Smarter Investment using Big Data, Data Science and Algorithmic Trading

Final Report

Chan Chun Hei (3035684908)

Supervisor: Prof SM Yiu

Department of Computer Science

The University of Hong Kong

April 21, 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. The project began by implementing a trend-following strategy using moving averages. With back testing, it reveals a significant dependency on market trends that led to inconsistent profitability. To address these limitations, the project explored the use of confidence intervals and momentum oscillators. The findings revealed the need of trend detection. Therefore, the project designed a sophisticated trend classification model featuring three key innovations including customisable threshold parameters, signal harmonization algorithms, and user-labelled training data. These adaptations allowed the system to better accommodate diverse market conditions and trader preferences. Complementing these quantitative improvements, the project explored the integration of qualitative data sources including financial news and social media sentiment. While demonstrating technical viability, these explorations also revealed inherent challenges in textual data processing for trading applications. 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
GAN	Generative adversarial network
GPT	Generative pre-trained transformer
GUI	Graphical user interface
HFT	High frequency trading
LLM	Large language model
LSTM	Long short-term memory
NLP	Natural language processing
RSI	Relative strength index
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.

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*.

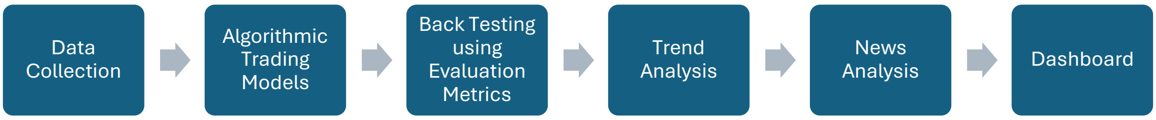

Figure 1 Sequential relationship of key components in the project

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. Trend analysis (*Section 3.4*) follows to provide trend insights for users. News analysis (*Section 3.5*) then provides qualitative insights to users. Finally, the results are then visualized in an interactive Dashboard (*Section 3.6*). 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

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

Inspired by the findings in the preliminary approach, more advanced strategies were also implemented and tested. They were based on moving average confidence interval and relative strength index local maximum and minimum. The idea behind each proposed strategy is discussed in *Section 4*.

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. Trend Analysis

The project explores various techniques in identifying and classifying trend into upward trend (+1), downward trend (-1) and no trend (0). It mainly explored the parallel ensemble model and LSTM neural network.

A parallel ensemble model was implemented as the baseline model. It aggregates predictions from multiple technical indicators and generate decision through consensus vote by summation. The indicators include simple & exponential moving averages, MACD (moving average convergence divergence), RSI (relative strength index), and rolling slope.

The trend calculation methods are defined as follows:

- Simple Moving Average
    - SMA_5 > SMA_10: Up (+1)
    - SMA_5 < SMA_10: Down (-1)
    - Otherwise: No trend (0)
- Exponential Moving Average
    - EMA_5 > EMA_10: Up (+1)
    - EMA_5 < EMA_10: Down (-1)
    - Otherwise: No trend (0)
- Moving average convergence divergence (MACD)
    - MACD > Signal: Up (+1)
    - MACD < Signal: Down (-1)
    - Otherwise: No trend (0)
- Relative strength index (RSI)
    - 50 < RSI <= 80 or RSI < 20: Up (+1)
    - 20 <= RSI < 50 or RSI > 80: Down (-1)
    - RSI = 50: No trend (0)
- Rolling slope
    - Slope > 0.05: Up (+1)
    - Slope < 0.05: Down (-1)
    - Otherwise: No trend (0)

The consensus vote by summation is illustrated below:

- Total votes >=3: Upward trend (Strong if total votes >=5)
- Total votes <=-3: Downward trend (Strong if total votes <=-5)
- Otherwise: No trend

On top of the baseline model, the project also applies a smoothing algorithm by calculating the average to reduce noise and adjust the degree of trend across different timeframe. This provides users with customisable trend classification. For each day, the project averages trend values across a predefined window (current day ± n surrounding days). For instance, if we set n=1, it means the project will apply a 3-day harmonisation (current day +- 1 surrounding day).

The final trend classification is determined by the positive and negative threshold. The rules are as follows:

- Upward trend: Harmonized value >= positive threshold
- Downward trend: Harmonized value <= negative threshold
- Neutral/No trend: Values between thresholds (exclusive)

This approach features adjustable parameters that let users to control sensitivity. Harmonisation filters random fluctuations while preserving sustained trends.

The hybrid LSTM neural network combines technical indicators and machine learning approach. In addition to the indicators mentioned above, this approach uses human-labeled training that tailored to user-defined trend thresholds features and an enhanced feature set including Bolinger Bands, volume change, volume simple moving average, and momentum.

The architecture of the model is as follows:

```
LSTM(128, return_sequences=True, input_shape=input_shape)
Dropout(0.4)
LSTM(64, return_sequences=False)
Dropout(0.3)
Dense(32, activation='relu')
Dense(num_classes, activation='softmax’)
```

Figure 3 LSTM model architecture

This approach uses human-defined label for training, enhancing its adaptability by offering customisable trend thresholds through adjustable training data.

3.5. News Analysis

This study investigates the integration of qualitative textual data into the trading system through a systematic pipeline comprising data acquisition, processing, and analysis. The methodology employs web scraping techniques for data collection and Large Language Models (LLMs) for content distillation and sentiment analysis.

