

$$(x_1, y_1), \ldots, (x_m, y_m) \in X \times \{\pm 1\}$$

X is a nonempty set of inputs and the y_i are called labels. In this example there are two classes of patterns, labeled as +1 and -1, referred to as binary classification. However, the patterns could be anything and no assumptions on X have been made other than it being a set. We want to generalize unseen data points. In the case of pattern recognition, this means that given some new pattern $x \in X$, we want to predict the corresponding $y \in \{\pm 1\}$. This means, y will be chosen in a way such that (x, y) is similar to the input used for training. Therefore, we need notions of similarity in X and in $\{\pm 1\}$. It is straightforward to describe the likeness of the outputs with a binary classification task, as there are only two possible outcomes: the labels can either match or not. However, determining the similarity metric for the inputs is a fundamental and complex inquiry that underpins the domain of machine learning. Statistical learning theory shows that it is crucial to restrict the class of functions that the learning machine can implement to one with a capacity that is suitable for the amount of available training data.

To design learning algorithms, we thus must come up with a class of functions whose capacity can be computed. Support Vector classifiers are based on the class of hyperplanes

$$(w \times x) + b \; = 0 \in \; \mathbb{R}^N, b \in \mathbb{R}$$

Corresponding to decision functions

$$f(x) = sign\big((w \times x) + b\big).$$

The optimal hyperplane, defined as the one with the maximal margin of separation between the two classes, has the lowest capacity. It can be uniquely constructed by solving a constrained quadratic optimization problem whose solution w has an expansion $w = \sum_i v_i \times x_i$ in terms of a subset of training patterns that lie on the margin. These training patters are called support vectors. They carry all relevant information about the classification problem. Both the quadratic programming problem and the final decision function $f(x) = sign(\sum_i v_i(x \times x_i) + b)$ depend only on dot products between patterns. This is precisely what lets us generalize to the nonlinear case. The idea of Support Vector Machines is to map data into some other dot product space, called feature space, F, via nonlinear map

$$\Phi: \mathbb{R}^N \rightarrow F,$$

and perform the above linear algorithm in F. As stated above, this only requires the evaluation of dot products.

$$k(x, y) := \left(\Phi(x) \times \Phi(y)\right).$$

If F is high-dimensional, the right-hand side of the equation will be very expensive to compute. But in some cases, there is a simple kernel k that can be evaluated efficiently. For instance, the polynomial kernel.

$$k(x, y) = (x \times y)^d$$

Can be shown to correspond to a map Φ into the space spanned by all products of exactly d dimensions of $\mathbb{R}^N$. For $d = 2$ and $x, y \in \mathbb{R}^2$, for example, we have

$$(x \times y)^2 = \left(\begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \end{pmatrix}\right)^2$$

$$= \left(\begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}\begin{pmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{pmatrix}\right)$$

$$\left(\Phi(x) \times \Phi(y)\right),$$

defining $\Phi(x) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$. More generally, we can prove that for every kernel that gives rise to a positive matrix $\left(k\left(x_i, x_j\right)\right)_{ij}$, we can construct a map Φ such that $k(x, y) := \left(\Phi(x) \times \Phi(y)\right)$ holds.

Besides polynominal kernels, radial basis function (RBF) kernels such as

$$k(x, y) = \exp\left(- \parallel x - y \parallel^2 / \left(2\sigma^2\right)\right),$$

and sigmoid kernels (with gain K and offset Θ)

$$k(x, y) = \tanh\left(K(x \times y) + \ \Theta\right.$$

are used by practitioners.

We now have all the tools to construct nonlinear classifiers. To this end, we substitute $\Phi(x_i)$ for each training example x_i, and perform the optimal hyperplane algorithm in F. Because we are using kernels, we will thus end up with a nonlinear decision function of the form

$$f(x) = sign(\sum_{i=1}^{1} v_i \times k(x, x_i) + b).$$

The parameters v_i are computed as the solution of a quadratic programming problem. In the input space, the hyperplane corresponds to a nonlinear decision function whose form is determined by the kernel. Figure 6 illustrates a simple binary classification example in a linear space.

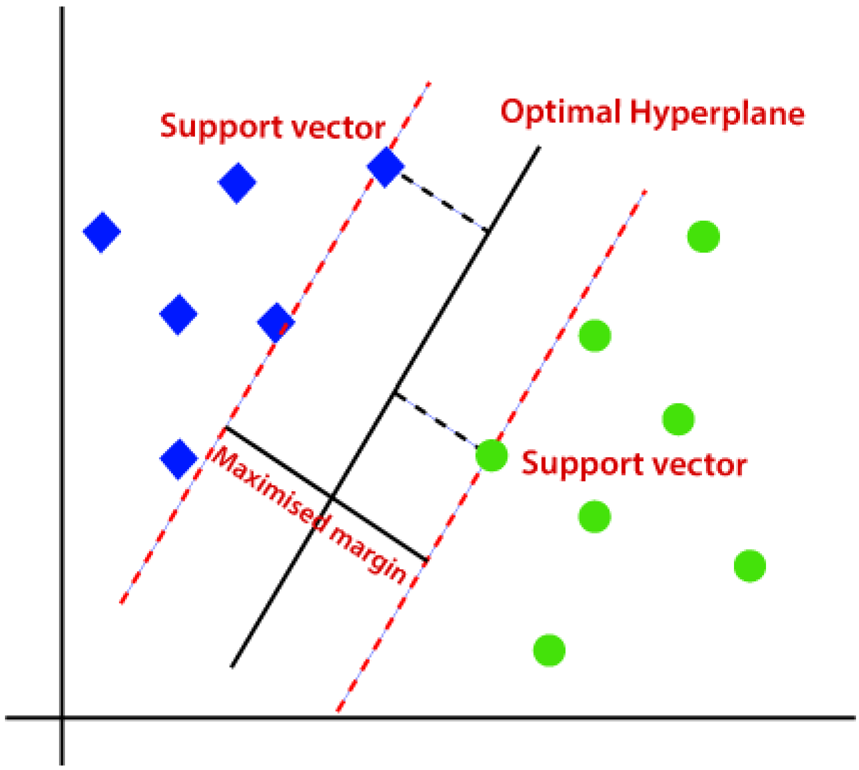

Figure 6: Optimal Hyperplane, Support Vector and Margin (Saini, 2023)

So far we have only considered the case of classification. A generalization to regression estimation, that is to $y \in \mathbb{R}$ can be given. In this case, the algorithm will construct a linear function in the feature space such that the training points lie within a distance $\varepsilon > 0$. Similar to the pattern-recognition, this can be written as a quadratic programming problem in terms of kernels. The nonlinear regression estimate takes the form

$$f(x) = \sum_{i=1}^{1} v_i \times k(x_i, x) + b$$

ε can either be applied in advance, or an upper bound can be specified on the fraction of training points allowed to lie outside of a distance ε from the regression estimate (asymptotically, the number of support vectors) and the corresponding ε is computed automatically.

Support Vector Regression Model

In Python, for each stock in the dataset and for each look-ahead period, a function is called that performs tasks related to the training and evaluation of a Support Vector Regression model. It standardizes the features and then trains an SVR model on them. There are 14 features that are used as inputs for the SVR model:

Feature	**Description**
MACD	Shows the relationship between two moving averages of the stocks's price.
MACD_Signal	Signal line for MACD.
MACD_Hist	The difference between the MACD and the signal line, illustrating when a crossover happens.
EMA_5	Exponential Moving Average for the past 5 days.
EMA_20	Exponential Moving Average for the past 20 days.
Upper_BB	The upper band in Bollinger Bands. Represents the level where the price is perceived as overbought.
Middle_BB	The middle band in Bollinger Bands.
Lower_BB	The lower band in Bollinger Bands. Represents the level where the price is perceived as oversold.
RSI	Relative Strength Index. Shows the magnitude of recent price changes to evaluate overbought or oversold conditions.
SlowK	The SlowK line in Stochastic Oscillator, which is a smoothed version of the FastK line.
SlowD	The SlowD line in Stochastic Oscillator, which is a 3-period moving average of the SlowK line.
Fibonacci_0.236	The 23.6% retracement level from a major price move.
Fibonacci_0.382	The 38.2% retracement level from a major price move.
Fibonacci_0.618	The 61.8% retracement level from a major price move.

Table 2: Input features for the SVR model

In this study, for each stock three different models will be trained. One where the target variable is the close prices 5 days in the future, one where the target variable is the close price 20 days in the future and one where the target variable is the close price 90 days in the future. In this paper, the different periods are also referred to as shifts (5, 20, 90). This approach results in three different models for each stock which will be used to predict prices for each shift.

