Smarter Investment using Big Data, Data Science and Algorithmic Trading

Interim Report

Chan Chun Hei (3035684908)

Supervisor: Prof SM Yiu

Department of Computer Science

The University of Hong Kong

January 26, 2025

i. Abstract

Algorithmic trading has gained popularity due to the rise of electronic systems in exchanges. Traditionally, algorithmic trading relied on quantitative data. With the advent of big data, qualitative data has become increasingly relevant for trading decisions. This project aims to critically evaluate and improve existing algorithmic trading strategies using traditional numeric data while exploring the integration of qualitative data to improve performance. Initially, a trend-following strategy using moving averages was implemented and its performance was evaluated through back testing. The results showed a significant dependency on market trends, leading to inconsistent profitability. Moving forward, the project explored the use of confidence intervals and momentum oscillators to address the identified limitations. Additionally, new qualitative data sources, such as financial news and social media sentiment will be explored to enhance the trading algorithms. The ultimate objective is to develop an integrated platform and interactive dashboard that facilitates the visualization of trading performance and supports smarter investment decisions, leveraging both traditional and innovative data sources.

ii. Acknowledgement

I would like to express my deepest gratitude to my supervisor, Prof. SM Yiu, for his invaluable patience, continuous support, and inspiring feedback throughout this project. His guidance has been instrumental in the completion of this research.

Additionally, I extend my heartfelt thanks to Mr. Nicholas Mo for his insightful feedback and suggestions regarding language and the professional presentation of this report.

Their contributions have significantly enhanced the quality of my project and work.

iii. Table of Contents

iv. List of Figures

v. List of Tables

vi. List of Equations

vii. Abbreviations

Abbreviation	Meaning
ANN	Artificial neural network
API	Application programming interface
ARR	Annualised rate of return
BERT	Bidirectional encoder representations from transformers
CAPM	Capital Asset Pricing Model
GPT	Generative pre-trained transformer
GUI	Graphical user interface
HFT	High frequency trading
LLM	Large language model
LR	Logistic regression
MLP	Multi-layer perceptron
NLP	Natural language processing
RF	Random forest
RSI	Relative strength index
SVM	Support vector machine
UAT	User acceptance test
VWAP	Volume Weighted Average Price

1. Introduction

1.1. Background

Practitioners and academics are continuously developing new and improved techniques to select stocks and increase returns in portfolios. The foundational work by Markowitz (1952) on the Efficient Trading Frontier and Sharpe (1964) on the Capital Asset Pricing Model (CAPM) has laid a solid groundwork for quantitative analysis in financial markets. The development of the Black-Scholes model by Black and Scholes (1973) for option pricing further laid the groundwork for quantitative finance and algorithmic trading.

Algorithmic trading, which involves the execution of orders using automatic pre-programmed trading rules, has been a significant development since its inception in the early 1970s when exchanges began using electronic trading systems rather than manual systems. Early algorithms were straightforward. Predefined instructions derived from price and volume data were executed. These basic algorithms served as the groundwork for the advanced and intelligent trading strategies that followed. Algorithmic trading has gained substantial traction over the past decades, accounting for approximately 92% of all equity volume in 2019 (Kissell, 2020). Recent studies indicate that the algorithmic trading market was valued at USD 3.1 billion in 2023 and is projected to grow at a rate exceeding 13% from 2024 to 2032 (Global Market Insights, 2024).

Algorithmic trading has significantly influenced financial market dynamics and presents both opportunities and challenges. It allows trades to be executed more efficiently and at better prices. However, it also introduces new challenges by increasing short-term volatility, making the financial market more risky. High frequency trading (HFT), a subset of algorithmic trading, exemplifies these effects by executing a large number of transactions within a short period. This leads to rapid price changes and increases market complexity (Boehmer et al., 2021).

The rise of data-driven investment strategies can be attributed to advancements in computational resources and artificial intelligence tools. The vast amount of data available on the Internet provides a unique opportunity to enhance investment decision-making processes.

Traditionally, algorithmic trading has relied on quantitative data such as historical prices and volumes. However, with the advancements in natural language processing (NLP) and large language models (LLMs), it has become increasingly feasible to develop trading rules and instructions based on textual data from social media posts and articles.

1.2. Problem Statement

Despite advancements in algorithmic trading, several challenges hinder its optimal implementation and effectiveness. The complexity of financial markets, influenced by economic indicators, geopolitical events, and market sentiment, makes it difficult to develop algorithms that consistently predict market movements and generate profitable trades. Most existing algorithms rely on historical numeric financial data, which may not fully capture current market dynamics. Additionally, the high-frequency nature of algorithmic trading demands robust computational resources to process vast amounts of data in real-time, posing a barrier for smaller firms or individual traders. Furthermore, the risk of overfitting in machine learning models, where algorithms perform well on historical data but fail to generalize new and unseen data, remains a critical issue.

Exploring the feasibility and performance of using new forms of data for transaction decisions in complex markets, and balancing overfitting and underfitting by evaluating and improving different models, are essential steps. Addressing these challenges is crucial for enhancing the reliability and profitability of algorithmic trading systems.

1.3. Motivation

The motivation behind this research is driven by the potential of algorithmic trading and big data to revolutionise the financial industry by enhancing trading efficiency and profitability. Algorithmic trading can execute trades at speeds and frequencies that are impossible for human traders, thereby capturing market opportunities more effectively. The integration of big data and data science allows for the analysis of vast datasets, uncovering patterns and insights that can inform trading strategies in real-time while adapting to dynamic and ever-changing market conditions.

1.4. Objectives

The primary objective of this project is to evaluate various algorithmic trading strategies that utilize numerical data. This involves comparing their performance across different markets and investigating their effectiveness in varying market conditions. By identifying the underlying reasons for their performance, the project aims to gain a deeper understanding of these algorithms.