The pipeline acts as a foundation and proof of concept to show it is feasible.

Financial news articles were programmatically retrieved from Google Finance using Python's requests and BeautifulSoup libraries. A two-stage refinement process was implemented:

Structural Cleaning:

- Removed HTML tags, advertisements, and non-article text elements
- Preserved semantic content through regular expression based extraction

Semantic Analysis:

- Deployed Azure OpenAI's GPT-4 model (API version 2024-02-01) with custom prompts to:
    - Summarize key market-relevant information
    - Classify sentiment polarity (bullish/bearish/neutral)
    - Identify sector-specific implications

System Prompt:

> You are an advanced financial analysis assistant specializing in market impact assessment. Your role is to analyse provided articles, news, or reports and generate insightful, data-driven assessments on how they may affect market conditions, sector performance, and specific stocks.
>
> Key Responsibilities:
>
> 1. Impact Analysis: Identify direct/indirect effects of the fed-in content on markets (e.g., sector trends, volatility, macroeconomic shifts).
> 2. Sentiment Evaluation: Assess bullish/bearish undertones and quantify potential market reactions (short/mid/long-term).
> 3. Stock-Specific Insights: Highlight companies likely to benefit/suffer, referencing fundamentals (P/E ratios, growth projections) when possible.
> 4. Actionable Advice: Provide concise investor recommendations (e.g., "Monitor X sector," "Consider hedging Y exposure").
> 5. Risk Awareness: Flag uncertainties, conflicting signals, or overhyped narratives.
>
> Output Guidelines:
>
> - Structure responses with: Summary (1-2 sentences), Key Impacts (bulleted list), Top Affected Stocks/Sectors, and Suggested Actions.
> - Use clear, professional language but avoid excessive jargon.

- Differentiate between high-confidence analysis and speculative trends.
- Cite specific passages from input content that justify your conclusions.
- Disclaimers: Always include: "AI generated content. Not financial advice. Investors should cross-validate with latest data and risk tolerance."

Figure 4 LLM system prompt for semantic analysis

3.6. 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.

4. Results

4.1. Algorithmic Trading

This section outlines the results of different algorithmic trading approaches.

4.1.1. 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 5*.

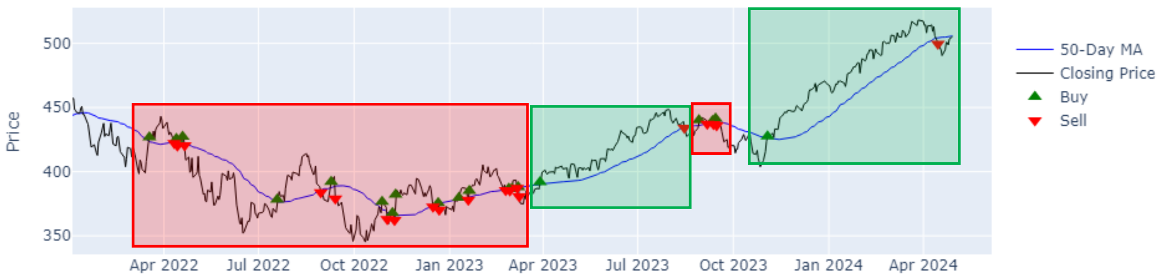

Figure 5 50-day moving average plotted on the SPY closing price, with trading signals generated by the Baseline

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 5*, 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 5*, 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 4.1.2*, *Section 4.1.3*, and *Section 4.1.4*.

The second major problem of the model is that it lags in capturing profits during bullish trends. Highlighted by the green boxes in *Figure 5*, the algorithm enters and leaves the market late. The project proposes adjusting the window size to reduce lag. The details are explained in *Section 4.1.5*.

4.1.2. 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 6*. 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 -* $1 \times \sigma$ *during crossover* ***then***   *Buy()*   ***else if*** *price < moving average +* $2 \times \sigma$ *during crossover* ***then***   *Sell()* ***ENDFor***

Figure 6 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 7*.

Figure 7 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 4.1.3*, aiming to profit from the sideways trends.

4.1.3. 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 8*.

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 8 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 9*.

Figure 9 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 9*. 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 9*. This issue will be addressed in Approach 3 in *Section 4.1.4*.

4.1.4. 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 10*.

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 10 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 11*.

Figure 11 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 11*. 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.

4.1.5. 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 12*.

Approach 4 – Simple moving average crossover (Window = 30)
For each trading day   ***if*** *price > moving average when crossover* ***then***   *Buy()*   ***else***   *Sell()* ***ENDFor***

Figure 12 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 13*.

Figure 13 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.

4.1.6. 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.

4.2. Trend Analysis

The evaluation of algorithmic trading models revealed significant performance variability across different market regimes, highlighting the need for more sophisticated trend detection capabilities. To address this limitation, the project developed and tested multiple trend analysis methodologies designed to provide traders with enhanced market regime insights prior to strategy selection. The following sections present empirical results from these investigations, comparing the effectiveness of various approaches in identifying and classifying market trends.

4.2.1. Baseline – Simple Parallel Ensemble Model

Given the inconsistent performance of algorithmic trading strategies under varying market conditions, this project proposes a classification model to categorize market trends into three states: upward, downward, or no trend.