Hyperparameter Tuning

A grid search strategy for hyperparameter tuning is applied to each model and shift. The hyperparameters in the grid include the SVR kernel type, C, epsilon, and gamma. Grid search involves training an SVR model for each possible combination of these hyperparameters, and selecting the best performing model based on cross-validated performance. Time series cross-validation with three splits is used to estimate the performance of each model. This type of cross-validation is suitable for time series data, as it respects the temporal ordering of observations.

As for the parameters in the grid, they represent different hyperparameters of the SVR model and their potential values:

The kernel parameter refers to the type of kernel function to be used in the SVR. A kernel function is a mathematical function used in SVR to transform the data into a suitable form. The Radial basis function (rbf) kernel is popular for non-linear problems. For performance reasons, no other kernels were used in the grid.

Kernel parameters in the grid: [rbf]

The C parameter is a regularization parameter, similar to those found in other machine learning models. It determines the trade-off between achieving a low training error and a low testing error, which is the bias-variance tradeoff. A smaller value of C creates a wider margin, which may result in more training errors. However, this wider margin may lead to better generalization on the test data. Conversely, a larger value of C aims to minimize the training error, which can sometimes lead to overfitting and poor generalization on the test data.

C parameters in the grid: [0.1, 1, 10, 100]

The epsilon parameter is specific to Support Vector Regression. It defines the width of the epsilon-insensitive tube within which no penalty is associated in the loss function with points predicted within a distance epsilon from the actual value. A smaller epsilon tightens the tube, making it narrower. This indicates a stricter tolerance for errors, and SVR will strive to fit the data more closely. As a result, the regression line may pass closer to the training data points, potentially leading to overfitting. A larger epsilon widens the tube, allowing more errors to be tolerated. This indicates a higher acceptance for errors and a more flexible fitting criterion. The regression line may not fit the training data as closely, but it will likely exhibit better generalization to unseen data.

Epsilon parameters in the grid: [0.1, 0.01]

The gamma parameter is crucial when using the rbf kernel. It is a parameter of the rbf function and can be thought of as the 'spread' of the kernel and therefore the decision region. If gamma is too large, the radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization with C will be able to prevent overfitting. When gamma is small, the influence of each training example extends farther, resulting in a smoother decision boundary or regression function. In this case, the model will tend to have a more generalized fit, as the impact of individual training examples decreases, and the model focuses on capturing the overall trend of the data. However, using a very small gamma value can also make the model more prone to underfitting, potentially leading to poor performance when dealing with complex or nonlinear data. On the other hand, a larger gamma value makes the influence of each training example more localized, resulting in a more complex and intricate decision boundary or regression function. The model becomes more sensitive to individual training examples and can potentially overfit the data, especially if the number of training examples is limited.

Gamma parameters in the grid: ['scale', 'auto', 0.1, 1, 10]

Performance Metrics

The following performance metrics will be calculated on the 81 trained Support Vector Regression models. They will be used to compare performance within each shift but also between them.

Mean Absolute Percentage Error (MAPE): This metric provides an understanding of the prediction error as a percentage, allowing for intuitive understanding of the model's performance. A lower MAPE value means a better fit of the data by the model. MAPE si the average of absolute percentage errors (APE). Let A_t and F_t denote the actual and forecast values at datapoint t , respectively. Then MAPE is defined as:

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t}\right|$$

Where N is the number of data points (Bowerman et al., 2005). MAPE is scale-independent and easy to interpret, which makes it popular with industry practitioners (Byrne, 2012).

Root Mean Squared Error (RMSE): It is a standard measure for the prediction error of a regression model. It measures the average magnitude of the residuals or prediction errors. The lower the RMSE, the better the model's performance. The RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}$$

The underlying assumption when presenting the RMSE is that the errors are unbiased and follow a normal distribution (Chai & Draxler, 2014).

R-squared Score: Also known as the coefficient of determination, the R-squared score indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, with 1 indicating perfect prediction capabilities. Consider a regression model of outcomes y and predictors X with predicted values $E(y|X,\theta)$, fit to data $(X,y)_n$, $n = 1,\ldots,N$. Ordinary least squares yields an estimated parameter vector θ with predicted values $\hat{y}_n = E(y \mid X_n, \hat{\theta})$ and residual variance $V_{n=1}^{N}\ \hat{y}_n$, where we are using the notation,

$$V_{n=1}^{N}\, z_n = \frac{1}{N-1}\sum_{n=1}^{N}(z_n - \bar{z})^2$$

for any vector z. The proportion of variance explained,

$$R^2 = \frac{V_{n=1}^{N}\, \hat{y}_n}{V_{n=1}^{N}\, y_n}$$

is a commonly used measure of model fit (Gelman et al., 2019).

Price prediction and trading signals

For each stock and each shift, the respective trained model will be applied to predict the stock price. The predicted prices are then used to generate Buy, Sell and Hold signals at each day of the time series. To account for varying market conditions the signals are calculated using the Average True Range (ATR), based on J. Welles Wilder Jr. (1978, p. 23). The equation is as follows:

$$ATR_{latest} = \frac{Previous\ ATR(n-1) + TR}{n}$$

where:

n = Number of periods

TR = True range

If no previous ATR is calculated, one must use:

$$\left(\frac{1}{n}\right)\sum_{i}^{n} TR_i$$

where:

TR_i = Particular true range, such as first day's TR, then second, then third.

n = Number of periods

The true range is calculated as follows:

$$TR = \max\left[(H - L), | H - C_p |, | L - C_p |\right]$$

where:

H = Today's high

L = Today's low

C_p = Yesterday's closing price

max = Highest value of the three terms

so that:

$(H - L)$ = Today's high minus the low

$| H - C_p |$ = Absolute value of today's high minus yesterday's closing price

$| L - C_p |$ = Absolute value of today's low minus yesterday's closing price

The ATR is calculated for each stock and each shift period. If the predicted price of the respective shift is 2ATR above the current price, a Buy signal will be generated. If the predicted price is 2ATR below the current price, a Sell signal will be generated. Values in between will be set as Hold signals.

After the above-described steps, for each of the stocks we have time series data frames in python, containing the following information:

- OHLCV Data
- 14 columns with values of calculated technical indicators
- Three columns with target values (future closing prices, one for each shift)
- Three columns with predicted future closing prices (one for each shift)
- Three columns with values of calculated ATR
- Three signal columns containing Buy, Hold or Sell (on for each shift)

The signal columns can now be used for stock selection.

Portfolio Stock Selection and Performance Metrics

For each time shift a portfolio is constructed that is rebalanced at the start of the corresponding period. This results in a total of three strategies: Weekly, Monthly and Quarterly. At each rebalancing date stocks with a Buy signal will be bought. If they were previously bought and their current signal is Hold, they will stay in the portfolio. If a stock has a Sell signal at the current rebalancing date, the stock will only be sold if it has been in the portfolio for the previous period. The portfolio will not enter any short positions. To measure the effectiveness of the active stock selection, the three strategies are compared to an equal weighted portfolio containing all 27 stocks of the remaining dataset. For the equal weighted portfolio, stocks will be bought at the beginning of the time period and held until the end.

To be able to compare the results of the different strategies to each other, the standard performance metrics as shown in table 3 will be used. The transaction cost rate for this study was set to 34.3 basis points per trade. Meaning, at each rebalancing date the net return of will be affected by the amount of trades (Buy, Sell) times the cost rate. The cost rate was taken from SIFMA's Global Equity Market Primer (Kolchin, 2021).

Metric	Description
Net Cumulative Return	It is the cumulative return on the investment at the end of the investment period and any changes in the value of the investment.
Total Transaction Cost	This is the total trading cost associated with executing the trading strategy.
Annualized Return	This metric represents the average amount of money earned by an investment each year over the given time period.
Annualized Volatility	It is a statistical measure of the dispersion of returns for a given security or market index. In stock markets, volatility is often seen as a measure of risk.
Sharpe Ratio	This is a measure used to understand the return of an investment compared to its risk. It is the average return earned in excess of the risk-free rate per unit of volatility or total risk.
Maximum Drawdown	Maximum drawdown represents the largest single drop from peak to bottom in the value of a portfolio, before a new peak is achieved. It's an indicator of downside risk over a specified time period.
Mean Return	This is the average return of the investment over the given time period.
Standard Deviation of Return	This is a statistical measurement that shows how much an investment's returns can vary from its average return (volatility).
Skewness	This is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. Negative skewness is often associated with the presence of more extreme negative returns.
Kurtosis	Kurtosis is a statistical measure used to describe the distribution of observed data around the mean. It is used in finance to identify the extreme values in one versus the other tail. It is a measure of the "tailedness" of the distribution.
p-value	The p-value is the probability of obtaining test results at least as extreme as the results actually observed, under the assumption that the null hypothesis is correct. A smaller p-value means that there is stronger evidence in favor of the alternative hypothesis.
t-statistic	The t-statistic is a type of test statistic in statistics used in hypothesis testing. In the context of finance and trading, it might be used to test whether the mean strategy performance differs significantly from zero.