Another key objective is to enhance existing algorithmic trading strategies and explore the effects by incorporating textual data. This involves justifying the principles behind these

enhancements and exploring the feasibility of using textual data respectively. The project also compares the advantages and disadvantages of the newly proposed strategies to ensure they offer tangible benefits.

Additionally, the project aims to visualise the results using dashboards, allowing an effective summary of key insights. This provides a comprehensive view of the collected data and its implications.

Ultimately, the project seeks to develop an integrated investment platform that combines statistical modelling, sentiment analysis, and algorithmic trading. By leveraging big data and AI techniques, the platform will provide personalized investment insights and minimize emotional biases, thereby enhancing decision-making processes.

1.5. Deliverables

1.5.1. Research

The research component of this project will focus on evaluating various algorithmic trading strategies using numerical data across different markets. This includes investigating the effectiveness of these algorithms in different market conditions and identifying the underlying reasons for their performance. Additionally, the project proposes improvements to existing algorithmic trading strategies, justifying the ideas behind these enhancements. The project also explores the feasibility and effects of incorporating textual data into the strategy.

1.5.2. Software Development

The software development component will deliver a user-friendly application with a graphical user interface (GUI). This application will feature a decision-making dashboard, an algorithmic trading module, and automated article collection and analysis capabilities. Additionally, a database management system will be developed to support the platform.

To ensure the application meets user needs, the project will conduct User Acceptance Testing (UAT) and Usability Testing. Feedback gathered during these tests will be used to refine the application, creating a robust and user-centric investment platform.

1.6. Report Outline

By combining the analysis of both quantitative and qualitative data, this project aims to develop a novel algorithmic trading strategy that adapts to various market scenarios and outperforms existing algorithms. The report reviews the literature on various aspects of algorithmic trading

in *Section 2* and outlines the methodologies of each component in *Section 3*. It then presents the results of various approaches in *Section 4*, discussing their performance and effectiveness. *Section 5* documents the difficulties encountered during the project and the mitigations implemented to overcome them. *Section 6* outlines the future work planned to expand and enhance the methodologies, and *Section 7* concludes the report by summarizing the key findings and their implications for developing robust and adaptable trading strategies.

2. Literature Review

This section provides an in-depth examination of existing research on algorithmic trading, covering various strategies, the nature of trading, and the application of machine learning techniques. The review aims to identify research gaps and potential areas for improvement.

2.1. Types of Algorithmic Trading Strategies

Previous work by Addy et al. (2024) has classified algorithmic trading into various types based on underlying motivations and principles.

One common strategy is trend following, which utilises the momentum of asset prices. This approach often uses moving averages or other technical indicators to identify and follow trends. Zhang et al. (2022) identified two major trend-following strategies, namely the Moving Average Crossover and Volume Weighted Average Price (VWAP).

Another widely used strategy is mean reversion, which is based on the premise that asset prices will revert to their historical mean over time, especially after significant price changes.

Arbitrage strategies exploit price discrepancies between different markets or instruments. The goal is to capture short-term market anomalies, assuming market inefficiencies (Ayala et al., 2021). Mean reversion strategies, cointegration analysis, and correlation-based models are common techniques used in statistical arbitrage.

However, there is limited research on the strengths and weaknesses of these models, and comparative analysis is scarce. This gap highlights the need for a systematic evaluation of these strategies to understand their relative effectiveness and adaptability in different market conditions.

Understanding these types of strategies is crucial for improving algorithmic trading and creating more consistent approaches under various market conditions.

2.2. Nature of Algorithmic Trading

High-frequency trading (HFT) is prevalent in algorithmic trading literature. This phenomenon is explained by Koo (2024). He has shown that traders are increasingly relying on algorithmic advisors for swing trading rather than long-term investing. The impact of HFT on market dynamics, such as price and volume, is also a frequent research topic. For instance, Dutta et al. (2023) discussed the influence of information flow on the behaviour of high-frequency traders and how certain HFT strategies notably affect market dynamics, such as asset prices and transaction volumes.

However, most existing research focuses on numeric data. In the era of big data, with improved computing power and algorithms, there is potential to also investigate unstructured data, such as text, within the context of trading.

The exploration of HFT naturally leads to the broader discussion of machine learning approaches, which leverage big data to refine and innovate algorithmic trading further.

2.3. Machine Learning Approaches

The advent of big data has significantly advanced algorithmic trading. Extensive financial data, including historical stock prices, company financial statements, financial news, social media sentiments, and macroeconomic indicators are now readily accessible online. Machine learning-based algorithmic trading has become a prominent research trend due to its ability to generalise complex patterns and adapt to ever-changing markets. Researchers focus on creating, analysing, and comparing algorithmic trading strategies. For instance, Hong et al. (2024) examined various deep learning models employed in stock market forecasting, while Majidi et al. (2024) introduced a new approach using reinforcement learning in algorithmic trading.

Large Language Models (LLMs) began gaining popularity around 2017 with the introduction of the transformer model by Google researchers (Vaswani et al., 2017). Delvin et al. (2018) further built on the transformer model and proposed the BERT model for language understanding. Conventional machine learning models find it challenging to efficiently process and interpret large amounts of textual data from articles and earnings reports. They often miss subtle details that can affect market trends. Ni et al. (2024) introduced an approach by employing LLMs to make stock predictions using company earnings reports. In the future, one of the research directions in algorithmic trading and stock predictions is likely to involve LLMs.

Machine learning models are prone to overfitting, where algorithms perform well on seen data but poorly on unseen data. This poor generalization typically results from training using market data from a specific market condition and distribution. However, existing research seldom addresses this problem. Zhang et al. (2022) proposed a reverse reinforcement learning model that adapts trading policies in real-time and accurately adjust to market changes. Inspired by this, the project will explore predicting market trends and classifying market conditions before applying specific algorithmic trading strategies to improve generalization and adaptability.

By exploring these three interconnected areas, the literature review establishes a comprehensive understanding of the current state of algorithmic trading research and identifies the critical areas for further investigation and development.