The baseline model, as outlined in *Section 3.4*, was evaluated using S&P 500 (SPY) data from 1 Jan 2018 to 30 Mar 2025. The results are illustrated in *Figure 14*.

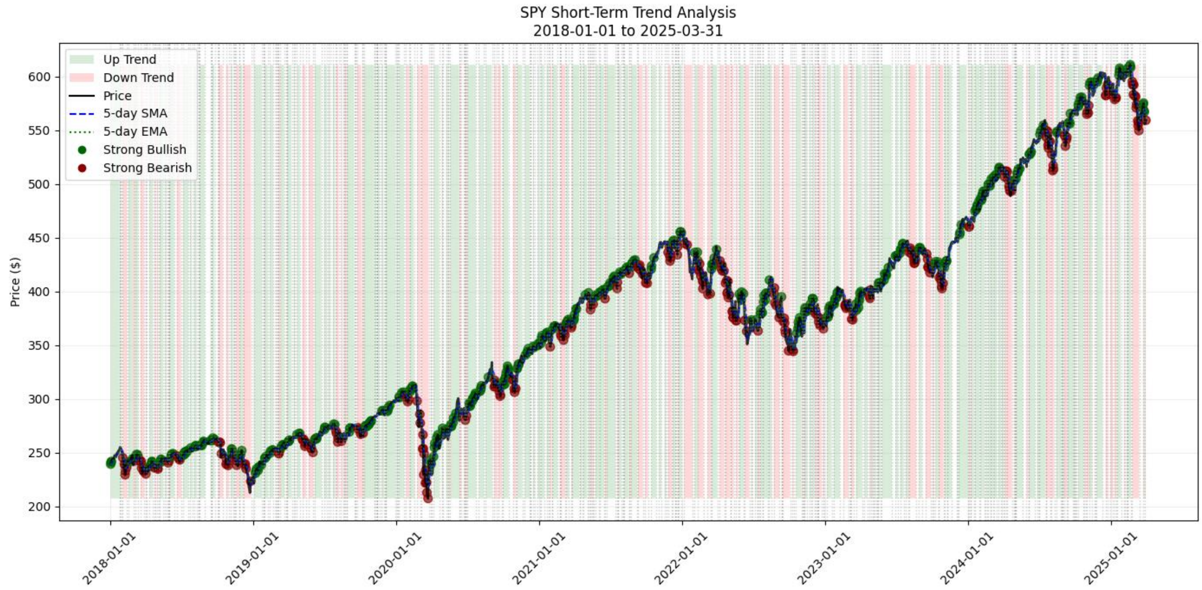

Figure 14 SPY closing price, with trend classification generated by the Baseline

The black line represents the closing price while the green and red regions represent the upward trend period and downward trend period respectively. The green and red dots mark the strong bullish and bearish signals respectively.

The figure reveals fragmented trend classifications, primarily due to the short evaluation window (5–10 days) relative to the entire seven-year timeframe. This fragmentation arises because the model treats each trading day independently, failing to capture multi-day momentum patterns. Consequently, the analysis lacks sequential context, leading to inconsistent trend identification over longer periods.

Additionally, the baseline model exhibits limited flexibility in adapting to different trading preferences. It does not account for varying asset volatilities or user-defined risk profiles. For instance, short-term traders may require finer-grained trend detection, while long-term investors might prefer broader trend classifications.

These limitations highlight the need for a more adaptive approach, which subsequent sections explore through alternative methodologies.

4.2.2. Approach 5 – Dynamic Parallel Ensemble Model

Building upon the baseline model, this approach introduces user-adjustable parameters and a dynamic smoothing algorithm to enhance trend classification robustness. The model applies harmonization with parameters n = 5, positive threshold = +0.5, and negative threshold = −0.5, refining trend signals by reducing short-term noise. The results shown in *Figure 15*, demonstrate improved trend continuity compared to the baseline.

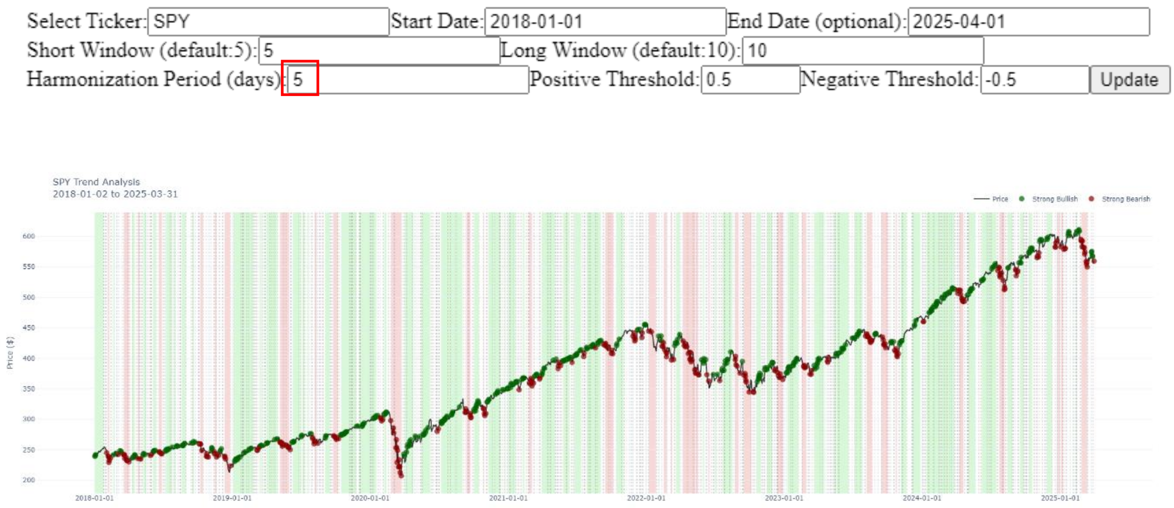

Figure 15 SPY closing price, with adjustable parameters and trend classification generated by Approach 5

