LA RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique
المدرسة الوطنية المتعددة التقنيات
Département d'Automatique
قسم الهندسة الآلية
Process Control Laboratory

Thesis for the end-of-study project to obtain the State Engineer's Degree in Automation
Specialty: Automation

Presented by:
DERDOUR Abdeslem

Title:

Metaheuristics Optimization of Financial Trading Strategies for Single Asset Trading

Defended on September 21, 2023, before the jury composed of:

Pr. M. TADJINE	ENP	, Supervisor
Pr. EM. BERKOUK	ENP	, President
Dr. M. CHAKIR	ENP	, Examiner

ENP 2023

LA RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique
المدرسة الوطنية المتعددة التقنيات
Département d'Automatique
قسم الهندسة الآلية
Process Control Laboratory

Thesis for the end-of-study project to obtain the State Engineer's Degree in Automation
Specialty: Automation

Presented by:
DERDOUR Abdeslem

Title:

Metaheuristics Optimization of Financial Trading Strategies for Single Asset Trading

Defended on September 21, 2023, before the jury composed of:

Pr. M. TADJINE	ENP	, Supervisor
Pr. EM. BERKOUK	ENP	, President
Dr. M. CHAKIR	ENP	, Examiner

ENP 2023

LA RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique

المدرسة الوطنية المتعددة التقنيات

Département d'Automatique

قسم الهندسة الآلية

Laboratoire de Commande des Processus

Mémoire de projet de fin d'études pour l'obtention du diplôme d'ingénieur d'État en automatique

Spécialité : Automatique

Présenté par :

DERDOUR Abdeslem

Intitulé :

Optimisation par métaheuristiques des stratégies de trading financier pour le trading d'actifs uniques

Soutenu en 21/09/2023, devant le jury composé de :

Pr. M. TADJINE	ENP	, Superviseur
Pr. EM. BERKOUK	ENP	, Président
Dr. M. CHAKIR	ENP	, Examinateur

ENP 2023

ملخص :
تركّزت الدراسة على تحسين أنظمة التداول الفني باستخدام تقنيات الميتاهيوريستيك، مثل تقنية تحسين تجمع الجسيمات وتقنيات تطوير الأهداف المتعددة. تشير النتائج إلى أن هذه الأساليب تعزز أداء وقوة الأنظمة التجارية بشكل كبير. تبين أن الطرق التقليدية للتدريب عرضة لخطر الضبط الزائد، وتم التخفيف من هذه المخاوف من خلال تقنية تحسين السير للأمام. تم التأكيد على أهمية اختيار وظيفة الهدف في تحسين قوة النظام. يُوصى بعناية بأن يختار الممارسون وظيفة الهدف والطريقة المثلى لتصميم وتقييم أنظمة التداول الفني .
كلمات مفتاحية : أنظمة التداول الفني ذات التردد المنخفض، تحسين السير للأمام، العملات الرقمية ، مُحسِنْ تجمع الجسيمات، خوارزمية تطورية متعددة الأهداف تعتمد على التحليل .

Résumé :
L’étude s’est concentrée sur l’optimisation des systèmes de trading techniques en utilisant des techniques métaheuristiques, telles que l’optimisation par essaim de particules et les algorithmes évolutifs multi-objectifs. Les résultats indiquent que ces méthodes améliorent significativement les performances et la robustesse des systèmes de trading. Il a été constaté que les approches de formation traditionnelles étaient susceptibles de surajuster, une préoccupation qui a été atténuée grâce à l’optimisation de la marche en avant. Le choix de la fonction objectif a été mis en avant comme crucial pour améliorer la robustesse du système. Il est recommandé aux praticiens de sélectionner soigneusement la fonction objectif et la méthode d’optimisation pour la conception et l’évaluation des systèmes de trading techniques.
Mots-clés : Systèmes de trading technique à basse fréquence (LFTTS), Optimisation de la marche en avant (WFO), Crypto-monnaies, Optimiseur de l’essaim de particules (PSO), Algorithme évolutif multi-objectifs basé sur la décomposition (MOEA/D).

Abstract :
The study focused on optimizing technical trading systems using metaheuristic techniques, such as Particle Swarm Optimization and Multi-Objective Evolutionary Algorithms. The results indicate that these methods significantly enhance the performance and robustness of the trading systems. Traditional training approaches were found to be susceptible to overfitting, a concern that was mitigated through WalkForward Optimization. The choice of the objective function was highlighted as crucial in improving system robustness. It is recommended that practitioners carefully select the objective function and optimization method for designing and evaluating technical trading systems.
Keywords: Low Frequency Technical Trading Systems (LFTTS), Walk Forward Optimization (WFO), Crypto-currencies, Particle Swarm Optimizer (PSO), Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D).

Acknowledgements

" I would like to express my deepest gratitude to my advisor, Professor TADJINE Mohamed, for his unwavering guidance, invaluable insights, and continuous support throughout the duration of this research. His expertise and dedication have been instrumental in shaping the direction of this thesis.

I want to express my heartfelt thanks to the members of the thesis jury for their time, expertise, and valuable insights during the evaluation of my master's thesis. Your feedback and support have been greatly appreciated.

I am also indebted to the faculty members of Control Systems for their knowledge-sharing and encouragement during my academic journey. Their lectures, discussions, and feedback have contributed significantly to my growth as a researcher.

I extend my heartfelt appreciation to my friends and colleagues who provided meaningful discussions, brainstorming sessions, and moral support. Your camaraderie and encouragement made the research process a truly collaborative and enjoyable experience.

I am thankful to my family for their unwavering encouragement, love, and belief in my abilities. Their support has been my constant motivation to strive for excellence.

Last but not least, I want to thank all the individuals who volunteered their time and expertise for interviews, surveys, and other data collection methods. Your contributions have enriched the quality of this study. This thesis would not have been possible without the collective efforts of all those who believed in me and contributed to my growth. Thank you all for being an integral part of this remarkable journey."

Contents

List of Figures

List of Tables

Acronyms

RSI Relative Strength Index

MACD Moving Average Convergence Divergence

LFTTS Low Frequency Technical Trading System

BB Bollinger Bands

CCI Commodity Channel Index

SMA Simple Moving Average

EMA Exponential Moving Average

AT Algorithmic Trading

OHLCV Open, High, Low, Close, and Volume

CAPM Capital Asset Pricing Model

ADX Average Directional Index

+DI Positive Directional Indicator

-DI Negative Directional Indicator

MDD Maximum Drawdown

PSO Particle Swarm Optimization

MOEA/D Multi-Objective Evolutionary Algorithm based on Decomposition

WFO WalkForward Optimization

BTC Bitcoin

ETH Ethereum

NEO NEO Cryptocurrency

USDT Tether (USD Tether)

API Application Programming Interface

TA Technical Analysis

BBW Bollinger Band Width

TTS Trading Technical System

SL Stop-Loss

General Introduction

General Introduction

The financial markets are a dynamic and ever-evolving ecosystem, where fortunes are made and lost in the blink of an eye. Successful trading in these complex environments demands more than just intuition and expertise; it requires the strategic use of cutting-edge tools and techniques. In recent years, the application of metaheuristics has emerged as a game-changing approach in the quest to optimize financial trading strategies.

Metaheuristics represent a class of algorithms inspired by natural processes, such as evolution, swarm behavior, and collective intelligence. These algorithms offer a fresh perspective on addressing the intricate challenges faced by traders and investors in modern financial markets. By harnessing the computational power and adaptability of metaheuristics, financial professionals can enhance their decision-making processes, manage risk more effectively, and ultimately strive for greater profitability.

In this exploration of the optimization of financial trading strategies using metaheuristics, we embark on a journey to uncover the transformative potential of these innovative methodologies. Our mission is to shed light on how metaheuristics, such as Genetic Algorithms, Particle Swarm Optimization, and Simulated Annealing, are reshaping the landscape of financial trading.

Throughout this journey, we will delve into the fundamental principles that underpin metaheuristics and discover how they can be applied to design and refine trading strategies. We will explore real-world case studies and empirical evidence showcasing the impact of metaheuristics on trading performance, risk management, and portfolio optimization.

Moreover, we will discuss the versatility of metaheuristics, highlighting their applicability across various financial instruments, including stocks, commodities, currencies, and cryptocurrencies. Whether you are a seasoned trader, an institutional investor, or an aspiring financial professional, this exploration will illuminate the potential benefits of integrating metaheuristics into your trading toolkit.

As we delve deeper into the world of optimizing financial trading strategies using metaheuristics, we invite you to join us on this exciting journey of discovery. Together, we will explore how these computational techniques are redefining the way we approach financial markets, offering new avenues for innovation, adaptability, and success in the fast-paced world of trading.

Chapter 1

Introduction

1.1 Introduction to Algorithmic Trading

In recent years, there has been significant interest in the use of metaheuristics algorithms in algorithmic trading (AT) among both finance and soft-computing researchers. There is a substantial body of published research on this topic, as evidenced by studies such as those conducted by Ponsich et al. (2013), Hu et al. (2015),. AT refers to computer programs that automate one or more stages of the trading process. These systems currently handle a significant portion of all stocks traded in the United States and the European Union and are a major source of innovation in computing and analytics, particularly with regard to machine learning and grid/GPU computing (Nuti et al., 2011; Hendershott and Riordan, 2013).

Algorithmic trading systems are designed to capitalize on momentary anomalies in market prices, take advantage of statistical patterns within or across financial markets, optimally execute orders, conceal traders' intentions, or detect and exploit rivals' strategies. Ultimately, these systems are driven by profits, whether in the form of cost savings, client commissions, or proprietary trading. The key feature of algorithmic trading systems is the sophistication of their analysis and decision-making capabilities (Nuti et al., 2011). They are used in highly liquid markets, including equities, futures, derivatives, bonds, and foreign exchange. Algorithmic trading can be applied to automate any stage of the trading process, resulting in a wide range of systems. For instance, in trade-execution programs, the algorithm may determine aspects such as which market to send the order to, the timing, price, and even the order's quantity splits (Nuti et al., 2011).

One of the significant benefits of algorithmic trading is that strategies can be tested on historical data. The ability to simulate a strategy is a major advantage of algorithmic trading. Back-testing provides insight into how well a strategy would have performed in the past. Although back-tested performance does not guarantee future results, it can be valuable when evaluating potential strategies. Back-tested results can be used to filter out strategies that do not suit the required investment style or are unlikely to meet risk/return performance goals.

1.2 Financial Time Series

Financial time series are initially constructed by capturing the details of each transaction, including the price, volume, and a timestamp. This information is asynchronously recorded at each tick. By resampling this tick data into equal time intervals, we obtain OHLCV (Open, High, Low, Close, Volume) data, where Open and Close represent the first and last prices within the time interval, and High and Low denote the highest and lowest prices recorded during that time. Volume indicates the total number of financial instruments exchanged between buyers and sellers. We can consider the price series for a particular market as a collection of vectors, consisting of Open, High, Low, Close, and Volume, sampled at regular time intervals τ (such as minutes, hours, days, etc.) where the Timestamp is the candle's reference.

Timestamp	Open	High	Low	Close	Volume
1575158460000	7554.2	7554.2	7546.14	7546.14	0.05786579
1575158520000	7546.02	7546.1	7546.02	7546.02	0.02604071
1575158580000	7546.02	7553.18	7545.09	7545.09	2.25604269
1575158640000	7543.63	7543.63	7543.63	7543.63	0.00778743
1575158700000	7543.5	7543.5	7532.83	7532.83	0.06064193
1575158760000	7535.92	7541.59	7535.92	7535.93	4.02772332
1575158820000	7543.05	7543.05	7543.05	7543.05	0.00519001
1575158880000	7536.15	7536.15	7536.15	7536.15	0.10575346
1575158940000	7541.85	7541.85	7532.67	7532.67	1.31539777
1575159000000	7529.6	7533.67	7529.6	7529.8	0.18649237
1575159060000	7527.43	7537.32	7527.43	7536.49	0.22423009

Figure 1.1: OHLCV data of BTCUSDT from Investopidia.com

1.3 Algorithmic Trading Components

Algorithmic trading strategies typically involve several components, including pre-trade analysis, trading signal generation, trade execution, and post-trade analysis.

1.3.1 Pre-Trade Analysis

The pre-trade analysis is a crucial process that encompasses three key components, each playing a vital role in ensuring informed and strategic decision-making within algorithmic trading systems. The first component is the alpha model, specifically designed to predict and gauge the future behavior of the financial instruments that the algorithmic system is intended to trade. By leveraging various data inputs, market trends, and historical patterns, the alpha model aims to generate insights into potential price movements, market dynamics, and investment opportunities. This predictive model serves as a foundation for identifying profitable trades and optimizing trading strategies.

The second component of the pre-trade analysis is the risk model, which serves as a critical tool for assessing and managing the levels of exposure and risk associated with the financial instruments being traded. Through sophisticated risk management techniques, this model evaluates factors such as volatility, liquidity, counterparty risk, and portfolio diversification to quantify the potential downside and upside risks of a particular trade. By identifying and understanding these risks, traders can make informed decisions to protect capital, optimize returns, and align their trading activities with risk tolerance and investment objectives.

The third component is the transaction cost model, an integral part of the pre-trade analysis that focuses on calculating the potential costs associated with trading the financial instruments. This model takes into account various factors that impact trading costs, including market liquidity, bid-ask spreads, order execution speed, and market impact. By quantifying these costs, traders can assess the profitability and efficiency of their trading strategies, optimize trade execution, and minimize the impact of transaction costs on overall portfolio performance.

Collectively, these three components of the pre-trade analysis provide a comprehensive framework for traders and investment professionals to make informed decisions in algorithmic trading. By leveraging the insights derived from the alpha model, managing risk through the risk model, and considering transaction costs through the transaction cost model, traders can enhance their trading strategies, mitigate risks, and maximize returns. This holistic approach to pre-trade analysis empowers market participants to navigate the complexities of financial markets, adapt to changing market conditions, and ultimately achieve their investment goals.

1.3.2 Trading Signal Generation

The process of generating trading signals and conducting pre-trade analysis often overlap and share similarities, but there are key differences between them. While pre-trade analysis is focused on analyzing financial data or news to identify potential trading opportunities, trading signal generation involves the use of algorithms to generate specific trade recommendations that include details such as price, quantity, and risk management strategies like stop-loss values. In contrast, pre-trade analysis recommendations are often less specific and intentionally vague, as they may be further refined and augmented by the trading signal generation process in more complex systems.

1.3.3 Trade Execution

During the trade execution phase, the order is sent to the appropriate trading venues. In cases where the order is excessively large, the system divides it into smaller orders and submits them over time, aiming to reduce market impact. Additionally, orders can be sent to various markets, including crossing markets or dark pools, where the current order book is not publicly displayed.

1.3.4 Post-Trade Analysis

Post-trade analysis involves analyzing the performance of the trading strategy after the trades have been executed. This can involve analyzing factors such as the total return, the Sharpe ratio, and the maximum drawdown.

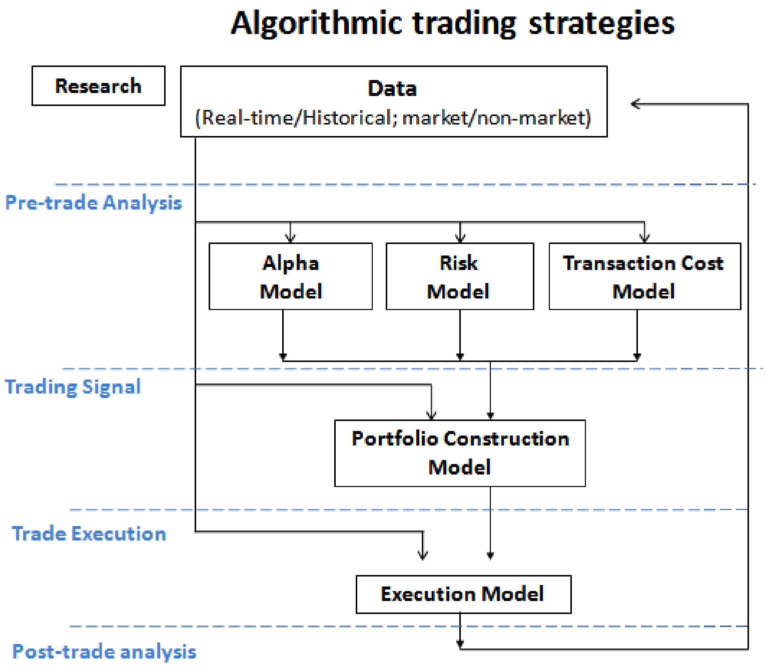


Figure 1.2: Algorithmic Trading Components

1.4 Pre-Trade Analysis Methodologies

There are several methodologies that can be used for pre-trade analysis, including fundamental analysis, quantitative analysis, and technical analysis.

1.4.1 Fundamental Analysis

Fundamental analysis involves analyzing financial and economic data to determine the intrinsic value of an asset. This comprehensive approach includes examining financial statements to assess a company's financial health and stability. It also encompasses analyzing industry trends to identify opportunities or risks within a specific sector. Additionally, fundamental analysis takes into account macroeconomic factors such as interest rates, inflation, and government policies to understand the broader economic environment. By considering all these factors, investors can make informed decisions aligned with their long-term financial goals.

1.4.2 Quantitative Analysis

Quantitative analysis involves using statistical and mathematical models to analyze financial data. This can involve analyzing asset returns, volatility, and other factors.

Asset Returns

Asset returns, representing the percentage change in an asset's price over a specific period, are essential in quantitative trading models. These models rely on mathematical algorithms and statistical analysis to analyze historical return data, identify patterns, and predict future price movements. By incorporating asset returns, these models enable the development of systematic strategies that can be backtested and validated. Asset returns are also crucial for calculating risk metrics such as volatility and Sharpe ratio, aiding in risk assessment and portfolio management. Ultimately, asset returns serve as a fundamental input in quantitative trading, shaping strategy development, risk evaluation, and performance analysis in the dynamic realm of financial markets.

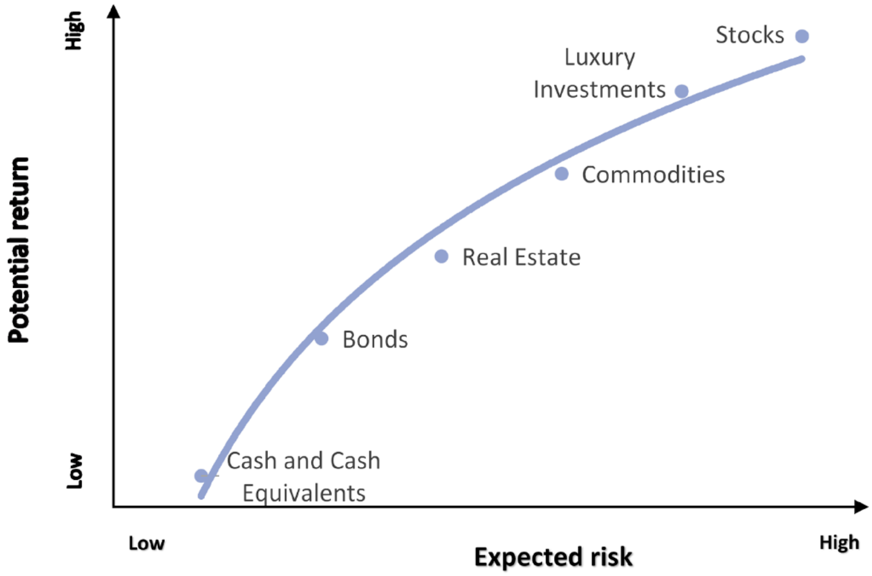


Figure 1.3: Asset Classes Graph from Investopidia.com

Volatility

Volatility, representing the degree of price variation in an asset over time, is a key measure of risk in trading strategies. It quantifies the potential price fluctuations and uncertainty associated with an asset, guiding traders and investors in assessing risk-reward profiles. Higher volatility implies greater risk and potential for both profits and losses. By incorporating volatility metrics into trading strategies, investors can adjust position sizes, set appropriate stop-loss levels, and implement risk management techniques. Volatility also impacts the pricing and profitability of derivative instruments like options. Understanding and effectively managing volatility is crucial for making informed decisions, identifying opportunities, and optimizing trading strategies in dynamic financial markets.

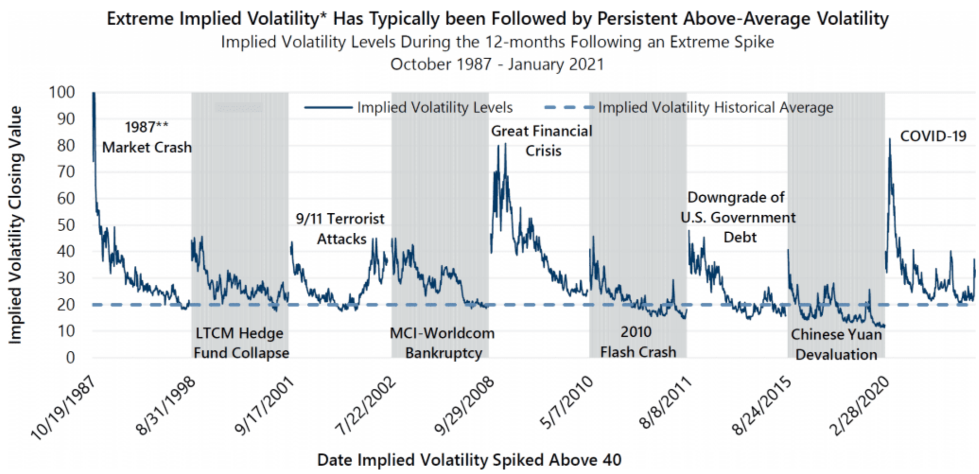


Figure 1.4: Implied Volatility Graph From Investopidia.com

Capital Asset Pricing Model (CAPM)

The Capital Asset Pricing Model (CAPM) is a widely used financial framework that helps investors determine the expected return on an investment based on its risk level. CAPM suggests that the expected return on an asset is a function of the risk-free rate, the asset's beta, and the market risk premium. The risk-free rate represents the return on a risk-free investment, while beta measures an asset's sensitivity to market movements. The market risk premium represents the excess return investors demand for holding a risky asset over the risk-free rate. By using CAPM, investors can estimate the appropriate expected return for an asset, facilitating investment decision-making and comparison with required returns. However, CAPM assumes rational and risk-averse investors and efficient markets, and relies on historical data for estimating betas. Nevertheless, CAPM remains a valuable tool for asset pricing, portfolio management, and understanding the risk-return relationship in financial markets.

$$\text{Expected Return} = \text{Risk-Free Rate} + \beta \times (\text{Market Risk Premium}) \tag{1.1}$$

1.4.3 Technical Analysis

Simple Moving Averages

Simple Moving Averages (SMA) are widely used trend-following indicators that smooth out price data over a specified period.Moving averages, as a statistical concept, have evolved over time with contributions from various individuals in different fields, including Charles Dow in the realm of financial analysis. The formula for calculating SMA is:

$$\text{SMA}(n) = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{1.2}$$

Figure 1.5: Simple Moving Average

Exponential Moving Averages

Exponential Moving Averages (EMA) give more weight to recent price data, making them more responsive to price changes compared to SMAs.The Exponential Moving Average (EMA) is a mathematical concept developed and refined over time by various mathematicians and researchers, commonly used in finance for its responsiveness to recent data. The formula for calculating EMA is:

$$\text{EMA} = (1 - \alpha) \times \text{Previous EMA} + \alpha \times \text{Current Price} \tag{1.3}$$

Such that, α represents the smoothing factor, Previous EMA is the previous EMA value, and Current Price is the current price value.