3. Methodology

The project is divided into several key components, shown sequentially below in *Figure 1*.

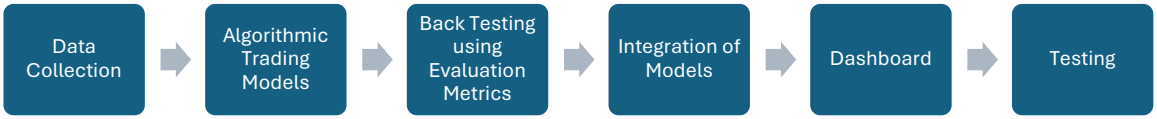

Figure 1 Sequential relationship of key components in the project

As shown in *Figure 1*, the process begins with Data Collection (Section 3.1), where data is gathered and prepared. The next step involves analysing Algorithmic Trading Models (Section 3.2), where trading signals are generated. Back Testing using Evaluation Metrics (Section 3.3) has been implemented to determine the effectiveness and robustness of the models. Integration of Models (Section 3.4) follows to combine multiple approaches to enhance performance. The results are then visualized in an interactive Dashboard (Section 3.5). Finally, Testing (Section 3.6) ensures the system meets user requirements. This section outlines the respective methodologies.

3.1. Data Collection

The project used historical numeric data (e.g., price, volume) primarily sourced from Yahoo Finance API, due to its comprehensive and frequently updated database, which is essential for accurate financial analysis. Python libraries such as Pandas and NumPy were employed for data cleaning, preprocessing, and aggregation because of their powerful data manipulation capabilities and efficiency in handling large datasets. Textual data, including news articles and social media posts, was collected via web scraping using BeautifulSoup, Selenium, and Scrapy, for their robustness and flexibility in extracting data from various online source. Native APIs

were utilized when available to ensure direct access to high-quality data, which enhances the project's overall reliability and accuracy.

3.2. Algorithmic Trading Models

The project explored the following strategies for generating buy/sell signals:

1. Traditional Approaches: Trend-following, mean reversion strategies.
2. Supervised Machine Learning: Logistic regression (LR), random forest (RF), support vector machines (SVM).
3. Reinforcement Learning: Q-learning.
4. Deep Learning: Multi-layer perceptron (MLP), artificial neural network (ANN).
5. Large Language Model (LLM): Generative Pre-trained Transformer (GPT).
6. Multi-Model Approach: Integration of multiple modalities.

In particular, a traditional trend-following strategy using moving averages was implemented as the preliminary approach due to its well-established methodology and ease of application. This serves as a baseline for comparing more advanced strategies, ensuring a structured progression in evaluating the effectiveness of different methodologies. The trading logic summary is shown below in *Figure 2*.

```
For each trading day
   if price > moving average when crossover then
      Buy()
   else
      Sell()
ENDFor
```

Figure 2 Trading algorithm of the trend-following approach using moving averages as the indicator

3.3. Back Testing using Evaluation Metrics

Back testing applies predictive models to historical data to evaluate their viability. The project used QuantConnect as the main back testing platform because it is open-sourced and offers built-in historical datasets.

Referencing the work by Cuthbertson et al. (2010) and Sukma et al. (2024), algorithmic trading models are compared and evaluated using the following metrics: Annualised Rate of Return (ARR), Sharpe Ratio, Win Rate, Maximum Drawdown, Profit Factor, and Alpha.

Evaluation metrics are crucial for assessing the performance and viability of algorithmic trading models. They provide standardised, objective criteria that allow for a clear comparison of different strategies. By using evaluation metrics, the project can quantify the effectiveness, risk, and profitability of each model in a consistent manner. This ensures that decisions are based on robust data rather than subjective judgment.

3.4. Integration Strategies

The project explores various model integration techniques:

1. Hybrid Models: Combine traditional and machine learning approaches.
2. Ensemble Methods: Aggregate predictions from multiple models.
3. Feature Engineering: Use dimension reduction technique to extract useful features.
4. Real-Time Adaptation: Implement reinforcement learning to adapt strategies in real-time based on market feedback.

Inspired by the reverse reinforcement learning model proposed by Zhang et al. (2022) mentioned in Section 2.3, the project will particularly focus on real-time adaptation to improve generalization and adaptability. This approach allows for continuous learning and adjustment, ensuring that the trading strategies remain effective even as market conditions evolve.

3.5. Dashboard

The dashboard is developed using Plotly and Dash, which are powerful tools for creating interactive web-based visualizations. These tools allow for the integration of various data sources and the creation of dynamic visualizations.

3.6. Testing

Testing includes User Acceptance Testing (UAT) and Usability Testing to ensure the system meets user requirements and is easy to use. Here are examples of key stakeholders:

1. End Users: Primary users of the app, such as general traders and investors.
2. Business Analysts: Professionals who understand the business and industry requirements. These professionals can be managers from brokers and banks.

Interviews and questionnaires will be designed to gather detailed feedback from users and stakeholders, providing valuable insights into their experiences and expectations. This helps identify any issues and areas for improvement, ensuring the system is user-friendly and meets all requirements. By incorporating both UAT and Usability Testing, the project ensures that the

system is not only technically sound but also practical and efficient for its intended users, thereby increasing the likelihood of adoption and success in the market.

4. Results

4.1. Baseline – Simple Moving Average Crossover

The project back tested the baseline approach detailed in Section 3.2 and *Figure 2* using S&P 500 (SPY) data from 1 Jan 2022 to 30 Apr 2024. The results are shown below in *Figure 3*.