The harmonization algorithm significantly reduces fragmentation in trend classification. Unlike the baseline model, which reacts sharply to daily price movements, this approach smooths trend signals by averaging classifications over multiple trading days. As a result, the trend zones in *Figure 15* appear more continuous, better capturing sustained upward or downward movements rather than transient fluctuations.

The model offers greater flexibility by allowing users to customize key parameters. Traders can adjust the indicator window to modify the sensitivity of technical signals, the harmonization period (n) to control smoothing intensity, and the classification thresholds to fine-tune the detection of bullish or bearish conditions. For example, a short-term trader might use a smaller harmonization window for rapid signal detection, while a long-term investor could increase the smoothing period to filter out market noise.

However, excessive indicator window and harmonization may introduce lag in trend detection. While smoothing improves trend continuity, an overly long harmonization period (e.g., n > 50) could delay the recognition of emerging trends, potentially causing missed entry or exit opportunities during rapid market shifts.

To assess the impact of parameter selection on trend classification, the model was tested with modified window sizes by increasing the short and long windows from (5, 10) to (50, 100) days. The results, presented in *Figure 16*, demonstrate two key effects.

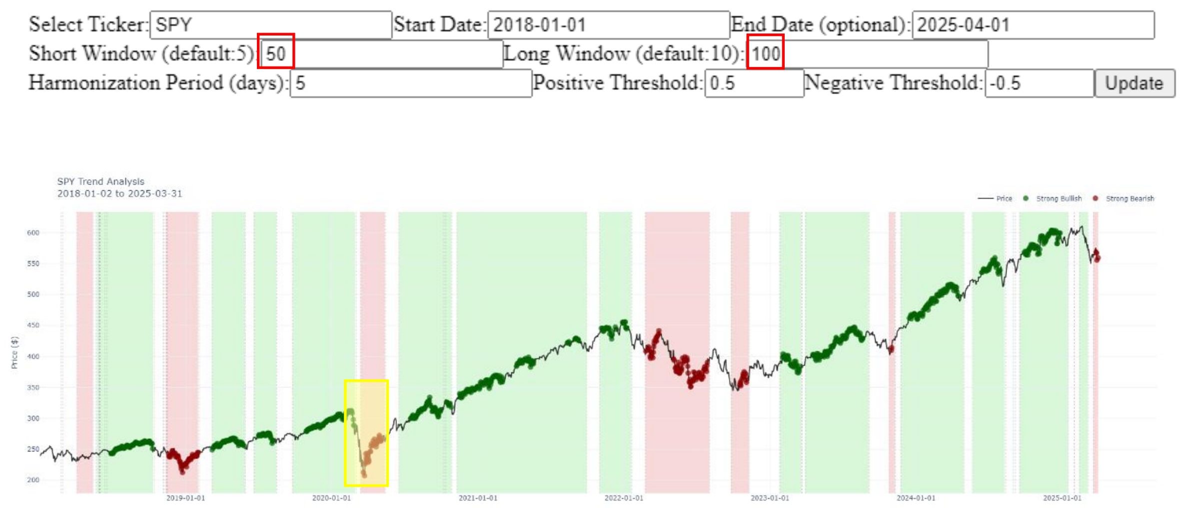

Figure 16 SPY closing price with trend classification generated by Approach 5 using extended windows

The expanded window size produced significantly less fragmented trend classifications compared to the baseline model. This improvement occurs because the larger calculation window smooths out short-term price fluctuations and noise, allowing the algorithm to focus on more sustained market movements. The resulting trend signals demonstrate greater consistency and better reflect longer-term price trajectories.

However, this enhanced stability comes at the cost of increased lag in trend identification. As clearly visible in the yellow-highlighted region of *Figure 16*, the model exhibits delayed responses to trend reversals when using the larger window settings. This lag occurs because moving averages with longer windows inherently respond more slowly to price changes, causing the algorithm to identify new trends only after they have become well-established.

These findings highlight the fundamental trade-off between signal stability and responsiveness in trend-following systems. While larger windows reduce false signals and provide more reliable trend identification, they may cause traders to miss early entry opportunities during market reversals. Conversely, smaller windows offer faster signal generation but may produce more erratic classifications during volatile market conditions.

The practical implications of these results suggest that window size selection should align with specific trading objectives and risk tolerances. Longer windows may be preferable for position traders and investors with longer time horizons who prioritize trend reliability over precise timing. Shorter windows could better serve active traders who require timely signals and can tolerate higher false positive rates. Future research could explore adaptive window sizing approaches that dynamically adjust based on current market volatility conditions to optimize this trade-off.

To further investigate the model's parameter sensitivity, the project examined the effects of adjusting the classification thresholds to more extreme values (+0.8 and -0.8) compared to the baseline settings (±0.5). The results of this configuration, presented in *Figure 17*, demonstrate significant changes in trend classification behaviour that provide valuable insights into the model's decision-making process.

