University of Essex

Research Repository

Strongly-typed genetic programming with an active trading operator for algorithmic trading using fundamental, sentiment, and technical analysis

Accepted for publication in Applied Soft Computing

Research Repository link: https://repository.essex.ac.uk/43936/

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the published version if you wish to cite this paper.
https://doi.org/10.1016/j.asoc.2026.116566

www.essex.ac.uk

University of Essex

Strongly-typed genetic programming with an active trading operator for algorithmic trading using fundamental, sentiment, and technical analysis

Eva Christodoulaki[a,*], Michael Kampouridis[a]

[a]*School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom*

Abstract

Algorithmic trading has advanced significantly in recent years, with investors typically relying on either fundamental analysis (FA) or technical analysis (TA) indicators. Sentiment analysis (SA), however, has only gained substantial attention in the last decade. Consequently, algorithms have become essential tools in creating pre-programmed trading strategies, yielding notable financial gains. Despite the extensive individual study of these three types of analyses, their combined potential remains underexplored. Given the success of FA, TA, and SA in identifying profitable trading strategies, this paper aims to examine whether merging these indicators can further enhance profitability. To this end, we introduce a novel strongly-typed Genetic Programming (GP) algorithm within a soft computing framework, designed to enhance robustness in trading strategy development. Each branch in the algorithm's tree structure is dedicated to one analysis type, meaning that

*Corresponding author.

Email addresses: christodoulakis.eva@gmail.com (Eva Christodoulaki), mkampo@essex.ac.uk (Michael Kampouridis)

one branch includes only fundamental indicators, another solely sentiment indicators, and the final branch exclusively technical indicators. This structured approach allows for better exploration and exploitation of the indicator search space. Additionally, the algorithm incorporates a unique fitness function, which evaluates not just overall trading performance but also the contributions of the individual FA, TA, and SA branches. Furthermore, we propose an innovative GP operator that promotes active trading by injecting trees into the GP population capable of executing numerous trades while maintaining high profitability and low risk. The algorithm's effectiveness is tested using the stocks of 42 international companies, where it is compared against eleven alternative algorithms across three key financial metrics: Sharpe ratio, rate of return, and risk. The findings show that the proposed GP algorithm, leveraging evolutionary computing principles, demonstrates practical financial benefits. The algorithm enhances the profitability and risk-adjusted performance of trading strategies, achieving statistically significant improvements over competing models and active trading potential. This work contributes to the advancement of decision support systems and financial analysis applications.

Keywords: Fundamental Analysis, Sentiment Analysis, Technical Analysis, Genetic Programming, Algorithmic Trading

---

1. Introduction

The use of algorithmic trading has become increasingly widespread in the financial industry, leveraging pre-programmed strategies to achieve profit. As with many fields, algorithms play a pivotal role in finance, where researchers

apply Machine Learning (ML) techniques to analyse historical market data. These techniques help detect patterns that can be used to generate signals forecasting future trends, thereby optimising potential returns.

Algorithmic trading commonly relies on three main methods: fundamental analysis (FA), which evaluates a company's financial and economic fundamentals; sentiment analysis (SA), which interprets and classifies the sentiments, attitudes, or opinions found in textual data; and technical analysis (TA), which scrutinises price patterns and trends to discover trading opportunities. While earlier studies primarily focused on fundamental and technical analysis indicators, such as earnings per share (EPS), volatility, and moving averages, sentiment analysis indicators like sentiment polarity have recently gained prominence in the field.

Each of these methods has demonstrated the ability to produce profitable, low-risk trading strategies, yet their combined use remains relatively rare, largely due to the differing assumptions and methodologies underlying each approach. Given the success of these individual methods, there is growing interest in combining them to enhance trading strategy performance.

Building on this foundation, we propose a novel strongly-typed genetic programming (GP) algorithm that, for the first time, fully integrates all three types of analysis: FA, SA, and TA. Each GP tree is designed with distinct branches dedicated to FA, SA, and TA indicators. This structural constraint enhances the GP's ability to explore and exploit the search space of each type of indicator more effectively. Furthermore, type constraints are applied during crossover and mutation operations to ensure that the offspring generated remain valid.

Our fitness function incorporates a form of structural sampling by evaluating not only the overall performance of each evolved tree but also the performance of its constituent FA, SA, and TA branches. This design encourages the discovery of models that perform robustly across different market signal types. Conceptually, this aligns with sampling-based evaluation strategies commonly used in machine learning, where model robustness is assessed across multiple data partitions or objective components to avoid overfitting and improve generalisation. By integrating performance feedback from multiple subtrees into a unified fitness score, our approach effectively "samples" the quality of different decision paths within the model, guiding the evolutionary process toward more stable and diverse solutions.

Lastly, we aim to encourage active trading strategies that regularly assess market conditions and capitalise on short-term opportunities. Such strategies benefit from the ability to respond to price movements within shorter timeframes, potentially enhancing profitability. To facilitate this, we introduce a new GP operator, which complements traditional crossover and mutation. This operator selects FA, SA, and TA subtrees that are executing highly active, profitable, low-risk trading strategies and combines them into new trees for the subsequent generation.

The goal of this study is to demonstrate the proposed GP algorithm's ability to create innovative and profitable trading strategies by integrating insights from fundamental, sentiment, and technical analysis. We evaluate its performance against a set of eleven benchmarks, which include several GP variants, a financial benchmark, and four algorithmic benchmarks: multi-layer perceptron, support vector machine, extreme gradient boosting

(XGBoost), and long short-term memory (LSTM) neural network. The experiments are conducted using stock data from 42 international companies, with performance assessed across three key financial metrics: Sharpe ratio, rate of return, and risk. This way, we validate the practical utility of the proposed GP algorithm and demonstrate the algorithm's adaptability and practical impact.

In summary, besides continuing to analyse our work in detail, we extend our research compared to previous papers in the following ways: (i) we present a novel strongly-typed genetic programming (GP) algorithm that integrates three analysis types, instead of two as in the past (ii) the fitness function of the proposed GP model considers the overall performance of the tree and the individual contributions of the three branches, (iii) we introduce an active trading operator that combines parts of the most profitable and low-risk trees, and (iv) we compare our algorithm with a total of 11 benchmarks.

The paper is structured as follows: Section 2 provides a review of past research on the application of fundamental analysis (FA), sentiment analysis (SA), and technical analysis (TA) indicators in algorithmic trading. Section 3 outlines the methodology used in this study, starting with an explanation of the features (indicators) employed, followed by a detailed description of the proposed GP algorithm. The experimental setup, which includes the dataset, performance metrics, and benchmarks, is discussed in Section 4. The results and analysis are presented in Section 5, where we compare the performance of the GP algorithm and assess the effectiveness of combining FA, SA, and TA indicators. Finally, Section 6 summarises the key findings and suggests

potential directions for future research.

2. Literature Review

This section reviews prior research on financial forecasting and algorithmic trading, with a specific focus on studies that utilise indicators from fundamental, technical, and/or sentiment analysis.

Fundamental analysis (FA) is a method used to assess and evaluate securities—such as stocks, bonds, or currencies—based on intrinsic factors that influence their value. This analysis considers both qualitative and quantitative factors related to the asset, including financial statements, industry trends, competitive positioning, management, economic indicators, and market conditions. The primary goal of FA is to estimate the intrinsic value[1] of an asset and determine whether it is overvalued or undervalued by the market. Investors commonly use FA to make long-term decisions and identify investment opportunities based on an asset's fundamentals. Previous studies have integrated FA indicators with deep neural networks, as demonstrated by [1] and [2], and hybrid models combining linear regression and XGBoost have also been employed for financial forecasting, as seen in [3]. Further examples include works that focus on financial models and trading strategies, such as those by [4] and [5].