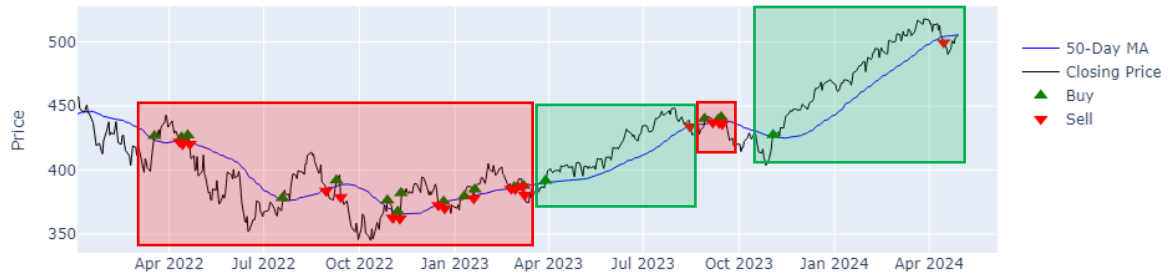

Figure 3 50-day moving average plotted on the SPY closing price, with trading signals generated by the Baseline

The blue and black lines indicate the 50-day moving average and the closing price respectively. If the price is above the moving average during their crossover, a buy signal is generated, indicated by green triangles. Otherwise, a selling signal is generated, indicated by red triangles.

The result of the baseline approach is concluded below in *Table 1* using QuantConnect.

Table 1 Back testing evaluation metrics of the Baseline

Baseline – Simple moving average crossover			
Evaluation Metrics	U.S. Stock	Jan'22 – Feb'23	Mar'23 – Apr'24
Annualized Rate of Return	3.566%	-9.424%	18.557%
Sharpe Ratio	-0.1	-0.815	0.927
Win Rate	26%	27%	29%
Average Win	6.23%	1.51%	13.33%
Average Loss	-1.50%	-1.94%	-1.00%
Profit-Loss Ratio	4.16	0.78	13.33
Maximum Drawdown	17.500%	15.900%	6.100%
Alpha	-0.01	-0.054	0

The model yields an annualised return of 3.566%, indicating slight profitability. However, with a slightly negative alpha of -0.01, the model slightly underperforms the market. The model has

a win rate of 26 % and a high profit-loss ratio of 4.16, but its performance is trend-dependent. During the bullish trend, highlighted using green rectangles in *Figure 3*, the strategy profits by buying low and selling high. However, in sideways markets, highlighted using red triangles, frequent transactions lead to negative profit. The model profits significantly during winning trades but incurs minor losses during frequent losing trades.

The evaluation metrics indicate that the strategy cannot consistently profit, requiring further improvement. Two types of trends are observed during the back testing period. The periods Jan 2022 – Feb 2023 and Mar 2023 – Apr 2024 are categorised into sideways and bullish trends respectively. Their respective evaluation metrics show that the effectiveness of the algorithm is highly dependent on the type of trend. It profits 18.557% and loses 9.424% respectively in bullish and sideways trends. Therefore, the project would propose improvements to the baseline algorithm by identifying major problems and evaluating the effectiveness of these improvements under the two trends.

The first major problem of the baseline model is that is generates unwanted signals during sideways markets. This is highlighted by the red boxes in *Figure 3*, where frequent unwanted signals are observed when the price moves close to the moving average. The project proposes a confidence interval approach to minimise unwanted signals and a mean reversion approach to capture peaks and toughs during sideway movements. The details are explained in *Section 4.2*, *Section 4.3*, and *Section 4.4*.

The second major problem of the model is that it lags in capturing profits during bullish trends. Highlighted by the green boxes in *Figure 3*, the algorithm enters and leaves the market late. The project proposes adjusting the window size to reduce lag. The details are explained in *Section 4.5*.

4.2. Approach 1 – Moving Average Confidence Interval

This approach addresses the frequent unwanted signals in the Baseline during sideways trends.

The trading logic uses moving average upper and lower bands, calculated using confidence interval, to avoid frequently unwanted trades. With the upper and lower bands, the price is less likely to crossover them during sideways. This minimises signal when the price moves along the moving average. The upper and lower bands for a trading day are calculated using *Equation 1* and *Equation 2*. Here, σ is the standard deviation of the closing price over the past n days and c a parameter adjusting the confidence interval width. Assuming a normal distribution, $c = 1$, $c = 2$, and $c = 3$ corresponds to the 68%, 95%, and 99% confidence intervals respectively.

$$Moving\ Average\ Lower\ Band = Moving\ Average - c\sigma$$

Equation 1

$$Moving\ Average\ Lower\ Band = Moving\ Average - c\sigma$$

Equation 2

The trading algorithm for Approach 1 is illustrated in *Figure 4*. It generates a buy signal when the price is above the moving average lower band during crossover and a sell signal when the price is below the moving average upper band. The lower band uses $c = 1$, while the upper band uses $c = 2$ to capture more profit by tightening sell signal conditions.

Approach 1 – Moving average confidence interval
For *each trading day* ***if*** *price > moving average* - 1 × σ *during crossover* ***then*** *Buy()* ***else if*** *price < moving average* + 2 × σ *during crossover* ***then*** *Sell()* ***ENDFor***

Figure 4 Trading algorithm of the moving average confidence interval approach

The project back tested this approach using the same S&P 500 (SPY) data from Jan 1 2022 to 30 Apr 2024. The results are shown below in *Figure 5*.

Figure 5 50-day moving average plotted on the SPY closing price confidence interval, with trading signals generated by Approach 1

The green and red dotted lines indicate the upper and lower bands of the moving average respectively. The blue line indicates the 50-day moving average. If the price is above the lower band during their crossover, a buy signal is generated, indicated by green triangles. Similarly, if the price is below the upper band during their crossover, a selling signal is generated, indicated by red triangles.

The result of Approach 1 is concluded below in *Table 2* using QuantConnect.