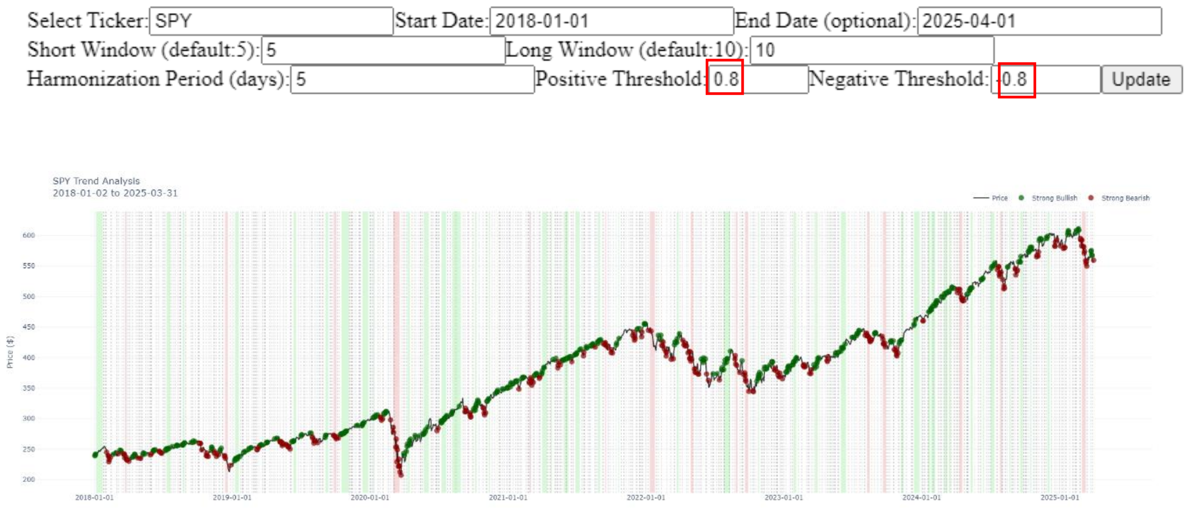

Figure 17 SPY closing price with trend classification generated by Approach 5 using more extreme harmonisation threshold

The implementation of stricter threshold criteria resulted in a notable increase in periods classified as having no trend, as evidenced by the expanded white regions in *Figure 17*. This outcome occurs because the more extreme thresholds require stronger evidence before assigning upward or downward trend classifications, effectively raising the confidence level required for trend determination. Consequently, the model becomes more selective in identifying trends, focusing only on those movements that exhibit greater magnitude and conviction.

This threshold adjustment creates a more conservative classification system that filters out marginal trends, potentially reducing false signals during periods of market indecision or low volatility. The model's increased selectivity may be particularly advantageous in sideways markets where clear trends are absent, as it avoids making premature classifications that could lead to suboptimal trading decisions. However, this benefit comes with the trade-off of potentially missing early signals of emerging trends that have not yet reached larger threshold levels.

The practical implications of these findings suggest that threshold selection should be carefully calibrated according to the trader's risk tolerance and market environment. More extreme thresholds may be preferable for conservative strategies that prioritize signal quality over coverage, while moderate thresholds might better suit approaches that value comprehensive trend detection. Future research directions could explore dynamic thresholding mechanisms that automatically adjust based on prevailing market volatility conditions, potentially offering an optimal balance between these competing considerations.

These results complement our earlier findings regarding window size adjustments, collectively demonstrating how parameter selection fundamentally shapes the model's behaviour and performance characteristics. The comprehensive analysis of both window sizes and classification thresholds provides practitioners with valuable guidance for customizing the model to their specific trading objectives and market conditions.

4.2.3. Approach 6 – LSTM Neural Network

This section presents the implementation and evaluation of a Long Short-Term Memory (LSTM) neural network for trend classification, which is based on the idea of allowing user-defined class. The LSTM architecture was specifically selected to address the limitations of traditional recurrent neural networks (RNNs), particularly their susceptibility to the vanishing gradient problem during extended sequence learning. This deep learning approach offers both technical advantages in pattern recognition and practical benefits in user customization.

The model was trained and tested on S&P 500 (SPY) data spanning from 1 Jan 2018 to 30 Mar 2025, with the classification results visualized in *Figure 18*.