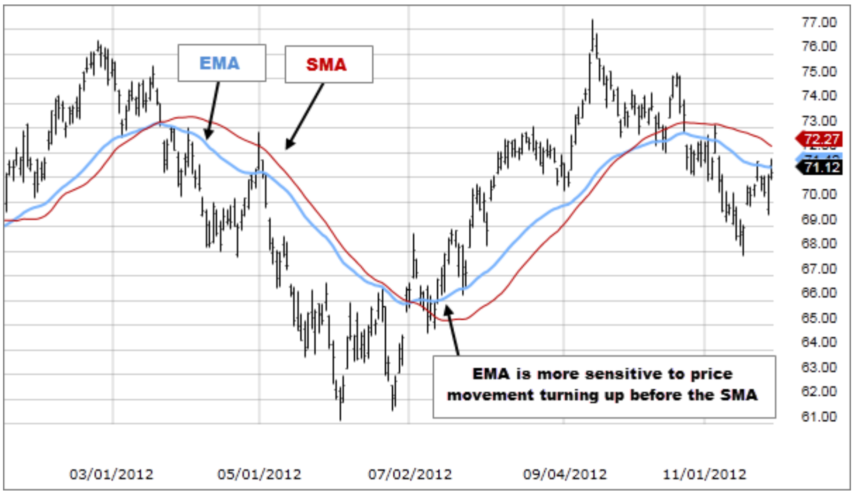


Figure 1.6: Exponential Moving Average

Bollinger Bands

Bollinger Bands consist of a middle band (SMA or EMA) and an upper and lower band that represent volatility levels around the middle band.John Bollinger, a financial analyst and author, created Bollinger Bands, a widely used technical indicator in the field of financial analysis, to help traders assess price volatility and potential price reversal points. The formulas for the upper and lower Bollinger Bands are:

$$\text{Upper Band} = \text{SMA} + (k \times \sigma) \tag{1.4}$$

$$\text{Lower Band} = \text{SMA} - (k \times \sigma) \tag{1.5}$$

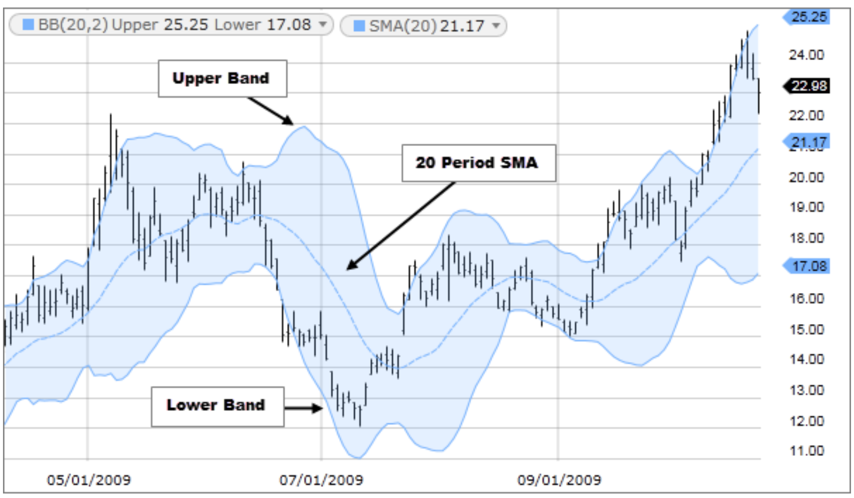


Figure 1.7: Bollinger Bands Indicator

Moving Average Convergence Divergence (MACD)

MACD is a trend-following momentum indicator that shows the relationship between two moving averages of an asset's price.Gerald Appel, a renowned technical analyst and author, created the Moving Average Convergence Divergence (MACD) indicator, a popular tool for analyzing price trends and momentum in financial markets. The formula for calculating MACD is:

$$\text{MACD} = \text{Short-term EMA} - \text{Long-term EMA} \tag{1.6}$$

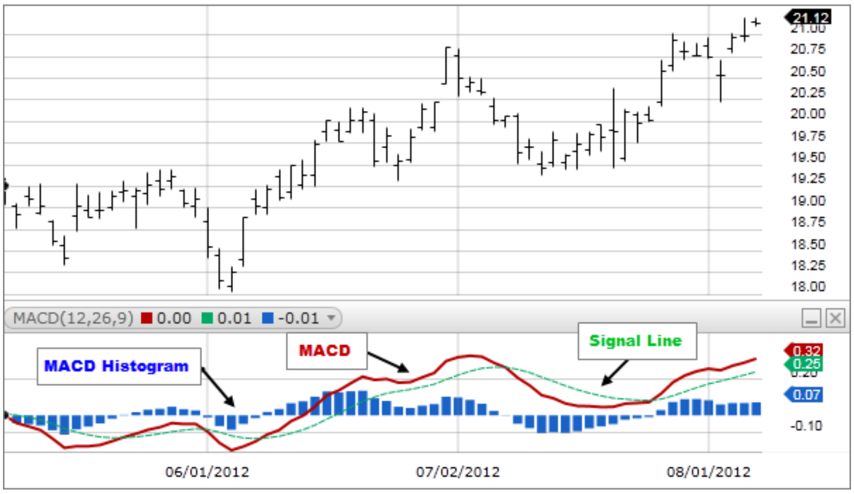


Figure 1.8: MACD Indicator

Relative Strength Index (RSI)

RSI is a momentum oscillator that measures the speed and change of price movements.J. Welles Wilder Jr., an American mechanical engineer and technical analyst, created the Relative Strength Index (RSI), a widely used momentum oscillator in financial analysis for assessing the speed and change of price movements. The formula for calculating RSI is:

$$\text{RSI} = 100 - \left(\frac{100}{1 + \frac{\text{Average Gain}}{\text{Average Loss}}} \right) \tag{1.7}$$

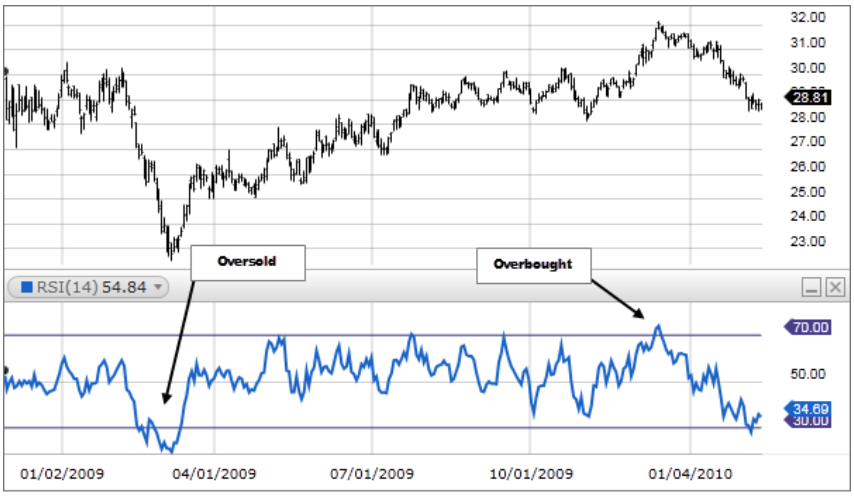


Figure 1.9: RSI Indicator

Commodity Channel Index (CCI)

CCI is a versatile indicator used to identify overbought and oversold levels in an asset's price.Donald Lambert, an American commodities trader and author, created the Commodity Channel Index (CCI), a popular technical indicator used to assess the cyclical trends and overbought/oversold conditions in financial markets. The formula for calculating CCI is:

$$\text{CCI} = \frac{\text{Typical Price} - \text{SMA of Typical Price}}{0.015 \times \text{Mean Deviation}} \tag{1.8}$$

Figure 1.10: CCI Indicator

Average Directional Index (ADX)

ADX is a technical indicator used to measure the strength of a trend, regardless of its direction.The Average Directional Index (ADX) was developed by J. Welles Wilder Jr., an American mechanical engineer and technical analyst, to measure the strength of price trends in financial markets. The formula for calculating ADX involves the calculations of the Positive Directional Indicator (+DI) and the Negative Directional Indicator (-DI), which are then used to calculate the ADX value.

The Positive Directional Indicator (+DI) is calculated as:

$$+DI = \left(\frac{\text{Smoothed Positive Directional Movement}}{\text{Average True Range}}\right) \times 100 \tag{1.9}$$

The Negative Directional Indicator (-DI) is calculated as:

$$-DI = \left(\frac{\text{Smoothed Negative Directional Movement}}{\text{Average True Range}}\right) \times 100 \tag{1.10}$$

The Average Directional Index (ADX) is then calculated as a smoothed average of the Absolute Directional Index (ADI):

$$\text{ADX} = \left(\frac{\text{Smoothed ADI}}{\text{Smoothing Period}}\right) \tag{1.11}$$

Note that the calculations involve the use of the True Range, which is a measure of market volatility. The smoothing periods and methods used may vary depending on the implementation.

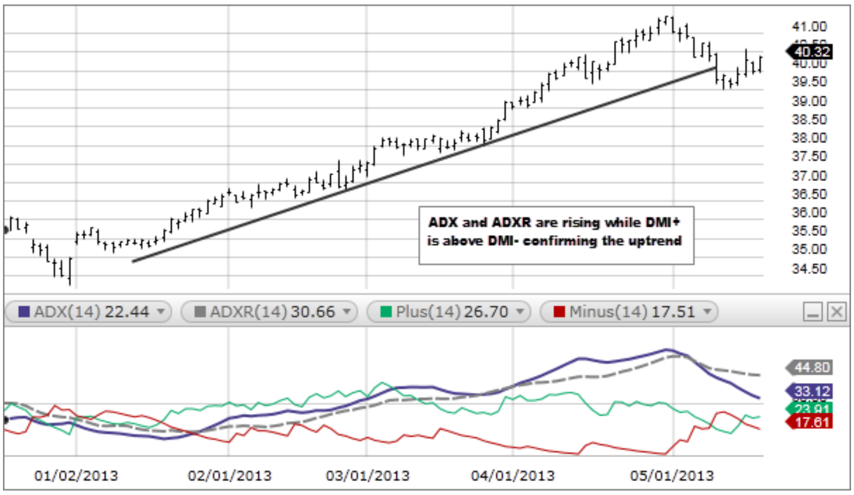


Figure 1.11: ADX Indicator

1.5 Trading strategy performance

1.5.1 Total Return

The total return measures the overall profit or loss generated by a trading strategy. It is calculated as the percentage change in the value of the investment over a given period. The formula for total return is:

$$\text{Total Return} = \frac{\text{Final Value} - \text{Initial Value}}{\text{Initial Value}} \times 100 \tag{1.12}$$

1.5.2 Sharpe Ratio

The Sharpe ratio is a measure of risk-adjusted return.The Sharpe Ratio was developed by William F. Sharpe, a Nobel laureate and economist, as a measure of the risk-adjusted return of an investment or portfolio in finance. It takes into account both the return and the volatility of the investment. The formula for the Sharpe ratio is:

$$\text{Sharpe Ratio} = \frac{\text{Average Return} - \text{Risk-Free Rate}}{\text{Standard Deviation of Return}} \tag{1.13}$$

1.5.3 Sortino Ratio

The Sortino Ratio is a risk-adjusted performance measure that focuses on the downside risk of an investment or portfolio.The Sortino Ratio was created by Dr. Frank A. Sortino, a finance professor and expert in the field of portfolio management, as a risk-adjusted performance measure that focuses on downside risk in investment portfolios. It is an extension of the Sharpe Ratio, but instead of considering the total volatility of returns, the Sortino Ratio takes into account only the downside volatility – that is, the volatility of returns below a certain target or minimum acceptable return.

The formula for the Sortino Ratio is:

$$\text{Sortino Ratio} = \frac{R - T}{D} \tag{1.14}$$

Where:

R is the average annualized return of the investment or portfolio.

T is the target or minimum acceptable return (usually 0 or the risk-free rate).

D is the downside deviation of the investment or portfolio's returns.

The Sortino Ratio measures the risk-adjusted return of an investment by considering only the downside volatility (volatility of returns below the target return) rather than the total volatility. A higher Sortino Ratio indicates a better risk-adjusted performance.

1.5.4 Calmar Ratio

The Calmar Ratio is a financial metric that assesses the risk-adjusted performance of an investment strategy or portfolio by comparing its average annualized rate of return (R) to its maximum drawdown (MDD).The Calmar Ratio, also known as the Drawdown Ratio, was not created by a single individual but is named after Terry W. Young, a trader and portfolio manager, who popularized its use in the field of risk management and investing. The ratio provides insights into how well an investment has performed in relation to the risks it has taken. It is especially useful for evaluating strategies with a focus on downside risk and volatility.

$$R = \frac{\text{Ending Portfolio Value}}{\text{Starting Portfolio Value}} - 1 \tag{1.15}$$

$$MDD = \max_{t \in [T_1, T_2]} \left(1 - \frac{V(t)}{V_{\text{peak}}} \right) \tag{1.16}$$

1.5.5 R-Square of Logarithmic Returns

In financial markets, the coefficient of determination (R^2) plays a pivotal role in evaluating the effectiveness of regression models employed to analyze and predict the behavior of financial assets based on their logarithmic returns.The R-Square of Logarithmic Returns is a statistical measure used in finance, but it doesn't have a single identifiable creator; it is derived from statistical techniques and is employed to assess the goodness of fit for a regression model in financial analysis. Logarithmic returns, which quantify the percentage change in an asset's value over time, are central to understanding the dynamic nature of financial markets. The R^2 metric offers a concise measure of how well a regression model captures the variability in these returns. The formula for R^2 elegantly encapsulates this concept:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{1.17}$$

In this formula, y_i represents the observed logarithmic returns at various time intervals, $\hat{y}_i$ signifies the predicted logarithmic returns derived from the regression model, and $\bar{y}$ denotes the mean of the observed logarithmic returns. The numerator of the formula computes the sum of squared differences between observed and predicted returns, while the denominator captures the total variability by summing squared deviations from the mean returns. Consequently, R^2 quantifies the proportion of variance in the observed logarithmic returns that the model can account for. A high R^2 value suggests that the regression model successfully explains a substantial portion of the return variability, indicating its potential efficacy in predicting asset movements. However, it's crucial to interpret R^2 judiciously, considering its limitations and the broader context of financial analysis. While R^2 offers a quantitative assessment of a model's fit, its interpretation should be complemented by domain expertise and an understanding of the economic significance of model outcomes.

In the context of investing, a substantial R-squared value, ranging from 85% to 100%, signifies a strong alignment between the performance of a stock or fund and the movements of the associated index. Conversely, a fund with a lower R-squared value, typically around 70% or less, suggests that its performance isn't consistently synchronized with the index fluctuations. A heightened R-squared value tends to yield a more informative beta metric. For instance, consider a scenario where a stock or fund boasts an R-squared value near 100%, yet its beta remains below 1. This situation implies that the investment is likely to offer enhanced risk-adjusted returns, despite its deviation from the index's behavior.

1.5.6 Maximum Drawdown

The Maximum Drawdown (MDD) is a metric used to quantify the largest percentage decline in the value of an investment portfolio over a certain time horizon. It is calculated as the maximum relative decrease between the portfolio's value at its peak V_{peak} and its subsequent minimum value V_{trough} during that period, expressed as a percentage:

$$MDD = \max_{t \in [T_1, T_2]} \left(1 - \frac{V(t)}{V_{\text{peak}}}\right) \tag{1.18}$$

where T_1 and T_2 define the time interval, $V(t)$ represents the value of the portfolio at time t, and the maximum is taken over all possible time intervals within $[T_1, T_2]$. This abstraction captures the most significant decline experienced by the portfolio and serves as a valuable measure for assessing its potential downside risk and overall performance resilience.

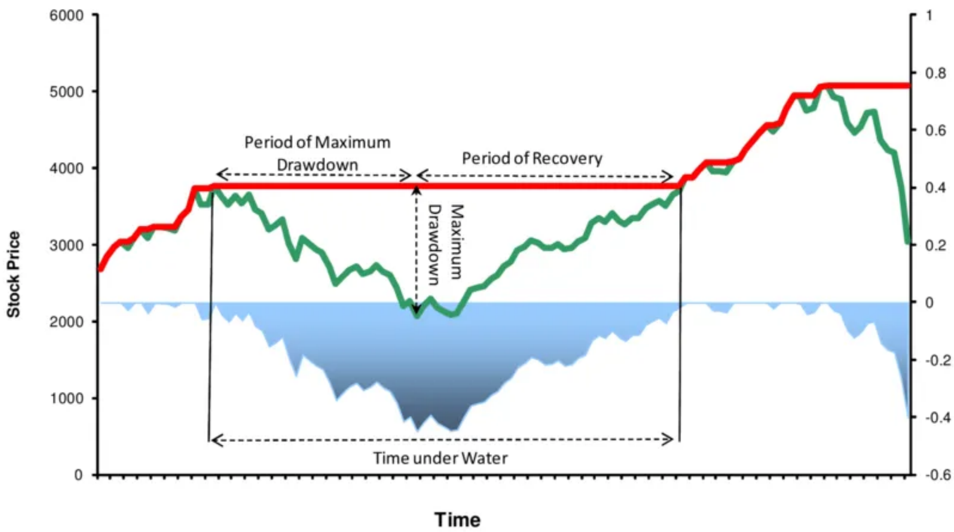


Figure 1.12: Maximum Drawdown

1.6 Conclusion

In conclusion, algorithmic trading has revolutionized the world of financial markets. Through the automation of trading strategies, it has brought efficiency, speed, and precision to the trading process. Algorithms analyze vast amounts of data and execute trades with split-second precision, allowing for the exploitation of even the smallest market inefficiencies. However, algorithmic trading is not without its challenges. It requires a deep understanding of financial markets, sophisticated modeling techniques, and continuous adaptation to changing market conditions. Risk management is paramount, as automated systems can amplify losses if not carefully monitored.Nonetheless, algorithmic trading has democratized access to financial markets, enabling a broader range of market participants to compete on a level playing field. As technology continues to advance, we can expect algorithmic trading to evolve further, pushing the boundaries of what is possible in the world of finance.

Chapter 2

Metaheuristics Optimization Approaches

2.1 Particle Swarm Optimizer (PSO)

2.1.1 Introduction

In 1995, Kennedy and Eberhart developed Particle Swarm Optimization (PSO), a method of stochastic optimization. It involves analyzing the search space of a problem to find the structure or parameters necessary for optimizing a critical target or defined objective. This technique draws inspiration from swarm intelligence in nature, specifically the swarming habits of creatures, animals, or insects. PSO has become increasingly popular in the field of computational intelligence and has been successfully applied to various optimization and search problems. As a result, it has gained a reputation as one of the most widely used and highly regarded algorithms in the literature of computational intelligence, metaheuristics, and optimization. PSO has been extensively utilized in several fields such as science and engineering.

2.1.2 Inspiration

PSO was developed by abstracting the working mechanism of natural phenomena, such as the navigation pattern and foraging swarms of creatures like birds or fish. The PSO algorithm is described as a group of particles or individuals that interconnect, link together, and interact using search directions or gradients. These particles fly over a search space to locate global optima positions, updating their location based on previous knowledge or experience, as well as information obtained from neighborhood searches. This means that particles, like birds, bats, or fish, maintain their position and learn from experience encountered while navigating as a flock or swarm. Effective communication during the navigational process is crucial, and feedback is received from both local and global best positions during the global search process.

2.1.3 Mechanism

In order to converge towards the global optimum, the PSO algorithm continuously updates the position and velocity of each particle based on its personal experience, as well as the collective experience of the swarm. This process of information sharing and communication among the particles allows them to efficiently explore the search space and avoid being trapped in local optima. By balancing the exploration and exploitation trade-off, PSO is able to effectively search for the optimal solution. Through repeated iterations and updates, the particles move towards the best solution found by the swarm so far, ultimately converging on the global optimal solution. This iterative process of updating the position and velocity of particles based on their past experience, combined with the collective experience of the swarm, allows PSO to find high-quality solutions to complex optimization problems.

The PSO algorithm can be described as follows:

1. Initialize the swarm of particles randomly in the search space.

2. Evaluate the fitness of each particle based on its position.

3. Update the personal best position and fitness of each particle.

4. Update the global best position and fitness based on the personal best positions of all particles.

5. Update the velocity and position of each particle according to the equations:

$$v_{i,j}^{t+1} = wv_{i,j}^{t} + c_1 r_1 (p_{i,j} - x_{i,j}^{t}) + c_2 r_2 (g_j - x_{i,j}^{t}) \tag{2.1}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \tag{2.2}$$

   where i is the particle index, j is the dimension index, t is the iteration number, $v_{i,j}^{t}$ is the velocity of particle i in dimension j at iteration t, $x_{i,j}^{t}$ is the position of particle i in dimension j at iteration t, $p_{i,j}$ is the personal best position of particle i in dimension j, g_j is the global best position in dimension j, w is the inertia weight, c_1 and c_2 are the acceleration coefficients, and r_1 and r_2 are random numbers between 0 and 1.

6. Evaluate the fitness of each particle based on its new position.

7. If the termination criterion is not met, go to step 3; otherwise, return the global best position as the solution[2].

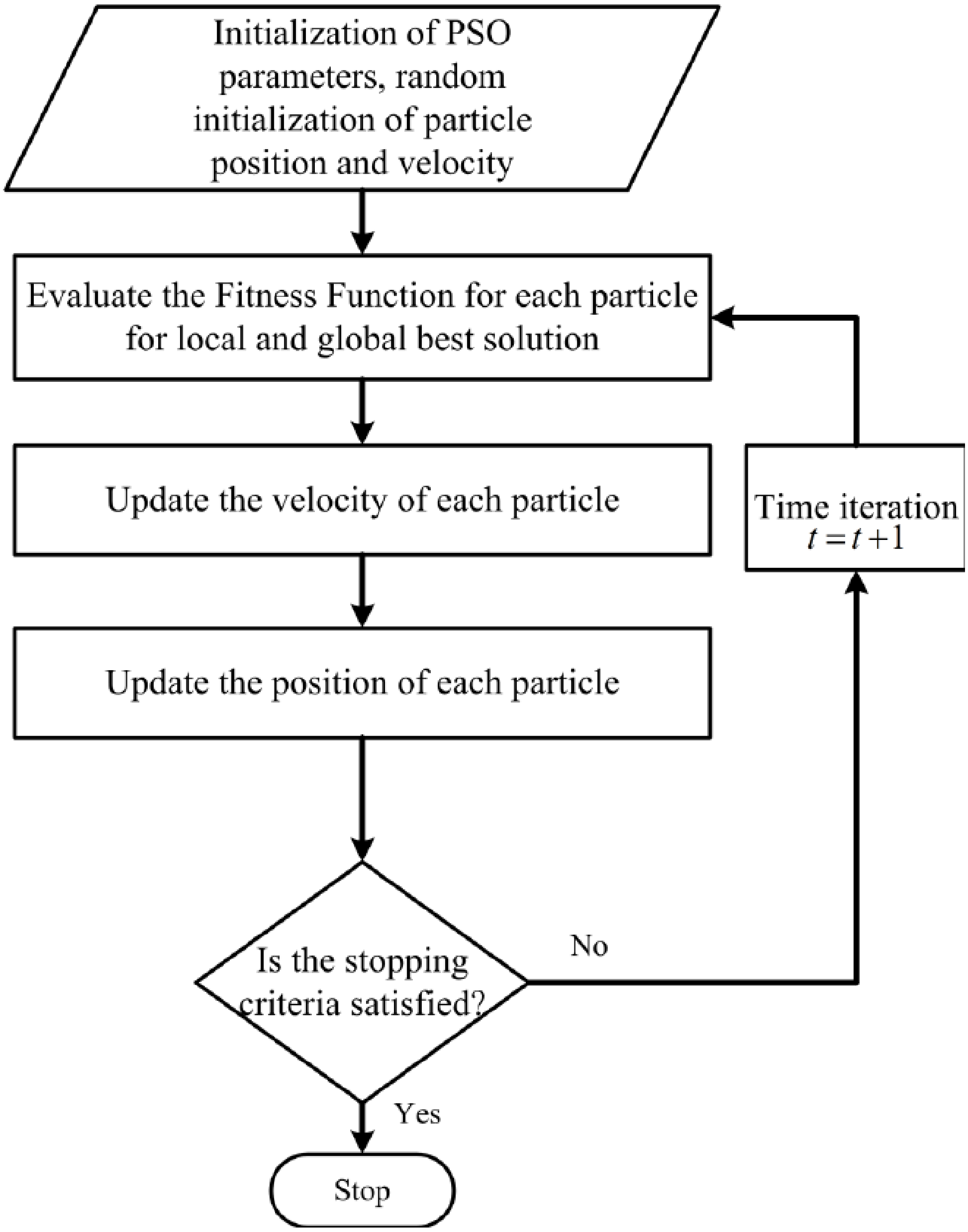


Figure 2.1: PSO Flow Chart

2.2 Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D)

2.2.1 Introduction

MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition) is a popular multi-objective optimization algorithm developed by Zhang and Li in 2007. It is a population-based optimization algorithm that solves multi-objective optimization problems by decomposing the problem into several scalar sub-problems, which are then optimized simultaneously.The algorithm uses a decomposition strategy to optimize the multiple objectives, where each sub-problem is solved using a simple single-objective optimization algorithm. The solution sets obtained from the sub-problems are combined to form the Pareto front, which represents the optimal trade-offs between the multiple objectives.MOEA/D starts by initializing a population of candidate solutions randomly within the search space. The algorithm then decomposes the multi-objective problem into T scalar sub-problems using a weight vector approach. Each sub-problem is solved independently by optimizing a weighted sum of the objectives, where the weight vector represents the relative importance of each objective.During the optimization process, each sub-problem maintains a separate population of solutions, which are evolved using a variation operator (e.g., crossover and mutation). The offspring solutions are then evaluated and added to the population of the corresponding sub-problem. To balance exploration and exploitation, the algorithm also employs a neighborhood search mechanism, where solutions from neighboring sub-problems are used to improve the diversity of the population.The algorithm terminates when a stopping criterion is met, typically after a fixed number of iterations or when the improvement in the Pareto front is below a certain threshold. At the end of the optimization process, the solutions from all sub-problems are combined to form the final Pareto front, which represents the set of trade-offs between the objectives that cannot be improved without compromising another objective. MOEA/D has been shown to be highly effective in solving multi-objective optimization problems in various fields, such as engineering, finance, and environmental management. It has been widely used and has inspired several variants and extensions, including MOEA/D-PCA, MOEA/D-DE, MOEA/D-DRA, and MOEA/D-STM.

2.2.2 Mechanism

This is an outline of the mechanism of a multi-objective optimization algorithm. The goal of such algorithms is to find a set of solutions that are optimal with respect to multiple objectives, rather than just a single objective. The algorithm proceeds through several steps:

1. **Initialization**: Generate an initial population of N individuals randomly.

2. **Decomposition**: Decompose the multi-objective problem into T single-objective subproblems using a scalarization method. Each subproblem aims to optimize a particular objective, and the scalarization method converts the multi-objective problem into a single-objective problem by assigning weights to each objective. Specifically, the scalarized subproblem for objective m can be formulated as:

$$\min g_m(\mathbf{x}) = f_m(\mathbf{x}) - \theta_m \sum_{l=1}^{M} \lambda_l f_l(\mathbf{x}), \tag{2.3}$$

where $\mathbf{x}$ represents the decision variables, $f_m(\mathbf{x})$ is the objective function to be minimized for the m-th subproblem, λ_l is a weight for the l-th objective, and θ_m is a normalization factor that ensures $\sum_{l=1}^{M} \lambda_l = 1$ and $\lambda_l \geq 0$ for all l.