Market fluctuations often result from events and decisions influenced by online information, causing stock prices to rise and fall. Consequently, there

[1]The intrinsic value of a stock is an estimate of the stock's true worth, calculated from fundamental factors such as earnings, assets, and growth prospects, independent of its current market price.

has been a growing interest in incorporating sentiment analysis (SA) into financial forecasting and algorithmic trading. SA involves extracting sentiment from online texts, such as articles or comments, and using this information to improve the accuracy of stock predictions and enhance the profitability of trading strategies. Several studies have explored SA's role in this context. For example, [6] applied neural networks, and [7] proposed an event-driven stock model. [8] used support vector machines (SVM) with tree kernels and semantic frame parsing to generalise from sentences to scenarios. Additional research includes [9], which evaluated the impact of news quality on stock movements, and [10], which analysed sentiments in tweets. Furthermore, [11] examined 8-scale emotions. SA has also been used in various studies, including [12], [13], [14], and [15], for financial forecasting and trading. A few studies, such as [16], [17], and [18], have incorporated SA indicators into GP algorithms for algorithmic trading, demonstrating that sentiment analysis can lead to profitable outcomes.

Technical analysis (TA) is another widely used tool in financial forecasting and algorithmic trading, focusing on analysing financial metrics to create indicators that help identify market trends and trading opportunities. TA leverages financial data, such as stock prices and trading volumes, to gain insights into market dynamics and trader behaviour. By studying historical price movements and other technical factors, traders can make informed buy or sell decisions. For instance, [19] incorporated TA indicators with linear models, while [20] used a long short-term memory (LSTM) model to forecast stock prices. Moreover, a hybrid deep learning model integrating TA for financial forecasting was proposed by [21], and [22] showed that GP algo-

rithms could surpass traditional TA methods. Similar results were reported in studies such as [23, 24, 25].

Compared to the studies utilising the algorithms individually, research combining two of the three analysis types has been relatively limited. Some studies, such as [26] and [27], combined fundamental analysis (FA) with technical analysis (TA), finding that this approach increased explanatory power in trading strategies. Similarly, in the Forex market, [28] used fused the two by first processing fundamental data with BERT1, combining them with technical indicators via an autoencoder, and refining using BERT2.Other studies, including [29], examined FA in combination with sentiment analysis (SA), focusing in particular on sentiment extracted from 10-K and 10-Q filings.[2] It is an essential document for investors making informed decisions.

Combining TA and SA indicators has received more attention in the literature. For example, [30] employed a hybrid model combining recurrent and convolutional neural networks, while [31] used reinforcement learning with stock price data and sentiment from news headlines. Additionally, [32] and [33] applied genetic programming (GP) to combine TA and SA, achieving strong results. However, few studies have explored the combination of all three analysis types, with notable exceptions being [34], [35], and [36]. More recent work further supports the value of multimodal integration, as [37] com-

[2] A 10-K filing is a comprehensive annual report submitted by publicly traded companies to the SEC, providing detailed information about their financial performance, operations, and risk factors. A 10-Q filing is the corresponding quarterly report, submitted for each of the three fiscal quarters not covered by the 10-K, and provides an unaudited but otherwise comparable update of financial performance for that quarter.

bined LSTM models with social media sentiment and technical indicators, showing improved predictive accuracy, while [38] used genetic algorithms to select sentiment and technical features for an LSTM forecaster. Additional studies, such as [39] and [40], demonstrate that deep-learning approaches can effectively exploit multiple indicator types to enhance trading performance.

In summary, while fundamental, sentiment, and technical analyses have all been used in algorithmic trading, studies combining these approaches remain limited. Notably, our prior work [32, 33, 41] has shown that combining SA and TA using GP outperforms using either type individually, while [42] and [43], have combined the three analysis types successfully. The former using non strongly typed GP and the latter using indicators from the three analysis types in the Forex market. Similarly, [44, 45] supported multimodal integration in financial prediction. The former demonstrates that fusing heterogeneous data sources (e.g., technical indicators, textual sentiment, and fundamental signals) yields superior accuracy compared to unimodal statistical and deep-learning baselines, while the latter shows that incorporating fundamental, technical, and sentiment variables within a unified neural-network framework improves forecasting performance compared to using any single analysis type alone.

In this paper, we introduce a novel GP algorithm that incorporates fundamental analysis (FA) in addition to SA and TA, aiming to further enhance the profitability and risk management of trading strategies.

3. Methodology

The research methodology is divided into two main sections. Section 3.1 outlines the three types of analysis along with the indicators selected for each. Following this, Section 3.2 details the GP methodology, highlighting the strongly-typed framework of the algorithm, as well as the introduction of a new fitness function and operator.

3.1. Financial analysis processes

In this section, we examine fundamental, sentiment, and technical analysis as distinct processes, with each one discussed in its own subsection.

3.1.1. Fundamental analysis

In our research, we use twelve widely recognised and frequently applied indicators for fundamental analysis (FA) [46], which include: Net Profit Ratio, Return on Equity, Quick Ratio, Debt to Equity Ratio, Price-Earnings Ratio, Price to Book Ratio, Price-Sales Ratio, Total Revenues, Levered Free Cash Flow, Diluted Earnings Per Share (EPS), Earnings Before Interest, Taxes, Depreciation, and Amortization (EBITDA), and Research and Development (R&D) expenses. A complete summary of these indicators can be found in Table 1.

The data for this study was sourced from 10-K and 10-Q filings, accessed via the financial content platform "Seeking Alpha," which links to each company's primary filing documents. Fundamental indicators were extracted directly from these primary filing documents, so that the values used reflect

Table 1 Fundamental Analysis indicators

Fundamental analysis indicators	
Indicators from 10-K/10-Q filings	Net Profit ratio Return on Equity, Quick ratio, Price-Earnings ratio, Price to Book ratio, Price-Sales ratio, Debt to Equity, Total Revenues, Levered Free Cash Flow, Diluted EPS, EBITDA, R&D Expenses

the figures as originally reported by the company. [3] Of the twelve indicators previously mentioned, Diluted EPS, Total Revenues, EBITDA, Levered Free Cash Flow, and R&D Expenses are directly available from the 10-K and 10-Q reports. The remaining seven indicators were calculated using the following equations, where y refers to the financial year, *price* denotes the stock value, and j represents a specific day within the five-year period.