Table 2 Back testing evaluation metrics of Approach 1

Approach 1 – Moving average confidence interval			
Evaluation Metrics	U.S. Stock	Jan'22 – Feb'23	Mar'23 – Apr'24
Annualized Rate of Return	6.333%	-0.294%	13.385%
Sharpe Ratio	0.108	-0.061	0.543
Win Rate	60%	50%	100%
Average Win	6.95%	4.44%	8.22%
Average Loss	-2.30%	-4.58%	0%
Profit-Loss Ratio	3.03	0.97	0
Maximum Drawdown	20.500%	20.500%	8.600%
Alpha	0.013	0.065	-0.021

This approach yields a greater annualised rate of return than the Baseline at 6.333% and only loses 0.294% during sideways trend (Jan '22 – Feb'23). In fact, the alpha is positive indicating this strategy outperforms the market during sideways market. However, the algorithm only earns 13.385% during bullish trend. The strategy slightly underperforms the market during upward trend evidenced by a slightly negative alpha.

The above metrics suggest that this algorithm can reduce loss during sideways trend by sacrificing some returns during bullish trend.

During sideways trend, the algorithm can successfully buy low and sell high. However, it still cannot profit. The project proposes Approach 2, in *Section 4.3*, aiming to profit from the sideways trends.

4.3. Approach 2 – Relative Strength Index Crossover

This approach leverages mean reversion, assuming asset prices and historical returns will revert to their long-term average. The project explores using relative strength index (RSI) to identify overbought or oversold conditions, anticipating price reversion to the mean.

RSI is a momentum oscillator measuring the speed and change of price movements over an n-day period. RSI values range from 0 to 100, with values above 70 indicating overbought conditions and values below 30 indicating oversold conditions.

The RSI is calculated using *Equation 3* and *Equation 4*.

$$RSI = 100 - \left(\frac{100}{1 + RS}\right)$$

Equation 3

$$RS = \frac{Average\ gain\ over\ n\ periods}{Average\ loss\ over\ n\ periods}$$

Equation 4

The averages in *Equation 4* are calculated using simple moving average where $n = 14$.

The trading algorithm assumes that prices will rise to the mean when oversold and fall to the mean when overbought. Thus, a buy signal is generated when it is oversold (RSI crosses below 30). Conversely, a sell signal is generated when it is overbought (RSI crosses above 70). This boosts the profit of the algorithm during sideways. The trading logic is presented below in Figure 6.

Approach 2 – Relative strength index crossover
For *each trading day*   ***if*** *rsi < 30 when crossover* ***then***   *Buy()*   ***else if*** *rsi > 70 when crossover* ***then***   *Sell()* ***ENDFor***

Figure 6 Trading algorithm of the relative strength index crossover approach

The algorithm was back tested using the same S&P 500 (SPY) data from Jan 1 2022 to 30 Apr 2024). The result is shown below in *Figure 7*.

Figure 7 Relative strength index plotted with the SPY closing price, with trading signals generated by Approach 2

The purple line indicates the 14-day RSI calculated using the simple moving average. The blue dotted lines represent the overbought and oversold thresholds set at 30 and 70 respectively. Green triangles indicate buy signals when RSI is below 30, while red triangles indicate sell signals when RSI is above 70.

The results of Approach 2 are concluded in *Table 3* using QuantConnect.

Table 3 Back testing evaluation metrics of Approach 2

Approach 2 – Relative strength index crossover			
Evaluation Metrics	U.S. Stock	Jan '22 – Feb '23	Mar '23 – Apr'24
Annualized Rate of Return	3.834%	-1.974%	13.155%
Sharpe Ratio	-0.026	-0.132	0.602
Win Rate	62%	40%	75%
Average Win	3.87%	4.02%	4.95%
Average Loss	-3.26%	-3.26%	0.01%
Profit-Loss Ratio	1.19	1.23	708.19
Maximum Drawdown	17.600%	17.600%	5.900%
Alpha	-0.004	0.053	-0.006

Approach 2 yields a higher annualised rate of return than the Baseline at 3.834% and only loses 1.974% during sideways trend (Jan '22 – Feb'23), indicated by the red rectangle in *Figure 7*. However, Approach 2 is slightly less effective than Approach 1. It outperforms the market during sideways trend, as indicated by a positive alpha, but only earns 13.155% during bullish trend, slightly underperforming the market, as evidenced by a slightly negative alpha.

These metrics suggest that Approach 2 can reduce losses during sideways trend by sacrificing some returns during bullish trend. However, it does not profit consistently from the sideways trend, despite outperforming the market during that period.

This limitation arises because the RSI signals cannot fully capture peaks and troughs, especially during periods highlighted by the yellow rectangle in *Figure 7*. This issue will be addressed in Approach 3 in *Section 4.4*.

4.4. Approach 3 – Relative Strength Index Local Maximum and Minimum

Approach 3 addresses the problem identified in Approach 2. It not only considers the crossover of the relative strength index (RSI) with a threshold but also takes into account the local

maximum and minimum values. This aims to solve the first major problem identified in the Baseline and profit from sideways trend.

The trading algorithm builds upon the algorithm in Approach 2. It detects local maximum and minimum values by examining the RSI from the previous day, When a change in RSI direction is identified, trading signals are generated accordingly. The algorithm is presented in *Figure 8*.

Approach 3 – Relative strength index local maximum and minimum
For *each trading day*   ***if*** *rsi < 30* ***and*** *previous_rsi < rsi* ***then***   *Buy()*   ***else if*** *rsi > 70* ***and*** *previous_rsi > rsi* ***then***   *Sell()* ***ENDFor***

Figure 8 Trading algorithm of the relative strength index local maximum and minimum approach

The algorithm was back tested to profit from the sideways trend, addressing the first major problem of the Baseline and the issue identified in Approach 2, using the same S&P 500 (SPY) data from Jan 1 2022 to 30 Apr 2024. The results are shown in *Figure 9*.

Figure 9 Relative strength index plotted with the SPY closing price, with trading signals generated by Approach 3

The purple line indicates the 14-day RSI calculated using the simple moving average. The blue dotted lines represent the overbought and oversold thresholds set at 30 and 70 respectively. Green triangles indicate buy signals when RSI is below 30 and the previous RSI is less than the current RSI. Red triangles indicate sell signals when RSI is above 70 and the previous RSI is greater than the current RSI.

The results of Approach 3 are summarised in *Table 4* using QuantConnect.