3. **Reproduction**: For each subproblem, generate a mating pool by selecting parents from the population using a selection method (e.g., tournament selection, roulette wheel selection). Perform crossover and mutation operations on the parents to create offspring.

4. **Improvement**: Apply an improvement method to improve the quality of the offspring solutions. This step may involve local search or other optimization techniques.

5. **Updating**: Replace the solutions in the current population with the offspring solutions generated in step 3 and improved in step 4.

6. **Environmental Selection**: Select the best solutions from the current population and the offspring solutions to form the next generation. The selection process is based on the subproblem contributions, which reflect the quality of the solutions with respect to the corresponding subproblems. Specifically, the contribution of solution $\mathbf{x}$ to subproblem m can be calculated as:

$$T_m(\mathbf{x}) = \frac{1}{d(\mathbf{x}, \mathbf{z}_m) + \epsilon}, \tag{2.4}$$

where $d(\mathbf{x}, \mathbf{z}_m)$ is the Euclidean distance between $\mathbf{x}$ and the reference point $\mathbf{z}m$, which represents the ideal point and the nadir point for each objective. The value of ϵ is a small positive constant added to the denominator to avoid division by zero. The subproblem contribution $C_m(\mathbf{x})$ of solution $\mathbf{x}$ is then computed as:

$$C_m(\mathbf{x}) = \frac{T_m(\mathbf{x})}{\sum i = 1^T T_i(\mathbf{x})}. \tag{2.5}$$

Finally, the new population is selected by maximizing the subproblem contributions of the solutions.

7. **Termination**: Check if the termination condition is satisfied (e.g., maximum number of generations reached, convergence criteria met). If not, go to step 2[2].

Overall, the algorithm attempts to balance the optimization of multiple objectives by decomposing the problem into several single-objective subproblems and selecting solutions based on their contributions to these subproblems. This can lead to a set of solutions that represents a trade-off between the different objectives, rather than just a single optimal solution.

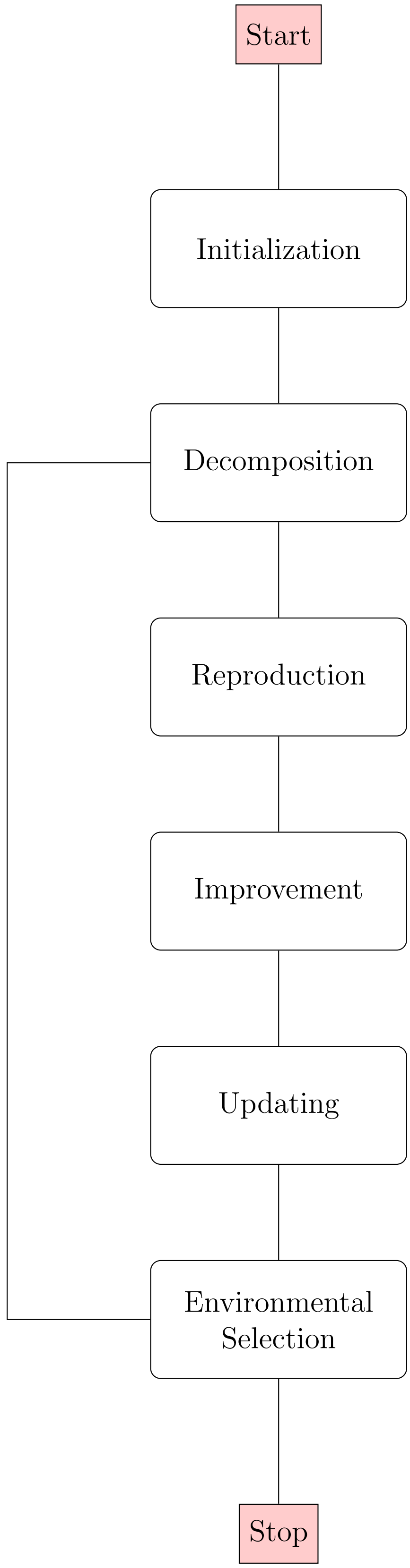


Figure 2.2: Flow chart of the MOEA/D algorithm.

2.3 Conclusion

In conclusion, metaheuristics, including techniques like Particle Swarm Optimization (PSO) and Multi-Objective Evolutionary Algorithms based on Decomposition (MOEA/D), stand as innovative and powerful approaches to optimization challenges. Particle Swarm Optimization, inspired by the collective behavior of birds or fish, offers an elegant way to navigate complex solution spaces. Its ability to balance exploration and exploitation makes it effective in finding high-quality solutions in various domains. On the other hand, MOEA/D, with its focus on solving multi-objective optimization problems, addresses the growing need for decision-making in scenarios with conflicting objectives. By decomposing multi-objective problems into simpler subproblems, MOEA/D provides valuable insights into trade-offs among objectives. Both PSO and MOEA/D exemplify the adaptability and versatility of metaheuristics. They have shown remarkable success in diverse fields, including engineering, finance, and artificial intelligence. As we continue to face complex, multi-dimensional challenges, these metaheuristic techniques offer valuable tools for finding solutions that balance conflicting goals and navigate intricate solution spaces efficiently. The future holds exciting opportunities for further advancements and applications of these powerful algorithms.

Chapter 3

Metaheuristics-Based Optimization of LFTTS

3.1 Introduction

LFTTS or Low-Frequency Technical Trading Systems are trading strategies that do not involve high-frequency, rapid-fire trading. Instead, they typically focus on longer timeframes, such as daily, weekly, or even monthly data. These systems often use technical analysis techniques to make trading decisions based on historical price data and patterns. Low-frequency trading is characterized by a slower pace compared to high-frequency trading, which involves rapid trading within milliseconds or microseconds. LFTTS strategies are more suitable for traders and investors who take a longer-term view of the market and do not engage in the frequent buying and selling of assets. Cryptocurrency markets are highly volatile and require effective trading strategies to maximize returns and manage risks. Technical trading systems that employ various technical indicators and trading rules are commonly used by traders and investors to analyze market trends and make trading decisions. However, designing and optimizing technical trading systems for cryptocurrency markets can be challenging due to the complexity and unpredictability of these markets. In recent years, metaheuristic algorithms have gained popularity for designing and optimizing trading systems due to their ability to handle complex and nonlinear optimization problems. In this study, we employ two different metaheuristic algorithms, particle swarm optimization (PSO) and multiobjective evolutionary algorithm based on decomposition (MOEAD), to design and optimize a technical trading system for three major cryptocurrencies, namely Bitcoin, Ethereum, and NEO. Two different approaches for training and optimization were adopted in this study. The first approach involved classical training, where 80% of the data was used for training and 20% for testing. This approach was used for both single and multiobjective optimization, with the objective of maximizing five different metrics: Sharpe ratio, Sortino ratio, Calmar ratio, R2 square of log returns, and log square of annual returns. The second approach utilized walk forward optimization (WFO) for single objective optimization only, using PSO to maximize the Sortino ratio. We aim to maximize various performance metrics to evaluate the effectiveness of the proposed trading strategy. The results of the study demonstrate the effectiveness of metaheuristic algorithms in designing and optimizing technical trading systems, suggesting potential applicability in real-world trading scenarios. Nonetheless, the issue of overfitting is a potential concern with the optimized trading strategy, and further research is needed to validate its performance on future data. The findings of this study could be of significant interest to investors, traders, and researchers interested in developing and optimizing technical trading systems for cryptocurrency markets.

3.2 Data Collection

We conducted a comprehensive data collection effort to gather historical OHLCV (Open, High, Low, Close, Volume) data for three of the most prominent cryptocurrencies in the market: BTCUSDT (Bitcoin/USDT), ETHUSDT (Ethereum/USDT), and NEOUSDT (NEO/USDT). This data, crucial for our research, was meticulously acquired from the Binance exchange utilizing their API. The dataset covers a substantial timeframe, spanning from August 7, 2017, to March 19, 2023, with an hourly resolution.

The resulting dataset comprised a substantial 48,831 candlesticks, each representing a snapshot of price and volume information during an hour of trading. However, our commitment to data quality did not stop at mere collection. To ensure the accuracy and reliability of the information we would base our analysis and optimization efforts upon, we rigorously performed a series of data preprocessing steps.

These preprocessing steps were essential in enhancing the integrity of our dataset. They involved tasks such as identifying and removing missing values, rectifying anomalies or inconsistencies that may have arisen during data collection, and meticulously filtering out any outliers. These meticulous efforts were not only a testament to our dedication to rigorous research but also a means of guaranteeing that our subsequent analysis would be grounded in high-quality, representative data. This representative dataset was pivotal in ensuring that our optimization and analysis efforts accurately reflected the real market conditions and trends that prevailed throughout the selected time period. In essence, the quality and reliability of our data formed the cornerstone of our research, enabling us to draw meaningful conclusions and make informed decisions in the realm of cryptocurrency trading strategies. In our data preprocessing efforts, we diligently addressed anomalies within the dataset. These anomalies primarily consisted of candles that were the outcome of "pump and dump" activities. "Pump and dump" refers to orchestrated efforts to inflate the price of a cryptocurrency (the "pump") followed by a swift and deliberate sell-off (the "dump"). Such irregularities in the data could potentially skew our analysis and optimization processes, making it imperative to rectify them. Our commitment to data integrity led us to identify and appropriately address these anomalous candles, ensuring that our subsequent analysis and optimization were grounded in a more accurate and representative dataset, free from artificially induced distortions.

3.3 Proposed Architecture of LFTTS

The technical trading system we use involves 19 parameters that must be optimized for maximum effectiveness. These parameters take into account changes in trading volume, price, and volatility to identify accurate and profitable entry points. The system is comprised of four stages:

3.3.1 Calculating Technical Indicators for Trading Analysis

In this section we will present the technical indicators and their corresponding parameters that are utilized in our trading strategy. These indicators are used to analyze different aspects of the market such as price movements, volume, volatility, and momentum, among others. By combining multiple indicators and sub-strategies, we aim to generate more accurate trade signals and improve our overall trading performance.

- **Bollinger Bands Indicator:**

  This indicator includes the Mid line of BB, Upper line of BB, and Lower line of BB with the following parameters:

  - Middle Bollinger Band:

  $$\mathrm{MA_{BB}} = \mathrm{SMA(Close, BBperiod)} \tag{3.1}$$

  - Upper Bollinger Band:

  $$\mathrm{BBup} = \mathrm{MA_{BB}} + \mathrm{nbdevup} \times \mathrm{MA_{BB}} \tag{3.2}$$

  - Lower Bollinger Band:

  $$\mathrm{BBlow} = \mathrm{MA_{BB}} - \mathrm{nbdevdn} \times \mathrm{MA_{BB}} \tag{3.3}$$

where $\mathrm{MA(Close, BBperiod)}$ represents the Simple moving average of the closing prices with a period of BBperiod, MABB is the middle Bollinger Band, BBup is the upper Bollinger Band, and $\mathrm{BB_{low}}$ is the lower Bollinger Band. nbdevup and nbdevdn are the deviation factors used to calculate the upper and lower Bollinger Bands, respectively.

- **Volume's EMAs**

  The crossover of fast and slow exponential moving averages (EMAs) of volume is a tool used by traders to identify potential shifts in volume trends.

  - $\mathrm{FastEMA_{Volume}}$:
  $$\mathrm{EMA_{FV}} = \mathrm{EMA(Volume, Fastvol)} \tag{3.4}$$
  - $\mathrm{SlowEMA_{Volume}}$:
  $$\mathrm{EMA_{SV}} = \mathrm{EMA(Volume, Slowvol)} \tag{3.5}$$

  where $\mathrm{EMA}(x, n)$ represents the EMA of a given data series x with a period of n.

- **Close Price EMAs**
  Using multiple exponential moving averages (EMA) in a technical trading system can improve trend identification across different time frames, confirm trading signals, smooth out noise, and provide flexibility to adjust the strategy to different market conditions.

  This includes five parameters for very fast, fast, mid, slow, and very slow EMA of close price: price

  - $\mathrm{VeryFastEMA_{Close}}$:
  $$\mathrm{EMA_{VF}} = \mathrm{EMA(Close, VF)} \tag{3.6}$$
  - $\mathrm{FastEMA_{Close}}$:
  $$\mathrm{EMA_{F}} = \mathrm{EMA(Close, F)} \tag{3.7}$$
  - $\mathrm{MidEMA_{Close}}$:
  $$\mathrm{EMA_{M}} = \mathrm{EMA(Close, M)} \tag{3.8}$$
  - $\mathrm{SlowEMA_{Close}}$:
  $$\mathrm{EMA_{S}} = \mathrm{EMA(Close, S)} \tag{3.9}$$
  - $\mathrm{VerySlowEMA_{Close}}$:
  $$\mathrm{EMA_{VS}} = \mathrm{EMA(Close, VS)} \tag{3.10}$$

  where $\mathrm{EMA}(x, n)$ represents the EMA of a given data series x with a period of n.

- **Bollinger Bands Width (BBW)**
  The Bollinger Bands width can be a useful tool for identifying periods of high and low volatility, as well as potential trading opportunities.

  When the BBW is narrow, it suggests that the market is experiencing low volatility and may be preparing for a breakout or a new trend. This can be a signal to watch for price movements that may indicate a change in the market's direction.

Conversely, when the BBW is wide, it suggests that the market is experiencing high volatility and that prices may be trending strongly in one direction or another. We use this information to identify potential entry or exit points for their trades.

$$\text{BBW} = \frac{\text{Upper Band} - \text{Lower Band}}{\text{Middle Band}} \tag{3.11}$$

This includes the following lines:

$$FastEMA_{\text{BBW}} = EMA(BBW, BbwFastperiod) \tag{3.12}$$

$$SlowEMA_{\text{BBW}} = EMA(BBW, BbwSlowperiod) \tag{3.13}$$

$$Vl_{\text{fast}} = \text{EMA}(BBW, \text{VlFastperiod}) \tag{3.14}$$

$$Vl_{\text{Slow}} = \text{EMA}(BBW, \text{VlSlowperiod}) \tag{3.15}$$

$$Vl_{\text{Top}} = \text{EMA}(BBW, \text{VlTopperiod}) + 2 \times \ \sigma(BBW, \text{VlTopperiod}) \tag{3.16}$$

Here, $\text{EMA}(x, n)$ represents the EMA of a given data series x with a period of n, and $\text{standardDeviation}(x, n)$ represents the standard deviation of the data series x with a period of n.

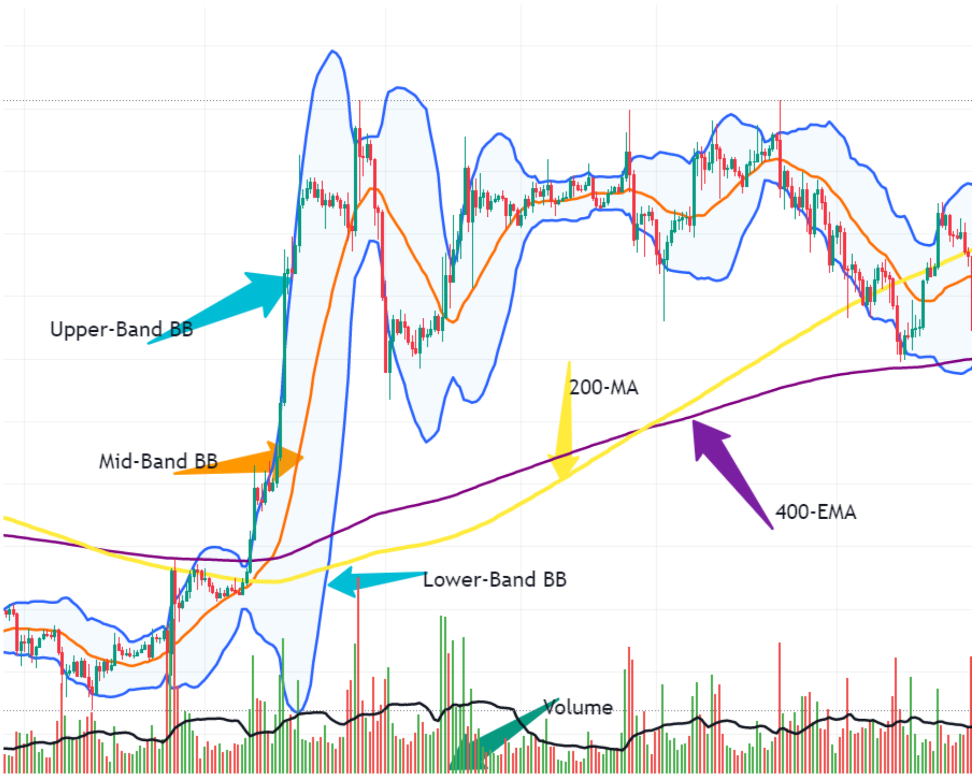


Figure 3.1: Technical Indicators (SMA_{200}, EMA_{400}, Volume, Bollinger Bands^{R})

The figure includes the four technical indicators that were mentioned earlier, namely the 200-MA, 400-EMA, upper band, lower band, and mid band of Bollinger Bands, along with the volume.

3.3.2 Establishing Effective Trading Conditions

Bollingers Bands Crossover conditions The Bollinger Bands crossover strategy utility involves using the Bollinger Bands to identify potential buying and selling opportunities. The strategy is based on the principle of buying when the price crosses above the upper band and selling when the price crosses below the lower band. A stop loss order should be used to limit potential losses, and a take profit order can be used to lock in gains. The parameters used in the strategy should be adjusted based on the asset's behavior and market conditions. It's important to note that the strategy is not a guarantee of profits and should be used with discretion and proper risk management techniques.

Algorithm 1 Bollinger Bands CrossOver Conditions Algorithm

INPUTS:
BB_{up} (BB_up), BB_{low} (BB_lw), $\text{Close}_{\text{price}}$ ($Close\_p$)
OUTPUTS:
CrossBBup ($CrossBBup\_condition$), CrossBBdn ($CrossBBdn\_condition$)

if $Close\_p[i-1] > BB\_up[i-1]$ *and* $Close\_p[i] < BB\_up[i]$ **then**
  $CrossBBup\_condition \leftarrow True$
else if $Close\_p[i-1] < BB\_up[i-1]$ *and* $Close\_p[i] > BB\_up[i]$ **then**
  $CrossBBup\_condition \leftarrow True$
else if $Close\_p[i-1] > BB\_dn[i-1]$ *and* $Close\_p[i] < BB\_dn[i]$ **then**
  $CrossBBdn\_condition \leftarrow True$
else if $Close\_p[i-1] < BB\_dn[i-1]$ *and* $Close\_p[i] > BB\_dn[i]$ **then**
  $CrossBBdn\_condition \leftarrow True$
else
  $CrossBBup\_condition \leftarrow False$,
  $CrossBBdn\_condition \leftarrow False$

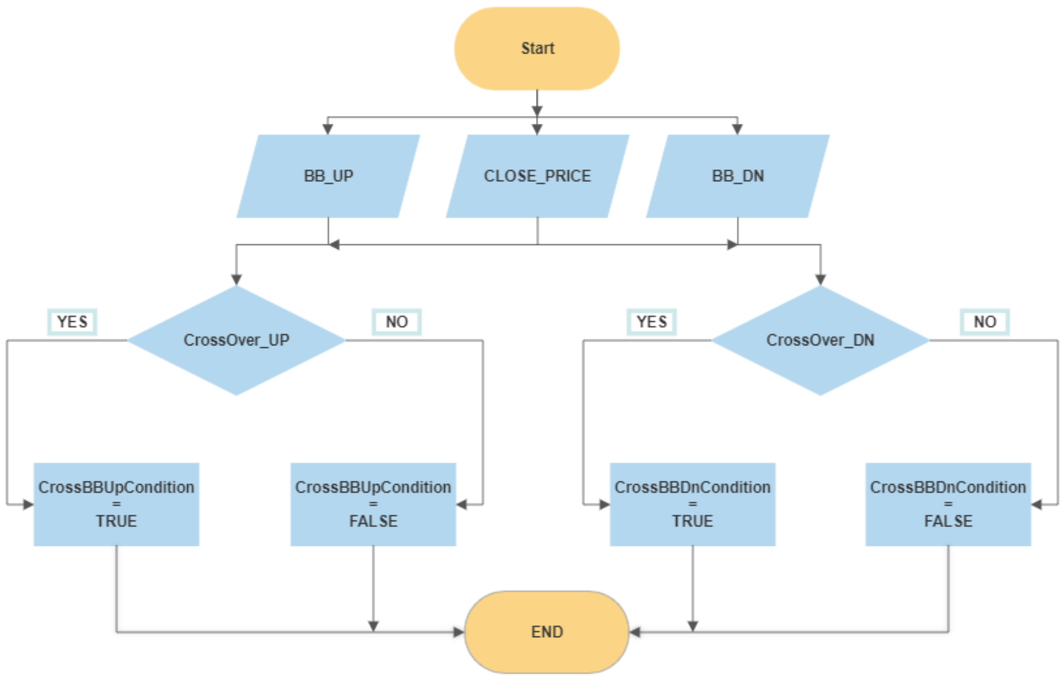


Figure 3.2: Flow Chart for BollingerBands CrossOver Conditions

The following figure shows how the Bollinger Bands condition is established:

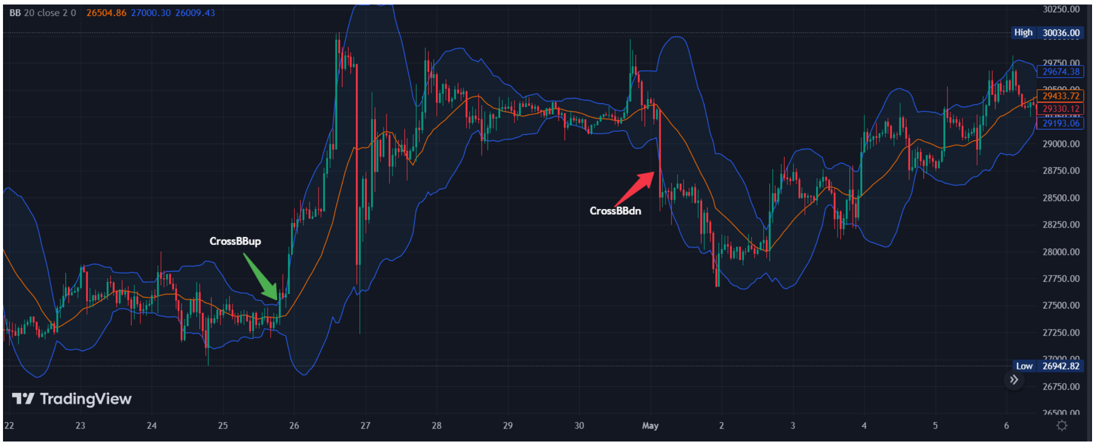


Figure 3.3: BollingerBands CrossOver Conditions

Volume Condition When the fast EMA of volume is above the slow EMA, it suggests that there has been an increase in trading volume in the recent past compared to the longer-term average. This can indicate increased market interest and potential trading opportunities. Conversely, if the fast EMA of volume is below the slow EMA, it suggests a decrease in trading volume compared to the longer-term average, which can indicate a lack of market interest and potential price decline.

Algorithm 2 Volume Condition Algorithm

INPUTS:

FastEMA$_{\text{Volume}}$ (EMA_FV), SlowEMA$_{\text{Volume}}$ (EMA_SV)

OUTPUTS:

Volume Condition $(volume\_condition)$

if $EMA\_FV[i] > EMA\_SV[i]$ **then**
  $volume\_condition \leftarrow True$
else
  $volume\_condition \leftarrow False$

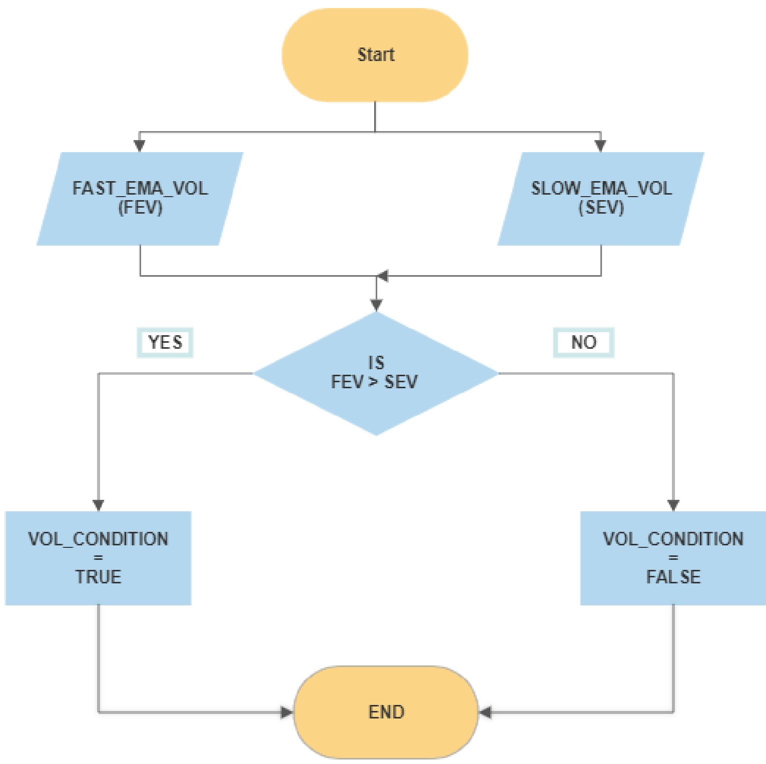


Figure 3.4: Flow Chart for Volume Condition

The following figure illustrates the establishment of the volume condition based on the relative positions of fast and slow exponential moving averages of volume:

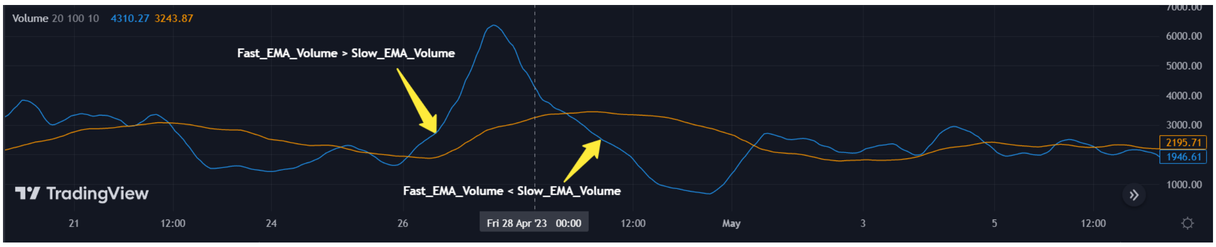


Figure 3.5: Volume Condition

Volatility Conditions To minimize the potential risks associated with volatility, a trading strategy can be designed to initiate trades during periods of low volatility, specifically when combined with Bollinger Bands crossover. By incorporating this approach, trades are opened when the market exhibits lower volatility levels. This strategy aims to avoid sudden price swings and unexpected market fluctuations that may result in increased risk. The Bollinger Bands crossover acts as a confirmation signal, providing additional validation for trade entry within the low volatility conditions.