The Net Profit ratio (NetProf) is a profitability metric calculated by dividing the Net Income by the Revenue.

$$\text{NetProf}(y) = \frac{NetIncome_y}{Revenue_y} \tag{1}$$

Return on Equity (RoE) is a profitability metric that measures the return

[3] https://seekingalpha.com. Last accessed: August 2023.

generated on shareholders' investments. It is usually calculated by dividing Net Income by Shareholders' Equity.

$$\text{RoE}(y) = \frac{NetIncome_y}{ShareholderEq_y} \tag{2}$$

The Quick Ratio (QuickR) is a liquidity metric used to evaluate a company's ability to cover its short-term liabilities with its most liquid assets. It is determined by subtracting Inventory from Current Assets and then dividing the result by Current Liabilities.

$$\text{QuickR}(y) = \frac{CurrentAssets_y - Inventory_y}{CurrentLiabilities_y} \tag{3}$$

The Debt to Equity ratio (DebtEq) measures the proportion of a company's assets that are financed through debt versus shareholders' equity. It is calculated by dividing Total Debt by Shareholders' Equity.

$$\text{DebtEq}(y) = \frac{TotalDebt_y}{ShareholderEq_y} \tag{4}$$

The Price-Earnings ratio (P/E) is a valuation metric used to assess whether a company's stock is overvalued or undervalued. It is calculated by dividing the Share Price (p) by the Earnings per Share (EPS).

$$\text{P/E}(j) = \frac{price_j}{EPS_y} \tag{5}$$

The Price to Book ratio (P/B) is a valuation metric used to assess potential investment opportunities. It is calculated by dividing the Share Price (p) by the Book Value per Share (BVPS).

$$\text{P/B}(j) = \frac{price_j}{BVPS_y} \tag{6}$$

The Price-Sales ratio (P/S) is a valuation metric that reflects how much the financial markets value each dollar of a company's revenue. It is calculated by dividing the Share Price (p) by the Revenue per Share.

$$\mathrm{P/S}(j) = \frac{price_j}{Revenue_y} \tag{7}$$

For each company, the accounting variables extracted from a given filing are treated as becoming available from the first trading day after the public release of that filing (based on the filing's SEC filing date) and are held constant until the next filing. This ensures that the GP strategies never use financial statement information that would not yet have been observable on the decision date, while remaining consistent with our daily, non–event-driven design. All FA indicators were normalised between $[-1, 1]$.

3.1.2. Sentiment analysis

Before implementing the sentiment analysis indicators, it was essential to gather the relevant data. In this study, we used the GoogleNews Python library, an interface to Google's News search results, together with newspaper3k for downloading and parsing full article text, to collect articles related to the selected companies. For each company and each day in the study period, we queried the library using the company's name as the keyword, restricted to that date, and retrieved the full paginated output returned by the

library, concatenating results across pages before filtering.[4] These articles were collected for the same time period as the technical analysis indicators. Additionally, we obtained each article's title and summary, which enabled the creation of polarity, subjectivity, and sentiment indicators for both the full text and its title and summary. The choice of using Google Search was driven by their broad coverage and ease of access for multiple companies across time. Alternative data sources were considered during preliminary stages, including Twitter. However, Twitter data was found to be excessively noisy for some companies included in this study, like Apple, particularly due to unrelated mentions and advertisements related to investment tools. Additionally, for several companies in our sample, the volume of Twitter data was insufficient to construct a reliable and consistent sentiment time series. For these reasons, we opted to rely on structured web-based article data, while acknowledging that future research may benefit from combining multiple sentiment sources, such as financial APIs, curated news feeds, or social media, to enhance coverage and robustness.

To reduce noise and ensure the relevance of the sentiment data, we applied strict filtering rules to the collected articles. Specifically, only articles with a minimum length of 500 characters were retained, which helped exclude brief or contentless entries such as headlines without context. In addition,

[4]This retrieval reflects Google's own ranking of News search results for the given keyword and date. We do not apply any further selection by rank or sentiment beyond the length and relevance filters described above (minimum 500 characters, company name and ticker both present in the text). All qualifying articles within the retrieved pages are included in the daily aggregation.

each article was required to contain both the company's full name and its stock ticker symbol, ensuring the content was specifically related to the target company and reducing the risk of irrelevant or misattributed sentiment. To properly align sentiment with market data, we synchronised each article's publication date with stock prices.For each trading day, the sentiment feature is computed as the average sentiment across all calendar days since the previous trading day's close, up to and including the current day. On ordinary weekdays this reduces to the sentiment of that single day. For a trading day immediately following a weekend (typically Monday), the feature aggregates sentiment from the preceding Friday together with the intervening Saturday and Sunday, since all of this information accumulates before the market reopens and only becomes actionable at that next session. This roll forward convention ensures that no sentiment feature is ever constructed from information dated later than the decision it informs, as seen in [47, 48]

This weekend-mapping convention is standard in daily-horizon sentiment studies and avoids assigning information to future dates, thereby preventing look-ahead bias in feature construction.

When multiple articles were published on the same day for a company, we averaged the sentiment values of the relevant articles. For days with no published articles, a neutral sentiment value of zero (0) was assigned, ensuring the continuity of our data.

Next, we present the sentiment analysis indicators. Two widely used indicators are sentiment polarity and subjectivity. Sentiment polarity indicates the overall sentiment direction, categorising the text as positive, negative, or neutral. Subjectivity, meanwhile, measures how much of the text re-

flects personal opinion rather than objective information. In our analysis, we utilise indicators based on these concepts, with different calculation methods explained below.

TextBlob is a Python library offering a simple API for sentiment polarity and subjectivity analysis. For each article, title, and summary, we extracted polarity (ranging from -1 for negative to 1 for positive) and subjectivity (ranging from 0 for completely objective to 1 for completely subjective). *SentiWordNet* 3.0 is a lexical resource built on the *WordNet* taxonomy, designed for sentiment classification and opinion mining. In our implementation, we first POS-tagged the text using the NLTK library and then computed sentiment scores by averaging the weighted positive and negative synsets. *AFINN sentiment*, developed by Finn Årup Nielsen, is a sentiment lexicon that assigns integer polarity scores to over 3,300 English words. All methods were implemented using open-source Python libraries to ensure transparency and reproducibility.

To ensure a diverse representation of sentiment, we employed 12 sentiment analysis (SA) indicators derived from three widely used sentiment analysis tools: TextBlob, SentiWordNet, and AFINN. Each tool was applied to three different textual components associated with each news article: the full article text, the title, and the summary, all retrieved via the Google Search API.

- TextBlob provides two measures: polarity (ranging from negative to positive sentiment) and subjectivity (ranging from objective to subjective content). Applied to the three textual components, this yields 6 indicators (2 metrics $\times$ 3 components).

- SentiWordNet provides a single polarity score, applied to each of the

three components, resulting in 3 indicators.

- AFINN also produces a single polarity score for each component, giving another 3 indicators.

Altogether, this results in 12 sentiment indicators. This design captures both lexical and semantic sentiment features across multiple content layers (title, summary, and full text), allowing the model to reflect sentiment as it may be differently expressed or emphasised in different parts of a document. All SA indicators were normalised between $[-1, 1]$. A summary of these indicators is presented in Table 2.

Table 2 Sentiment Analysis Indicators. TEXT corresponds to a complete article, TITLE to the title of that article, and SUMM to the summary of that article, as provided by the Google Search results.

Analysis type	**Indicator**
Sentiment Analysis (TextBlob)	TEXTpol, TEXTsub
	TITLEpol, TITLEsub
	SUMMpol, SUMMsub
Sentiment Analysis (SentiWordNet)	TEXTsenti, TITLEsenti, SUMMsenti
Sentiment Analysis (AFINN)	TEXTafinn, TITLEafinn, SUMMafinn
	ERC

3.1.3. Technical analysis

In this study, we use six well-established technical analysis (TA) indicators: Moving Average, Momentum, Rate of Change (ROC), Williams %R, Midprice, and Volatility (Equations (8) - (13)). They are calculated using historical data from the selected companies, including the (adjusted) closing prices, as well as the daily highest and lowest prices, sourced from Yahoo! Finance[5]. Each indicator is evaluated using two different lookup windows of $n = 5$ and $n = 10$ days, producing a total of 12 TA indicators, summarised in Table 3. Further details are provided in Section 4.1.