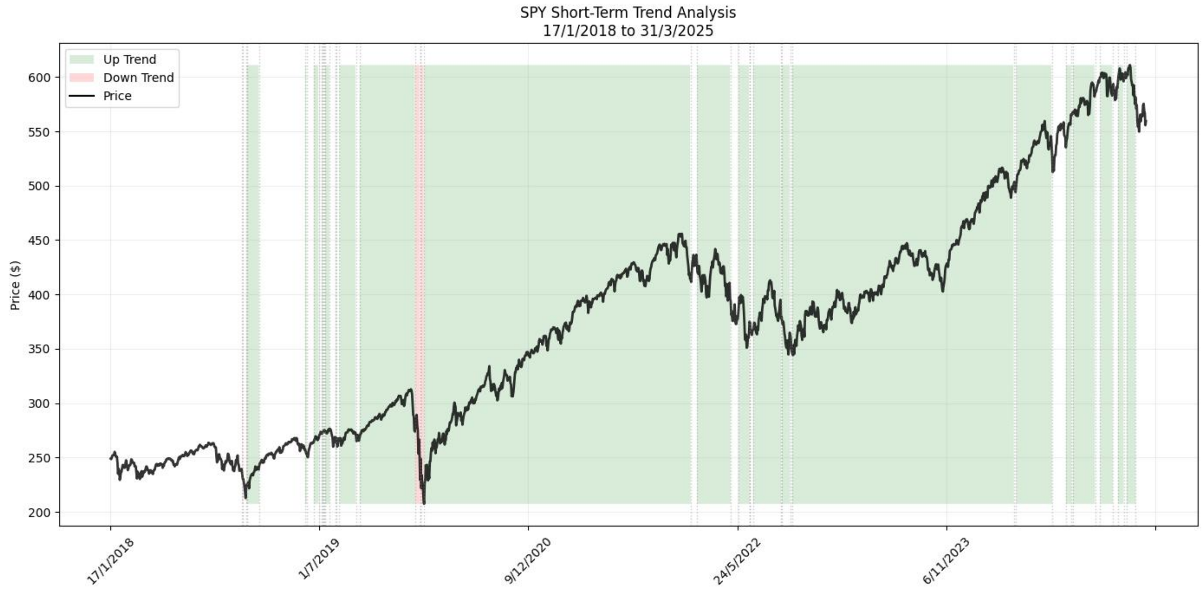

Figure 18 SPY closing price, with trend classification generated by Approach 6

The visualization maintains consistent formatting with previous approaches for comparative analysis, where the black line tracks closing prices while green and red regions denote upward and downward trend classifications respectively.

The LSTM-generated classifications demonstrate less fragmentation compared to the baseline, producing more coherent and sustained trend identifications. This enhanced performance stems from two key factors. First, the LSTM's inherent ability to process and learn from sequential data patterns rather than treating individual data points in isolation. Second, the model benefits from the more consistent labelling patterns in the training data, as human-labelled classifications naturally relate to individual preferences and exhibit less short-term variability.

The LSTM approach offers unique advantages in terms of customisation and adaptability. Unlike fixed-parameter models, this architecture permits users to define their own trend classification criteria (upward/downward/no trend) during the training process. This feature enables the model to adapt to individual interpretation styles and trading philosophies intuitively and directly.

The successful implementation of this LSTM model demonstrates the potential of deep learning techniques in financial time series analysis. By effectively capturing complex temporal dependencies in price movements and accommodating user-specific classification preferences, this approach addresses several limitations identified in the baseline model.

4.2.4. Summary

In summary, the baseline approach revealed two fundamental limitations in its simple parallel ensemble model. First, the model produced fragmented trend classifications due to its narrow evaluation window and daily independent analysis, failing to capture sustained market movements. Second, the rigid framework lacked adaptability to different trading styles or market conditions, offering no customization for user-specific needs.

Approach 5 addressed these issues by introducing a dynamic parallel ensemble model with three key innovations: (1) harmonization to smooth classifications and reduce fragmentation, (2) configurable window sizes for technical indicators, and (3) adjustable thresholds to control sensitivity. While this significantly improved trend continuity, experiments revealed an inherent trade-off. Larger windows reduced fragmentation but introduced lag in trend detection, while extreme thresholds (±0.8) filtered noise at the cost of fewer classified trends.

Approach 6 represented a paradigm shift through its LSTM neural network architecture, which fundamentally redefines trend analysis through human-defined classification label. By processing sequential data rather than isolated points, the LSTM model achieved more coherent classifications that better reflect actual market behaviour. Its standout advantage lies in customizable training, allowing users to define their own trend criteria (upward/downward/no trend) and dynamically adjust to personal risk profiles, which is a capability absent in the baseline approach. The model's ability to learn temporal patterns from human-labelled data enhanced classification quality while maintaining interpretability.

Collectively, these approaches demonstrate an evolution from rigid, rule-based systems (Baseline) to parameter-adaptive models (Approach 5) and intelligent, learning-based solutions (Approach 6). Approach 5 added necessary flexibility to traditional technical analysis, while Approach 6's LSTM use user labelled classification and overcame the fundamental limitation of isolated-day analysis through sequence learning. However, the approaches demonstrate some trade-offs, and careful consideration is needed before selecting a model and setting the parameters.

4.3. News Analysis

This section presents empirical validation of the proposed qualitative data processing pipeline, demonstrating its feasibility for extracting and analyzing financial news relevant to S&P 500 ETF (SPY) trading decisions.