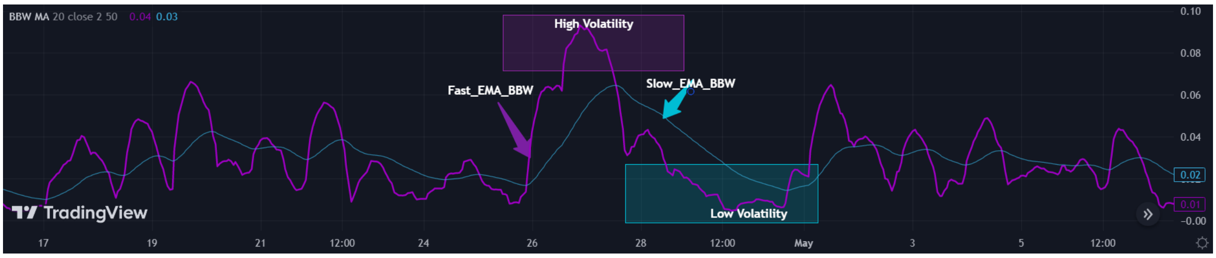


Figure 3.6: Volatility Condition

Algorithm 3 Volatility Conditions Algorithm

INPUTS:
BBW(BBW),
FastEMA$_{\text{BBW}}$ (BBW_F),
SlowEMA$_{\text{BBW}}$ (BBW_S),
Vl$_{\text{fast}}$ (Vl_f),
Vl$_{\text{slow}}$ (Vl_s),
Vl$_{\text{Top}}$ (Vl_T)

OUTPUTS:
BBW condition ($bbw_{condition}$),
Low$_{\text{Volatility}}$ condition ($Low_{volatility}$),
High$_{\text{Volatility}}$ condition ($High_{volatility}$),
Extreme$_{\text{Volatility}}$ condition ($Extreme_{volatility}$)

if $BBW\_F[i] > BBW\_S[i]$ **then**
  $bbw\_condition \leftarrow$ True
end
else
  $bbw\_condition \leftarrow$ False
end
if $Vl\_f[i] < Vl\_s[i]$ **then**
  $Low\_volatility \leftarrow$ True
end
else if $Vl\_f[i] > Vl\_s[i]$ **then**
  $High\_volatility \leftarrow$ True
end
else if $BBW[i] > Vl\_T[i]$ **then**
  $Extreme\_volatility \leftarrow$ True
end
else
  $Low\_volatility \leftarrow$ False,
  $High\_volatility \leftarrow$ False,
  $Extreme\_volatility \leftarrow$ False,
end

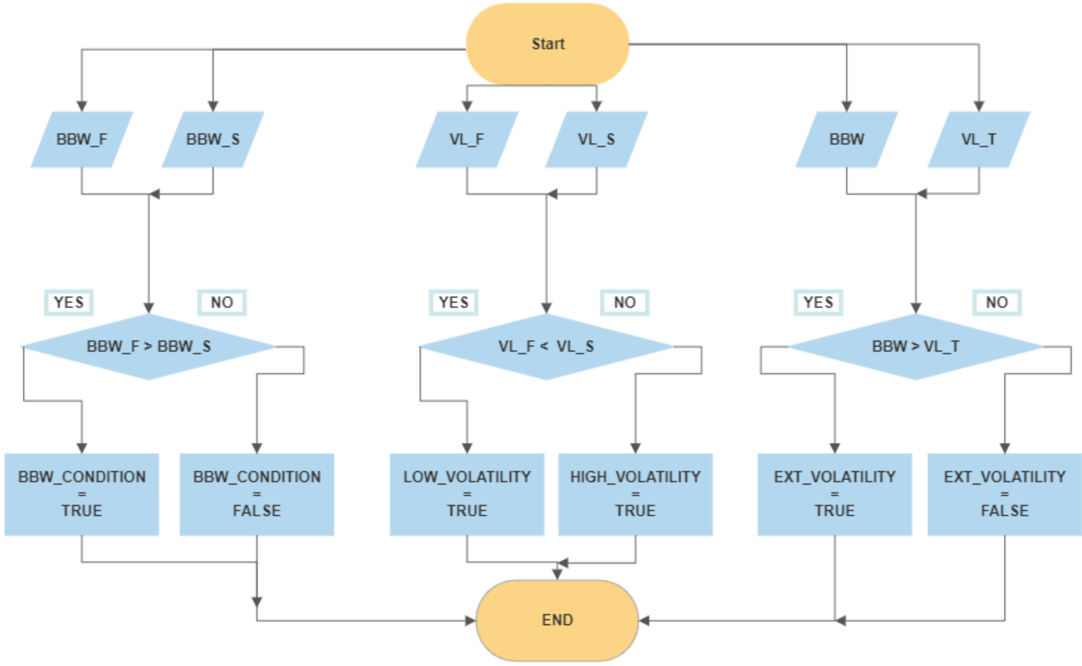


Figure 3.7: Flow Chart for Volatility Conditions

3.3.3 Establishing Rules for the Technical Trading System (TTS)

To elaborate further, we will utilize the Bollinger bands crossover to identify potential buy or sell signals. We will also consider the level of volatility and volume in the market to confirm the strength of the signal. By combining all of these factors, we aim to generate more reliable and accurate trade signals, allowing us to make informed trading decisions. Ultimately, the goal is to increase our profitability and minimize risks associated with trading in financial markets.

Algorithm 4 Trading Algorithm for Long & Short Signals

```
for each data point in the time series do
    if no position is open then
        if for long position: CrossBBup_condition is True and volume_condition is True then
            if data_Close > FastEMA_Close then
                if BBW < VL_top then
                    if low_volatility is True then
                        if MidEMA_Close > VerySlowEMA_Close then
                            Generate LONG/BUY Limit Order at : data_Close
                        end
                    end
                end
            end
        end
        if for short position: CrossBBdn_condition is True and volume_condition is True then
            if data_Close < fastEMA_Close then
                if BBW < VL_top then
                    if low_volatility is True then
                        if fastEMA_Close < slowEMA_Close then
                            Generate SELL/Short Limit Order at : data_Close
                        end
                    end
                end
            end
        end
    end
end
```

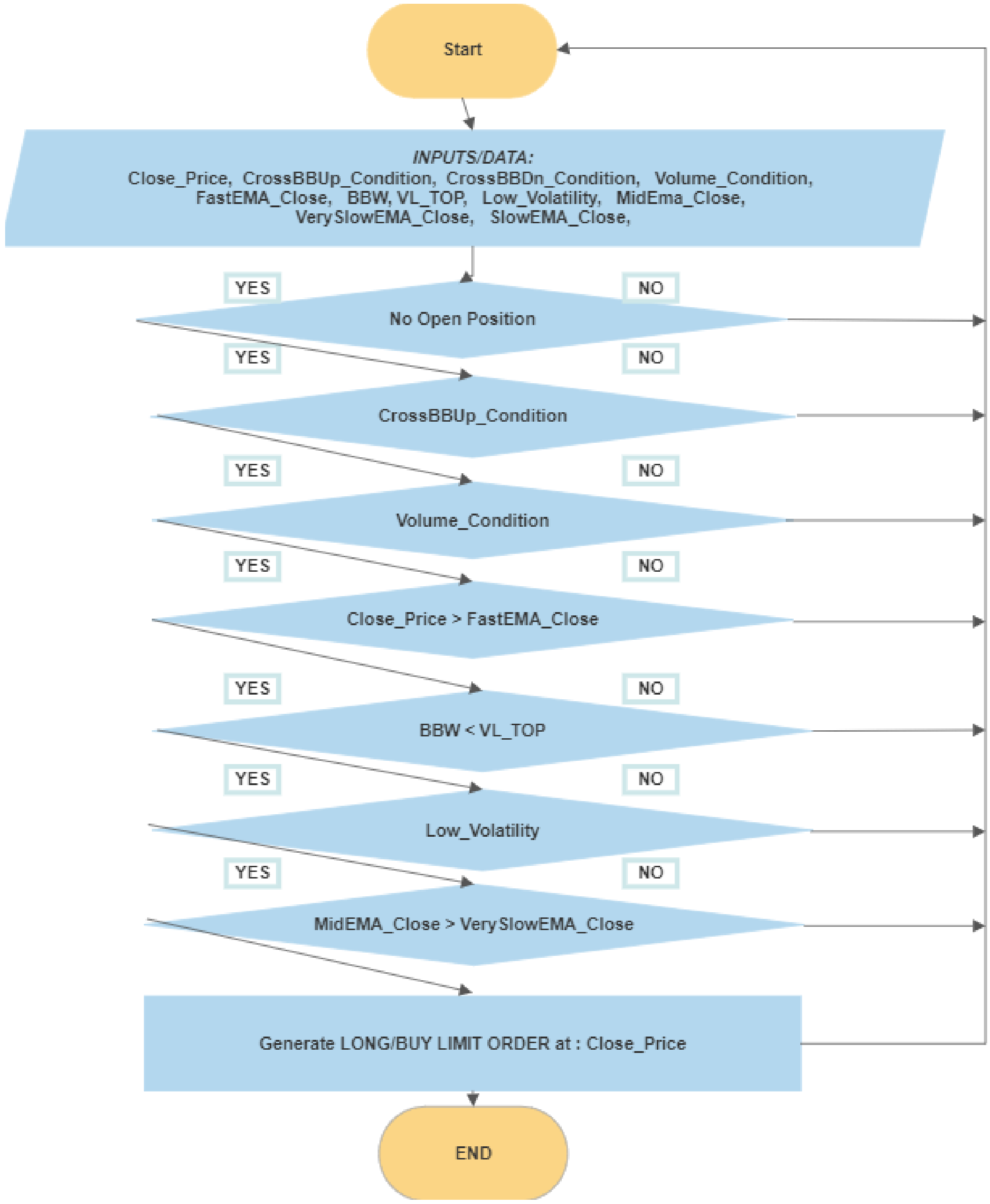


Figure 3.8: Flow Chart for LONG/BUY Signal

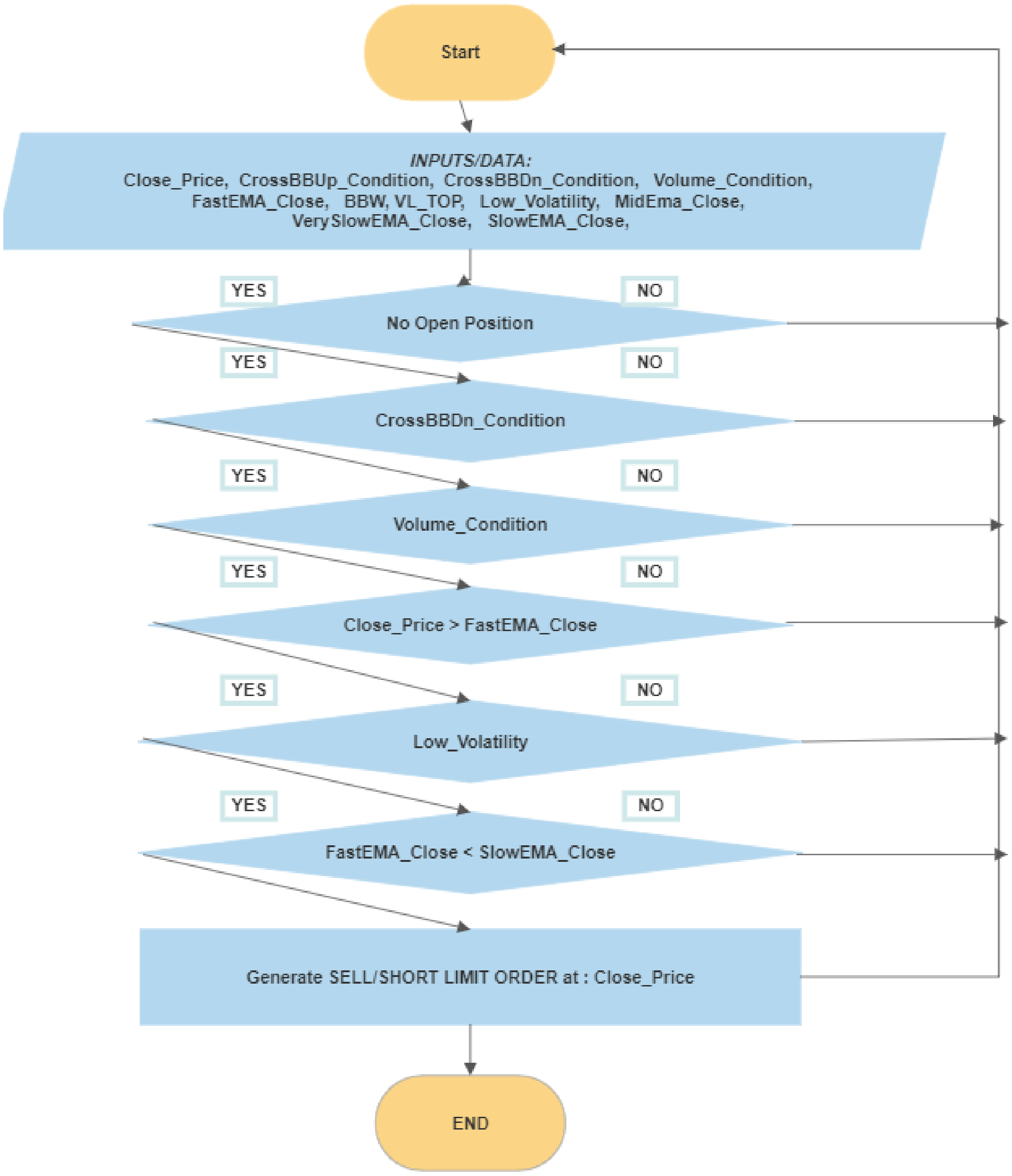


Figure 3.9: Flow Chart for SELL/SHORT Signal

3.3.4 Dynamic StopLoss/TakeProfit for the Technical Trading System(TTS)

The strategy we developed incorporates a dynamic stop-loss/take-profit algorithm aimed at optimizing trade exit levels based on current market conditions. The algorithm is designed to adjust the stop-loss and take-profit levels of a trade based on the current market conditions.

The algorithm starts by iterating through each data point in the time series. It checks whether there is an open position, and if not, it sets the maximum and minimum prices reached to 0.

For long positions, if the data close price is greater than or equal to the very slow exponential moving average (EMA), the algorithm updates the maximum price reached to the highest data high price seen so far. If the dynamic stop-loss (SL) level multiplied by the maximum price reached is greater than or equal to the data low, the algorithm closes the long position with a market order.

For short positions, if the data close price is less than or equal to the very slow EMA, the algorithm updates the minimum price reached to the lowest data low price seen so far. If the dynamic SL level multiplied by the minimum price reached is less than or equal to the data high, the algorithm closes the short position with a market order.

The algorithm continues to iterate through the time series and adjusts the stop-loss and take-profit levels for each new data point based on the current market conditions. Overall, this algorithm is designed to minimize losses and maximize profits by dynamically adjusting the exit levels of trades.

There are four parameters that require optimization in the last stage of the system. These parameters include:

- $\mathtt{dynamic}_{\mathrm{SL}_{\mathrm{Long}_{\mathtt{Bull}}}}$: the dynamic stop-loss multiplier for a long position when the close price is above $\mathrm{VerySlowEMA}_{\mathrm{Close}}$.
- $\mathtt{dynamic}_{\mathrm{SL}_{\mathrm{Long}_{\mathtt{Bear}}}}$: the dynamic stop-loss multiplier for a long position when the close price is below $\mathrm{VerySlowEMA}_{\mathrm{Close}}$.
- $\mathtt{dynamic}_{\mathrm{SL}_{\mathrm{Short}_{\mathtt{Bear}}}}$: the dynamic stop-loss multiplier for a short position when the close price is below $\mathrm{VerySlowEMA}_{\mathrm{Close}}$.
- $\mathtt{dynamic}_{\mathrm{SL}_{\mathrm{Short}_{\mathtt{Bull}}}}$: the dynamic stop-loss multiplier for a short position when the close price is above $\mathrm{VerySlowEMA}_{\mathrm{Close}}$.

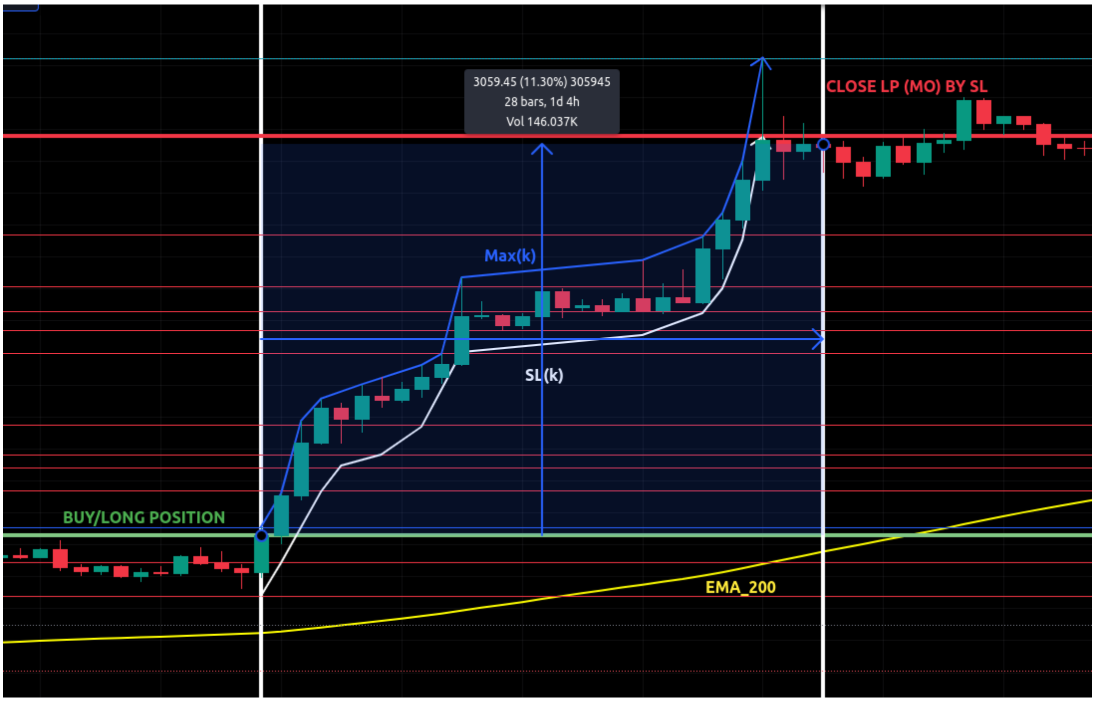


Figure 3.10: Illustration of the proposed Dynamic SL for LONG/BUY POSITION

Algorithm 5 Dynamic StopLoss/TakeProfit for Long and short Positions

for *each data point in the time series* **do**
- **if** *no position is open* **then**
  - Set Max = 0;
  - Set Min = 0;
- **end**
- **if** *Long position is True* **then**
  - **if** $data_{close} >= VerySlow_{EMA}$ **then**
    - **if** $Max <= data_{High}$ **then**
      - Set Max = $\mathrm{data}_{\mathrm{high}}$;
    - **end**
    - **if** $dynamic_{SL_{Long_{Bull}}} * Max >= data_{low}$ **then**
      - Close long position by dynamic SL/TP with Market Order;
    - **end**
  - **end**
  - **else if** $data_{close} <= VerySlow_{EMA}$ **then**
    - **if** $Max <= data_{High}$ **then**
      - Set Max = $\mathrm{data}_{\mathrm{high}}$;
    - **end**
    - **if** $dynamic_{SL_{Long_{Bear}}} * Max >= data_{low}$ **then**
      - Close long position by dynamic SL/TP with Market Order;
    - **end**
  - **end**
- **end**
- **if** *short position is True* **then**
  - **if** $data_{close} <= VerySlow_{EMA}$ **then**
    - **if** $Min >= data_{low}$ **then**
      - Set Min = $\mathrm{data}_{\mathrm{low}}$;
    - **end**
    - **if** $dynamic_{SL_{Short_{Bear}}} * Min <= data_{high}$ **then**
      - Close short position by dynamic SL/TP with Market Order;
    - **end**
  - **end**
  - **else if** $data_{Close} >= VerySlow_{SMA}$ **then**
    - **if** $Min >= data_{Low}$ **then**
      - Set Min = $\mathrm{data}_{\mathrm{low}}$;
    - **end**
    - **if** $dynamic_{SL_{Short_{Bull}}} * Min <= data_{high}$ **then**
      - Close short position by dynamic SL/TP with Market Order;
    - **end**
  - **end**
- **end**

end

3.4 Optimization Problem Formulation

Let $\boldsymbol{x} = [x_0, x_1, x_2, ..., x_{18}]$ represent a vector of the 19 parameters to be optimized for the technical trading system, where:

$$\begin{cases} \mathrm{x}_0 = \mathrm{BBperiod} & \mathrm{x}_1 = \mathrm{nbdevup} & \mathrm{x}_2 = \mathrm{nbdevdn} \\ \mathrm{x}_3 = \mathrm{Fastvol} & \mathrm{x}_4 = \mathrm{Slowvol} & \mathrm{x}_5 = \mathrm{VF} \\ \mathrm{x}_6 = \mathrm{F} & \mathrm{x}_7 = \mathrm{M} & \mathrm{x}_8 = \mathrm{S} \\ \mathrm{x}_9 = \mathrm{VS} & \mathrm{x}_{10} = \mathrm{BbwFastperiod} & \mathrm{x}_{11} = \mathrm{BbwSlowperiod} \\ \mathrm{x}_{12} = \mathrm{VlFastperiod} & \mathrm{x}_{13} = \mathrm{Vlslowperiod} & \mathrm{x}_{14} = \mathrm{VlTopperiod} \\ \mathrm{x}_{15} = \mathrm{dynamic}_{\mathrm{SL}_{\mathrm{Long}_{\mathrm{Bull}}}} & \mathrm{x}_{16} = \mathrm{dynamic}_{\mathrm{SL}_{\mathrm{Long}_{\mathrm{Bear}}}} & \mathrm{x}_{17} = \mathrm{dynamic}_{\mathrm{SL}_{\mathrm{Short}_{\mathrm{Bear}}}} \\ \mathrm{x}_{18} = \mathrm{dynamic}_{\mathrm{SL}_{\mathrm{Short}_{\mathrm{Bull}}}} \end{cases} \tag{3.17}$$

3.4.1 Single Objective Optimization Problem Formulation

Therefore, the optimization problem can be mathematically formulated as:

$$\begin{cases} \max\limits_{\boldsymbol{x}} \quad f(\boldsymbol{x}) \quad \text{subject to} \begin{cases} 5 \leq x_0 \leq 30 & 1.5 \leq x_1 \leq 3 & 1.5 \leq x_2 \leq 3 \\ 5 \leq x_3 \leq 25 & 25 \leq x_4 \leq 70 & 5 \leq x_5 \leq 20 \\ 20 \leq x_6 \leq 30 & 30 \leq x_7 \leq 70 & 70 \leq x_8 \leq 100 \\ 150 \leq x_9 \leq 250 & 5 \leq x_{10} \leq 20 & 20 \leq x_{11} \leq 70 \\ 100 \leq x_{12} \leq 300 & 800 \leq x_{13} \leq 1200 & 800 \leq x_{14} \leq 1200 \\ 0.75 \leq x_{15} \leq 0.98 & 0.85 \leq x_{16} \leq 0.99 & 1.05 \leq x_{17} \leq 1.30 \\ 1.05 \leq x_{18} \leq 1.20 & x_{15} > x_{16} & x_{17} > x_{18} \end{cases} \end{cases} \tag{3.18}$$

where $\boldsymbol{x} = [x_0, x_1, x_2, ..., x_{18}]$ represents the 19 parameters of the technical trading system, and $f(\boldsymbol{x})$ is one of the objective functions to be maximized: Sharpe ratio, Sortino ratio, Calmar ratio, or R2 of log returns. The optimization problem is subject to 19 constraints on the parameter values, where $x_0, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}$ are integers, and $x_1, x_2, x_{15}, x_{16}, x_{17}, x_{18}$ are real numbers (floats).