Table 3 Technical Analysis indicators. Each indicator is considered for two different lookup windows (n).

lookup windows $n = 5$ **and** $n = 10$	
Technical analysis indicators	Moving Average
	Momentum
	ROC
	Williams %R
	Volatility
	Midprice

The Moving Average is a technical analysis tool used to smooth stock price data, helping to identify trends by filtering out short-term fluctuations

[5]https://finance.yahoo.com. Last accessed: August 2023

or noise. In this context, p_j denotes the adjusted closing price of the stock on the j-th day.

$$\text{Moving Average}(n, j) = \frac{\sum_{i=j-n}^{j} p_i}{n}, \text{ for } j \geq n. \tag{8}$$

The Momentum indicator measures the difference between the current adjusted closing price and the adjusted closing price from n days earlier, as shown below.

$$\text{Momentum}(n, j) = p_j - p_{j-n}, \tag{9}$$

while the Rate of Change (ROC) normalises the momentum.

$$\text{ROC}(n, j) = \left(\frac{p_j}{p_{j-n}} - 1 \right) \cdot 100 \tag{10}$$

Volatility is a statistical metric that quantifies the variation in returns over a given period. The relevant indicator is calculated as follows.

$$\text{Volatility}(n, j) = \sqrt{\text{Var}\left(\left\{ \frac{p_{j-i}}{p_{j-n}} - 1 \right\}_{i \in \{0,\ldots,n-1\}} \right)}, \tag{11}$$

where Var defines the sample variance over a dataset.

The Williams %R indicator, defined in Equation (12), compares the most recent closing price cl_j (on day j) to the highest price $hh_{n,j}$ recorded within the lookup window ending on day j, while also taking into account the lowest price $ll_{n,j}$ observed within the same period.

$$\text{Williams \%R}(n, j) = -100 \cdot \frac{hh_{n,j} - cl_j}{hh_{n,j} - ll_{n,j}} \tag{12}$$

The *Midprice*, as defined in Equation (13), represents the midpoint between the highest price, $hh_{n,j}$, and the lowest price, $ll_{n,j}$, over all days within the lookup window ending on day j.

$$\text{Midprice}(n, j) = \frac{hh_{n,j} - ll_{n,j}}{2} \tag{13}$$

All TA indicators were normalised between $[-1, 1]$.

3.2. Proposed ActTrade GP algorithm

GP algorithms have been very popular algorithms for many financial tasks, including algorithmic trading [49]. In this work, we propose a novel strongly-typed Genetic Programming algorithm that has three main novel components: (i) a strongly-typed nature, (ii) a novel fitness function, and (iii) a novel genetic operator.

For the rest of this section, we will first present the model representation and the strongly-typed nature of the algorithm (Section 3.2.1), and then explain how trading takes place (Section 3.2.2). Then, in Section 3.2.3 we will discuss the novel fitness function of the proposed GP algorithm, as well as the metrics used for the evaluation of the GP trees. Next, Section 3.2.4 will present the GP operators used in our experiments, along with our novel operator that encourages active trading. Lastly, Section 3.2.5 will discuss the selection process of individuals to go through the GP operators.

3.2.1. Model representation

The function nodes are based on a 3-input `AND` function with a boolean output (we call it `3-AND`), as well as binary `AND`, `OR`, Greater than (`GT`) and Less than (`LT`) functions, with different variants allowing for each indicator type. In particular, our algorithm uses $\texttt{AND}_{\texttt{FA}}$, $\texttt{OR}_{\texttt{FA}}$, $\texttt{GT}_{\texttt{FA}}$, $\texttt{LT}_{\texttt{FA}}$ function nodes in the FA branch, $\texttt{AND}_{\texttt{SA}}$, $\texttt{OR}_{\texttt{SA}}$, $\texttt{GT}_{\texttt{SA}}$, $\texttt{LT}_{\texttt{SA}}$ function nodes in the SA branch and it uses $\texttt{AND}_{\texttt{TA}}$, $\texttt{OR}_{\texttt{TA}}$, $\texttt{GT}_{\texttt{TA}}$, $\texttt{LT}_{\texttt{TA}}$ function nodes in the TA branch.

The `3-AND` function takes 3 children. The first is of type FA, the second of type SA, and the third of type TA. Thus, the first branch can take any of the

relevant FA-type functions, i.e. $\texttt{AND}_{\texttt{FA}}$, $\texttt{OR}_{\texttt{FA}}$, $\texttt{GT}_{\texttt{FA}}$, or $\texttt{LT}_{\texttt{FA}}$. The same principle applies to the second (SA) and third (TA) branches. All three branches of the `3-AND` function need to evaluate to TRUE in order for the function to yield a TRUE outcome.

The function set used in our algorithm is summarised in Table 4. This carefully designed function set ensures that the algorithm generates models that fully utilise all indicator types (FA, SA and TA), while at the same time enforcing type consistency. In doing so, we can prevent type errors and enhance the exploration of the search space.

Table 4 Function set for the GP algorithm.

Explanation	**Function nodes**
Root node	`3-AND`
FA, SA and TA type for `AND`	$\texttt{AND}_{\texttt{FA}}$, $\texttt{AND}_{\texttt{SA}}$, $\texttt{AND}_{\texttt{TA}}$
FA, SA and TA type for `OR`	$\texttt{OR}_{\texttt{FA}}$, $\texttt{OR}_{\texttt{SA}}$, $\texttt{OR}_{\texttt{TA}}$
FA, SA and TA type for Greater Than	$\texttt{GT}_{\texttt{FA}}$, $\texttt{GT}_{\texttt{SA}}$, $\texttt{GT}_{\texttt{TA}}$
FA, SA and TA type for Less Than	$\texttt{LT}_{\texttt{FA}}$, $\texttt{LT}_{\texttt{SA}}$, $\texttt{LT}_{\texttt{TA}}$

Regarding the terminal sets, different sets are allowed at different branches of the corresponding trees. In the FA branch (Table 1), the terminal set consists of specific indicators that are permitted to be used within that branch. Similarly, the SA branch (Table 2) and the TA branch (Table 3) have their own terminal sets of allowed indicators. Additionally, all terminal sets in-

clude an Ephemeral Random Constant (ERC) variable, which serves as a threshold value for the indicators and consists of random values between -1 and 1.

Compared to a non-strongly typed GP, our proposed algorithm has the advantage of fully utilising the search space of each individual indicator type. The three branches corresponding to the FA, SA and TA indicators are connected using the `3-AND` function at the root of the tree. This integration creates a foundation for better exploration and exploitation of the search space. As a result, the model can generate more diverse, effective, and adaptable trading strategies.

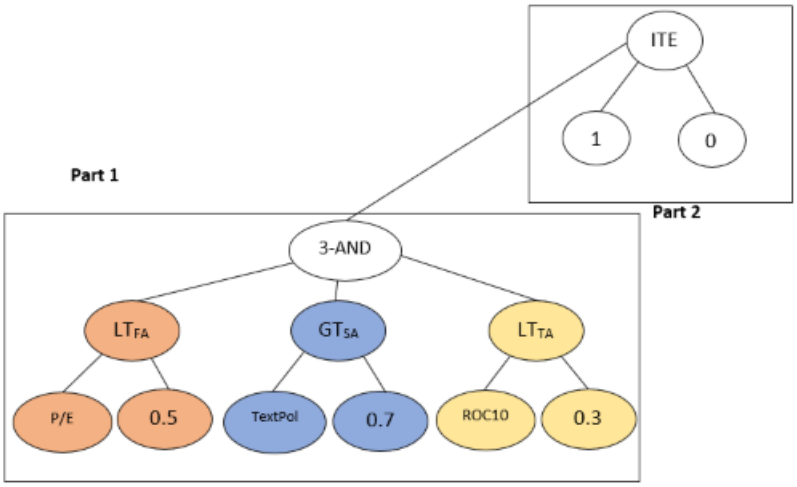
3.2.2. Trading signals and trading strategy

Trading signals are generated by the GP trees, where the root `3-AND` function, which outputs a TRUE/FALSE value, is used to decide whether to issue a buy recommendation or hold the stock. This decision is implemented using an If-Then-Else (ITE) node at the root of another tree structure. The second branch represents the buy action (1), while the third branch signifies the hold action (0), as shown in the example GP tree in Figure 1. It is important to note that only Part 1 of Figure 1 undergoes evolution via GP operations.