The web scraping framework successfully retrieved six representative news articles from Google Finance. The sample captures diverse SPY-related content including:

- Macroeconomic policy analysis (Geo-political impacts)
- ETF comparison strategies
- Options market anomalies
- Scheduled market-moving events

[1] Title: Market Outlook: Next Week's Most Important Events (NYSEARCA:SPY) (31 分鐘前)
By: Seeking Alpha
Website: https://seekingalpha.com/article/4776438-market-outlook-next-week-most-important-events
[2] Title: The Smartest S&P 500 ETF to Buy With $500 Right Now (18 小時前)
By: The Motley Fool
Website: https://www.fool.com/investing/2025/04/20/the-smartest-sp-500-etf-to-buy-with-500-right-now/
[3] Title: Fed Policy Shift and Global Supply Chain Concerns: Impact on $SPY Trading (3 天前)
By: Blockchain News
Website: https://blockchain.news/flashnews/fed-policy-shift-and-global-supply-chain-concerns-impact-on-spy-trading
[4] Title: SPY ETF News, 4/18/2025 (2 天前)
By: The Globe and Mail
Website:
https://www.theglobeandmail.com/investing/markets/stocks/HSY/pressreleases/31947199/spy-etf-news-4182025/
[5] Title: SPY, QQQ Call Volumes Spiked Minutes Before Tariff Pause Announcement: Alexandria Ocasio-Cortez Demands Disclosure From Congress Members (1 週前)
By: Benzinga

Website: https://www.benzinga.com/government/regulations/25/04/44728276/spy-qqq-call-volumes-spiked-minutes-before-tariff-pause-announcement-alexandria-ocasio-cortez-demands-disclosure-from-congress-members

[6] Title: Only 1 Of These 3 Unusually Active SPY Put Options Makes For A Good Bull Put Spread (3 天前)

By: Barchart.com

Website: https://www.barchart.com/story/news/31938797/only-1-of-these-3-unusually-active-spy-put-options-makes-for-a-good-bull-put-spread

Figure 19 Articles related to SPY from the web scraping pipeline

All extracted articles maintained complete metadata (title, publisher, timestamp, and source URL), with temporal relevance ranging from 31 minutes to one week prior to analysis. This confirms the pipeline's capability to gather timely, structured financial information from heterogeneous web sources.

The project also explored semantic analysis using large language model by processing Article 2. Here is the result:

Analysis:

Summary: The article discusses the benefits of investing in the Vanguard S&P 500 ETF amid market volatility caused by tariffs and economic concerns.

Key Impacts:
- Historic average return rate of 10% annually for the S&P 500.
- Vanguard's S&P 500 ETF offers exposure to 500 large U.S. publicly traded companies with a low expense ratio of 0.03%.
- The S&P 500 index has shown consistent long-term growth despite short-term fluctuations.
- Investing in a diversified ETF like VOO can provide stability during market turmoil caused by tariffs.
- Market volatility is expected due to uncertain tariff policies, potentially affecting short-term gains in the S&P 500.

Top Affected Stocks/Sectors:

- Vanguard S&P 500 ETF (VOO) - The highlighted ETF likely to benefit from investors seeking stability amid market uncertainties.
- Large-cap U.S. publicly traded companies across sectors like technology, financials, healthcare, consumer goods, and energy.

Suggested Actions:
- Monitor Market Volatility: Keep track of tariff-related developments and their impact on market stability.
- Consider Investing in Diversified ETFs: Look into ETFs like VOO for exposure to a broad range of companies during uncertain market conditions.
- Focus on Long-Term Growth: Emphasize long-term investment strategies given the historical growth trends of the S&P 500.
- Stay Informed: Keep abreast of changing economic conditions and tariff policies to make informed investment decisions.

AI generated content. Not financial advice. Investors should cross-validate with the latest data and risk tolerance.

Figure 20 Output of the LLM news analysis

While the pipeline demonstrates technical feasibility, some areas require refinement. More research is needed to fine tune the model for practical use.

The current implementation applies uniform processing to all news items regardless of publication time, despite the well-documented decay in market impact of financial news. Macroeconomic news loses its price-moving effect after some time (Engelberg & Parsons, 2011), yet the system weights a week-old article equally with a 30-minute alert. Future iterations should incorporate exponential time-decay factors aligned with established market microstructure research. The system currently lacks differentiation between established financial publishers and less regulated sources, potentially compromising analysis quality. Additionally, the AI-generated recommendations remain too generic for direct integration into trading systems, requiring further development to produce actionable signals.

Important ethical considerations emerge regarding AI's role in financial decision-making, particularly concerning the risk of hallucination. This phenomenon occurs when AI systems detect false patterns and generate inaccurate or nonsensical outputs. Users must remain aware of these limitations, including potential biases stemming from skewed training data. To promote transparency, all AI-generated content not verified by human analysts should carry clear disclaimers identifying it as machine-produced.

5. Difficulties and Mitigations

5.1. Algorithmic Trading Implementation

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.

5.2. Trend Analysis Limitations

Historical analysis reveals significant variability in trend patterns across different market regimes. The 2008 financial crisis produced prolonged bearish trends lasting over 12 months, while the COVID-19 pandemic saw a compressed V-shaped recovery within just 3 months. Recent trade war volatility in 2025 further demonstrates how geopolitical events can create unpredictable, policy-driven market movements.

These observations suggest that traditional trend classification models struggle to account for exogenous macroeconomic shocks. While extending the training window improves generalisability, no purely algorithmic solution can anticipate all potential market disruptions. It is necessary to maintain human oversight to adjust models during extraordinary events and customize parameters for specific market conditions.

The project has implemented two primary mitigation strategies to address these challenges. First, the framework incorporates user-configurable parameters that allow traders to adjust trend sensitivity thresholds according to their risk profiles and market outlooks. Second, the system supports integration of user-labelled training data, enabling customisation of trend classifications to align with individual interpretation styles and investment horizons. These adaptive features help bridge the gap between rigid algorithmic classification and the dynamic judgment of experienced market participants.