The choice of bounds is intended to balance the need to explore a wide range of possible values to find the global maximum of the objective function with the need to avoid searching in regions that are unlikely to yield good results. The bounds are also constrained by the specific requirements of the technical trading system being optimized, such as the need for certain parameters to be within a specific range to produce desired trading signals.

Note that we will need to run four separate optimization experiments, one for each objective function.

In this project, we consider four different metrics as objective functions for our optimization problem. These metrics are commonly used in finance to evaluate the performance of investment portfolios.

The first metric is the **Sharpe Ratio** , which measures the excess return earned per unit of risk. It is defined as:

$$SharpeRatio = \frac{R_p - R_f}{\sigma_p} \tag{3.19}$$

where R_p is the portfolio return, R_f is the risk-free rate, and σ_p is the portfolio standard deviation[6].

The second metric is the **Sortino Ratio** , which is similar to the Sharpe Ratio, but it only considers the downside risk. It is defined as:

$$SortinoRatio = \frac{R_p - R_f}{\sigma_d} \tag{3.20}$$

where R_p is the portfolio return, R_f is the risk-free rate, and σ_d is the downside risk[6].

The third metric is the **Calmar Ratio**, which measures the ratio of the average annual rate of return over a certain period to the maximum drawdown over that period. It is defined as:

$$CalmarRatio = \frac{R_{avg}}{DD} \tag{3.21}$$

where R_{avg} is the average annual rate of return and DD is the maximum drawdown[6].

The fourth metric is the **R-squared of log returns** , which measures the proportion of the variance in the portfolio returns that can be explained by the variance in the benchmark returns. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(r_{p,i} - \hat{r_p})(r_{b,i} - \hat{r_b})}{\sqrt{\sum_{i=1}^{n}(r_{p,i} - \hat{r_p})^2}\sqrt{\sum_{i=1}^{n}(r_{b,i} - \hat{r_b})^2}} \tag{3.22}$$

where $r_{p,i}$ and $r_{b,i}$ are the portfolio and benchmark log returns, respectively, at time i, and $\hat{r_p}$ and $\hat{r_b}$ are their respective mean values[6].

Each of these metrics represents a single objective function for our optimization problem. In the next step, we will run four separate optimization experiments, each time maximizing one of these objective functions.

3.4.2 Multi-Objective Optimization Problem Formulation

$$\max_{\boldsymbol{x}} \quad \left[f_1(\boldsymbol{x}) \ f_2(\boldsymbol{x})\right] \quad \text{subject to } C^{19}(\boldsymbol{x}) \tag{3.23}$$

where $\boldsymbol{x} = [x_0, x_1, x_2, ..., x_{18}]$ represents the 19 parameters of the technical trading system, $C^{19}(\boldsymbol{x})$ is the set of 19 constraints mentioned previously, and $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are the Sortino and Calmar ratios, respectively.

3.5 Results and Analysis

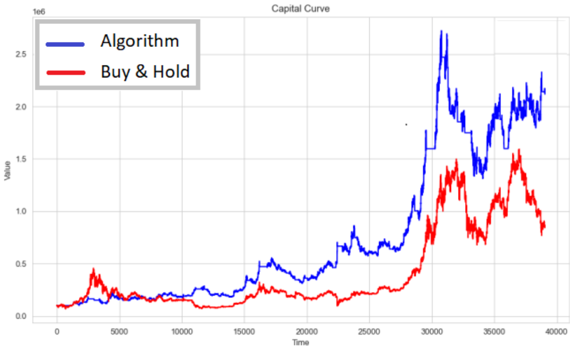
3.5.1 Classical Training Approach

We utilized the Particle Swarm Optimization (PSO) algorithm to optimize the strategy parameters of five different trading strategies for each of the three cryptocurrencies (BTCUSDT, ETHUSDT, and NEOUSDT). The dataset was divided into training and testing data, with 80% of the data used for training from 17/08/2017 to 04/02/2022, and 20% used for testing from 04/02/2022 to 19/03/2023. Each strategy was optimized using a specific metric, such as the Sharpe ratio, Sortino ratio, Calmar ratio, or R-squared of log returns, while the fifth strategy was optimized using a multi-objective optimization algorithm (MAEOD) to maximize both the Sortino and Calmar ratios simultaneously.

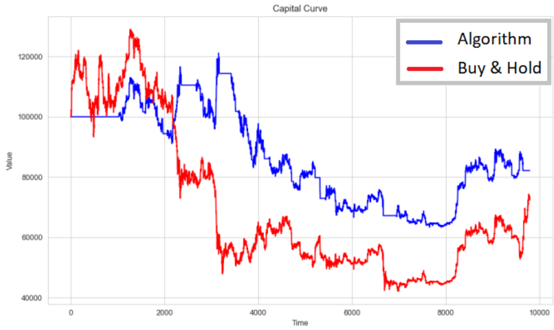
After optimizing each strategy, we traded the testing data and calculated the metrics for both in-sample and out-of-sample data. Tables were created to evaluate the performance of each strategy and compare them across different cryptocurrencies.The backtesting was conducted using the backtrader professional framework with a commission of 0.1% per trade and starting capital of 100,000 USD to simulate a realistic trading environment that accounted for trading costs and initial capital requirements. We assess our results by concentrating on the right side of our graphs, which signifies the conclusion of our simulations. Here, we conduct a comparative analysis, evaluating the effectiveness and profitability of our trading strategies in comparison to the default buy-and-hold approach. The blue line in our subsequent graphical representations represents our trading strategy's performance, while the red line depicts the buy-and-hold strategy. This visual distinction facilitates a clear and immediate comparison, allowing us to gauge the strengths and weaknesses of each strategy throughout the simulation.

3.5.2 Simulation Results Of The Classical Training Approach

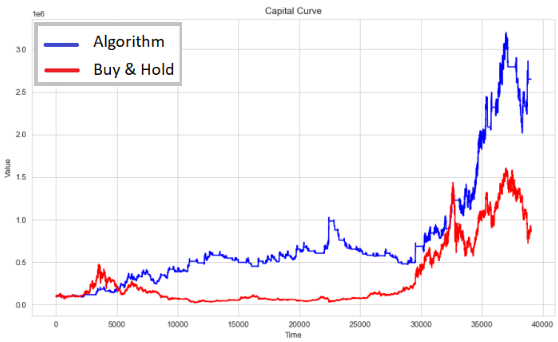
Sharpe$_{\text{Ratio}}$-Optimized Strategy



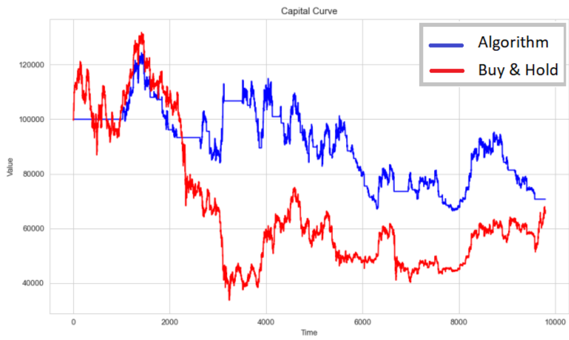
(a) BTCUSDT Capital Curve for Training Set



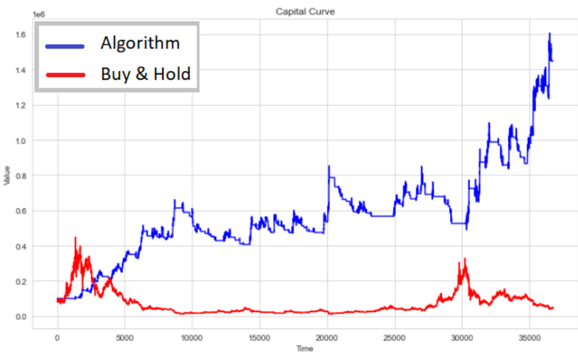
(b) BTCUSDT Capital Curve for Testing Set



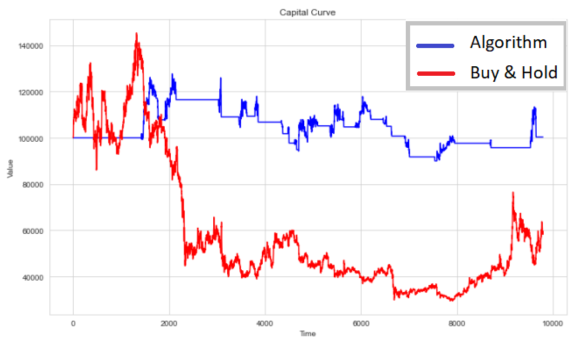
(c) ETHUSDT Capital Curve for Training Set



(d) ETHUSDT Capital Curve for Testing Set



(e) NEOUSDT Capital Curve for Training Set

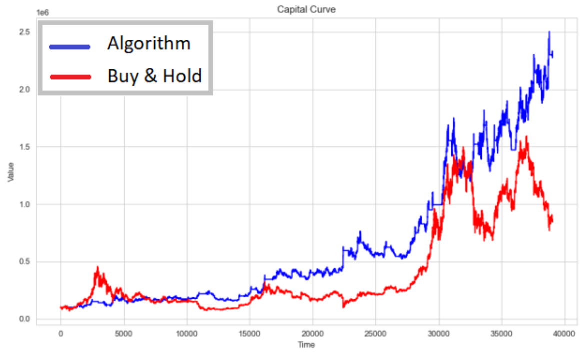


(f) NEOUSDT Capital Curve for Testing Set

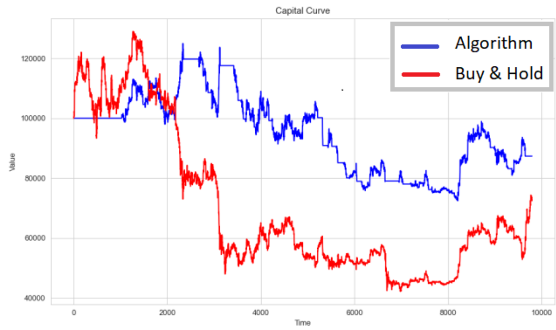
Figure 3.11: Capital Curves for Sharpe$_{\text{Ration}}$-Optimized Strategy vs Buy & Hold

we can see that our strategy performed better than 'buy and hold' for both in and out of sample

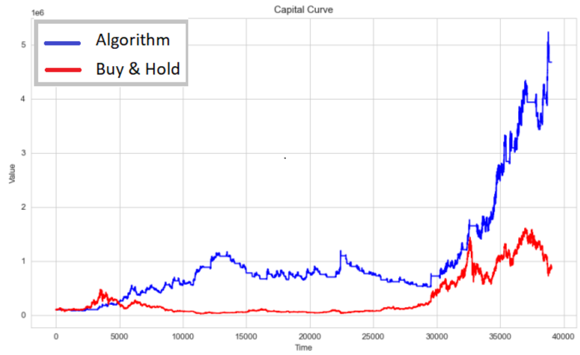
$\text{Sortino}_{\text{Ration}}$-Optimized Strategy



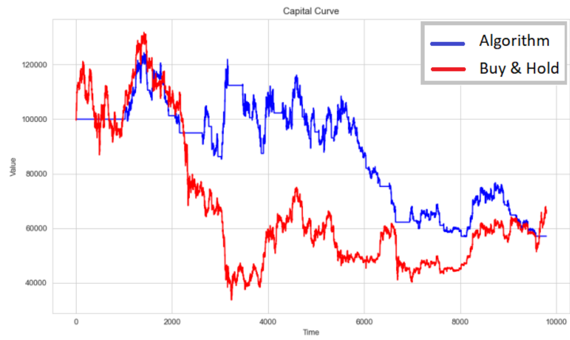
(a) BTCUSDT Capital Curve for Training Set



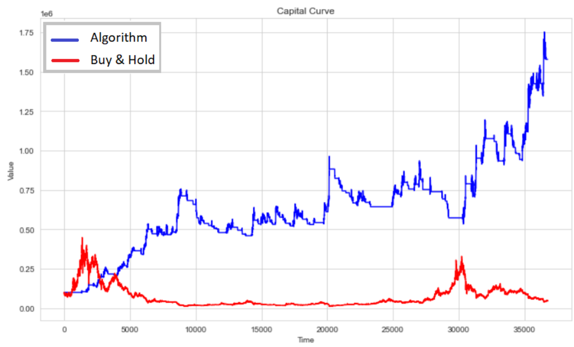
(b) BTCUSDT Capital Curve for Testing Set



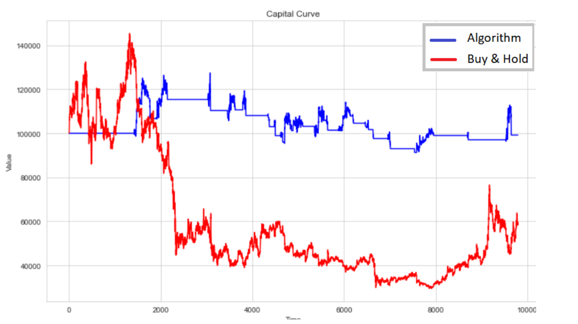
(c) ETHUSDT Capital Curve for Training Set



(d) ETHUSDT Capital Curve for Testing Set



(e) NEOUSDT Capital Curve for Training Set

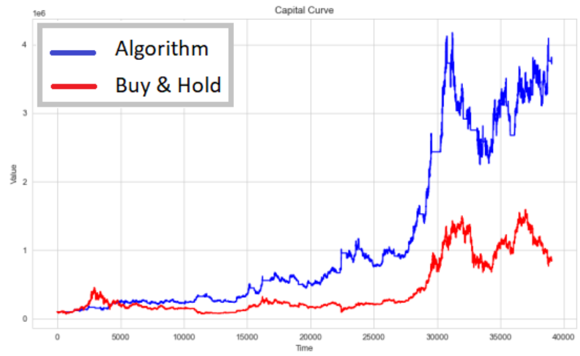


(f) NEOUSDT Capital Curve for Testing Set

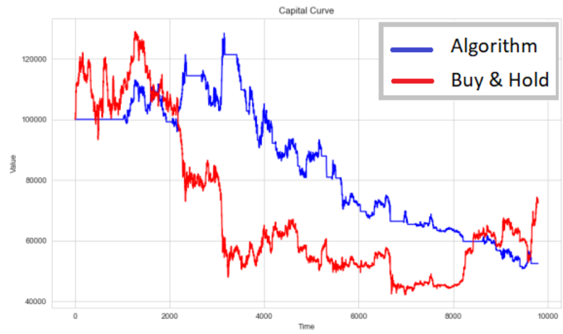
Figure 3.12: Capital Curves for $\text{Sortino}_{\text{Ratio}}$-Optimized Strategy vs Buy & Hold

We observed that our strategy outperformed the ’buy and hold’ approach in both in-sample and out-of-sample scenarios, exept for the ETHUSDT pair when out of sample.

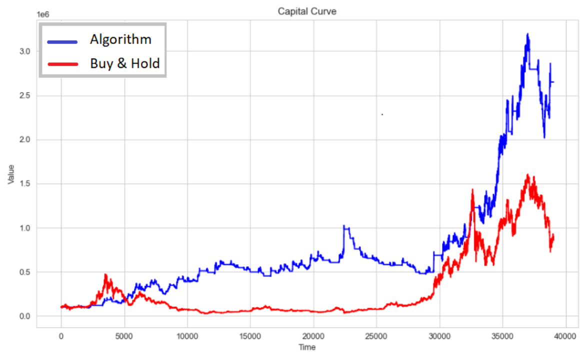
$\mathbf{Calmar_{Ratio}}$-Optimized Strategy



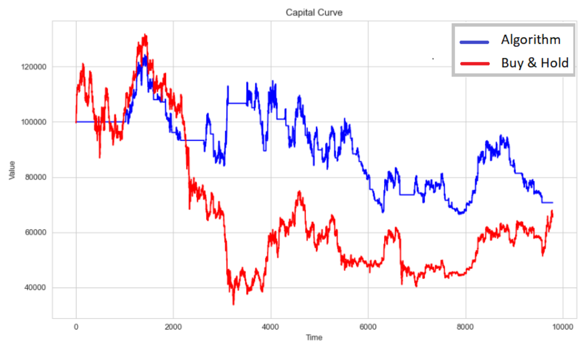
(a) BTCUSDT Capital Curve for Training Set



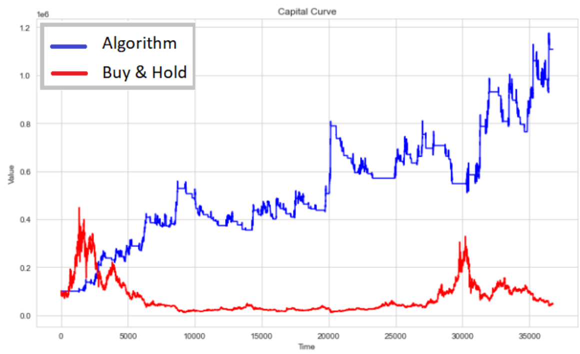
(b) BTCUSDT Capital Curve for Testing Set



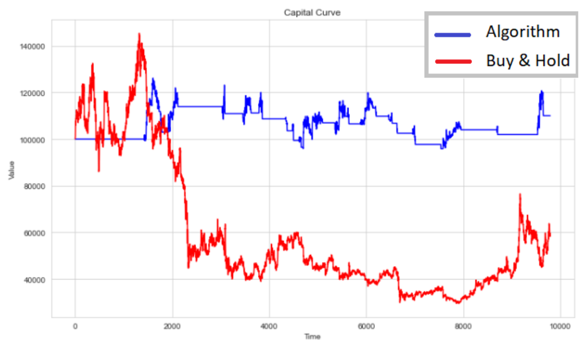
(c) ETHUSDT Capital Curve for Training Set



(d) ETHUSDT Capital Curve for Testing Set



(e) NEOUSDT Capital Curve for Training Set

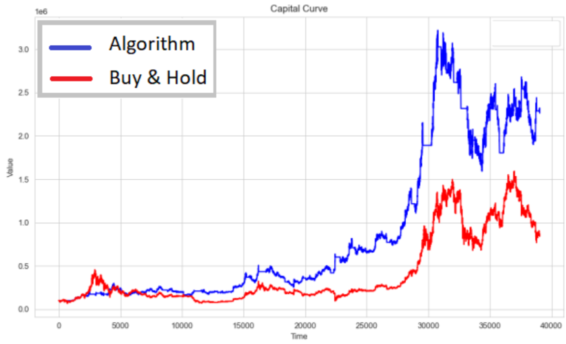


(f) NEOUSDT Capital Curve for Testing Set

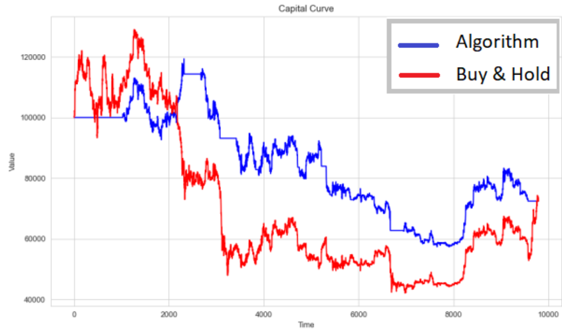
Figure 3.13: Capital Curves for $\mathrm{Calmar_{Ratio}}$-Optimized Strategy vs Buy & Hold

Our strategy demonstrated superior performance compared to the 'buy and hold' approach in both in-sample and out-of-sample scenarios, with the exception being the BTCUSDT pair during testing.

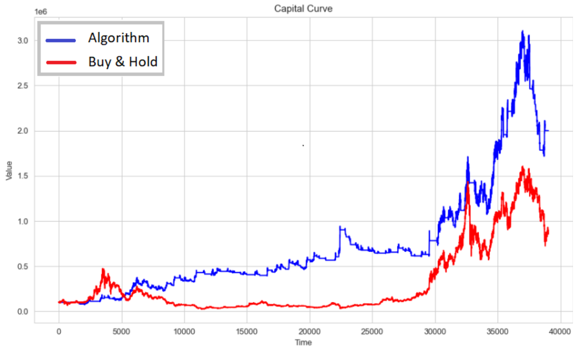
R2$_{\text{LogReturns}}$-Optimized Strategy



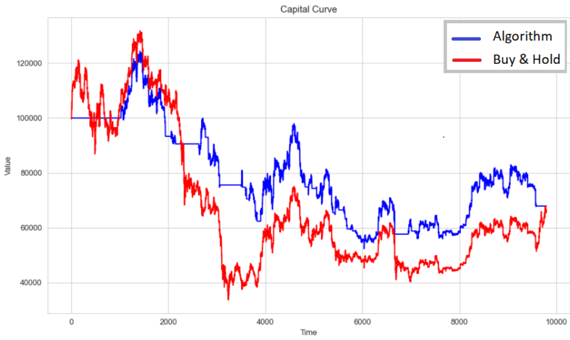
(a) BTCUSDT Capital Curve for Training Set



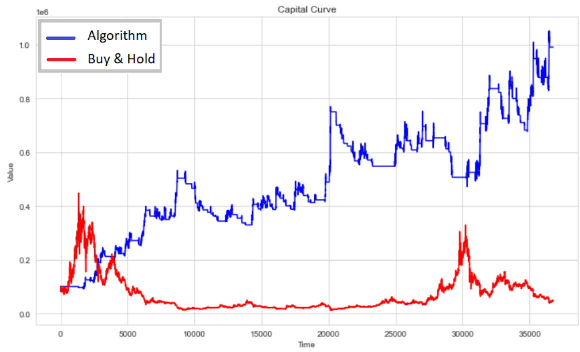
(b) BTCUSDT Capital Curve for Testing Set



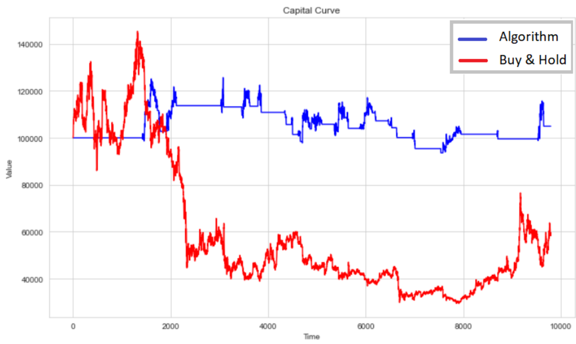
(c) ETHUSDT Capital Curve for Training Set



(d) ETHUSDT Capital Curve for Testing Set



(e) NEOUSDT Capital Curve for Training Set

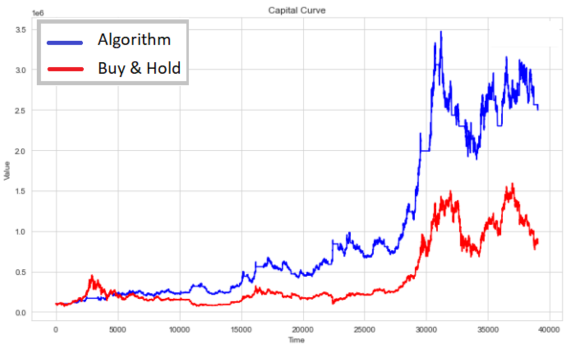


(f) NEOUSDT Capital Curve for Testing Set

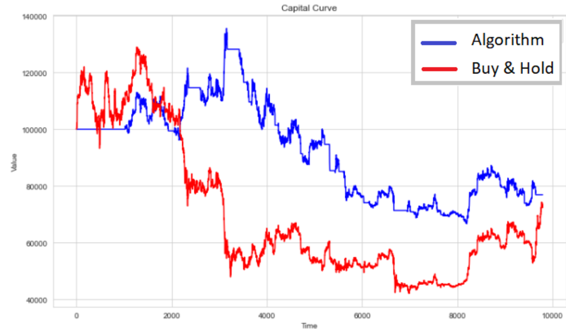
Figure 3.14: Capital Curves for R2$_{\text{LogReturns}}$-Optimized Strategy vs Buy & Hold

Our strategy exhibited superior performance when compared to the 'buy and hold' approach in both in-sample and out-of-sample scenarios. However, it's worth noting that this outperformance did not extend to the BTCUSDT pair during our testing, resulting in a more even set of results.

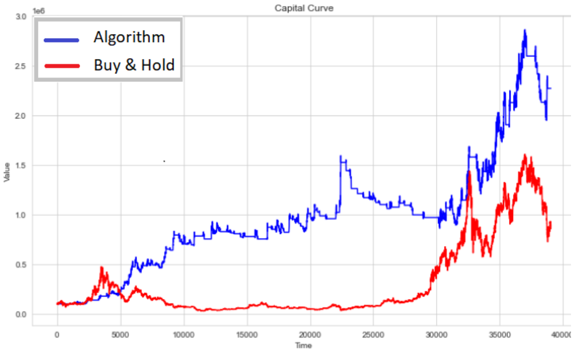
Multi$_{\text{Objective}}$-Optimized Strategy



(a) BTCUSDT Capital Curve for Training Set

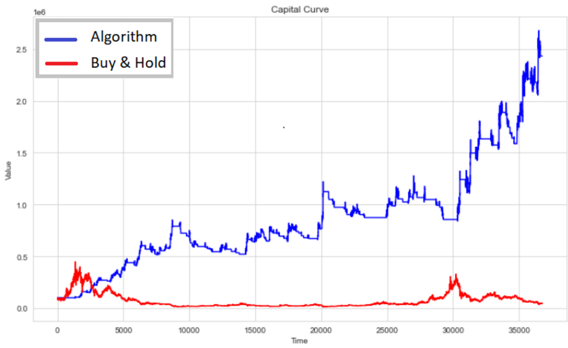


(b) BTCUSDT Capital Curve for Testing Set

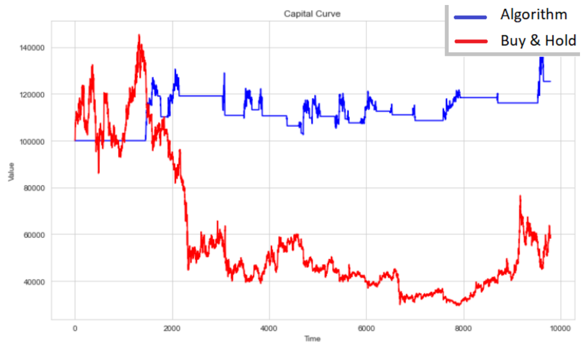


(c) ETHUSDT Capital Curve for Training Set

(d) ETHUSDT Capital Curve for Testing Set



(e) NEOUSDT Capital Curve for Training Set



(f) NEOUSDT Capital Curve for Testing Set

Figure 3.15: Capital Curves for Multi$_{\text{Objective}}$-Optimized Strategy vs Buy & Hold

Our strategy displayed superior performance in both in-sample and out-of-sample scenarios when contrasted with the 'buy and hold' approach.