In Part 1 of Figure 1, the evolved tree begins with the root node `3-AND`. The fundamental analysis (FA) branch starts with the $\texttt{LT}_{\texttt{FA}}$ node (depicted by orange-coloured nodes), the sentiment analysis (SA) branch begins with the $\texttt{GT}_{\texttt{SA}}$ node (also in orange), and the technical analysis (TA) branch is represented by the $\texttt{LT}_{\texttt{TA}}$ node (shown in yellow).

If the GP tree (Part 1) returns a TRUE value, it activates the second

Figure 1 GP sample tree. The first child of the 3-AND function is enforced to be FA-related, the second child is SA-related and the third is enforced to be TA-related. This sample tree checks if the P/E indicator is less than the ERC of 0.5, and if the TEXTpol indicator is greater than the ERC 0.7, and if the ROC10 indicator is less than the ERC 0.3. If all three of them are TRUE, the recommendation will be to buy (1), otherwise it will be to hold (0).



branch (1) of the ITE tree, signaling a buy action. In this scenario, the GP algorithm purchases the stock. To determine when to sell, the algorithm follows this rule: given a predefined increase rate r and a time period of d days, if the stock's return within the next d days exceeds r, the stock is sold on that day. If the return does not surpass r within the d-day window, the stock is sold at the end of that period. For instance, assuming $d = 30$ and $r = 0.05$, the rule becomes: "If the stock price increases by more than 5% of the purchase price within the next 30 days, sell the stock on that day. Otherwise, sell it at the end of the 30-day period." On the other hand, if the

ITE tree signal is 0, indicating a hold decision, no trades are executed.

The parameters d and r are optimised during the validation phase and remain constant across all GP algorithms considered, though they may differ for different companies (see Section 4.3).

3.2.3. Fitness function and Metrics

Our analysis evaluates the metrics of return, risk, and the Sharpe ratio, which are defined as follows.

The *return*, R, of a trade reflects the profit made as a percentage of the invested amount. The profit calculation incorporates transaction costs of 0.025% of the selling price (c_t). Specifically, return is computed as detailed in Equation (14), where V_f represents the final value, or the price at which the stock was sold, and V_i represents the initial value, or the price at which the stock was purchased

$$R = \frac{(1 - c_t)V_f - V_i}{V_i} \tag{14}$$

This simplified, uniform transaction cost model is commonly used in daily-frequency trading studies to ensure comparability across strategies and to avoid introducing additional assumptions about intraday liquidity. It captures the principal cost of executing a trade, while keeping the focus on the relative performance differences between the proposed GP approach and the ML baselines. Future work could include more sophisticated costs, such as round-trip costs (buy and sell), bid-ask spreads, and slippage effects.

The *rate of return* (RoR) is the average return of all trades over a given period. It is calculated by averaging the individual returns from each trade.

Risk is quantified as the standard deviation of the returns, expressed as $\sqrt{var[R]}$.

The *Sharpe ratio*, SR, is defined as the ratio of the expected excess return, relative to the risk-free rate, R_f, to the level of risk. Formally,

$$SR = \frac{\mathrm{E}[R - R_f]}{\sqrt{var[R]}}, \tag{15}$$

where R_f represents the risk-free return, set at a value of 0.022%, an average figure reported by Fernandez et al. [50] and referenced by Long et al. [51].

The proposed *fitness function*, denoted as f_{sum}, evaluates both global and local performance within the GP tree structure. It is defined as the weighted sum of four components: the Sharpe ratio (SR^C) of the entire tree, which integrates FA, SA, and TA indicators, the Sharpe ratio (SR^{FA}) of the subtree containing only FA indicators, the Sharpe ratio (SR^{SA}) of the subtree with SA indicators, and the Sharpe ratio (SR^{TA}) of the subtree focusing on TA indicators. Each of these components is weighted by wc, wFA, wSA, and wTA, respectively, to balance the influence of the global and the branch performances. Formally,

$$f_{sum} = w_c \cdot SR^C + w_{FA} \cdot SR^{FA} + w_{SA} \cdot SR^{SA} + w_{TA} \cdot SR^{TA} \tag{16}$$

For instance, in Figure 1, SR^C represents the subtree with the root node `3-AND` (i.e., Part 1), while SR^{FA} refers to the subtree with the root node $\texttt{LT}_{\texttt{FA}}$ (represented by orange-colored nodes), SR^{SA} corresponds to the subtree rooted at $\texttt{GT}_{\texttt{SA}}$ (blue-colored nodes), and SR^{TA} refers to the subtree with the root node $\texttt{LT}_{\texttt{TA}}$ (yellow-colored nodes).

The benefit of this fitness function is that it enables the evolution of trees that optimise all four Sharpe ratios, thereby allowing the GP to guide the search toward trading strategies with strong performance across all components. This is crucial because, if the fitness function were based solely on the Sharpe ratio of the entire tree (as is common in the literature), the GP could identify well-performing trees, but it might not fully leverage the strongly-typed structure that ensures the inclusion of FA, SA, and TA indicators.[6][7]

3.2.4. GP operators

This section is divided into two parts. First, we introduce what we refer to as "traditional GP operators", including elitism, subtree crossover, and point mutation. Then, we present our proposed operator designed to promote

[6] In fact, early experiments we conducted confirmed this issue: when the fitness function was limited to the Sharpe ratio of the entire tree, the best-performing trees often had a large, well-performing subtree focused on just one analysis type, such as SA, while the FA and TA subtrees were small and underperforming. Moreover, the overall tree's performance did not surpass that of a non-strongly typed GP that allowed FA, SA, and TA indicators to be mixed, rendering the strongly-typed approach unnecessary. Our proposed fitness function resolves this issue.

[7] It is also worth noting that during the development of ActTrade, we experimented with incorporating parsimony pressure in the GP fitness function (specifically, lexicographic parsimony pressure to penalise larger trees). However, we observed that imposing a strong size penalty often led to a deterioration in trading performance: the GP would converge prematurely to overly simplistic models that under-utilized the rich set of indicators. As a result, we opted not to include an explicit parsimony term in the final runs. Instead, we controlled tree complexity by enforcing a maximum tree depth of 6 in our experiments (this constraint ensures trees do not grow unreasonably deep). More detail about the maximum tree depth and the other GP parameter values can be found in Section 4.

active trading.

Traditional GP operators. In elitism, the individual with the highest fitness value in each generation is directly carried over to the next generation. For subtree crossover, we apply it independently to each analysis type branch. It is crucial to ensure that any operation is valid, meaning that it adheres to the three distinct types (FA, SA, TA) within the GP trees. Consequently, exchanged nodes must match in both type (e.g., terminal node with terminal node) and data type (e.g., `FA` branch with `FA` branch) to preserve the integrity of the tree. To ensure the validity of this exchange, we first swap the `FA` branches between the two selected parents, followed by the `SA` branches, and finally, the `TA` branches.