Table 3: Description of Performance Metrics and Statistics

Results

This section presents the key results obtained from the evaluation of the support vector regression (SVR) models in the context of stock price prediction. The performance of these models is assessed after an extensive grid search for the best combination of parameters. Additionally, the distribution of trading signals generated by the models, based on predicted prices and a threshold of 2 Average True Range (ATR), is analyzed. Then, the performance of a stock selection strategy, which utilizes the trading signals of all stocks, is evaluated against an equal-weighted portfolio of all stocks. The results provide valuable insights into the effectiveness and potential profitability of utilizing SVR models for stock price prediction and subsequent trading decisions. By examining the performance, this analysis offers a comprehensive evaluation of the trading strategies employed. The following sections will present and describe the results obtained, shedding light on the predictive capabilities of the SVR models, the generated trading signals, and the comparative performance of the trading and stock selection strategies. These findings will contribute to a deeper understanding of the applicability and performance of SVR models in the realm of stock market analysis and trading strategies.

Model performances

The following figures shows the result metrics for the best performing SVR models after grid search for each of the 27 stocks and each time shift.

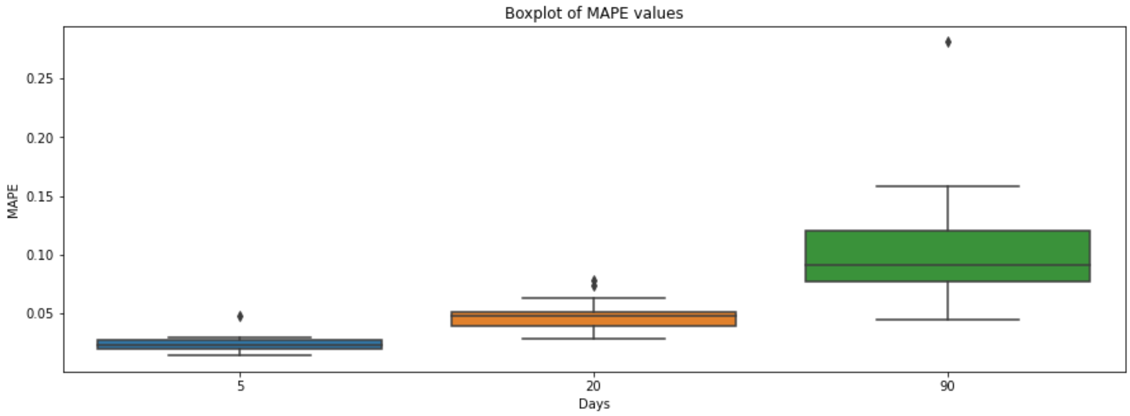

Figure 7:MAPE Boxplots over all stocks for each time shift

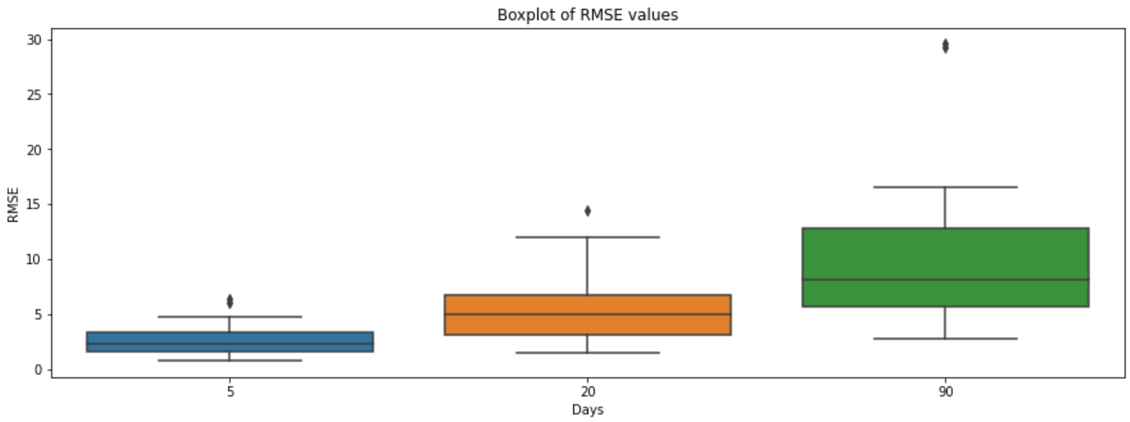

Figure 8: RMSE Boxplot over all stocks for each time shift

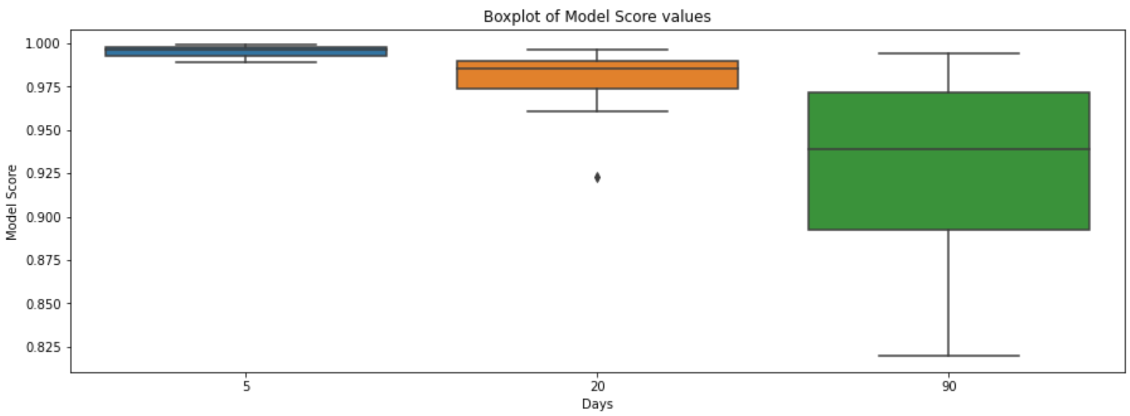

Figure 9: Model Scores Boxplot over all stocks for each shift

The time shift 5 models demonstrate the highest accuracy, with a relatively low average MAPE of 0.0234 and average RMSE of 2.5871. This indicates a close approximation to actual prices and a smaller deviation from observed values. The average model score for this time shift is 0.9953, reflecting strong predictive performance. The time shift 20 models show slightly lower accuracy, with a higher MAPE of 0.0.466 and larger average RMSE of 5.2636. This suggests a greater percentage difference from actual prices and a larger deviation from observed values. As a result, the average model score decreases to 0.9805, indicating a slightly less accurate prediction compared to the time shift 5 models. The time shift 90 models exhibit the lowest average accuracy among the models for the three look-ahead periods, with a higher average MAPE of 0.1028 and larger average RMSE of 10.1909. These results suggest a significant deviation from actual prices and a lower prediction accuracy. Consequently, the average model scores decrease to 0.9263, reflecting their weaker performance. These differences can be attributed to the time shift intervals and the inherent nature of stock price prediction. The shorter time shift interval in the time shift 5 models allow them to capture recent and relevant market trends, resulting in higher accuracy. Conversely, the longer time shift interval in the time shift 90 models makes it more challenging to capture and incorporate dynamic market changes, leading to lower accuracy. The time shift 20 models strike a balance between short-term fluctuations and longer-term market trends, resulting in moderate accuracy. MAPE, RMSE and model scores for all individual stocks can be found in the appendices, as well as the best parameters for each of the models, after grid search.