In-Sample Analysis

Performance Metrics of the Optimized Trading Strategies

Based on the in-sample results, the Sharpe Ratio-Optimized and Sortino Ratio-Optimized strategies performed well for all three assets, with Sharpe Ratio-Optimized outperforming Sortino Ratio-Optimized for BTCUSDT and NEOUSDT. The Calmar Ratio-Optimized strategy performed well for BTCUSDT and NEOUSDT, while R2LogReturns-Optimized strategy performed well for BTCUSDT and ETHUSDT. The MultiObjective-Optimized strategy had mixed performance across the three assets.

Strategy	Asset	$\text{Sharpe}_{\text{Ratio}}$	$\text{Sortino}_{\text{Ratio}}$	$\text{Calmar}_{\text{Ratio}}$	$\text{R2}_{\text{LogReturns}}$
$\text{Sharpe}_{\text{Ratio}}$-Optimized	BTCUSDT	1.50	1.56	1.61	0.94
	ETHUSDT	1.61	1.43	1.63	0.82
	NEOUSDT	1.51	1.17	1.78	0.71
$\text{Sortino}_{\text{Ratio}}$-Optimized	BTCUSDT	1.57	1.57	2.31	0.95
	ETHUSDT	1.78	1.69	1.82	0.68
	NEOUSDT	1.55	1.20	1.74	0.70
$\text{Calmar}_{\text{Ratio}}$-Optimized	BTCUSDT	1.71	1.79	2.09	0.95
	ETHUSDT	1.77	1.56	1.95	0.8
	NEOUSDT	1.43	1.00	1.84	0.77
$\text{R2}_{\text{LogReturns}}$-Optimized	BTCUSDT	1.50	1.56	1.70	0.91
	ETHUSDT	1.47	1.25	1.84	0.90
	NEOUSDT	1.38	0.96	1.68	0.75
$\text{Multi}_{\text{Objective}}$-Optimized	BTCUSDT	1.56	1.63	1.91	0.94
	ETHUSDT	1.65	1.36	1.78	0.76
	NEOUSDT	1.77	1.36	2.26	0.79

Table 3.1: Performance Metrics of the Optimized Trading Strategies on BTCUSDT, ETHUSDT, and NEOUSDT

Comparaison with Buy and Hold Strategy

It can be observed that all optimized trading strategies outperformed the buy and hold strategy for all three assets. Among the optimized strategies, the Calmar Ratio-Optimized strategy had the highest total returns for BTCUSDT and NEOUSDT, while the Sortino Ratio-Optimized strategy had the highest annual returns for ETHUSDT. On the other hand, the Multi-Objective-Optimized strategy had the lowest total returns for NEOUSDT, indicating that this strategy might not be the best option for this particular asset. Overall, these results suggest that the optimized trading strategies can potentially provide better returns than a simple buy and hold strategy for cryptocurrency trading. However, it is important to note that these results are based on historical data and may not necessarily reflect future performance.

Out-Of-Sample Analysis

Performance Metrics of the Optimized Trading Strategies

Overall, the performance of the strategies for out-of-sample data is worse compared

Asset	$Sharpe_{optimized}$	$Sortino_{optimized}$	$Calmar_{optimized}$	Multi-Objective	$R2_{optimized}$	Buy and Hold
BTCUSDT	2024.42%	2181.92%	3630.74%	2170.34%	2405.17%	731.66%
ETHUSDT	2550.2%	4574.45%	2106.5%	1900%	2169.22%	267.94%
NEOUSDT	1347.10%	1478.11%	1007.56%	2333.75%	891.3%	-18.65%

Table 3.2: Total Cumulative Returns of the BTCUSDT, ETHUSDT, and NEOUSDT Trading Strategies Compared to Buy and Hold

to the in-sample data. The Sharpe Ratio-Optimized and Sortino Ratio-Optimized strategies had negative Sharpe and Sortino ratios for most assets, indicating that they did not perform well in generating returns relative to their risk. The Calmar Ratio-Optimized strategy had negative Sharpe ratios for BTCUSDT and ETHUSDT but performed relatively well for NEOUSDT. The R2LogReturns-Optimized strategy also had negative Sharpe ratios for BTCUSDT and ETHUSDT but performed well for NEOUSDT. The MultiObjective-Optimized strategy had mixed performance across the three assets, with positive Sharpe ratios for NEOUSDT and ETHUSDT but negative Sharpe ratios for BTCUSDT.

Overall, it can be concluded that the optimized trading strategies did not perform well in generating returns for out-of-sample data.

Strategy	Asset	$Sharpe_{Ratio}$	$Sortino_{Ratio}$	$Calmar_{Ratio}$	$R2_{LogReturns}$
$Sharpe_{Ratio}$-Optimized	BTCUSDT	-0.30	-0.32	-0.21	-0.54
	ETHUSDT	-0.57	-0.37	-0.38	-0.57
	NEOUSDT	0.14	0.11	0.20	0.24
$Sortino_{Ratio}$-Optimized	BTCUSDT	-0.18	-0.18	-0.13	-0.51
	ETHUSDT	-0.75	-0.77	-0.68	-0.70
	NEOUSDT	0.11	0.08	0.17	-0.24
$Calmar_{Ratio}$-Optimized	BTCUSDT	-1.47	-1.43	-0.84	-0.84
	ETHUSDT	-0.36	-0.37	-0.38	-0.57
	NEOUSDT	0.39	0.30	0.58	0.02
$R2_{LogReturns}$-Optimized	BTCUSDT	-0.61	-0.66	-0.42	-0.66
	ETHUSDT	-0.40	-0.40	-0.35	0.52
	NEOUSDT	0.26	0.20	0.38	0.11
$Multi_{Objective}$-Optimized	BTCUSDT	-0.47	-0.48	-0.32	-0.61
	ETHUSDT	0.18	0.18	0.22	-0.01
	NEOUSDT	0.75	0.55	1.18	0.25

Table 3.3: Performance Metrics of the Optimized Trading Strategies on BTCUSDT, ETHUSDT, and NEOUSDT

Comparaison with Buy and Hold Strategy The table shows the total returns of trading strategies for three assets, BTCUSDT, ETHUSDT, and NEOUSDT, compared to a buy and hold strategy. The trading strategies are optimized using different metrics, including Sharpe ratio, Sortino ratio, Calmar ratio, Multi-Objective, and R2 score.

For BTCUSDT, all optimized strategies except for NEOUSDT underperformed compared to buy and hold, with total returns ranging from -17.84% to -47.58%. The Calmar-optimized strategy had the worst performance, while the Multi-Objective and R2-optimized strategies had relatively better but still negative returns. For ETHUSDT, the Sharpe, Sortino, and Multi-Objective optimized strategies all had negative total returns ranging from -21.19% to -42.91%. However, the Calmar and R2-optimized strategies had slightly better but still negative returns compared to buy and hold, with returns of -29.29% and -1.88% respectively.

For NEOUSDT, the Calmar, Multi-Objective, and R2-optimized strategies outperformed buy and hold, with total returns of 10.03%, 4.93%, and 25.32%, respectively. The Sharpe and Sortino-optimized strategies had negative returns.

Overall, the table shows that optimized strategies did not always outperform buy and hold, and in some cases, they had significantly worse performance. It is also important to note that past performance is not necessarily indicative of future results,

Asset	$\text{Sharpe}_{\text{optimized}}$	$\text{Sortino}_{\text{optimized}}$	$\text{Calmar}_{\text{optimized}}$	Multi-Objective	$\text{R2}_{\text{optimized}}$	Buy and Hold
BTCUSDT	-17.84%	-12.70%	-47.58%	-27.58%	-23.08%	-46.12%
ETHUSDT	-21.19%	-42.91%	-29.29%	-32.09%	-1.88%	-42.93%
NEOUSDT	0.35%	-0.80%	10.03%	4.93%	25.32%	-18.65%

Table 3.4: Total Cumulative Returns of the BTCUSDT, ETHUSDT, and NEOUSDT Trading Strategies Compared to Buy and Hold

3.5.3 Walk Forward Optimization (WFO)

In this section, we have divided the data into in-sample and out-of-sample subsamples using the walk-forward optimization approach. The starting date of the walk-forward optimization is August 17th, 2017, and the end date is February 17th, 2023. We used a rolling period of 6 months, with each in-sample period consisting of 1 year of data and each out-of-sample period consisting of 6 months of data.During the walk-forward optimization, we used the Sortino ratio as an objective function.

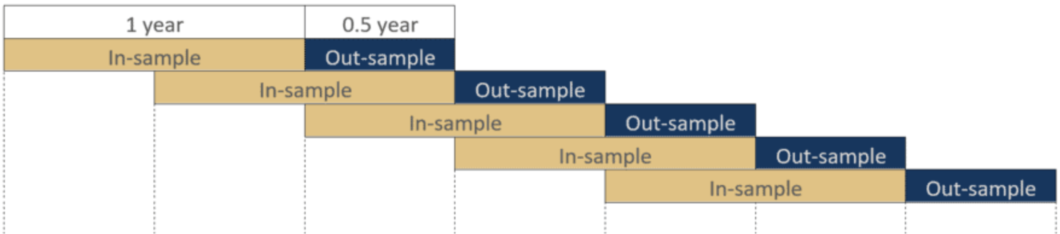
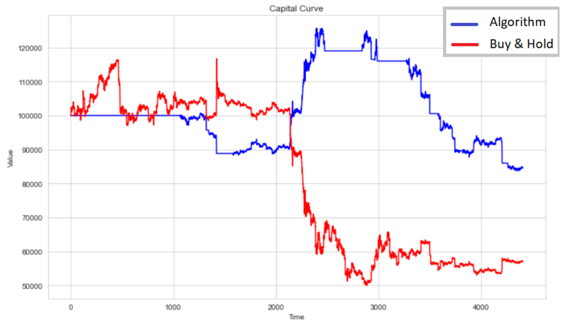


Figure 3.16: Walk Forward Optimization

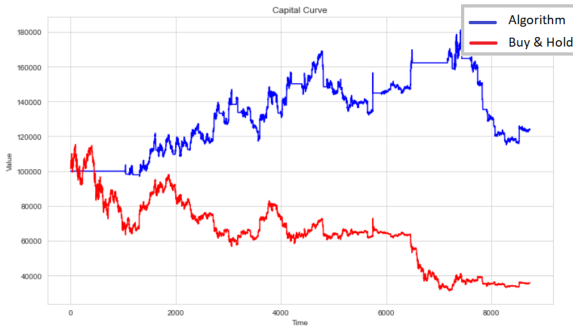
3.5.4 Simulation Results Of The WFO

BTCUSDT

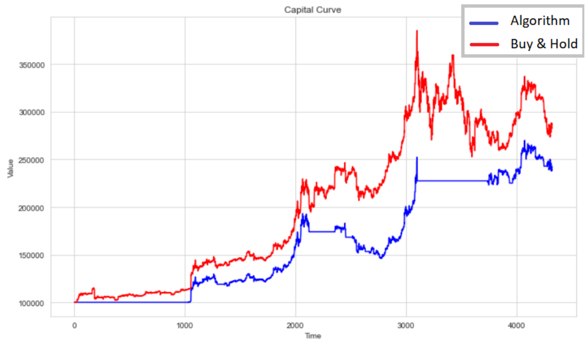
(a) Capital Curve for In-Sample period 1



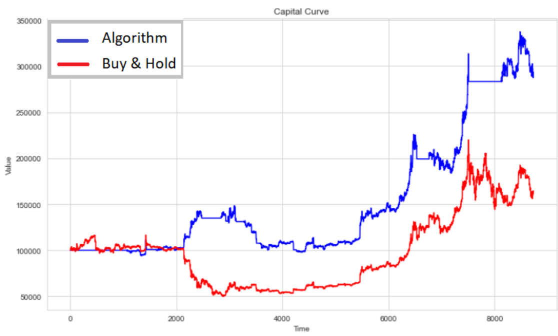
(b) Capital Curve for Out-Of-Sample period 1



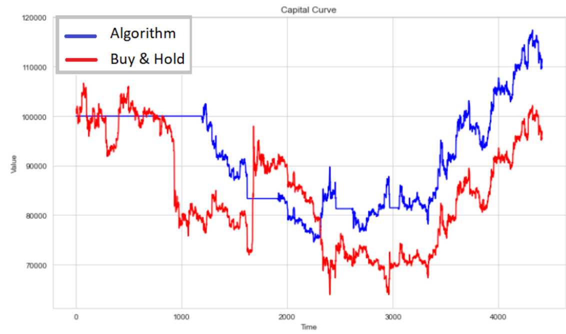
(c) Capital Curve for In-Sample period 2



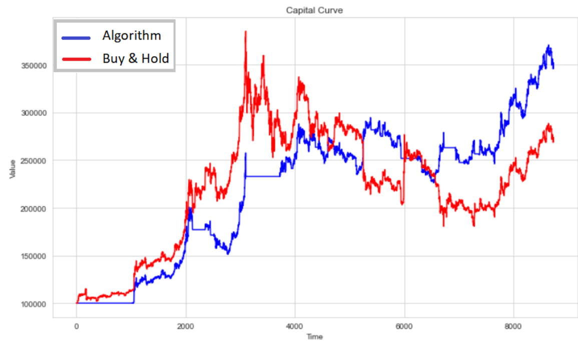
(d) Capital Curve for Out-Of-Sample period 2



(e) Capital Curve for In-Sample period 3

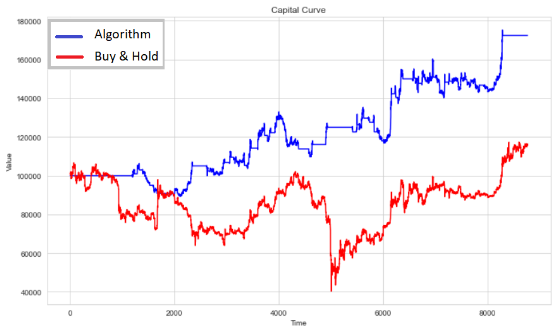


(f) Capital Curve for Out-Of-Sample period 3

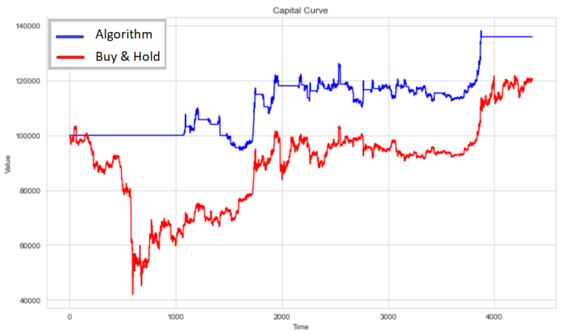


(g) Capital Curve for In-Sample period 4

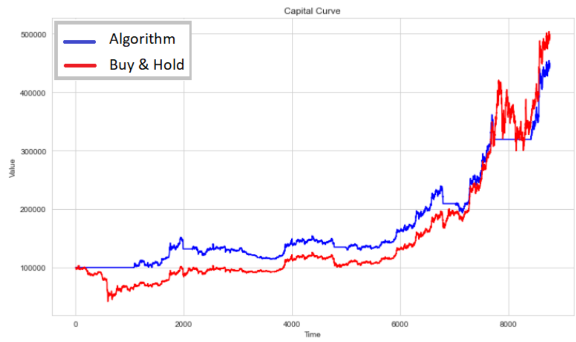
(h) Capital Curve for Out-Of-Sample period 4



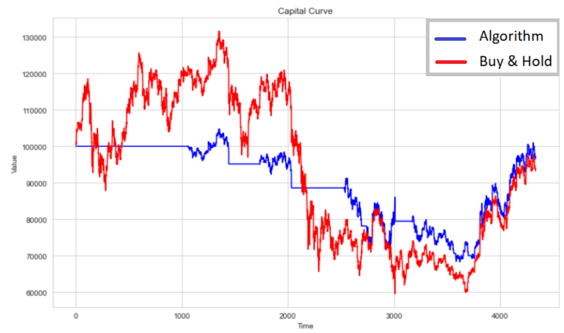
(i) Capital Curve for In-Sample period 5



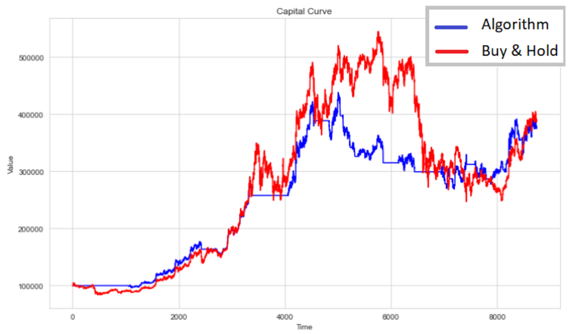
(j) Capital Curve for Out-Of-Sample period 5



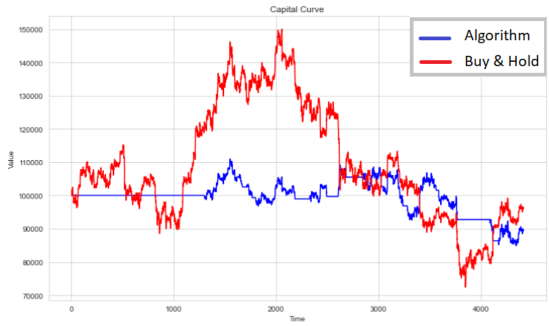
(k) Capital Curve for In-Sample period 6



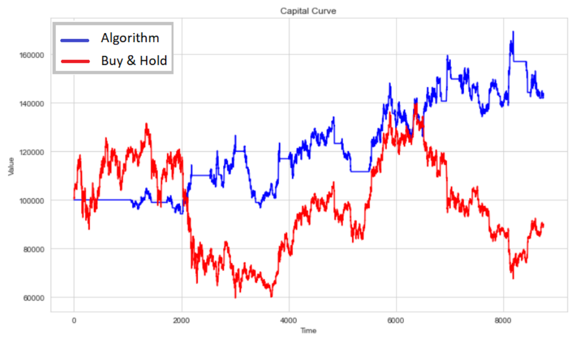
(l) Capital Curve for Out-Of-Sample period 6



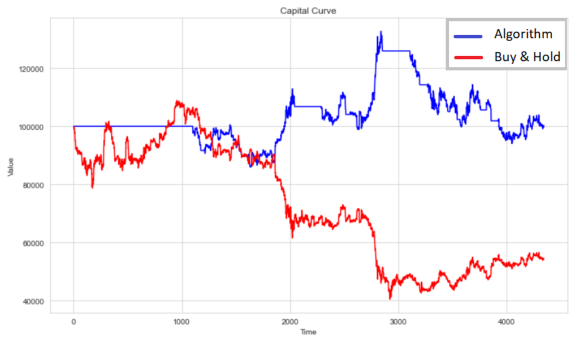
(m) Capital Curve for In-Sample period 7



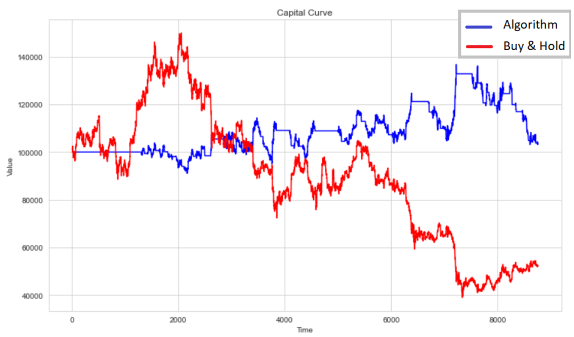
(n) Capital Curve for Out-Of-Sample period 7



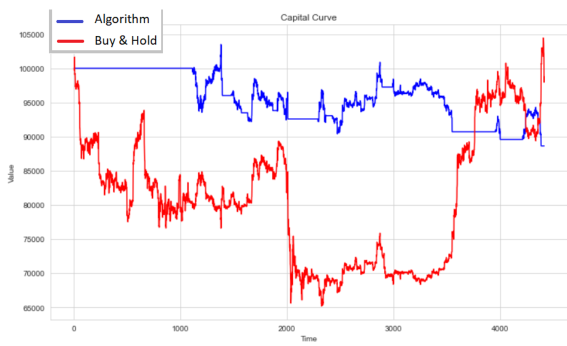
(o) Capital Curve for In-Sample period 8



(p) Capital Curve for Out-Of-Sample period 8



(q) Capital Curve for In-Sample period 9



(r) Capital Curve for Out-Of-Sample period 9

Figure 3.19: WFO for $\text{Sortino}_{\text{Ratio}}$-Optimized Strategy for BTCUSDT vs Buy & Hold

In seven out of nine cases, our algorithm outperformed the "buy and hold" strategy in the in-sample scenario. However, it's important to note that our strategy surpassed the "buy and hold" strategy only five out of nine times. Additionally, it's worth mentioning that losses were amortized in the cases where they occurred.

WFO Analysis for BTCUSDT

The table shows that the strategy performed well in some in-sample periods, but poorly in some out-of-sample periods. This indicates that the strategy was inconsistent in generating profits. In terms of in-sample results, the Sortino-optimized strategy outperformed the Buy and Hold strategy in all but one period. However, the out-of-sample results were mixed, with the Sortino-optimized strategy sometimes outperforming Buy and Hold and sometimes underperforming. For example, in the first out-of-sample period, the Sortino-optimized strategy underperformed the Buy and Hold strategy. While the Sortino-optimized strategy can generate higher returns in certain market conditions, it may not always outperform the Buy and Hold strategy in out-of-sample periods.