In a strongly-typed setting, point mutation also comes with specific constraints. For instance, a function node such as $\texttt{OR}_{\texttt{SA}}$ can only be substituted with $\texttt{AND}_{\texttt{SA}}$, and a function node like $\texttt{GT}_{\texttt{TA}}$ can only be replaced by $\texttt{LT}_{\texttt{TA}}$ (and similarly for other function nodes). Additionally, an ERC can only be replaced with another ERC, and a terminal variable must be swapped with another variable of the same indicator type. The algorithm ensures that mutated nodes are replaced with valid data types. Point mutation occurs in one of the three branches of the tree.

The parents selected for these operations are chosen through tournament selection. An individual may undergo crossover with a probability of p and mutation with a probability of $1 - p$.

A novel GP operator for active trading. As outlined in Section 1, one of our key objectives is to develop an algorithm that implements highly active trading strategies, which can lead to greater profit potential. To achieve

this, we concentrated on producing offspring that aim to maximise both the number of trades and the Sharpe ratio.

To accomplish this, we follow the process outlined below: for each generation, we iterate through the population and separate the three subtrees (FA, SA, TA) of each individual into three respective lists—one for each analysis type. We then calculate the number of trades and the Sharpe ratio for each subtree. After processing all trees in the current generation, we identify the FA subtree with the highest number of trades and the one with the highest Sharpe ratio; the same steps are repeated for the SA and TA subtrees. Additionally, to encourage simpler, more interpretable trees, if subtrees have the same number of trades, we select the smaller one.

Next, we generate three offspring by combining subtrees that optimise both the Sharpe ratio and the number of trades. Specifically, the first offspring is formed by combining the FA subtree with the highest Sharpe ratio, the SA subtree with the most trades, and the TA subtree with the most trades, placing them under the `3-AND` function. This structure is then carried over into the next generation. Similarly, the second offspring combines the SA subtree with the highest Sharpe ratio and the FA and TA subtrees with the most trades. Finally, the third offspring consists of the TA subtree with the highest Sharpe ratio and the FA and SA subtrees with the most trades.

It is important to note that this operator is applied at every generation during training, meaning these three offspring are copied into the next generation before crossover and mutation take place. This process is outlined in Algorithm 1.

By using this approach, the GP operator instills two key characteristics in the population's individuals: a high number of trades and a high Sharpe ratio.[8] This strategy enhances the quality of the evolving trees in subsequent generations, favouring individuals that exhibit both traits to achieve more effective and profitable trading outcomes.

[8]During early experimentation, we explored alternative strategies, such as constructing trees where all three subtrees were optimised either for maximising the number of trades or the Sharpe ratio. However, in those scenarios, only a small number of cases evaluated to TRUE, leading to very few trading actions due to the root function being `3-AND`. This restrictive condition made the GP overly passive, often resulting in no trades. Other alternatives we considered were (i) using an `OR` function as the root, which didn't ensure simultaneous consideration of all three indicator types during trading, and (ii) employing a non-strongly-typed GP approach. Neither of these alternatives produced strong financial results. Further investigation of these alternatives has been left for future work.

Algorithm 1 Active trading genetic operator procedure

Require: Population of individuals, Analysis types: FA, SA, TA

Initialise empty lists for FA subtrees, SA subtrees, and TA subtrees.

for each individual in the population **do**
    Extract FA, SA, and TA subtrees from the individual.
    Append these subtrees to the respective lists for the analysis type.
end for

for each subtree in the corresponding analysis type (FA, SA, TA) list **do**
    Calculate number of trades and Sharpe ratio.
end for

for each list (FA subtrees, SA subtrees, TA subtrees) **do**
    Identify subtree with highest number of trades (`max_trades_subtree`) and subtree with highest Sharpe ratio (`max_sharpe_subtree`).
end for

Create three offspring (named Offspring1, Offspring2, Offspring3):
for each offspring **do**
    Select corresponding subtrees:
    if *Offspring1* **then**
        FA subtree: `max_sharpe_subtree` from FA subtrees
        SA subtree: `max_trades_subtree` from SA subtrees
        TA subtree: `max_trades_subtree` from TA subtrees
    else if *Offspring2* **then**
        FA subtree: `max_trades_subtree` from FA subtrees
        SA subtree: `max_sharpe_subtree` from SA subtrees
        TA subtree: `max_trades_subtree` from TA subtrees
    else
        # Offspring3
        FA subtree: `max_trades_subtree` from FA subtrees
        SA subtree: `max_trades_subtree` from SA subtrees
        TA subtree: `max_sharpe_subtree` from TA subtrees
    end if

    Combine the three subtrees (FA, SA, TA) under the ternary `3-AND` function and enter them into the population of the next generation.
end for

3.2.5. Selection

In the previous section, we explained how our proposed genetic operator creates three offspring, which are then placed into the population of the next generation. We call them *Offspring1, Offspring2,* and *Offspring3*. As these three individuals might not always make large impact in the search, we decided to also probabilistically allow them to act as parents during crossover and mutation. In this way, we allow their genetic material to be copied into more trees in the population.

To achieve the above, we assign four probabilities p_1, p_2, p_3, and p_4,[9] to select the first parent for crossover, and the single parent for mutation. More specifically, with probability p_1, we use tournament selection, where k individuals (trees) are randomly selected from the population and then the individual with the highest fitness value in the tournament is selected as a parent to undergo crossover/mutation. In addition, we assign a probability p_2, p_3, and p_4, to each one of Offspring1, Offspring2, and Offspring3, respectively. In this way the proposed genetic operator can also affect the creation of new offspring by carrying part of its genetic material to the new offspring.

For the second parent, we always use tournament selection i.e. $p_1 = 1$. Algorithm 2 summarises the selection process.

To conclude, the proposed GP algorithm is a strongly typed GP that aims to maximise the Sharpe ratio. The fitness function is determined by the weighted sum of the FA, SA, TA subtree, and the complete tree comprising of the three subtrees. The strongly typed structure of the GP enables more

[9]The sum of $p_1 + p_2 + p_3 + p_4$ is always equal to 1.

Algorithm 2 Parents selection procedure

Require: Probabilities p_1, p_2, p_3, p_4, Offspring1, Offspring2, Offspring3

if Selecting first parent for crossover or parent for mutation **then**
    With probability p_1: Perform tournament selection to select parent
    With probability p_2: Select Offspring1 as parent
    With probability p_3: Select Offspring2 as parent
    With probability p_4: Select Offspring3 as parent
else
    Perform tournament selection to select second parent for crossover
end if

effective exploration and utilisation of all three indicator types, while the fitness function emphasises the contribution of each indicator in enhancing the trading performance of the algorithm. Lastly, the proposed operator allows for offspring to be created that focus both on maximising the Sharpe ratio, as well as maximising the number of trades. The relevant Python code for these experiments is available on GitHub.[10]

4. Experimental Setup

4.1. Data

In this study, we gathered data from 42 international companies, from different stock exchanges. The research period covered five years, from January 1st, 2015, to January 31st, 2020, intentionally excluding the COVID-19

[10]https://github.com/avechris/ActTrade

pandemic period. This exclusion was necessary to maintain consistency in the training and validation sets, ensuring reliable parameter tuning.