Trading signals for each time shift

The best model for each stock and time shift is applied to predict the prices 5 days, 20 days or 90 days into the future. Based on these price predictions and by using the Average True Range, three different trading signals were generated. If the predict price was two times the ATR bigger than the current price, the stock would be a Buy at the current date. Vice versa, if the predicted price was two times the ATR smaller than the current price, the stock would be a Sell at the current date. Everything between is a Hold. Over all 27 stocks the resulting amount of Buy, Sell and Hold is as follows, for each time shift:

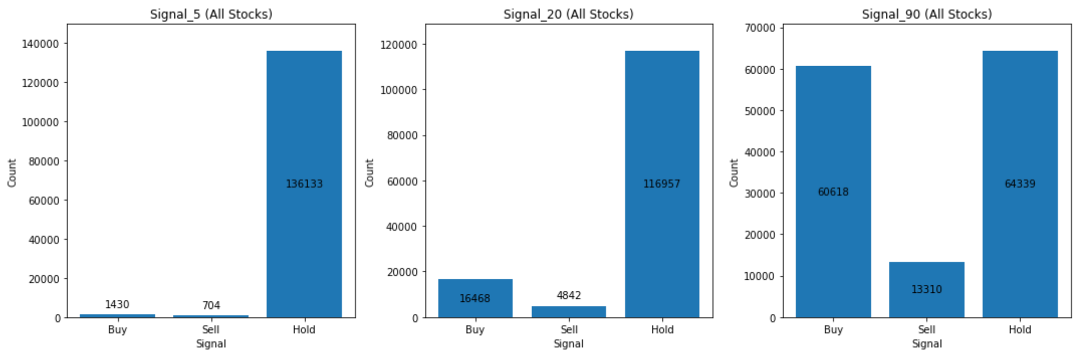

Figure 10: Overview of Trading Signals generated based on predicted prices and 2 ATR threshold

For the time shift of 5, most signals indicate a "Hold" recommendation with 136’133 occurrences. This suggests that the predicted price is within an acceptable range, aligning with a strategy that aims to maintain existing positions. The occurrence of "Buy" signals (1’430 occurrences) indicates opportunities for initiating new positions when the predicted price deviates significantly above the current price. Conversely, "Sell" signals (704 occurrences) suggest opportunities to exit positions when the predicted price deviates significantly below the current price. With a time shift of 20, the distribution of signals changes. The number of "Hold" signals remains the highest with 116’957 occurrences, reflecting the strategy's focus on maintaining existing positions. The increase in "Buy" signals (16’468 occurrences) compared to the time shift of 5 implies that the predicted price tends to deviate further above the current price, potentially indicating greater bullish market expectations. The occurrence of "Sell" signals (4’842 occurrences) suggests opportunities for profit-taking or risk management when the predicted price deviates significantly below the current price. In the case of a time shift of 90, the distribution of signals continues to evolve. The frequency of "Buy" signals increases substantially to 60’618 occurrences, indicating a higher prevalence of predicted price deviations above the current price. This suggests a stronger bullish bias in the market and potential opportunities for initiating new positions. The number of "Sell" signals (13’310 occurrences) remains relatively high, suggesting the need to exit positions when the predicted price deviates significantly below the current price. The occurrence of "Hold" signals (64’339 occurrences) suggests a balanced approach, potentially indicating that the predicted price tends to fluctuate within a moderate range.

Overall, the results demonstrate that the strategy based on 2 ATR deviation generates a considerable number of "Hold" signals across all time shifts. This aligns with a risk-averse approach, maintaining existing positions when the predicted price remains within an acceptable range. The varying frequencies of "Buy" and "Sell" signals highlight opportunities for initiating new positions or exiting existing ones based on the deviation of the predicted price from the current price. The implications of these results suggest that the strategy could be effective in capturing potential price movements when the predicted price deviates significantly from the current price. Traders utilizing this strategy may benefit from taking action based on the signals generated, such as initiating new positions during bullish market expectations or managing risk through position exits during bearish market conditions.

Constructed Portfolios and Their Performances

This section showcases the construction of portfolios based on generated trading signals and their subsequent rebalancing, taking into account the time shift of price predictions. For instance, portfolios relying on a 5-day price prediction were rebalanced on a weekly basis. The visualization of these constructed portfolios is presented here, accompanied by a comprehensive comparison of their performance metrics against each other and an equal-weighted benchmark portfolio. The benchmark portfolio comprises all 27 stocks of the DJIA, and it follows a buy-and-hold strategy. This analysis offers valuable insights into the effectiveness of different trading signals and rebalancing strategies, shedding light on their relative performance and potential advantages over traditional investment approaches.

Weekly Rebalanced Portfolio

For this strategy, at the beginning of each week the portfolio was rebalanced. Figure 11 shows the constructed portfolios over time. Areas in light blue mean the stock was held at this time, while areas in grey mean the stock was currently not part of the portfolio.

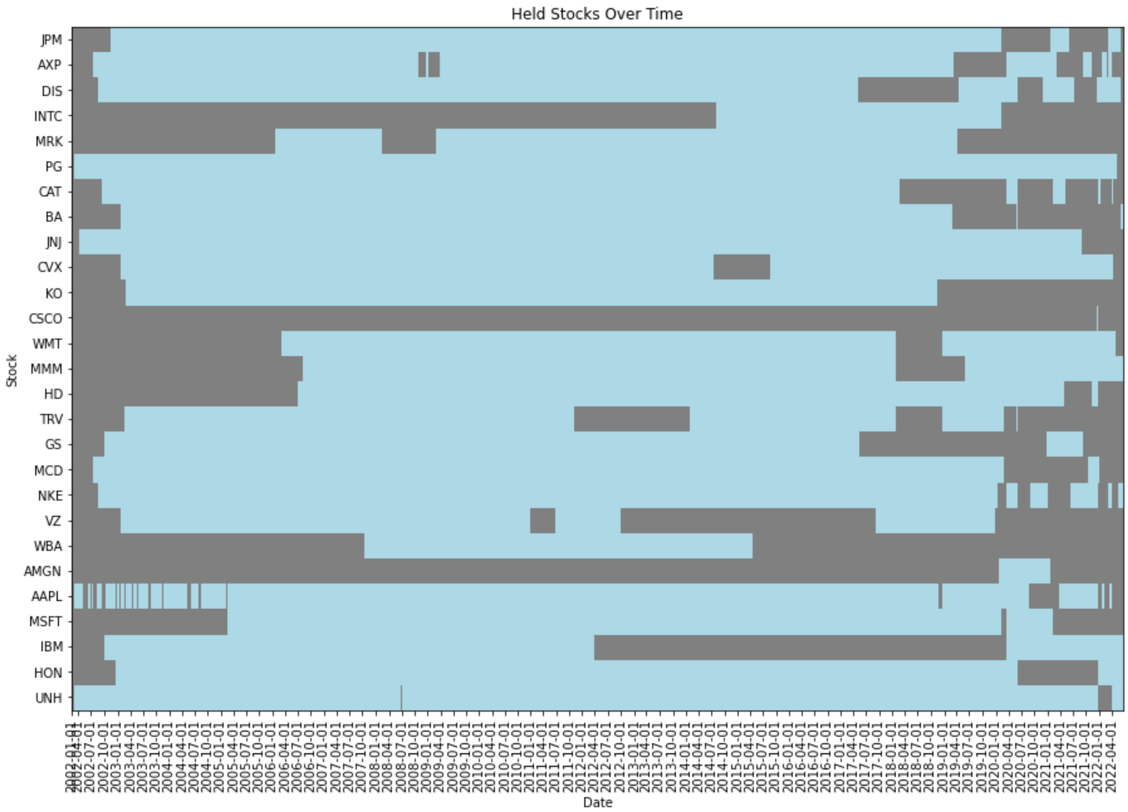

Figure 11: Held Stocks over Time for Weekly Strategy

The visualized portfolio holding strategy reveals critical insights into the investment behavior over the given time. Of note, two specific stocks, CSCO and AMGN, were predominantly excluded from the portfolio, suggesting almost no Buy signals were generated using the 2 ATR method. A significant shift in trading activity is observed from around 2017, escalating further in 2020. This increased dynamism is indicative of changing market conditions, increased volatility, or improved predictive power of the trading signals during this period, necessitating more frequent rebalancing and stock rotation. Additionally, the investment strategy shows a tendency for long-term holding in numerous stocks. This can be explained by the predominant amount of Hold signals as seen in Figure 11. The results also suggest that trading costs may not be as high as one could expect from a weekly rebalanced portfolio.

Monthly Rebalanced Portfolio

For this strategy, at the beginning of each month the portfolio was rebalanced. Figure 12 shows the constructed portfolios over time. Areas in light blue mean the stock was held at this time, while areas in grey mean the stock was currently not part of the portfolio.