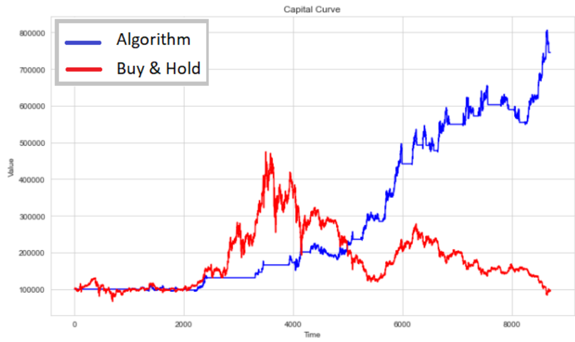
Dataset	Period	Sharpe	Sortino	Calmar	R2
In-Sample	Aug 17, 2017 - Aug 17, 2018	2.57	2.31	8.77	0.96
Out-of-Sample	Aug 17, 2018 - Feb 17, 2019	-0.92	-0.69	-0.84	-0.01
In-Sample	Feb 17, 2018 - Feb 17, 2019	0.67	0.65	0.88	0.5
Out-of-Sample	Feb 17, 2019 - Aug 17, 2019	3.72	3.36	7.91	0.93
In-Sample	Aug 17, 2018 - Aug 17, 2019	2.41	2.34	3.54	0.71
Out-of-Sample	Aug 17, 2019 - Feb 17, 2020	0.71	0.72	0.99	0.04
In-Sample	Feb 17, 2019 - Feb 17, 2020	2.86	3.03	5.56	0.79
Out-of-Sample	Feb 17, 2020 - Aug 17, 2020	1.03	0.95	1.43	0.01
In-Sample	Aug 17, 2019 - Aug 17, 2020	1.86	1.58	3.40	0.87
Out-of-Sample	Aug 17, 2020 - Feb 17, 2021	2.05	1.64	4.75	0.76
In-Sample	Feb 17, 2020 - Feb 17, 2021	3.31	3.41	6.52	0.81
Out-of-Sample	Feb 17, 2021 - Aug 17, 2021	0.03	0.03	0.07	-0.54
In-Sample	Aug 17, 2020 - Aug 17, 2021	2.83	2.94	3.75	0.74
Out-of-Sample	Aug 17, 2021 - Feb 17, 2022	-0.55	-0.51	-0.70	-0.19
In-Sample	Feb 17, 2021 - Feb 17, 2022	1.07	1.16	1.87	0.86
Out-of-Sample	Feb 17, 2022 - Aug 17, 2022	0.19	0.20	0.31	0.14
In-Sample	Aug 17, 2021 - Aug 17, 2022	0.22	0.22	0.34	0.64
Out-of-Sample	Aug 17, 2022 - Feb 17, 2023	-1.07	-0.86	-1.51	-0.61

Table 3.5: WFO Performance Metrics for BTCUSDT

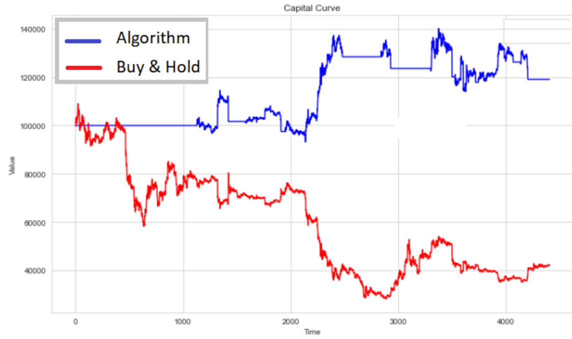
Dataset	Period	$\text{Sortino}_{\text{Optimized}}$	Buy and Hold
In-Sample	Aug 17, 2017 - Aug 17, 2018	290.43%	50.1%
Out-of-Sample	Aug 17, 2018 - Feb 17, 2019	-15.40%	-47.64%
In-Sample	Feb 17, 2018 - Feb 17, 2019	23.79%	-64.14%
Out-of-Sample	Feb 17, 2019 - Aug 17, 2019	142.74%	171.15%
In-Sample	Aug 17, 2018 - Aug 17, 2019	193.48%	53.91%
Out-of-Sample	Aug 17, 2019 - Feb 17, 2020	10.88%	-5.84%
In-Sample	Feb 17, 2019 - Feb 17, 2020	250.14%	181.74%
Out-of-Sample	Feb 17, 2020 - Aug 17, 2020	20.19%	19.17%
In-Sample	Aug 17, 2019 - Aug 17, 2020	72.41%	18.21%
Out-of-Sample	Aug 17, 2020 - Feb 17, 2021	35.90%	20.11%
In-Sample	Feb 17, 2020 - Feb 17, 2021	348.55%	352.47%
Out-of-Sample	Feb 17, 2021 - Aug 17, 2021	-3.49%	-6.74%
In-Sample	Aug 17, 2020 - Aug 17, 2021	274.51%	269.12%
Out-of-Sample	Aug 17, 2021 - Feb 17, 2022	-10.35%	-4.55%
In-Sample	Feb 17, 2021 - Feb 17, 2022	43.52%	-10.11 %
Out-of-Sample	Feb 17, 2022 - Aug 17, 2022	0.31%	-48.22%
In-Sample	Aug 17, 2021 - Aug 17, 2022	3.17%	-47.13%
Out-of-Sample	Aug 17, 2022 - Feb 17, 2023	-11.37%	-7.34%

Table 3.6: Performance Comparison of Buy and Hold and Optimized Strategy for BT-CUSDT

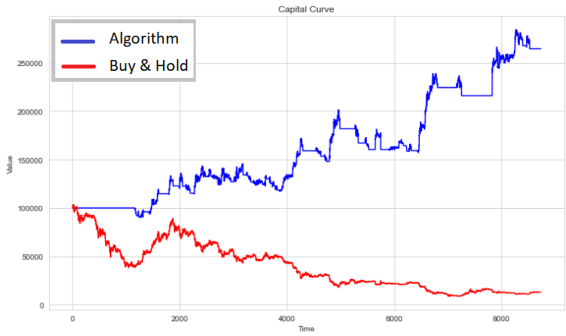
ETHUSDT



(a) Capital Curve for In-Sample period 1

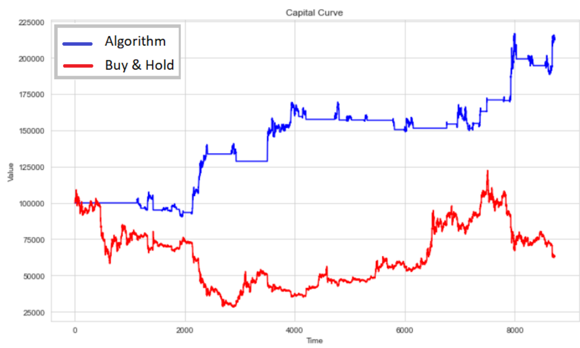


(b) Capital Curve for Out-Of-Sample period 1

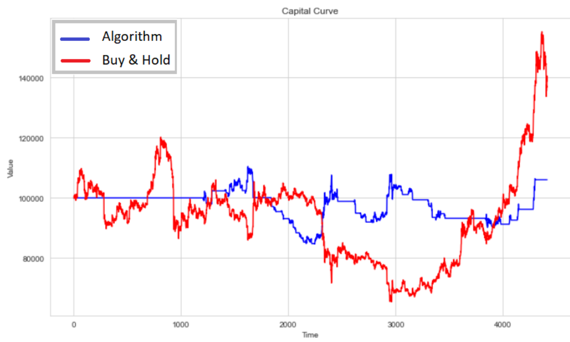


(c) Capital Curve for In-Sample period 2

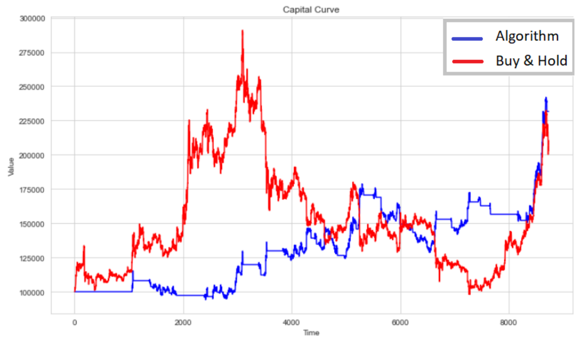
(d) Capital Curve for Out-Of-Sample period 2



(e) Capital Curve for In-Sample period 3

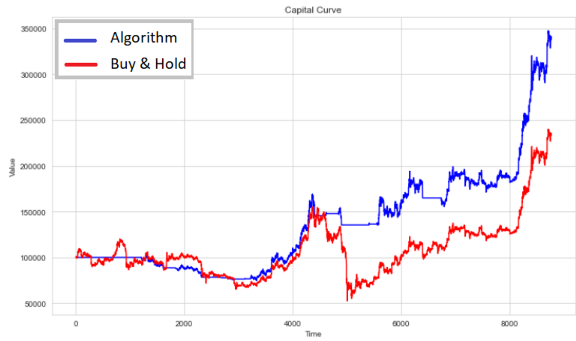


(f) Capital Curve for Out-Of-Sample period 3

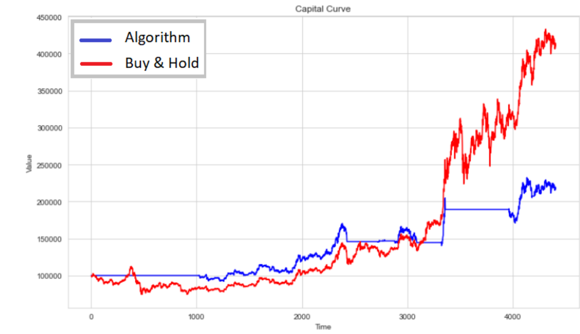


(g) Capital Curve for In-Sample period 4

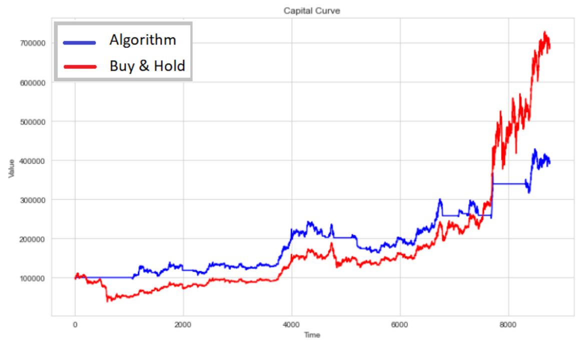
(h) Capital Curve for Out-Of-Sample period 4



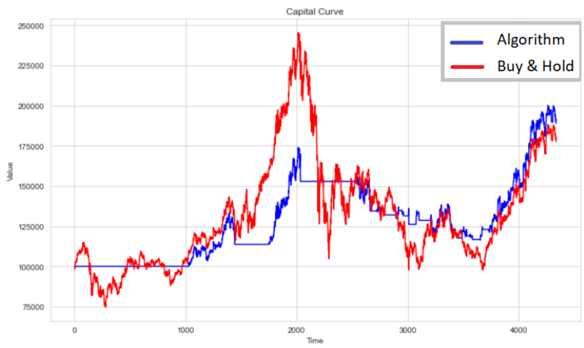
(i) Capital Curve for In-Sample period 5



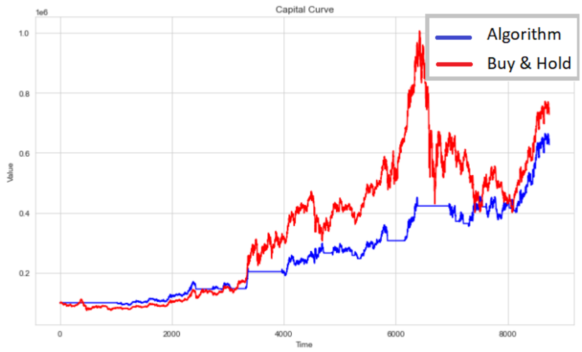
(j) Capital Curve for Out-Of-Sample period 5



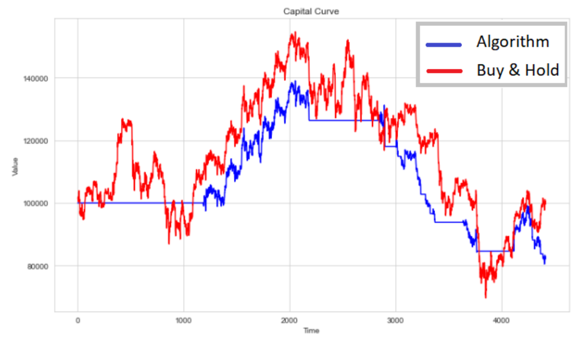
(k) Capital Curve for In-Sample period 6



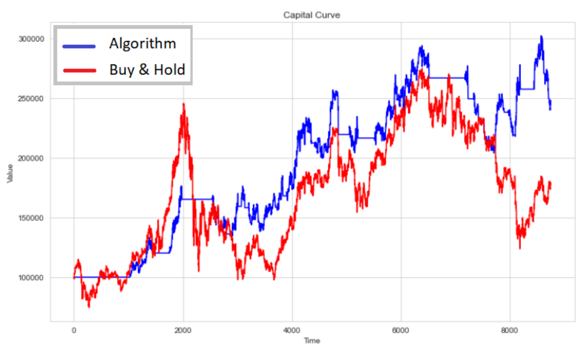
(l) Capital Curve for Out-Of-Sample period 6



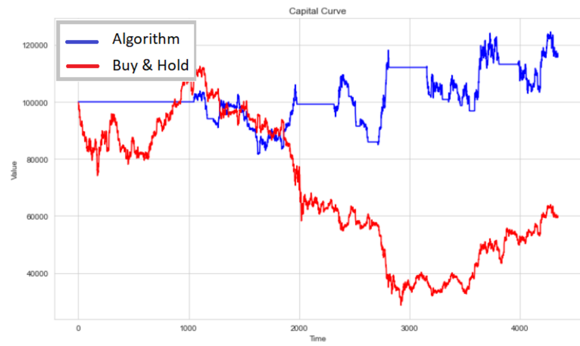
(m) Capital Curve for In-Sample period 7



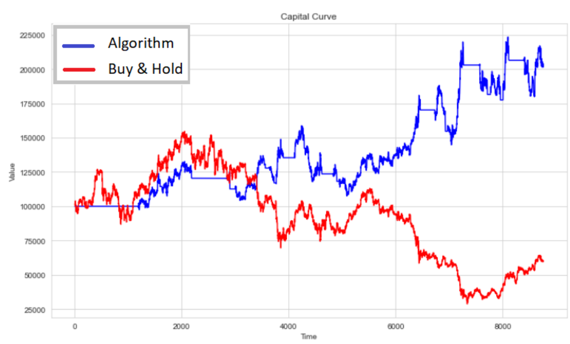
(n) Capital Curve for Out-Of-Sample period 7



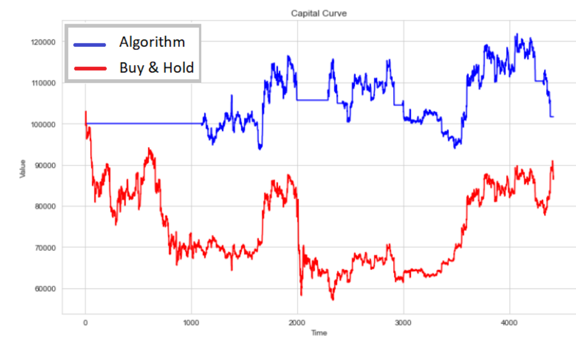
(o) Capital Curve for In-Sample period 8



(p) Capital Curve for Out-Of-Sample period 8



(q) Capital Curve for In-Sample period 9



(r) Capital Curve for Out-Of-Sample period 9

Figure 3.22: WFO for $\text{Sortino}_{\text{Ratio}}$-Optimized Strategy for ETHUSDT vs Buy & Hold

In seven out of nine cases, our algorithm outperformed the "buy and hold" strategy in the in-sample scenario. It's important to note, however, that our strategy surpassed the "buy and hold" strategy only five out of nine times. Additionally, it's worth mentioning that in most of the cases, there were no capital losses, but rather underperformance in terms of gains. Furthermore, losses were amortized in the cases where they occurred.

WFO Analysis for ETHUSDT

Looking at the first table, we can see that the performance metrics for the in-sample periods are generally better than those for the out-of-sample periods. This suggests that the algorithm may have overfit to the in-sample data and may not perform as well on new, unseen data.

However, there are some exceptions to this general trend. For example, the performance metrics for the out-of-sample period from Aug 17, 2020 to Feb 17, 2021 are better than those for the in-sample period immediately preceding it. Similarly, the performance metrics for the out-of-sample period from Aug 17, 2021 to Feb 17, 2022 are worse than those for the in-sample period immediately preceding it.

Looking at the second table, we can see that the Sortino-optimized performance metrics generally outperform the Buy and Hold performance metrics for the same periods. This suggests that the algorithm is able to identify and exploit market inefficiencies to generate excess returns compared to a simple Buy and Hold strategy. However, there are some periods where the Buy and Hold strategy outperforms the Sortino-optimized strategy, particularly in the out-of-sample periods from Feb 17, 2019 to Aug 17, 2019 and from Aug 17, 2022 to Feb 17, 2023.

Overall, the performance of the algorithmic trading system appears to be mixed. While it is able to generate excess returns compared to a simple Buy and Hold strategy in many periods, there are also periods where it underperforms. Additionally, the out-of-sample performance is generally worse than the in-sample performance, which suggests that the algorithm may be overfitting to the data.

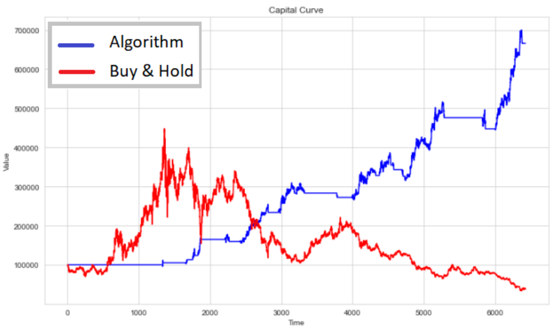
Dataset	Period	Sharpe	Sortino	Calmar	R2
In-Sample	Aug 17, 2017 - Aug 17, 2018	3.43	3.24	10.56	0.94
Out-of-Sample	Aug 17, 2018 - Feb 17, 2019	0.94	0.71	2.48	0.65
In-Sample	Feb 17, 2018 - Feb 17, 2019	2.18	1.88	5.03	0.92
Out-of-Sample	Feb 17, 2019 - Aug 17, 2019	1.71	1.42	3.32	0.37
In-Sample	Aug 17, 2018 - Aug 17, 2019	2.16	1.55	5.26	0.82
Out-of-Sample	Aug 17, 2019 - Feb 17, 2020	0.49	0.38	0.70	0.13
In-Sample	Feb 17, 2019 - Feb 17, 2020	2.08	2.05	3.36	0.82
Out-of-Sample	Feb 17, 2020 - Aug 17, 2020	-0.08	-0.07	-0.07	-0.38
In-Sample	Aug 17, 2019 - Aug 17, 2020	2.61	2.53	4.83	0.75
Out-of-Sample	Aug 17, 2020 - Feb 17, 2021	3.06	2.87	9.57	0.90
In-Sample	Feb 17, 2020 - Feb 17, 2021	2.56	2.66	4.68	0.90
Out-of-Sample	Feb 17, 2021 - Aug 17, 2021	2.39	1.99	4.44	0.57
In-Sample	Aug 17, 2020 - Aug 17, 2021	3.12	3.30	9.56	0.96
Out-of-Sample	Aug 17, 2021 - Feb 17, 2022	-0.73	-0.63	-0.68	-0.04
In-Sample	Feb 17, 2021 - Feb 17, 2022	1.81	1.81	3.51	0.85
Out-of-Sample	Feb 17, 2022 - Aug 17, 2023	0.84	0.86	2.03	0.27
In-Sample	Aug 17, 2021 - Aug 17, 2022	1.55	1.74	2.67	0.82
Out-of-Sample	Aug 17, 2022 - Feb 17, 2023	0.26	0.27	0.62	0.36

Table 3.7: WFO Performance Metrics for ETHUSDT

Dataset	Period	$\text{Sortino}_{\text{Optimized}}$	Buy and Hold
In-Sample	Aug 17, 2017 - Aug 17, 2018	645.18%	0.18%
Out-of-Sample	Aug 17, 2018 - Feb 17, 2019	19.09%	-58.11%
In-Sample	Feb 17, 2018 - Feb 17, 2019	164.51%	-80.74%
Out-of-Sample	Feb 17, 2019 - Aug 17, 2019	39.57%	49.11%
In-Sample	Aug 17, 2018 - Aug 17, 2019	113.75%	-40.22%
Out-of-Sample	Aug 17, 2019 - Feb 17, 2020	6.03%	33.08%
In-Sample	Feb 17, 2019 - Feb 17, 2020	131.62%	101.1%
Out-of-Sample	Feb 17, 2020 - Aug 17, 2020	-6.87%	65.11%
In-Sample	Aug 17, 2019 - Aug 17, 2020	238.41%	137.22%
Out-of-Sample	Aug 17, 2020 - Feb 17, 2021	117.12%	306.49%
In-Sample	Feb 17, 2020 - Feb 17, 2021	294.17%	611.12%
Out-of-Sample	Feb 17, 2021 - Aug 17, 2021	88.91%	77.2%
In-Sample	Aug 17, 2020 - Aug 17, 2021	526.94%	654.55%
Out-of-Sample	Aug 17, 2021 - Feb 17, 2022	-17.16%	-1.46%
In-Sample	Feb 17, 2021 - Feb 17, 2022	146.67%	74.19%
Out-of-Sample	Feb 17, 2022 - Aug 17, 2022	17.07%	-40.09%
In-Sample	Aug 17, 2021 - Aug 17, 2022	103.94%	-45.22%
Out-of-Sample	Aug 17, 2022 - Feb 17, 2023	1.64%	-11.33%

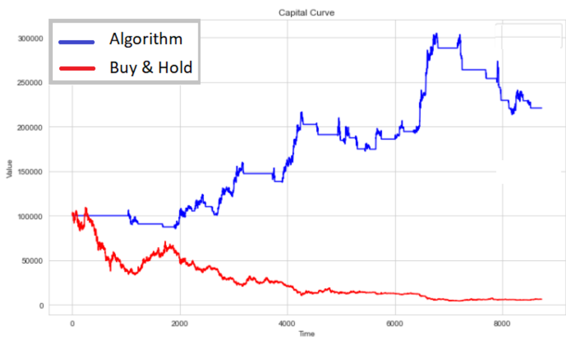
Table 3.8: Performance Comparison of Buy and Hold and Optimized Strategy for ETHUSDT

NEOUSDT



(a) Capital Curve for In-Sample period 1

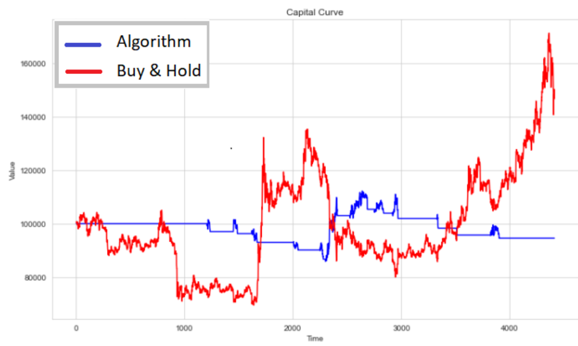
(b) Capital Curve for Out-Of-Sample period 1



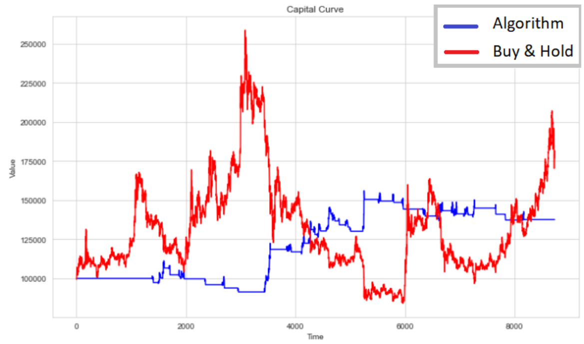
(c) Capital Curve for In-Sample period 2

(d) Capital Curve for Out-Of-Sample period 2

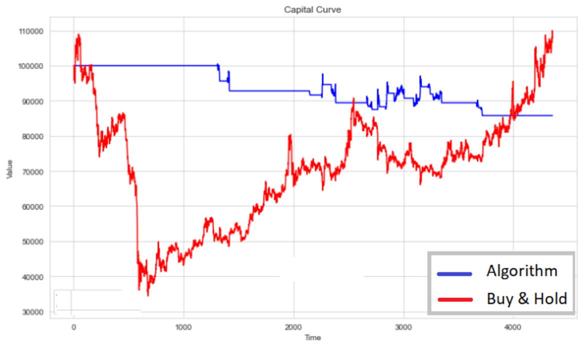
(e) Capital Curve for In-Sample period 3



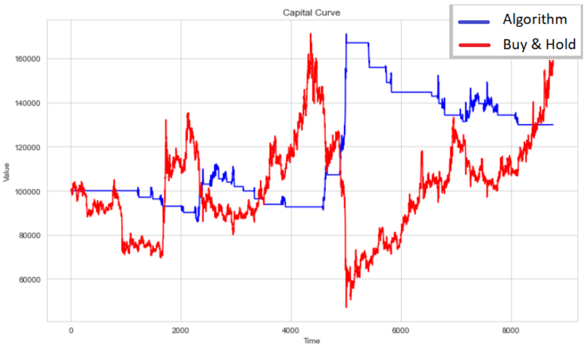
(f) Capital Curve for Out-Of-Sample period 3



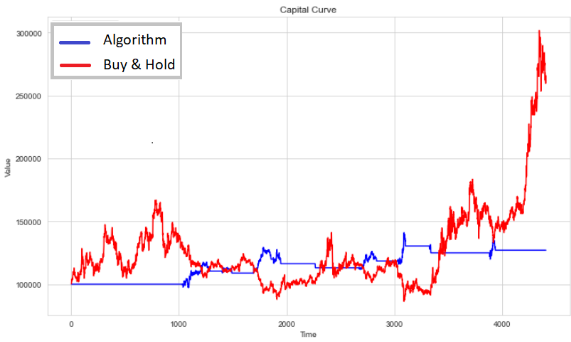
(g) Capital Curve for In-Sample period 4



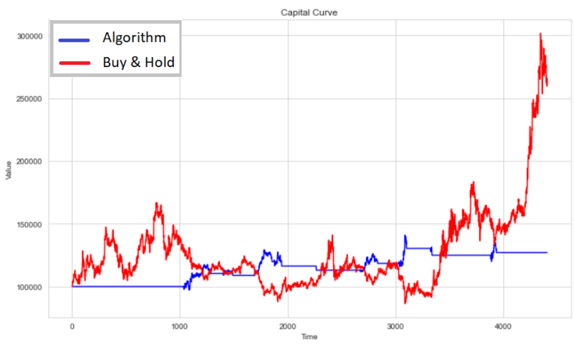
(h) Capital Curve for Out-Of-Sample period 4



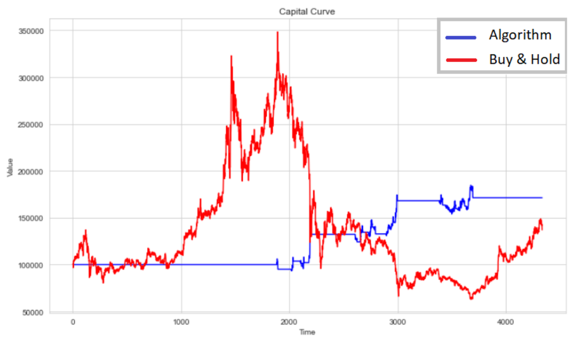
(i) Capital Curve for In-Sample period 5



(j) Capital Curve for Out-Of-Sample period 5

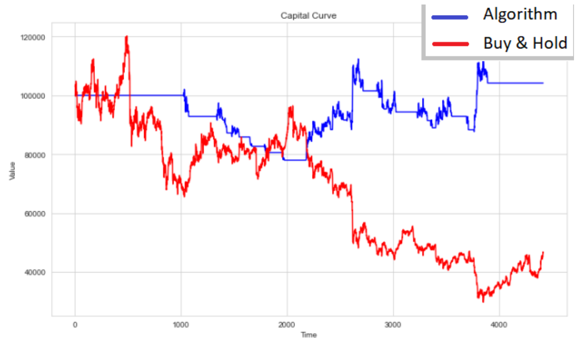


(k) Capital Curve for In-Sample period 6

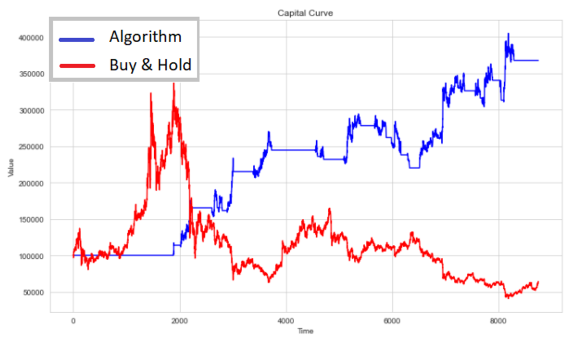


(l) Capital Curve for Out-Of-Sample period 6

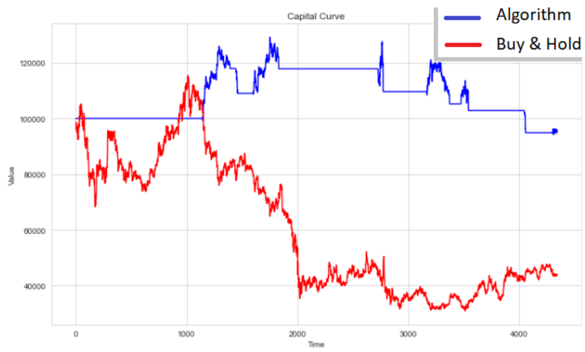
(m) Capital Curve for In-Sample period 7



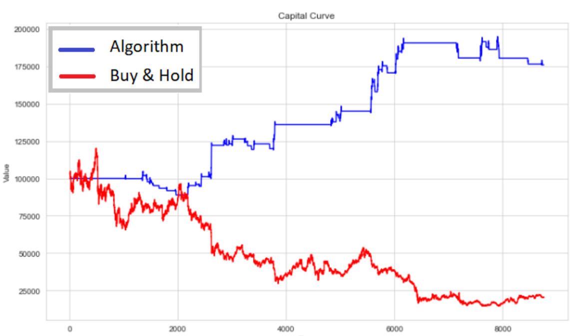
(n) Capital Curve for Out-Of-Sample period 7



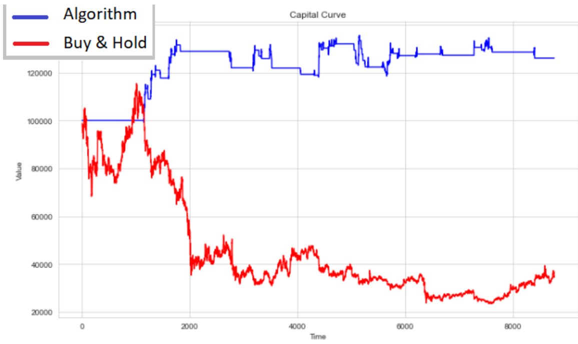
(o) Capital Curve for In-Sample period 8



(p) Capital Curve for Out-Of-Sample period 8



(q) Capital Curve for In-Sample period 9



(r) Capital Curve for Out-Of-Sample period 9

Figure 3.25: WFO for $\text{Sortino}_{\text{Ratio}}$-Optimized Strategy for NEOUSDT vs Buy & Hold

In five out of nine instances, our algorithm demonstrated better performance than the "buy and hold" strategy during the in-sample evaluation, without any capital losses. It's essential to emphasize that our strategy outperformed the "buy and hold" approach only five times out of nine, and capital losses were encountered in just two of those scenarios

WFO Analysis for NEOUSDT

Looking at the first table, it is apparent that the in-sample performance of the trading strategy was better than the out-of-sample performance. The Sharpe Ratio, Sortino Ratio, and Calmar Ratio are significantly higher in the in-sample period compared to the out-of-sample period. The Returns also show a similar trend, with higher returns achieved during the in-sample period.