The fundamental analysis data was sourced from the 10-K and 10-Q filings available on the financial content platform "Seeking Alpha". For sentiment analysis, we employed the GoogleNews Python library, together with newspaper3k, to collect articles, titles, and summaries. The technical analysis was performed using daily closing price data obtained from Yahoo! Finance. After gathering all the data, we generated the 12 fundamental analysis (FA) indicators, 12 sentiment analysis (SA) indicators, and 12 technical analysis (TA) indicators as outlined in Section 3.1. It is important to note that our study focuses exclusively on individual company stocks rather than index-level assets such as the VIX or NASDAQ$\tilde{1}$00, so we can isolate and evaluate company specific signals derived from fundamental, sentiment, and technical indicators. We therefore restrict our evaluation to 42 international companies across diverse geographies and industries, which provides heterogeneity while keeping the analysis aligned with our objective of measuring the benefits of integrating FA, SA, and TA at the stock level. The dataset was split chronologically into training, validation, and testing sets, ensuring that all training data precedes validation and all validation data precedes testing. We split the datasets sequentially into three portions: 60% for training, 20% for validation, and 20% for testing. This design prevents future information leakage across phases, which is a critical issue in financial time series. It provides a clear measure of its ability to generalise to new, out-of-sample market data. More sophisticated validation schemes, such as walk-forward or rolling-window evaluation, are not as feasible with daily data over a five-

year horizon per stock (roughly 1,200–1,300 trading days), as GP requires reasonably long contiguous periods for evolution to converge.

It is important to note that our study is not a high-frequency trading (HFT) framework. The algorithm is designed for daily decision making, reflecting the resolution of the available data, where sentiment indicators are aggregated daily and fundamental indicators are released quarterly. While technical indicators can be computed intraday, excluding FA and SA at those horizons would undermine the purpose of integrating the three analysis types. Hence, our contributions are positioned within the context of daily trading strategies, consistent with prior work in quantitative finance, as in [52] and [53].

Lastly, it is worth noting that the datasets used in this work have varied daily trading volume and return volatility. When looking at the one year test set, we find a range of market conditions, where companies like Apple and Tesla have very high trading volumes (above 10^8 shares on average), Amazon and Walmart (above 10^7 shares on average), while firms such as Suzuki, Fujitsu, and Kodak operate at volumes three to four orders of magnitude smaller, evidencing low volume days. In parallel, daily return volatility varies substantially across companies, with technology and automotive stocks in particular showing elevated volatility levels compared to common consumer companies. More specifically, we computed the standard deviation of daily prices across the test set, which is a year. The results show companies in all volatility regimes, for example Ford ($\sigma = 0.54$) and IBM ($\sigma = 4.05$) in the low-volatility group (¡10), Apple ($\sigma = 10.3$) and Adidas ($\sigma = 30.7$) in the moderate group (10–50), Amazon ($\sigma = 93$) in the high-volatility group

(50–100), and extremely volatile stocks such as Suzuki (σ = 458.3). This demonstrates that the one year test period includes both stable and highly volatile markets, thereby ensuring that the reported annualised performance metrics inherently account for different volatility regimes.

These descriptive statistics confirm that our evaluation period indeed spans both low volume and high volatility conditions, and that the reported performance metrics already integrate the model's behaviour in such condition, as we will highlight in more detail in Section 5.

4.2. Benchmarks

We refer to our proposed GP algorithm as ActTrade, named for its active trading operator. We compare its performance against six GP variants, four machine learning algorithms, and one financial-based trading strategy. These benchmarks are introduced in the following section.

- **GP-FA** is a GP algorithm that exclusively uses fundamental analysis indicators in its terminal set. The purpose of this benchmark is to assess the trading performance of a non-strongly-typed GP relying solely on fundamental analysis indicators.

- **GP-SA** is a GP algorithm that includes only sentiment analysis indicators in its terminal set. Similar to GP-FA, this benchmark is designed to evaluate the performance of a GP algorithm that operates exclusively with sentiment analysis indicators.

- **GP-TA** is a GP algorithm that solely utilises technical analysis indicators in its terminal set. The objective, as with GP-FA and GP-SA,

is to examine the trading performance of a GP that operates only with technical analysis indicators.

- **GP-FASATA** is a (non-strongly-typed) GP algorithm that combines fundamental, sentiment, and technical analysis indicators. The goal of this benchmark is to assess how well the GP performs when it integrates all three types of indicators, without any typing constraints. Since there are no defined branches, the GP is free to select any combination of indicators, regardless of whether they come from fundamental, sentiment, or technical analysis.

- **STGP-FASATA** is a strongly-typed GP algorithm that incorporates fundamental, sentiment, and technical analysis indicators. This algorithm does not use the novel fitness function or the active trading operator introduced in ActTrade, making it a valuable benchmark to understand the impact of these innovations on the GP's performance.

- **STGP-FASATA-S** is similar to STGP-FASATA but includes the same fitness function used in ActTrade (the summation of Sharpe ratio fitness functions). The only distinction between STGP-FASATA-S and ActTrade is the absence of the active trading operator, serving as a benchmark to highlight the added value of the active trading mechanism.

Furthermore, the study also included two additional algorithmic benchmarks, namely:

- **Multilayer perceptron (MLP)**

- **Support vector machine (SVM)**
- **eXtreme Gradient Boosting (XGBoost)**
- **Long short-term memory (LSTM)**

All the algorithms mentioned above were used to tackle a binary classification problem: predicting whether a stock's price will rise by a certain percentage ($r\%$) within the following d days. A Class 1 prediction signals a buy action, while a Class 0 prediction indicates a hold action. As discussed in Section 3.2.2, the sell action is executed as part of the trading strategy. and the same trading rules, transaction-cost assumptions, and (d,r) parameters are applied to both GP-based and ML-based methods. In this sense, the ML algorithms serve as predictive reference models whose signals are fed into the identical downstream trading protocol as ActTrade, rather than as an alternative GP strategy with evolution mechanisms. Our aim is therefore to compare the quality of the induced trading performance under a common evaluation pipeline, not to claim that the GP and ML benchmarks operate in exactly the same way.

Furthermore, the proposed ActTrade is evaluated against two financial benchmarks:

- **Buy and Hold (BnH)**: The Buy and Hold strategy is a widely used investment approach and serves as a financial benchmark. It relies on historical price data, where one unit of stock is purchased at the start of the trading period and sold at the end. The strategy is based on the idea that, over time, the value of investments generally increases.

- **Trading-Strategy$_{d,r}$ ($\mathbf{TS}_{d,r}$)**: In this strategy, we purchase stock at the start of each trading period and sell it when the price increases by more than a specified rate r, or after d days have elapsed.

The $\mathrm{TS}_{d,r}$ benchmark serves as a baseline trading strategy without a learning component, as it follows the same trading rules employed by the GP algorithms. This allows us to isolate and assess the added value of the GP algorithm, independent of the underlying trading strategy.

4.3. Parameter Tuning

To mitigate the risk of overfitting and ensure the robustness of our results, we employed a structured experimental design. Initially, we split the data into three subsets: training, validation, and testing. The validation set was utilised exclusively to tune and select appropriate parameters (such as GP population size, crossover/mutation rates, and tree depth). After identifying optimal parameters via the validation set, we combined the training and validation sets to retrain the final GP model, maximising the available data for model learning. Importantly, model performance was assessed on a test set that remained entirely unseen during training and validation, providing a clear measure of its ability to generalise to new, out-of-sample market data.