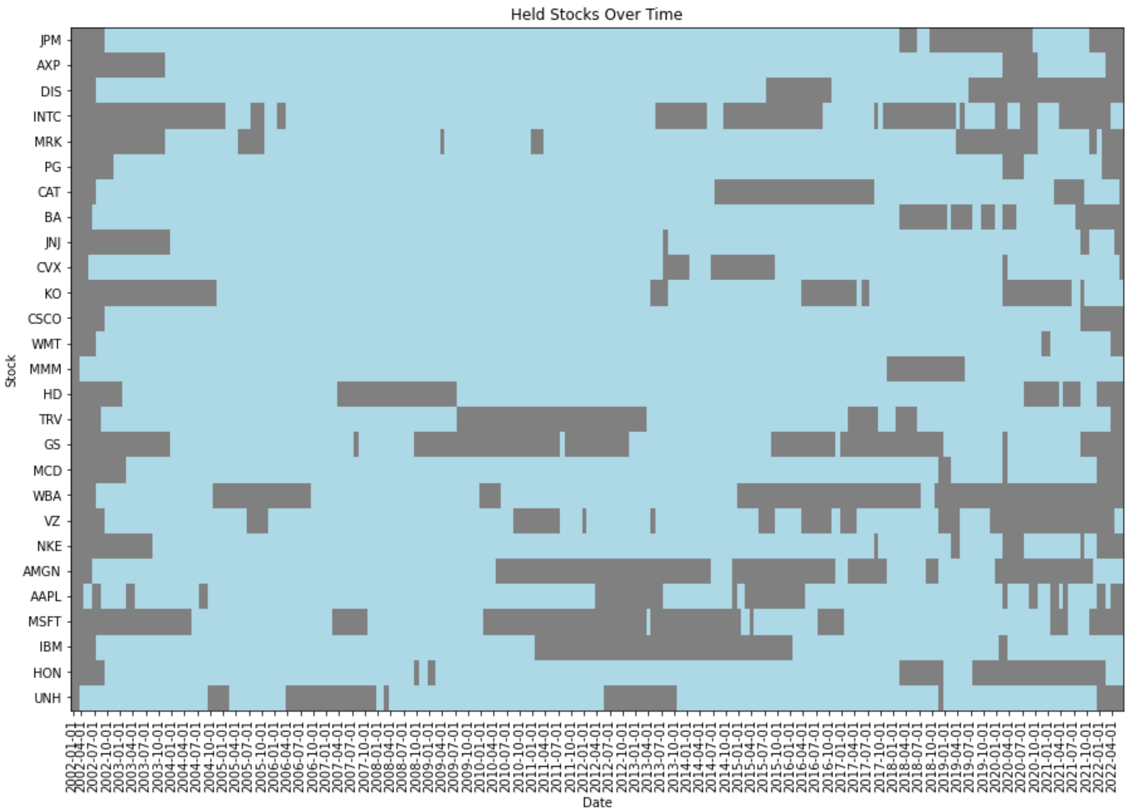

Figure 12: Held Stocks over Time for Monthly Strategy

The portfolio holding strategy visualized monthly reveals unique facets of the applied investment approach. Contrary to the weekly strategy, all stocks in the portfolio have been held for a significant portion of the observed period, highlighting the monthly strategy's wider diversification across individual securities. There's a noticeable increase in trading activity during the middle of the examined timeline. This increased activity could be attributed to either evolving market conditions that warranted more frequent adjustments, or an enhanced effectiveness of the trading signals, stimulating more portfolio rebalancing during this period. In contrast to the weekly strategy, the stock CSCO has been consistently held for extended periods in the monthly strategy, almost perpetually.

This pattern might suggest that the monthly strategy identified a more consistent long-term value in CSCO, which was not picked up in the weekly rebalancing. AMGN, on the other hand, still presents lengthy periods where it is not included in the portfolio, similar to its behavior under the weekly strategy. This could imply that both the weekly and monthly strategies evaluate AMGN similarly, leading to its less frequent presence in the portfolio. The visualized monthly strategy portrays a more frequent turnover of securities compared to the weekly strategy. Considering the strong increase of Buy (and Sell) signals, as seen in Figure 12, compared to the weekly strategy this is not unexpected. The increase in Buy and Sell signals should also lead to more trading costs in this strategy.

Quarterly Rebalanced Portfolio

For this strategy, at the beginning of each quarter the portfolio was rebalanced. Figure 13 shows the constructed portfolios over time. Areas in light blue mean the stock was held at this time, while areas in grey mean the stock was currently not part of the portfolio.

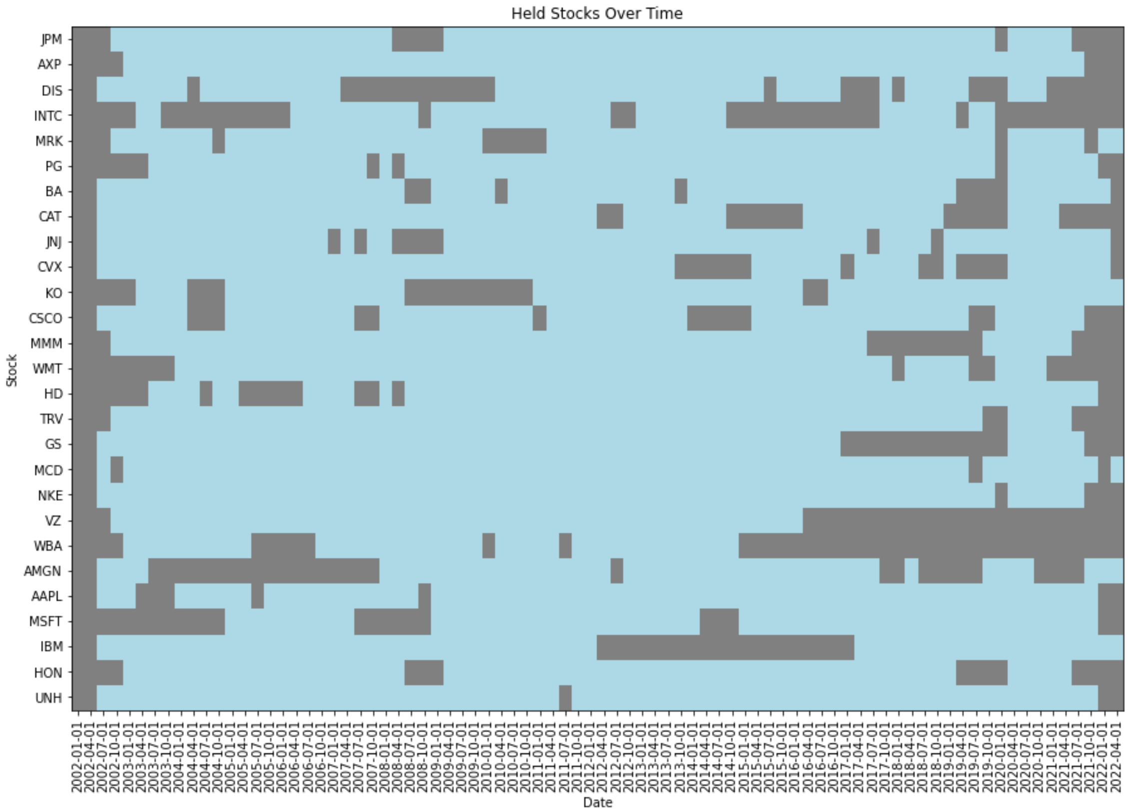

Figure 13: Held Stocks over Time for Quarterly Strategy

The quarterly portfolio holding strategy reveals interesting nuances in the investment approach, particularly when contrasted with the monthly strategy. At a first glance, it appears that the quarterly strategy might lean towards a more 'buy and hold' approach, reflecting a tendency towards longer-term investments. This could be indicative of a higher confidence in the long-term viability of selected stocks. However, it is challenging to definitively ascertain whether the quarterly strategy involves more or less trading activity compared to the monthly strategy without quantitative analysis. An increase in trading activity is observed from 2020, a characteristic shared with the other strategies, indicating shifting market dynamics. The quarterly strategy showcases a consistent preference for certain stocks, such as UNH and MCD, which have been almost perpetually held. This implies consistent Buy or Hold signal over long periods for these two stocks. Interestingly, the first two quarters of the observed period display no trading activity, suggesting that the portfolio was not populated during this time. This means, no Buy signals were generated for any of the 27 stocks during the first two beginning of quarters.

Performance and comparison

It is essential to evaluate the performance metrics of these strategies to discern their effectiveness in delivering returns over the observed period. The analysis will not only include the weekly, monthly, and quarterly strategies, but also consider an equally weighted (EW) benchmark portfolio. The EW benchmark portfolio serves as a crucial point of comparison; it represents a simplistic, yet often effective, strategy where all stocks are purchased at the beginning of the period and held until the end, each with an equal proportion of the total investment. By comparing the active strategies against this benchmark, we can gauge the value generated by their dynamic trading decisions. The following figure shows the cumulative return of all strategies and the benchmark portfolio.