Table 4 Back testing evaluation metrics of Approach 3

Approach 3 – Relative strength index local maximum and minimum			
Evaluation Metrics	U.S. Stock	Jan '22 – Feb '23	Mar '23 – Apr'24
Annualized Rate of Return	5.105%	3.597%	2.148%
Sharpe Ratio	0.041	0.099	-0.718
Win Rate	67%	50%	100%
Average Win	3.59%	3.36%	2.51%
Average Loss	-1.22%	-1.22%	0%
Profit-Loss Ratio	2.96	2.76	0
Maximum Drawdown	14.600%	14.600%	8.600%
Alpha	0.004	0.089	-0.068

Approach 3 yields a higher annualised rate of return than the Baseline and Approach 2 at 5.105% and even earns 3.597% during sideways trend (Jan '22 – Feb'23), as indicated by the red rectangle in *Figure 9*. This strategy outperforms the market during sideways market as indicated by a positive alpha. However, the algorithm only earns 2.148% during bullish trend, underperforming the market, as evidenced by a slightly negative alpha.

These results suggest that this algorithm can profit during the sideways trend by sacrificing returns during the bullish trend.

4.5. Approach 4 – Moving average crossover (Window = 30)

Approach 4 addresses the lagging property of the moving average during bullish trends.

The project investigates the impact of reducing the window size to 30 to mitigate lag. The algorithm is shown in *Figure 10*.

Approach 4 – Simple moving average crossover (Window = 30)

```
For each trading day
  if price > moving average when crossover then
    Buy()
  else
    Sell()
ENDFor
```

Figure 10 Trading algorithm of the simple moving average approach with reduced window size

The algorithm was back tested to reduce lag during bullish trend, addressing the second major problem of the Baseline, using the same S&P 500 (SPY) data from Jan 1 2022 to 30 Apr 2024. The results are shown in *Figure 11*.

Figure 11 30-day moving average plotted on the SPY closing price, with trading signals generated by Approach 4

Similar to the Baseline, the blue line indicates the 30-day moving average. Green triangles indicate buy signals when the price is above the moving average during crossover, while red triangles indicate sell signals.

The evaluation metrics of Approach 4 are summarised in *Table 5*.

Table 5 Back testing evaluation metrics of Approach 4

Approach 4 – Simple moving average crossover (Window = 30)			
Evaluation Metrics	U.S. Stock	Jan '22 – Feb '23	Mar '23 – Apr'24
Annualized Rate of Return	8.394%	-2.899%	20.958%
Sharpe Ratio	0.253	-0.333	1.194
Win Rate	42%	40%	43%
Average Win	3.75%	2.10%	4.86%
Average Loss	-1.17%	-1.93%	-0.60%
Profit-Loss Ratio	3.20	1.09	8.06
Maximum Drawdown	12.600%	12.600%	4.200%
Alpha	0.023	-0.004	0.026

Approach 4 generates higher profit than the Baseline during bullish trends, yielding 20.958% compared to the Baseline's 18.557% between Jan 2022 and Feb 2023. By using a smaller window size, the moving average becomes more sensitive to recent price changes. This reduces lag and captures more profit during bullish trends. However, the increased sensitivity also results in more frequent unwanted signals during sideways trends, reducing performance between Mar 2023 and Apr 2024.

These results suggest that this algorithm can profit more during the bullish trends at the cost of more frequent unwanted signals during the sideways trends.

4.6. Summary

In summary, two major problems were identified in the Baseline approach. The first problem is the algorithm's inability to profit due to frequent unwanted signals during sideways trends. The second problem is the algorithm's failure to effectively capture profits from opportunities due to the lagging property of moving averages during bullish trends.

Approach 1 aimed to address the first major problem by minimising unwanted trades using confidence intervals. While it was better than the Baseline in reducing losses during sideways trends, it sacrificed returns during bullish trends and failed to profit from sideways trends.

Approach 2 attempted to capture profits from sideways trends by using the relative strength index (RSI) to identify overbought or oversold conditions, assuming prices would revert to the average over time. Like Approach 1, it reduced losses during sideways trends by sacrificing returns during bullish trends. However, it still could not fully capture peaks and troughs and failed to profit consistently from sideways trends.

Approach 3 solved the first major problem and issue identified in Approach 2 by capturing peaks and troughs. It built upon the idea in Approach 2 by considering local maximum and minimum values, not solely relying on RSI crossovers with thresholds. The results suggest that this algorithm can profit during sideways trends by sacrificing returns during bullish trends.

Approach 4 aimed to mitigate the lagging property of moving averages by reducing the window size to 30. It successfully captured more profit during bullish trends by making the moving averages more sensitive to recent price changes, thereby reducing lag. However, it resulted in more frequent unwanted signals when the price moves along the moving average during sideways market.

Overall, the proposed approaches demonstrate varying degrees of effectiveness in addressing the major problems identified in the Baseline approach, with trade-offs between profitability during bullish trends and the frequency of unwanted signals during sideways trends.

5. Difficulties and Mitigations

The project encountered some difficulties when using the back testing platform QuantConnect, introduced in Section 3.3, for performance evaluation. For instance, some prior knowledge is needed to transfer the algorithm written in Python on local machines to the online platform.

Key functions and syntax of the platform need to be understood before migrating the code. This is mitigated by spending time learning about the key functions and syntax of the online platform. To ensure a comprehensive understanding, all algorithms were also implemented independently using Python in addition to the established tools like QuantConnect. The trading signals generated by both implementations were compared to ensure accurate implementation.

Furthermore, developing and improving algorithmic trading strategies proved challenging due to a lack of familiarity with algorithmic trading and no prior experience in algorithmic trading and technical analysis. This initial unfamiliarity made it difficult to design robust strategies and understand the intricate details of trading algorithms. To mitigate these challenges, the team is committed to continuous research and learning throughout the project. This involved gaining a deeper understanding of the working principles of various algorithms and staying updated with the latest advancements in the field. The team also seeks to improve their knowledge of quantitative trading to create more effective strategies.