5.3. News Analysis Challenges

The semantic processing of financial news presents multiple technical hurdles. Linguistic complexities such as sarcasm in headlines (Haripriya & Patil, 2024) and domain-specific semantics reduce model accuracy, while multilingual coverage requires additional preprocessing. affects the model performance. The temporal dimension introduces further complications, as latency between news release, NLP processing, news analysis and trading signals diminishes the actionable value of insights.

6. Future Work

This section outlines the possible future directions of the project. The goal is to enhance the trading system's sophistication, robustness, performance and adaptability. The research will pursue four key directions to address current limitations and explore emerging methodologies.

6.1. Advanced Model Architecture

The project will investigate next-generation modelling techniques beyond current technical indicator-based approaches. Reinforcement learning frameworks are useful for dynamic hyperparameter optimization, automatically adjusting to changing market regimes. Additionally, the integration of large language models (LLMs) could enable more advanced interpretation of complex market patterns by combining quantitative signals with qualitative economic context.

6.2. Adaptive Ensemble Framework

Current algorithmic training models demonstrate regime-specific effectiveness, performing well in either bullish or sideways markets but lacking universal adaptability. Future development will focus on creating an intelligent ensemble system that dynamically weights constituent models based on real-time market condition classification. This will incorporate black swan event detection mechanisms to improve resilience during extreme market disruptions.

6.3. Generative Adversarial Network for Trend Analysis

Building on the foundational work by Zhang et al.'s (2019), the project proposes implementing a specialized Generative Adversarial Network (GAN) architecture for financial time series generation.

The generator forecasts the future trend of stocks while the discriminator aims to evaluate the authenticity of generated data against historical market behaviour. This approach could significantly improve trend prediction by learning the underlying data distribution of market movements during different economic regimes. GAN has shown exceptional contribution in image generation nowadays. It would be interesting to explore this technique in the field of financial trend prediction.

6.4. Research on Textual Data

While the current pipeline demonstrates basic feasibility of news analysis, substantial work remains to operationalise textual data for trading. The project can expand research into advanced textual data processing techniques to overcome current limitations in news analysis.

Future work can develop specialized fine-tuning protocols for financial large language models to improve factual accuracy and reduce hallucination rates in market analysis. The research can implement sophisticated temporal decay algorithms that quantitatively model the diminishing impact of news events over time based on their category and source credibility.

Further development can focus on creating multi-modal analysis frameworks that effectively combine linguistic sentiment signals with traditional technical indicators for more robust trading signals.

These combined advancements aim to transform qualitative data from a supplementary signal into a core component of algorithmic decision-making while maintaining rigorous validation standards against market outcomes.

7. Conclusion

This project has systematically evaluated and enhanced algorithmic trading strategies and trend analysis through an integrated approach combining traditional technical indicators with modern machine learning techniques. The research successfully achieved its core objectives by: (1) evaluating different algorithmic trading strategies across diverse market regimes, (2) enhancing baseline algorithmic trading models through statistical methods, (3) enhancing trend analysis algorithm through technical indicator refinement and LSTM integration, (4) demonstrating the feasibility of textual data integration while analysing its challenges, and (5) developing an interactive dashboard prototype for strategy visualization.

Key findings reveal fundamental trade-offs in strategy performance across market conditions. Trend-following approaches based on moving averages proved effective in bullish markets but generated excessive false signals during sideways trends, while mean-reversion strategies showed the inverse pattern. The dynamic parallel ensemble model and the LSTM hybrid model addressed these limitations by improving trend continuity through adjustable parameters and sequential analysis. The project also shows the technical viability and inherent challenges of text analysis.

The project's significance lies in its threefold contribution to the field: First, it bridges classical technical analysis with contemporary AI techniques through its hybrid modelling approach. Second, it provides empirical evidence of market-regime dependencies that inform both academic research and practical strategy development. Third, it establishes a framework for personalized algorithmic trading through configurable interfaces and adaptive model architectures.

Future research will pursue four key directions to address current limitations: (1) implementing reinforcement learning for dynamic hyperparameter optimization and large language model for more advanced interpretation of complex market data, (2) developing ensemble classifiers that combine LSTM, technical indicators, and NLP outputs, (3) exploring GAN for more advanced trend analysis, and (4) advancing natural language capabilities for financial text analysis.

The ultimate goal remains the development of a self-adapting trading system that maintains robustness across market regimes, balance trend capture and sideways-market resilience. By continuing to integrate numeric and textual data sources within an investor-centric framework, this research aims to create an all-rounded platform for traders that is capable of navigating today's dynamic financial markets.

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.

Engelberg, J., & Parsons, C. (2011). The Causal Impact of Media in Financial Markets. *The Journal of Finance (New York)*, 66(1), 67–97. https://doi.org/https://doi.org/10.1111/j.1540-6261.2010.01626.x

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

Haripriya, V., & Patil, P. (2024). Ambiguity and Sarcasm Detection on Social Media Data Using Ensemble Techniques. *SN Computer Science*, 5(7), 881-. https://doi.org/https://doi.org/10.1007/s42979-024-03233-1

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

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)	<u>2 Oct 2024</u> • 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)	<u>27 Jan 2024</u> • 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) • Prepare the interim report (20)	
6 Fine-tuning	Feb 2025	• Implement remaining functionalities (30) • Improve the product according to feedback from stakeholders (30)	22 Apr 2025 • 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	• 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.