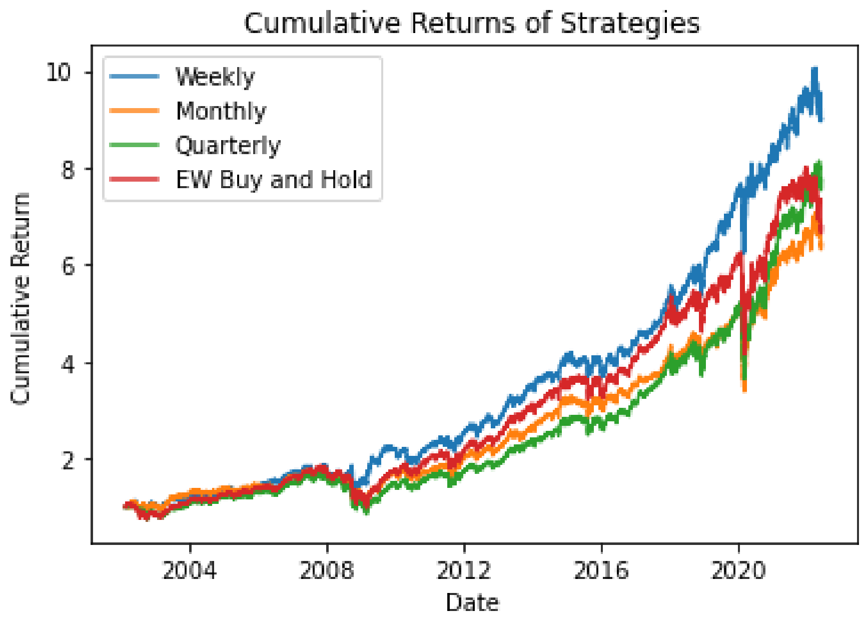

Figure 14: Cumulative Returns of Strategies

From 2002 until 2009 the strategies seem to perform almost identically. It is only from 2009 where differences in cumulative returns are noticeable., aligning with the overall market movements. Notably, the periods during the Global Financial Crisis (GFC) and the COVID-19 stock market crisis have had significant impacts across all strategies. This is evident through sharp declines in the cumulative returns during these periods, underlining the systemic risk that such global events can impart. Comparing the individual strategies, the weekly rebalancing strategy has outperformed the others. It shows the highest cumulative returns, indicating superior performance over the studied period. This could suggest that more frequent rebalancing may have been beneficial in capturing market gains and minimizing losses during volatile periods. The quarterly strategy follows next in terms of cumulative returns. The performance of the monthly strategy and the EW benchmark strategy are almost identical. Both lines are close together, indicating that their returns have been similar over the entire period. The fact that the passive EW benchmark strategy performed on par with the monthly active strategy raises questions about the efficacy of monthly rebalancing in creating excess returns.

Table 4 summarizes several key performance metrics of the weekly, monthly, quarterly, and equal-weighted portfolio strategies.

	Weekly	**Monthly**	**Quarterly**	**Equal Weighted**
Net Cumulative Return	9.0059	6.4328	7.7472	6.7830
Total Transaction Cost	0.5625	0.8541	0.6757	0.0926
Annualized Return	0.1142	0.0959	0.1060	0.0988
Annualized Volatility	0.1614	0.1818	0.1854	0.1891
Sharpe Ratio	0.7076	0.5277	0.5717	0.5223
Maximum Drawdown	0.3421	0.4349	0.4980	0.4705
Mean Return	0.0005	0.0004	0.0005	0.0004
Std Return	0.0102	0.0115	0.0117	0.0119
Skewness	0.3433	0.2079	0.1331	0.1265
Kurtosis	15.2758	13.9819	14.2154	13.3623
p-value	0.0000	0.0000	0.0000	0.0076
t-statistic	104.9631	124.8283	106.2368	2.6717

Table 4: Results of Performance Metrics of Strategies.

The weekly strategy achieved the highest net cumulative return of 9.0059, exceeding that of the monthly (6.4328), quarterly (7.7472), and equal-weighted strategies (6.7830). This means an initial investment in the weekly yields a 900% portfolio return. The total transaction cost was, as one would expect, the lowest for the equal weighted strategy at 0.00926, followed by the weekly strategy. If it was not for the transaction costs, the monthly would have had higher cumulative returns than the equal weighted portfolio. The annualized return, which offers a consistent basis for comparing returns over different periods, was the highest for the weekly strategy at 11.42%, followed by the quarterly strategy at 10.60%. The monthly and equal-weighted strategies lagged with returns of 9.59% and 9.88% respectively.

In terms of risk-adjusted performance, the weekly strategy again outperformed, posting a Sharpe ratio of 0.7076. This was superior to the Sharpe ratios of the monthly (0.5277), quarterly (0.5717), and equal-weighted strategies (0.5223). Interestingly, considering the higher net cumulative return and Sharpe ratio of the weekly strategy, it displayed the lowest standard deviation of returns (0.0102) suggesting lower risk relative to other strategies. The weekly strategy also experienced the lowest maximum drawdown at 0.3421, indicating that a more frequent rebalancing could be a better protection from losses in downside market. The mean return was similar across all strategies.

The skewness values showed that returns for all strategies were positively skewed, with the weekly strategy (0.3433) more so than the others. Kurtosis values were relatively high for all strategies, indicating heavy tails and a higher likelihood of extreme return values. In statistical significance terms, all strategies yielded a p-value of 0.0000, except for the equal-weighted strategy with a p-value of 0.0076, indicating that the null hypothesis of zero mean excess return could be rejected at a conventional significance level. Finally, the t-statistics showed that all active strategies had larger t-statistics compared to the passive equal-weighted strategy. Overall, the weekly strategy displayed superior performance across several metrics, despite having the highest potential for losses during market downturns.

Intuitively, one would expect the highest transaction costs occurring in the strategy with the most frequent rebalancing. This is not the case for the examined strategies in this study. As seen in figure 15, the held stocks within the portfolio are relatively stable at the beginning of the observed period. When looking at the cumulative transaction costs over time for all three strategies, we can see the following:

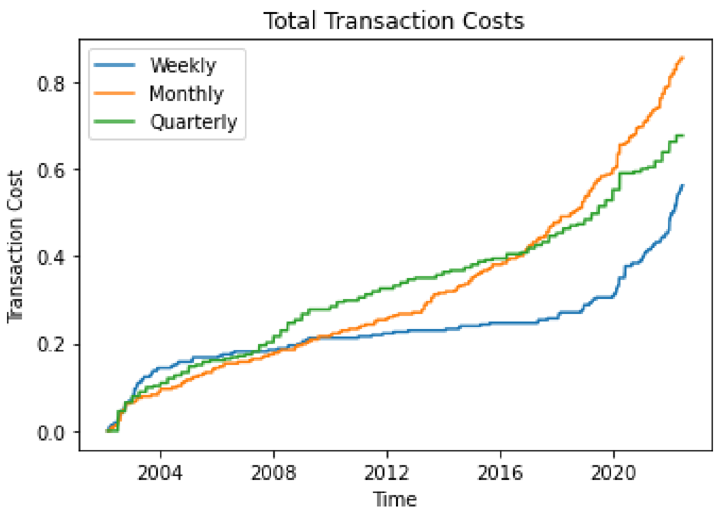

Figure 15: Total Transaction Costs over Time

The weekly strategy accumulates a total of 56.25% transaction cost, the monthly a total of 85.41% and the quarterly 67.57%. The transaction cost of the equal weighted portfolio is omitted in this graph, since all of its trading occur at the initial date, where all of the stocks are bought. The monthly and quarterly strategies seem to have a more stable accumulation of transaction costs over time. The weekly strategy accumulates almost all of its costs before 2004 and after 2020. Especially after the years following the COVID-19 pandemic all strategies show steeper curves. In this study, transaction costs were considered to be 34.3 bps following a study from the Securities Industry and Financial Markets Association SIFMA.

Discussion and Outlook

The preceding analysis has highlighted several noteworthy results regarding the performance of different portfolio rebalancing strategies. Primarily, the study underscores the significant role that the frequency of rebalancing and the utilization of machine learning predictions can have on portfolio performance. Particularly, the weekly rebalancing strategy demonstrated superior performance across a host of metrics, including net cumulative returns, annualized returns, and the Sharpe ratio. However, these findings need to be contextualized and interpreted with other considerations that are intrinsic to portfolio management.

For example, as can be seen the best in the quarterly strategy, there are certain periods where no stocks were selected.

Machine learning models are known for their ability to learn and adapt from data. However, the study raised pertinent questions regarding the reliability of these models in generating trading signals, especially given the inherent unpredictability and noise in financial markets. While the outcomes of this research suggest that the strategy based on 2 ATR deviation of the predicted price from the current price generated a significant number of actionable signals, the effectiveness of these signals is predicated on the accuracy of the machine learning model's predictions. As seen in the quarterly strategy, there are certain periods where no stocks were selected. This might not be a realistic real-world stock selection approach in portfolio management. No stocks bought would result in 100% cash held, while the amount of maximum cash held is often restricted by investment guidelines. As a result, further studies could explore the robustness of machine learning models in financial forecasting and the factors influencing their accuracy. Additionally, there are several parameters in parts of this study that could be optimized in further research, such as the time windows for the technical indicators, the time shift for the price predictions, the factor used for ATR to generate trading signals and the weights given to the selected stocks at the rebalancing dates.