By focusing on these learning and research activities, the project team gradually overcame the initial difficulties and enhanced their algorithmic trading strategies, leading to better performance and adaptability.

6. Future Work

This section outlines the planned future directions of the project. The goal is to expand and enhance the implemented methodologies to improve trading strategies' performance and adaptability. The project is on schedule, with the Minimum Viable Product (MVP) expected to be ready by January.

6.1. Model Implementation

The current models use technical indicators. Further work will explore other types of models, such as reinforcement learning and large language models (LLM), as discussed in Section 3.2. For instance, reinforcement learning could be used to find the optimal hyperparameters for the algorithm. This is scheduled for completion by February.

6.2. Model Enhancement and Ensemble

The baseline approach is inconsistent and cannot adapt to different market conditions as reflected in Section 4. While the implemented models address one of the major problems and only work well in either sideways or bullish trends, more research is needed to profit from

various market conditions consistently. Moreover, black swan analysis is required to make the research more comprehensive.

Currently, the models perform well on one type of trend (bullish or sideways). Future enhancements will focus on the algorithm's strengths and weaknesses. Model ensemble techniques will be explored to balance these strengths and weaknesses. For example, the project could first classify the type of trend and then use the respective algorithm that is effective for that trend. This part is expected to be completed by February.

6.3. Exploration of Textual Data

Existing research and models usually focus on numeric data, as discussed in Section 2.2. The project will explore processing textual data from news articles and social media posts using natural language processing (NLP) techniques. Additionally, the use of large language models (LLMs) for advanced NLP tasks will be explored. This will be completed by March.

7. Conclusion

This project has laid the groundwork for improving algorithmic trading strategies by implementing a trend-following model and conducting an initial evaluation. While effective in capturing market momentum, this approach shows dependency on market conditions, leading to inconsistent profitability. This highlights the need for more robust and adaptive strategies.

The project identified two major problems: frequent unwanted signals during sideways trends and the lagging property of the moving average during bullish trends. Approaches 1-3 tried to address the first problem to profit from sideways trends but sacrificed profit during bullish trends. Approach 4 aimed to address the second problem to profit more during bullish trends but sacrificed profit from sideways trends.

The project's significance lies in its potential to enhance trading performance through diverse data types and advanced machine learning techniques. It offers a structured methodology for integrating traditional and modern trading models, providing valuable insights for both academic research and practical applications in financial trading. However, the current model's limitations, such as reliance on specific market conditions and overfitting, necessitate further development. Future work will focus on developing a balanced strategy that addresses both frequent unwanted transactions during sideways trends and the lagging property during bullish trends. This can be achieved by exploring other types of models, such as reinforcement learning for hyperparameter optimization, using ensemble techniques to classify trends and apply

effective algorithms accordingly, and incorporating textual data from news and social media using natural language processing (NLP) for richer insights.

By building on the initial implementation and addressing its limitations, the project aims to develop a robust and adaptable trading system. This system will leverage both numeric and textual data to enhance decision-making, ultimately contributing to more effective and reliable trading strategies in dynamic market environments.

8. References

Addy, W. A., Ajayi-Nifise, A. O., Bello, B. G., Tula, S. T., Odeyemi, O., & Falaiye, T. (2024). Algorithmic Trading and AI: A Review of Strategies and Market Impact. *World Journal of Advanced Engineering Technology and Sciences, 11(1)*, 258–267.

Ayala, J., García-Torres, M., Noguera, J. L., Gómez-Vela, F., & Divina, F. (2021). *Technical analysis strategy optimization using a machine learning approach in stock market indices*. Retrieved from https://doi.org/10.1016/j.knosys.2021.107119

Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy, 81(3)*, 637–654.

Boehmer, E., Fong, K., & Wu, J. (2021). Algorithmic Trading and Market Quality: International Evidence. *Journal of financial and quantitative analysis, 2021-12, Vol.56 (8)*, 2659-2688.

Cuthbertson, K., Nitzsche, D., & O'Sullivan, N. (2010). Mutual Fund Performance: Measurement and Evidence. *Financial Markets, Institutions & Instruments, 19(2)*, 95–187.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Retrieved from https://doi.org/10.48550/arxiv.1810.04805

Dutta, C., Karpman, K., Basu, S., & Ravishanker, N. (2023). Review of Statistical Approaches for Modeling High-Frequency Trading Data. *Sankhyā. Series B (2008), 85(Suppl 1*, 1–48.

Global Market Insights. (2024). *Algorithmic Trading Market Size*. Retrieved from https://www.gminsights.com/industry-analysis/algorithmic-trading-market

Hong, N., Wang, C., Aoshi, S., Hitomi, S., Li, Y., & Debopriyo, R. (2024). AI in Stock Market Forecasting: A Bibliometric Analysis. *SHS Web of Conferences, 2024, Vol.194*, 1003.

Kissell, R. (2020). *Algorithmic Trading Methods: Applications Using Advanced Statistics, Optimization, and Machine Learning Techniques.* Elsevier Science & Technology.

Koo, J. (2024). AI is not careful: approach to the stock market and preference for AI advisor. *International Journal of Bank Marketing*. Retrieved from https://doi.org/10.1108/IJBM-10-2023-0568

Majidi, N., Shamsi, M., & Marvasti, F. (2024). Algorithmic trading using continuous action space deep reinforcement learning. *Expert systems with applications, 2024-01, Vol.235*, 121245.

Markowitz, H. (1952). Portfolio Selection. *Journal of Finance*.

Ni, H., Meng, S., Chen, X., Zhao, Z., Chen, A., Li, P., . . . Chan, Y. (2024). *Harnessing Earnings Reports for Stock Predictions: A QLoRA-Enhanced LLM Approach*. Retrieved from https://doi.org/10.48550/arxiv.2408.06634

Sharpe, W. (1964). Capital Asset Prices: A Theory of Market Equilibrium. *Journal of Finance, Vol. 19, No. 3*, 425-422.