The second table provides further insights into the performance of the trading strategy by comparing its Sortino-optimized returns to a buy and hold approach. In general, the Sortino-optimized returns are much higher than the buy and hold returns, which indicates that the trading strategy is successful in generating positive returns. However, the trading strategy's performance was inconsistent, with some out-of-sample periods showing negative returns.

In conclusion, the tables suggest that the trading strategy performed well in the in-sample periods but struggled to maintain the same level of performance in the out-of-sample periods. Additionally, the trading strategy's performance was inconsistent, with some out-of-sample periods showing negative returns.

Dataset	Period	Sharpe	Sortino	Calmar	R2
In-Sample	Aug 17, 2017 - Aug 17, 2018	4.27	3.32	15.22	0.95
Out-of-Sample	Aug 17, 2018 - Feb 17, 2019	0.79	0.50	0.88	0.35
In-Sample	Feb 17, 2018 - Feb 17, 2019	1.86	1.44	3.04	0.86
Out-of-Sample	Feb 17, 2019 - Aug 17, 2019	0.92	0.59	2.02	0.03
In-Sample	Aug 17, 2018 - Aug 17, 2019	1.16	0.71	1.15	0.01
Out-of-Sample	Aug 17, 2019 - Feb 17, 2020	-0.28	-0.16	-0.45	-0.02
In-Sample	Feb 17, 2019 - Feb 17, 2020	1.16	0.64	2.04	0.73
Out-of-Sample	Feb 17, 2020 - Aug 17, 2020	-1.17	-0.5	-1.89	-0.86
In-Sample	Aug 17, 2019 - Aug 17, 2020	0.91	0.60	1.31	0.54
Out-of-Sample	Aug 17, 2020 - Feb 17, 2021	1.45	0.93	3.61	0.83
In-Sample	Feb 17, 2020 - Feb 17, 2021	0.77	0.41	1.50	0.27
Out-of-Sample	Feb 17, 2021 - Aug 17, 2021	2.23	1.48	9.40	0.83
In-Sample	Aug 17, 2020 - Aug 17, 2021	1.76	1.35	2.79	0.50
Out-of-Sample	Aug 17, 2021 - Feb 17, 2022	0.38	0.34	0.75	0.6
In-Sample	Feb 17, 2021 - Feb 17, 2022	2.59	2.61	5.78	0.87
Out-of-Sample	Feb 17, 2022 - Aug 17, 2022	-0.16	-0.1	-0.17	-0.002
In-Sample	Aug 17, 2021 - Aug 17, 2022	2.32	1.27	3.94	0.89
Out-of-Sample	Aug 17, 2022 - Feb 17, 2023	1.11	0.58	2.03	0.42

Table 3.9: WFO Performance Metrics for NEOUSDT

Dataset	Period	$\text{Sortino}_{\text{Optimized}}$	Buy and Hold
In-Sample	Aug 17, 2017 - Aug 17, 2018	566.05%	-15.4%
Out-of-Sample	Aug 17, 2018 - Feb 17, 2019	12.41%	-50.1%
In-Sample	Feb 17, 2018 - Feb 17, 2019	120.92%	-98.1%
Out-of-Sample	Feb 17, 2019 - Aug 17, 2019	15.81%	15.91%
In-Sample	Aug 17, 2018 - Aug 17, 2019	44.43%	-40.77%
Out-of-Sample	Aug 17, 2019 - Feb 17, 2020	-5.47%	40.06%
In-Sample	Feb 17, 2019 - Feb 17, 2020	37.59%	72.47%
Out-of-Sample	Feb 17, 2020 - Aug 17, 2020	-14.22%	11.1%
In-Sample	Aug 17, 2019 - Aug 17, 2020	29.91%	61.47%
Out-of-Sample	Aug 17, 2020 - Feb 17, 2021	27.00%	124.64%
In-Sample	Feb 17, 2020 - Feb 17, 2021	22.24%	174.44%
Out-of-Sample	Feb 17, 2021 - Aug 17, 2021	71.35%	47.43%
In-Sample	Aug 17, 2020 - Aug 17, 2021	113.77%	291.11%
Out-of-Sample	Aug 17, 2021 - Feb 17, 2022	4.18%	-55.14%
In-Sample	Feb 17, 2021 - Feb 17, 2022	267.51%	-52.91%
Out-of-Sample	Feb 17, 2022 - Aug 17, 2022	-5.06%	-57.27%
In-Sample	Aug 17, 2021 - Aug 17, 2022	76.00%	-77.12%
Out-of-Sample	Aug 17, 2022 - Feb 17, 2023	26.05%	-64.11%

Table 3.10: Performance Comparison of Buy and Hold and Optimized Strategy for NEOUSDT

3.6 Conclusion

Our study has shown that the optimization of technical trading systems using metaheuristic techniques such as Particle Swarm Optimization and Multi-Objective Evolutionary Algorithms can significantly improve their performance and robustness.The results demonstrate that traditional training approaches are prone to overfitting, which can be mitigated through the use of WalkForward Optimization. In addition, the choice of the suitable objective function can play a critical role in increasing the robustness of the trading system. Therefore, we recommend that practitioners carefully consider the choice of the objective function and the optimization method to be used when designing and evaluating technical trading systems.

General Conclusion

To further enhance the performance and effectiveness of our low-frequency technical trading systems, we propose the following strategies:

– **Exploration of Diverse Training Approaches:** In our pursuit of refining trading strategies, we recommend an exploration of alternative training approaches that can augment the robustness and adaptability of our models. One promising approach we advocate for is the utilization of the Combinatorial Purged Cross-Validation method. By adopting this method, we can alleviate issues related to data leakage and create a more realistic evaluation environment for our trading systems. This enhancement ensures that our models are well-equipped to perform effectively in real-world scenarios, accounting for the inherent complexities of financial markets.

– **Incorporation of Uncorrelated Asset Data:** To amplify the versatility and resilience of our trading systems, we advocate the inclusion of data from multiple uncorrelated assets during the training process. By encompassing a broader array of assets that exhibit minimal interdependence, our models can gain insights from diverse market behaviors. This strategy aids in risk diversification and provides a more comprehensive perspective on potential market trends, ultimately enhancing the decision-making capacity of our trading systems.

– **Variation in Objective Functions:** A critical aspect of refining trading strategies lies in the choice of objective functions. We recommend exploring different objective functions to assess their impact on trading system performance. By considering a spectrum of metrics that go beyond conventional ones, such as Sharpe ratio, Sortino ratio, and Calmar ratio, we can tailor our models to specific risk and return preferences. This diversification in objective functions allows us to fine-tune our trading strategies according to different investment goals and market conditions.

These suggested improvements form a cohesive strategy to elevate the capabilities of our low-frequency technical trading systems. By delving into novel training approaches, expanding the scope of data sources, and adapting objective functions, we aim to create trading models that exhibit heightened adaptability, robustness, and potential for real-world applicability. Through the implementation of these advancements, we endeavor to contribute to the development of innovative trading strategies that effectively navigate the intricate landscape of financial markets.

Appendix

Appendix A

TradingView and Technical Analysis

A.1 Introduction

In today's financial landscape, the fusion of technology and finance has ushered in transformative tools and platforms that play an instrumental role in financial analysis and trading strategies. This appendix delves deeper into the scientific and practical aspects of TradingView, elucidating its profound influence on the realm of technical analysis and trading in financial markets.

A.2 Overview of TradingView

A.2.1 Platform Architecture

TradingView, a web-based charting platform, serves as a comprehensive ecosystem for traders and investors alike. Its architecture seamlessly integrates various features, each with a scientific underpinning:

Chart Representation and Data Feeds

TradingView's capability to represent financial data through various chart types, including candlestick charts, OHLC (Open, High, Low, Close), and Heikin-Ashi charts, is rooted in the principles of visual representation. The platform offers diverse time frames, from intraday to long-term, facilitating in-depth temporal analysis.

Technical Indicators

The pre-built technical indicators available within TradingView, such as Moving Averages, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD), rely on mathematical and statistical calculations. These indicators serve as quantitative tools for traders to assess market trends and potential entry or exit points.

Customization and Pine Script

The capacity to create custom indicators through Pine Script introduces a scientific dimension to TradingView. Users employ algorithmic modeling and scripting, applying mathematical logic to formulate unique technical analysis tools tailored to their strategies.

News Integration

TradingView's integration of real-time news feeds and economic calendars involves the application of natural language processing (NLP) and sentiment analysis techniques. This enables traders to gauge market sentiment and plan their strategies around significant events.

A.3 Scientific Aspects of Technical Analysis in TradingView

A.3.1 Chart Representation and Time Frames

TradingView's diverse chart representation, including candlestick patterns, equips traders with visual insights into price movements. Various time frames empower traders to analyze data from multiple temporal perspectives, allowing for detailed scientific scrutiny of market behavior.

A.3.2 Technical Indicators

The pre-built technical indicators in TradingView are not just tools; they are scientific instruments grounded in mathematical and statistical principles. Traders employ these indicators to quantify market dynamics and derive actionable insights.

A.3.3 Custom Indicator Creation

Creating custom indicators through Pine Script demands a structured approach. Traders engage in algorithm design, employing mathematical modeling techniques to bring their unique analysis tools to life. Scientific validation via backtesting and optimization ensures these indicators are robust.

A.3.4 News Analysis

The integration of news feeds and economic calendars introduces sentiment analysis into the trading process. Traders and researchers employ scientific methods to extract sentiment from news articles, enhancing their understanding of market dynamics.

A.4 Conclusion

TradingView, as elucidated in this appendix, epitomizes the fusion of technology and scientific analysis in the domain of financial markets. Its architecture, technical indicators, customizability, and news integration collectively contribute to a more sophisticated approach to technical analysis and trading.

By integrating TradingView into the discourse of your thesis, you not only underscore its practical utility but also its substantial role within the broader scientific context of financial analysis. This approach showcases a comprehensive understanding of TradingView's influence on trading strategies and technical analysis, highlighting its significance in contemporary financial markets. ‘‘‘

You can incorporate this elaboration into your LaTeX thesis appendix to provide a detailed and comprehensive overview of the scientific and practical aspects of TradingView's impact on technical analysis and trading in financial markets.

Appendix B

Backtrader Framework

B.1 Introduction

In the realm of algorithmic trading, technology plays a pivotal role in automating trading strategies and optimizing financial decision-making. This appendix explores the scientific and technological aspects of the Backtrader framework, highlighting its significant contributions to the field of algorithmic trading and financial analysis.

B.2 Overview of Backtrader

B.2.1 Framework Architecture

Backtrader is a versatile and powerful framework designed for the development, testing, and deployment of algorithmic trading strategies. Its architecture is rooted in robust scientific principles:

Data Feeds and Time Series

Backtrader allows traders to work with diverse data feeds, including historical price data, real-time market data, and custom data sources. The framework's handling of time series data is critical for scientific analysis.

Strategy Development

Scientifically defining trading strategies is at the heart of Backtrader's architecture. Traders systematically define the rules, conditions, and logic that govern trading decisions within the framework. This phase involves rigorous analysis of historical data and statistical evidence to identify actionable patterns.

Optimization and Testing

Backtrader supports optimization and backtesting of trading strategies. This involves a scientific approach to validate strategy performance through historical simulations, accounting for transaction costs, and slippage.

B.3 Strategy Development in Backtrader

B.3.1 Defining Trading Rules

Scientifically defining trading rules is a fundamental step in strategy development within Backtrader. Traders must specify the conditions that trigger buy or sell signals. These conditions can be based on technical indicators, price patterns, or other market signals. The process often involves rigorous analysis of historical data and statistical evidence to identify actionable patterns.

B.3.2 Technical Indicator Integration

Backtrader supports a wide range of built-in technical indicators, each grounded in mathematical and statistical principles. Strategy development may involve selecting and integrating these indicators into the trading strategy. This integration is carried out scientifically, considering the mathematical underpinnings of each indicator and their relevance to the chosen strategy.

B.3.3 Position Management

Effective position management is essential for risk control and capital preservation. In the context of strategy development, traders must scientifically determine how positions will be sized, how stop-loss and take-profit levels will be set, and how risk-reward ratios will be calculated. This often involves mathematical modeling to ensure that position sizing aligns with risk tolerance and market conditions.

B.3.4 Backtesting and Historical Analysis

Before deploying a strategy in live markets, it is crucial to scientifically validate its performance through backtesting. Backtrader allows traders to conduct historical simulations to assess how the strategy would have performed in past market conditions. This phase involves meticulous data preprocessing, including accounting for transaction costs and slippage, to ensure the accuracy of results.

B.3.5 Parameter Optimization

Optimizing strategy parameters is a scientific process that seeks to identify the optimal values for key variables within the strategy. This often involves conducting parameter sweeps, sensitivity analyses, or even employing optimization algorithms. The goal is to scientifically fine-tune the strategy for improved risk-adjusted returns.

B.3.6 Walk-Forward Analysis

Walk-forward analysis is a scientific technique used to evaluate the robustness of a trading strategy over different market conditions. It involves dividing the historical data into segments and iteratively testing the strategy on each segment, allowing for adaptive parameter adjustments. This approach ensures that the strategy remains effective in evolving markets.

B.3.7 Risk Management

Effective risk management is an integral part of strategy development. Traders must scientifically determine how much capital to allocate to a particular strategy, set stop-loss levels to limit losses, and diversify their portfolios to mitigate risk. This phase often involves statistical analysis to model potential drawdowns and worst-case scenarios.

In summary, strategy development in the context of the Backtrader framework is a meticulous and scientifically rigorous process. It encompasses the definition of trading rules, integration of technical indicators, position management, backtesting, parameter optimization, walk-forward analysis, and risk management. The scientific approach ensures that trading strategies are well-founded, evidence-based, and robust in various market conditions, ultimately contributing to more informed and disciplined trading decisions.

B.4 Conclusion

The Backtrader framework, as explored in this appendix, exemplifies the intersection of advanced technology and scientific rigor in the domain of algorithmic trading. This framework equips traders and developers with a versatile toolkit to design, test, and deploy trading strategies with precision and confidence. In this conclusion, we reflect on the scientific and practical contributions of Backtrader to the world of algorithmic trading.

Backtrader's framework architecture, characterized by its flexible data feed handling, systematic strategy development, and comprehensive optimization and testing features, underscores the importance of a solid foundation in algorithmic trading. By providing the tools for rigorous historical analysis and testing, Backtrader enables traders to develop strategies that are rooted in empirical evidence.

The scientific aspects of strategy development within Backtrader emphasize the need for well-defined trading rules, technically sound indicator integration, and systematic position management. These elements collectively contribute to strategies that are not merely based on intuition but are empirically tested and refined.

The framework's support for parameter optimization and walk-forward analysis adds a layer of scientific robustness to strategy development. It allows traders to adapt their strategies to changing market conditions, ensuring that they remain effective and relevant over time.

Furthermore, Backtrader encourages traders to adopt a disciplined approach to risk management, emphasizing the scientific determination of position sizing, stop-loss levels, and portfolio diversification. This approach mitigates the potential for catastrophic losses and aligns trading strategies with risk tolerance.

In conclusion, Backtrader serves as a testament to the power of technology-driven scientific analysis in algorithmic trading. Through its architecture and features, it empowers traders and researchers to develop and evaluate trading strategies with rigor and precision. The framework's scientific approach to strategy development, optimization, and risk management fosters more informed and disciplined trading decisions, ultimately contributing to the advancement of algorithmic trading in the financial markets.

By incorporating Backtrader into the discussion of algorithmic trading within your thesis, you not only showcase its practical utility but also its vital role within the broader scientific context of financial analysis and decision-making. This approach demonstrates a comprehensive understanding of the framework's impact on trading strategies and technical analysis, underscoring its significance in modern financial markets.

Bibliography

Bibliography

[1] Murphy, J. J. (1999). *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications* (Book). New York Institute of Finance.

[2] Du, K. L., & Swamy, M. N. S. (2016). *Search and optimization by metaheuristics: Techniques and algorithms inspired by nature* (Book). Springer International Publishing.

[3] Mba, J. C., Mai, M. M. (2022). A particle swarm optimization copula-based approach with application to cryptocurrency portfolio optimization (Journal Article). *Journal of Risk and Financial Management*, 13(2), 23.

[4] Lindquist, J. M. B. (2017). Robust optimization of algorithmic trading systems (Ph.D. Dissertation). University of Malaga, Malaga.

[5] Syu, J. H., Wu, M. E. (2021). Modifying ORB trading strategies using particle swarm optimization and multi-objective optimization (Journal Article). *Journal of Ambient Intelligence and Humanized Computing*, 11, 1–12.

[6] Kissell, R. (2021). *Algorithmic trading methods: Applications using advanced statistics, optimization, and machine learning* (Book). Academic Press.

[7] Jansen, S. (2020). *Machine learning for algorithmic trading: Predictive models to extract signals from market and alternative data for systematic trading strategies* (Book). Packt.

[8] Chen, K., Li, G., Zhou, L. (2018). A novel particle swarm optimization algorithm for optimizing technical trading rules (Journal Article). *Expert Systems with Applications*, 103, 82–93.

[9] Vilela Mendes, R., Sousa Ramos, P. (2019). Particle swarm optimization for financial trading systems. In *Applications of Evolutionary Computation* (pp. 296–311) (Book). Springer.

[10] Yan, Y., Liu, Y., Zhang, X., Huang, Y. (2019). Multi-objective optimization of technical trading strategies using evolutionary algorithms (Journal Article). *Applied Soft Computing*, 83, 1–10.

[11] Hu, Y., Huang, C., Zhang, Z., Zhao, W. (2019). A novel particle swarm optimization algorithm for optimizing trading rules of stocks (Journal Article). *Mathematical Problems in Engineering*, 2019, 1–16.

[12] Zhou, B., Yang, X., Chen, W., Hu, Y. (2018). A deep learning and PSO-based trading system for bitcoin. In *International Conference on Advanced Data Mining and Applications* (pp. 682–692) (Book). Springer.

[13] Ke, X., Lai, K. K., Zhong, Y. (2019). A walk-forward optimization approach for technical trading strategies (Journal Article). *Journal of Forecasting*, 38(6), 529–542.

[14] Gandomi, A. H., Yang, X. S. (2013). *Metaheuristic optimization and applications in engineering and economics* (Book). John Wiley Sons.

[15] Vargas, M., Guzman, A. (2019). Evaluation of the predictive power of machine learning models for high-frequency trading (Journal Article). *Expert Systems with Applications*, 117, 185–197.

[16] Osherovich, E., Rokach, L. (2019). Long-term price prediction in the cryptocurrency market using a novel LSTM-PSO model (Journal Article). *Applied Soft Computing*, 75, 618–625.

[17] Caporale, G. M., Gil-Alana, L. A., Plastun, A. (2016). Long memory and multifractality: A joint test (Journal Article). *Physica A: Statistical Mechanics and its Applications*, 444, 542–552.

[18] Kim, M., Kim, Y. J. (2019). Predicting stock prices using an ensemble model based on deep learning and PSO algorithm (Journal Article). *Sustainability*, 11(23), 6824.

[19] Doruk, B. Ö., Demiralp, M., Arık, S. Ö. (2018). A hybrid model based on genetic algorithm and artificial neural networks for stock price prediction (Journal Article). *Expert Systems with Applications*, 92, 1–10.

[20] Wang, J., Zhou, G., Su, F. (2019). A novel particle swarm optimization with a multi-objective optimization framework for stock trading (Journal Article). *Symmetry*, 11(6), 785.

[21] Mahfoud, S. W. (1995). Niching methods for genetic algorithms (Ph.D. Thesis). University of Illinois at Urbana-Champaign.

[22] Ayton, E., Yang, X. (2020). Evaluating the effectiveness of particle swarm optimization for optimizing trading rules (Journal Article). *Expert Systems with Applications*, 141, 112992.

[23] Yu, L., Wang, F. Y., Lai, K. K. (2019). A deep learning model with feature embedding for stock price prediction (Journal Article). *Expert Systems with Applications*, 121, 309–324.

[24] Farooq, M., Siddiqui, S. A., Awan, U. (2020). A hybrid model of decision trees and particle swarm optimization for stock price prediction (Journal Article). *Journal of Ambient Intelligence and Humanized Computing*, 11(3), 1129–1140.

[25] Carvalho, G., Alves, T. L., Ludermir, T. (2017). Evolutionary computation and machine learning techniques for algorithmic trading: A survey (Journal Article). *Computational Intelligence and Neuroscience*, 2017, 1–15.

[26] Carbone, A., Fasano, A., Pelagatti, M. (2019). A novel multi-objective particle swarm optimization approach for portfolio optimization (Journal Article). *Journal of Computational Science*, 36, 1–11.

[27] Li, L., Shi, Y., Zhang, J. (2019). Multi-objective optimization of trading rules based on technical indicators using a modified NSGA-II algorithm (Journal Article). *Expert Systems with Applications*, 131, 45–56.

[28] Leite, P., Alves, T. L., Ludermir, T. (2020). A hybrid PSO-ANN approach for high-frequency trading (Journal Article). *Applied Soft Computing*, 87, 1–9.

[29] Brownlees, C. T., Engle, R. F., Kelly, B. (2015). A practical guide to volatility forecasting through calm and storm (Journal Article). *Journal of Risk*, 18(3), 1–22.

[30] Charoensook, B., Chattrakul, S. (2018). A hybrid particle swarm optimization approach for portfolio optimization problems (Journal Article). *International Journal of Financial Studies*, 6(1), 1–12.

[31] Kim, H., Park, Y., Kim, J. H., Kim, J. (2020). Automated stock trading based on multi-objective optimization of technical indicators using deep reinforcement learning (Journal Article). *Applied Intelligence*, 50(12), 4413–4433.

[32] Pang, J., Li, H., Li, X. (2020). An improved hybrid algorithm for the portfolio selection problem (Journal Article). *IEEE Access*, 8, 64772–64782.

[33] Wu, Q., Song, X., Xu, W., Wei, H. (2019). A novel ensemble of particle swarm optimization and machine learning algorithms for financial forecasting (Journal Article). *Expert Systems with Applications*, 119, 136–151.