An important finding from this research is the outperformance of the weekly rebalanced portfolio compared to its monthly and quarterly counterparts. While this suggests the potential benefits of more frequent rebalancing in capturing market gains and minimizing losses, it simultaneously raises concerns about higher transaction costs that often accompany increased trading frequency. Consequently, future research might investigate how to optimize the frequency of rebalancing in consideration of transaction costs, which can significantly erode portfolio returns. Further studies could be done to find the transaction cost rate at which and actively managed portfolios is no longer favorable to a simple buy and hold strategy. The strategies developed could also be applied and examined under certain circumstances, i.e. times with high volatility or very low volatility.

The analysis revealed that the passive equal-weighted strategy performed similarly to the active monthly strategy. This provokes a deeper discussion about the circumstances under which passive strategies might be preferable to active strategies. The relative simplicity, lower transaction costs, and comparable performance of passive strategies can make them an attractive choice for certain investors. An area for future investigation might be to understand the conditions under which passive investment strategies can outperform active strategies.

While these results appear promising, it's important to discuss some of the limitations inherent in backtesting-based studies. Backtesting, while providing valuable insights, is essentially a simulation based on historical data, and its predictive power for future performance can be overstated. This limitation is further compounded by potential market microstructure issues such as slippage, i.e., the difference between the expected price of a trade and the price at which the trade is executed. Frequent trading, as seen in the weekly rebalancing strategy, could be particularly susceptible to slippage, potentially eroding the returns seen in the backtesting environment.

Lastly, the significant impacts of major global events, such as the Global Financial Crisis and the COVID-19 pandemic, on the stock market and, by extension, the performance of these trading strategies are evident. This underlines the systemic risk that these events pose to investment strategies. Future research can examine how investors might adapt their strategies in response to such events, potentially incorporating elements of risk management and portfolio diversification.

Conclusion

The aim of this study was to answer the question “Can a Support Vector Regression model, trained on technical indicators, be utilized to implement a trading strategy that outperforms a benchmark?”. The results presented in the previous chapter lead to the conclusion, that SVR can be a powerful tool in the area of stock price prediction and lead to outperformance over an equal weighted benchmark portfolio. While the study provides valuable insights into the performance of different rebalancing strategies, the complex dynamics of financial markets necessitate a multifaceted approach to portfolio management that considers factors such as transaction costs, risk tolerance, and major market events. Thus, in the pursuit of effective investment strategies, it is essential to balance the quest for high returns with a thorough understanding of the associated risks.

References

Altan, A., & Karasu, S. (2019). The effect of kernel values in support vector machine to forecasting performance of financial time series and cognitive decision making. 4, 17–21.

Anghel, G. D. I. (2015). Stock Market Efficiency and the MACD. Evidence from Countries around the World. Procedia Economics and Finance, 32, 1414–1431. https://doi.org/10.1016/S2212-5671(15)01518-X

Bollinger, J. (1992). Using bollinger bands. Stocks & Commodities, 10(2), 47–51.

Bowerman, B. L., O'Connell, R. T., & Koehler, A. B. (2005). Forecasting, time series, and regression: An applied approach (Vol. 4). South-Western Pub.

Brock, W., Lakonishok, J., & Lebaron, B. (1992). Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. Journal of Finance, 47(5), 1731–1764.

Byrne, R. F. (2012). Beyond Traditional Time-Series: Using Demand Sensing to Improve Forecasts in Volatile Times. Journal of Business Forecasting, 31(2), 13–19. Business Source Premier.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geoscientific Model Development, 7(3), 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

Chen, S., Zhang, B., Zhou, G., Qin, Q., & others. (2018). Bollinger bands trading strategy based on wavelet analysis. Applied Economics and Finance, 5(3), 49–58.

Chong, T. T.-L., & Ng, W.-K. (2008). Technical analysis and the London stock exchange: Testing the MACD and RSI rules using the FT30. Applied Economics Letters, 15(14), 1111–1114. https://doi.org/10.1080/13504850600993598

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29, 103–130. https://doi.org/10.1023/A:1007413511361

Dow Jones Industrial Average®. (n.d.). S&P Dow Jones Indices. Retrieved April 9, 2023, from https://www.spglobal.com/spdji/en/indices/equity/dow-jones-industrial-average/

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support Vector Regression Machines. In M. C. Mozer, M. Jordan, & T. Petsche (Eds.), Advances in Neural Information Processing Systems (Vol. 9). MIT Press. https://proceedings.neurips.cc/paper_files/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf

Duan, K., Keerthi, S. S., & Poo, A. N. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing, 51, 41–59. https://doi.org/10.1016/S0925-2312(02)00601-X

Edwards, R. D., Magee, J., & Bassetti, W. C. (2018). Technical analysis of stock trends. CRC press.

Fama, E. F. (1965). The Behavior of Stock-Market Prices. The Journal of Business, 38(1), 34–105. JSTOR.

Gaucan, V. & others. (2011). How to use Fibonacci retracement to predict forex market. Journal of Knowledge Management, Economics and Information Technology, 1(2), 1–14.

Gelman, A., Goodrich, B., Gabry, J., & Vehtari, A. (2019). R-squared for Bayesian Regression Models. The American Statistician, 73(3), 307–309. https://doi.org/10.1080/00031305.2018.1549100

Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., & Weinberger, K. Q. (Eds.). (2014). Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. https://proceedings.neurips.cc/paper/2014

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. Expert Systems with Applications, 38(8), 10389–10397. https://doi.org/10.1016/j.eswa.2011.02.068

Han, J., Pei, J., & Tong, H. (2022). Data mining: Concepts and techniques. Morgan kaufmann.

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Systems with Applications, 129, 273–285. https://doi.org/10.1016/j.eswa.2019.03.029

Jadhav, S. D., & Channe, H. (2016). Comparative study of K-NN, naive Bayes and decision tree classification techniques. International Journal of Science and Research (IJSR), 5(1), 1842–1845.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 4, 237–285.

Khashei, M., & Bijari, M. (2010). An artificial neural network (p,d,q) model for timeseries forecasting. Expert Systems with Applications, 37(1), 479–489. https://doi.org/10.1016/j.eswa.2009.05.044

Kolchin, K. (2021). Global Equity Markets Primer (p. 59). SIFMA. https://www.sifma.org/resources/research/insights-global-equity-markets-primer/

Kumar, D., Sarangi, P. K., & Verma, R. (2022). A systematic review of stock market prediction using machine learning and statistical techniques. Materials Today: Proceedings, 49, 3187–3191. https://doi.org/10.1016/j.matpr.2020.11.399

Kumar, R. (2014). Magic of Fibonacci sequence in prediction of stock behavior. International Journal of Computer Applications, 93(11).

LeCun, Y., Bengio, Y., & others. (1995). Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks, 3361(10), 1995.

Malkiel, B. G. (2003). The Efficient Market Hypothesis and Its Critics. Journal of Economic Perspectives, 17(1), 59–82. https://doi.org/10.1257/089533003321164958

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). Machine learning: Algorithms and applications. Crc Press.

Murphy, J. J. (1999). Technical analysis of the financial markets: A comprehensive guide to trading methods and applications (First). New York Institute of Finance.

Neely, C., Weller, P., & Dittmar, R. (1997). Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach. The Journal of Financial and Quantitative Analysis, 32(4), 405–426.

Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W., & Wills, G. (2020). A survey on machine learning for stock price prediction: Algorithms and techniques. 2nd International Conference on Finance, Economics, Management and IT Business (05/05/20 - 06/05/20), 63–71. https://eprints.soton.ac.uk/437785/

Pan, J., Zhuang, Y., & Fong, S. J. (2016). The Impact of Data Normalization on Stock Market Prediction: Using SVM and Technical Indicators. International Conference on Soft Computing in Data Science.

Park, C.-H., & Irwin, S. H. (2007). WHAT DO WE KNOW ABOUT THE PROFITABILITY OF TECHNICAL ANALYSIS? Journal of Economic Surveys, 21(4), 786–826. https://doi.org/10.1111/j.1467-6419.2007.00519.x

Rish, I. & others. (2001). An empirical study of the naive Bayes classifier. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 3(22), 41–46.

S. Jones, J., & Kincaid, B. (2014). Can the correlation among Dow 30 stocks predict market declines? Managerial Finance, 40(1), 33–50. https://doi.org/10.1108/MF-08-2012-0179

Saini, A. (2023, October 12). Support Vector Machine (SVM): A Complete guide for beginners. https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 2(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Sarker, I. H., Kayes, A., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: An overview from machine learning perspective. Journal of Big Data, 7, 1–29.