Sukma, N., & Namahoot, C. S. (2024). Enhancing Trading Strategies: A Multi-indicator Analysis for Profitable Algorithmic Trading. *Computational Economics*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Zhang, L., Wu, T., Lahrichi, S., Salas-Flores, C., & Li, J. (2022). *A Data Science Pipeline for Algorithmic Trading: A Comparative Study of Applications for Finance and Cryptoeconomics*. Retrieved from https://arxiv.org/pdf/2206.14932

Zhang, W., Zhang, N., Yan, J., Li, G., & Yang, X. (2022). Auto uning of price prediction models for high-frequency trading via reinforcement learning. *Pattern Recognition, 125, 108543*. https://doi.org/https://doi.org/10.1016/j.patcog.2022.108543

9. Appendices

9.1. Definition of the evaluation metrics

1. Annualised Rate of Return (ARR)
   The Annualised Rate of Return is a typical and generally understood metric to measure investment performance. It measures the yearly return of an investment strategy. Different strategies may have different evaluation period. Therefore, the rate of return (R_t) at t is annualised to make it more consistent. A higher annualised rate of return indicates a more profitable strategy.

$$R_{annualised} = (1 + R_t)^{1/n} - 1$$

Equation 5

   Referring to *Equation 5*, the annualised rate of return ($R_{annualised}$) is calculated by annualising the rate of return (R_t) according to the number of years (n) in the evaluation period.
2. Sharpe Ratio
   The Sharpe Ratio is a widely used metric in evaluating investment performance. It evaluates the risk-adjusted return of an investment strategy, helping to determine if returns are due to smart investment decisions or excessive risk. A higher Sharpe Ratio indicates better reward-to-risk ratio.

$$Sharpe\ Ratio = \frac{R - R_f}{\sigma}$$

Equation 6

   In *Equation 6*, the Sharpe Ratio is measured. In the equation, *R* refers to the return of an investment while R_f refers to the risk-free rate, typically estimated using U.S. Treasury Bond interest rates. σ refer to the standard deviation of the investment return.
3. Win Rate
   The Win Rate is the percentage of profitable trades out of the total number of trades, indicating the competence of a trading strategy disregarding the return of the trades. The Win Rate is calculated using *Equation 7*. A higher win rate suggests a more successful strategy.

$$Win\ Rate = \frac{Number\ of\ Profitable\ Trades}{Total\ Number\ of\ Trades}$$

Equation 7

4. Maximum Drawdown
   The Maximum Drawdown measures the largest loss from a peak to a trough before a new peak is achieved, as defined in *Equation 8*. It assesses the potential downside risk of a trading strategy. A higher maximum drawdown indicates greater potential loss.

$$Maximum\ Drawdown = \frac{Trough\ Value - Peak\ Vakue}{Peak\ Value}$$

Equation 8

5. Profit Factor
   The Profit Factor measures the ratio of gross profit to gross loss, indicating the profitability of a trading strategy. A profit factor greater than 1 indicates a profitable strategy.

$$Profit\ Factor = \frac{Gross\ Profit}{Gross\ Loss}$$

Equation 9

6. Alpha
   The Alpha (α) measures the excess return of an investment relative to the return of a market benchmark, indicating the value that an algorithmic trading strategy add to or subtracts from the market return. A positive alpha indicates outperformance, while a negative alpha indicates underperformance.

$$\alpha = R - [R_f + \beta(R_m - R_f)]$$

Equation 10

   Alpha (α) measures the unsystematic return of the strategy in *Equation 10*. R is the rate of return of the algorithmic trading strategy, R_f is the risk-free rate, R_m is the return of the market portfolio, and β is its beta. Beta (β) is proportional to the covariance between the strategy return and market return.

9.2. Project Schedule

Phase/Milestone	Month	Tasks	Deliverables
1 Preparation	Aug 2024	• Brainstorm ideas and confirm topic (10) • Perform background research on algorithmic trading (20) • Conduct market research on investment apps (15)	2 Oct 2024 • Detailed project plan • Project web page
2 Planning	Sep 2024	• Consult project supervisor (5) • Define project scope (5) • Define project objectives (5) • Define main features (5) • Research on various algorithmic trading strategies (20) • Prepare the detailed project plan (15) • Prepare the project web page (5)	
3 Implementation	Oct 2024	• Research and implement various algorithmic trading strategies (40) • Perform back testing in different markets and various market conditions (20)	27 Jan 2024 • Preliminary implementation and prototype • Prototype testing • Interim Report
	Nov 2024	• Propose enhancements on existing algorithmic trading strategies (30) • Explore the feasibility of incorporating different textual data (30)	
4 Prototyping	Dec 2024	• Explore the feasibility of related articles collection and summarisation (10) • Design and implement the dashboard to summarize key insights (20) • Design and implement the integrated investment app (20)	

5 **Testing**	Jan 2025	• Minimal Viable Product (MVP) prototype ready (10) • Perform User Acceptance Testing (UAT) to validate the functionality, allowing stakeholders to identify issues or discrepancies and provide feedback for necessary adjustments (20) • Perform Usability Testing to ensure user satisfaction by evaluating the prototype's ease of use involving the user interface (UI) and user experience (UX) (20) • Prepare the interim report (20)	
6 **Fine-tuning**	Feb 2025	• Implement remaining functionalities (30) • Improve the product according to feedback from stakeholders (30)	<u>22 Apr 2025</u> • Implementation of the final product • Final Report
	Mar 2025	• Implement the final algorithmic trading strategy (20) • Cut off back testing and organise the results (20)	
	Apr 2025	• Forward test the final algorithmic trading strategy (20) • Prepare the final report (20) • Prepare for the final presentation (10) • Prepare the poster (5)	

Table 6 Project Schedule

Note: Estimated learning hours for each milestone are indicated inside the parenthesis.