Schölkopf, B., & Smola, A. J. (2001). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

Shynkevich, Y., McGinnity, T. M., Coleman, S. A., Belatreche, A., & Li, Y. (2017). Forecasting price movements using technical indicators: Investigating the impact of varying input window length. Neurocomputing, 264, 71–88. https://doi.org/10.1016/j.neucom.2016.11.095

Tay, F. E. H., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. Omega, 29(4), 309–317. https://doi.org/10.1016/S0305-0483(01)00026-3

Vapnik, V. (2000). The Nature of Statistical Learning Theory. Springer: New York.

Wilder, J. W. (1978). New Concepts in Technical Trading Systems. Trend Research. https://books.google.ch/books?id=WesJAQAAMAAJ

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent Neural Network Regularization. ArXiv E-Prints, arXiv:1409.2329. https://doi.org/10.48550/arXiv.1409.2329

Zhang, G. P., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. European Journal of Operational Research, 160(2), 501–514. https://doi.org/10.1016/j.ejor.2003.08.037

Zhang, K., Zhong, G., Dong, J., Wang, S., & Wang, Y. (2019). Stock Market Prediction Based on Generative Adversarial Network. Procedia Computer Science, 147, 400–406. https://doi.org/10.1016/j.procs.2019.01.256

Zhu, Y. (2020). Stock price prediction using the RNN model. Journal of Physics: Conference Series, 1650(3), 032103. https://doi.org/10.1088/1742-6596/1650/3/032103

Appendices

Table of model parameters after grid search

Time shift 5

Model	Shift	C	Epsilon	Gamma
MMM	5	100	0.01	auto
AXP	5	100	0.1	auto
AMGN	5	100	0.01	auto
AAPL	5	100	0.1	auto
BA	5	100	0.01	auto
CAT	5	100	0.1	auto
CVX	5	100	0.01	auto
CSCO	5	10	0.01	auto
KO	5	100	0.01	auto
DIS	5	100	0.1	auto
GS	5	100	0.1	auto
HD	5	100	0.01	auto
HON	5	100	0.01	auto
IBM	5	10	0.01	auto
INTC	5	10	0.1	auto
JNJ	5	100	0.01	auto
JPM	5	100	0.1	auto
MCD	5	100	0.01	auto
MRK	5	100	0.01	auto
MSFT	5	100	0.1	auto
NKE	5	100	0.01	auto
PG	5	100	0.1	auto
TRV	5	100	0.1	auto
UNH	5	100	0.1	auto
VZ	5	100	0.01	auto
WBA	5	10	0.01	auto
WMT	5	100	0.1	auto

Table 5: Best Parameters for Time Shift 5

Time shift 20

Model	Shift	C	Epsilon	Gamma
MMM	20	10	0.1	auto
AXP	20	10	0.01	auto
AMGN	20	10	0.01	auto
AAPL	20	100	0.1	auto
BA	20	100	0.01	auto
CAT	20	10	0.01	auto
CVX	20	100	0.01	auto
CSCO	20	1	0.01	auto
KO	20	10	0.01	auto
DIS	20	10	0.01	auto
GS	20	100	0.1	auto
HD	20	100	0.01	auto
HON	20	10	0.1	auto
IBM	20	10	0.1	auto
INTC	20	100	0.01	auto
JNJ	20	10	0.01	auto
JPM	20	10	0.1	auto
MCD	20	100	0.01	auto
MRK	20	100	0.01	auto
MSFT	20	10	0.1	auto
NKE	20	100	0.1	auto
PG	20	10	0.1	auto
TRV	20	10	0.01	auto
UNH	20	100	0.1	auto
VZ	20	100	0.1	auto
WBA	20	0.1	0.01	auto
WMT	20	10	0.1	auto

Table 6:Best Parameters for Time Shift 20

Time shift 90

Model	Shift	C	Epsilon	Gamma
MMM	90	1	0.01	auto
AXP	90	1	0.01	auto
AMGN	90	100	0.1	auto
AAPL	90	10	0.1	auto
BA	90	100	0.01	auto
CAT	90	1	0.1	auto
CVX	90	10	0.01	auto
CSCO	90	1	0.1	auto
KO	90	1	0.1	auto
DIS	90	1	0.01	auto
GS	90	10	0.1	auto
HD	90	10	0.01	auto
HON	90	10	0.1	auto
IBM	90	1	0.01	auto
INTC	90	1	0.01	auto
JNJ	90	1	0.1	auto
JPM	90	10	0.1	auto
MCD	90	100	0.1	auto
MRK	90	10	0.01	auto
MSFT	90	10	0.01	auto
NKE	90	10	0.01	auto
PG	90	100	0.1	0.1
TRV	90	10	0.01	auto
UNH	90	10	0.1	auto
VZ	90	0.1	0.1	auto
WBA	90	0.1	0.1	auto
WMT	90	1	0.01	auto

Table 7: Best parameters for Time Shift 90

Table of model performances

	Time shift 5			*Time shift 20*			*Time shift 90*		
Stock	**MAPE**	**RMSE**	**Model Score**	**MAPE**	**RMSE**	**Model Score**	**MAPE**	**RMSE**	**Model Score**
MMM	0.0192	3.3119	0.9958	0.0395	6.7513	0.9824	0.0855	13.9158	0.9240
AXP	0.0249	2.4596	0.9955	0.0517	5.2422	0.9795	0.1261	10.9494	0.9150
AMGN	0.0237	3.8291	0.9966	0.0489	7.5035	0.9872	0.0756	10.5202	0.9757
AAPL	0.0475	1.3161	0.9990	0.0786	2.4427	0.9965	0.2811	4.5644	0.9889
BA	0.0288	6.0517	0.9960	0.0577	14.4703	0.9771	0.1281	29.2597	0.9053
CAT	0.0288	3.5715	0.9946	0.0626	7.5782	0.9758	0.1578	16.5056	0.8879
CVX	0.0215	2.7233	0.9913	0.0420	5.4806	0.9647	0.0821	11.1235	0.8601
CSCO	0.0272	1.0120	0.9929	0.0588	2.1240	0.9687	0.1099	3.7148	0.9048
KO	0.0147	0.7443	0.9958	0.0296	1.5362	0.9824	0.0608	2.8276	0.9433
DIS	0.0235	2.3515	0.9975	0.0500	5.2471	0.9874	0.1205	13.0287	0.9217
GS	0.0276	6.4345	0.9918	0.0513	11.9960	0.9718	0.1338	29.5554	0.8348
HD	0.0248	3.3917	0.9986	0.0453	6.7200	0.9947	0.1006	12.5667	0.9824
HON	0.0230	2.5652	0.9980	0.0475	5.6214	0.9904	0.0974	10.5080	0.9673
IBM	0.0220	3.7830	0.9889	0.0420	7.1237	0.9604	0.0906	13.8207	0.8477
INTC	0.0293	1.2866	0.9900	0.0491	2.0784	0.9738	0.1204	4.8193	0.8591
JNJ	0.0144	1.9627	0.9972	0.0297	3.8825	0.9893	0.0632	7.1419	0.9652
JPM	0.0268	2.1962	0.9963	0.0525	4.4207	0.9852	0.1086	9.0667	0.9387
MCD	0.0181	2.4893	0.9986	0.0331	5.0062	0.9944	0.0676	8.1981	0.9857
MRK	0.0206	1.4420	0.9924	0.0389	2.6703	0.9743	0.0883	5.5309	0.8973
MSFT	0.0234	2.3832	0.9990	0.0498	5.7994	0.9939	0.0827	5.8224	0.9944
NKE	0.0230	1.6452	0.9983	0.0436	3.1683	0.9939	0.0900	6.2902	0.9762
PG	0.0138	1.5077	0.9971	0.0285	3.1759	0.9871	0.0447	5.1783	0.9671
TRV	0.0198	2.1328	0.9973	0.0405	4.5150	0.9881	0.0744	8.0312	0.9640
UNH	0.0272	4.7128	0.9985	0.0486	7.4982	0.9963	0.1114	13.6310	0.9892
VZ	0.0172	0.9815	0.9908	0.0318	1.8078	0.9689	0.0788	3.9471	0.8516
WBA	0.0252	1.7430	0.9903	0.0733	4.9271	0.9226	0.1224	7.5362	0.8194
WMT	0.0173	1.8228	0.9960	0.0338	3.3304	0.9869	0.0718	7.0993	0.9432

Table 8: MAPE, RMSE and Model Scores for Price Prediction 5 days, 20 days and 90 days into the future