Table 5 provides the GP parameters used in the experiments conducted for this study. A two-step grid search was performed on the validation set to identify the optimal GP parameters. In the first step, key parameters such as population size, crossover probability (p_c)[11], number of generations,

[11]Since the mutation probability equals $1 - p_c$, it was excluded from the parameter tuning process.

tournament size, and maximum tree depth were tuned. Additionally, Table 5 lists the probabilities p_1, p_2, p_3, and p_4 for selecting the first parent during crossover or mutation (p_1: tournament selection; p_2, p_3, and p_4 are for the three offspring generated by the active trading operator). The trading parameters d and r were fixed at 30 and 0.05, respectively, to expedite the tuning process and ensure a fair comparison by identifying parameter combinations that performed well across all GP variants.

Table 5 GP Parameters for the eight GP algorithms.

GP Parameters	
Population size	1000
Crossover probability	0.95
Mutation probability	0.05
Generations	50
Tournament size	4
Maximum tree depth	6
Tournament probability (p_1)	0.4
Active trading offspring probability (p_2, p_3, p_4)	(0.2, 0.2, 0.2)

In the second phase of parameter tuning, we concentrated on optimising the trading strategy parameters, d and r. This tuning was conducted individually for each company across the algorithms, enabling improved trading performance by tailoring the parameters to the specific characteristics of each company. The optimal values for the two trading parameters were

determined by selecting the combination that produced the best overall performance across all GP algorithms on the validation set.

The parameter tuning for MLP, SVM, XGBoost, and LSTM was performed separately using a binary classification approach, where one class represents a certain percentage increase in the stock price for the following day, and the other class represents a different price change (see Section 4.2). The goal was to identify the optimal parameter settings that maximise the models' predictive accuracy on the validation set. After tuning, the model with the best predictive performance on the validation set was selected. The predicted class from this model was then used as a trading signal for the testing dataset, which was applied to the trading strategy. The trading parameters, d (days) and r (percentage increase), were set to the same values used in the GP variants to ensure consistency in evaluating the performance of the different algorithms.

After the parameter-tuning stage, we run 50 independent GP runs for each company, using the fixed parameter values identified during tuning and are trained only on the training and validation sets. After evolution, all strategies are evaluated on the unseen test set, and all performance statistics reported in Section 5 are computed across these these 50 evaluations. This procedure ensures that the reported results do not rely on a single run and that model selection remains robust.

5. Results and Analysis

In this section, we present the results of our experiments, comparing the proposed ActTrade algorithm with the benchmarks outlined in Section 4.2.

It is important to note that our analysis is conducted per company, since strategies are evolved and evaluated individually for each stock rather than at the portfolio level.For each GP algorithm, we conducted 50 independent runs across the 42 companies. Each run generated a unique trading strategy, which was subsequently evaluated on the test set.

All experiments were conducted on the CERES cluster of the University of Essex. CERES is a computational cluster built using the Rocks Clustering Solution with CentOS Linux. For computational purposes, the cluster has 1632 processing cores (3232 with hyperthreading) provided by servers with a mix of Intel E5-2698, Intel Gold 5115, 6152 and 6238L processors, and between 500Gb and 6Tb RAM each.

To assess the statistical significance of our findings, we applied the two-sample Kolmogorov-Smirnov (KS) test to the performance results of all runs across companies for each algorithm. This non-parametric test evaluates whether the distributions of performance metrics differ significantly between ActTrade and each benchmark.

Given the multiple comparisons involved, (algorithms $\times$ metrics), we control the false discovery rate (FDR) using the Benjamini–Hochberg (BH) procedure with $q = 0.05$. This choice balances the Type I error control with statistical power across a large set of related tests.

5.1. Sharpe ratio

Table 6 shows the average Sharpe ratio values across 50 runs for the 42 companies. A value of zero (0) for a company indicates that the respective algorithm chose not to trade, likely to avoid potential losses. As observed, GP-SA has the highest occurrence of best Sharpe ratio values, leading in 11

companies, followed by STGP-FASATA with 8 companies, GP-TA with 7, and ActTrade with 6. GP-FA achieves the highest values in 5 companies, while STGP-FASATA-S and GP-FASATA lead in 4 and 2 companies, respectively. It is also worth noting that both STGP-FASATA-S and ActTrade display several 0 Sharpe ratio values, suggesting that these algorithms are highly conservative. As mentioned earlier, this conservatism allows them to avoid trades in unfavourable situations, potentially improving their average and median performance, as will be discussed next.

Table 7 presents summary statistics for the 50 GP runs conducted for each company across the seven GP algorithms. Runs with a Sharpe ratio of 0, where no trades were executed, were excluded from the calculation of these statistics. When interpreting the Sharpe ratio results, it is important to remember that this metric can be sensitive to the number of trades. It can yield disproportionately large values (positive or negative) when only a few trades are made. This occurs because the risk, which is computed as the standard deviation of returns over a given period, can become extremely small if an algorithm performs only a few trades (e.g., 2-4 trades over the entire test set). Since risk is the denominator in the Sharpe ratio calculation, a very small risk value can result in an unusually high Sharpe ratio.

As we can observe from Table 7, the proposed ActTrade algorithm has the highest mean and median values, indicating a strong risk-adjusted performance compared to the other algorithms and highlighting its robustness and consistency in delivering favourable risk-adjusted returns. Generally, all strongly-typed GP algorithms experience higher mean values, which indicates the advantage of the architecture in comparison to non strongly-typed GPs.

Table 6 Average values for Sharpe ratio per company. Boldface is used to denote the best value for the particular dataset.

Company	*GP-FA*	*GP-SA*	*GP-TA*	*GP-FASATA*	*STGP-FASATA*	*STGP-FASATA-S*	*ActTrade*
AAPL	5.5915	2.9952	6.2435	6.0608	2.6744	**8.4403**	7.2507
ADIDAS	0.0455	-0.8810	0.1087	0.1709	**0.6694**	0	0
ALIBABA	1.3241	**12.8724**	2.7733	4.4762	2.7338	2.3999	11.8146
AMAZON	1.1791	1.7928	**14.7694**	5.5078	3.7653	0	0
ASUS	-1.3757	3.4320	0.5216	0.8423	-0.2138	-0.5439	**6.3323**
ATVI	0.5347	2.5614	**3.5067**	0.7391	3.0072	0.3750	2.6987
BLACKB	-1.2496	-0.0329	**2.8176**	1.3741	1.6736	1.0647	1.0846
COKE	**8.7271**	-0.5819	0.6859	1.2089	1.8917	2.6953	1.1766
EBAY	0.6157	**6.3670**	0	-0.5972	-0.0702	0.4302	0.1362
ESTEE	0.2138	2.0982	1.2340	0.7858	1.8651	**8.7272**	1.6824
FORD	-0.1341	**8.3010**	-0.5957	0.1256	5.2550	0	3.2462
FUJIFILM	**1.0442**	0.5517	-1.5904	-0.5409	1.0232	-1.7277	0.1871
FUJITSU	0.7533	3.7077	13.7997	15.5743	9.9302	**24.6731**	15.8708
GM	1.0112	**4.0144**	-20.9106	-0.2385	-0.6598	0	0
GOOGL	0.0520	2.4776	2.3801	3.7570	1.1243	-0.1268	**14.1991**
HITACHI	-0.1452	**0.2454**	0	0.0971	-1.2785	0	0
HONDA	1.2830	**6.6325**	2.5088	1.6852	2.0304	2.4846	0
HSBC	2.8015	**2.9204**	0.1729	0.0370	1.9430	0	0.0372
HYUNDAI	2.4618	0.7194	-0.6452	**31.4885**	10.8667	1.1241	2.3572
IBM	0.4644	0.9346	**10.5784**	1.8172	6.8871	3.2115	1.7280
INTC	0.8131	1.9789	1.4645	0.9523	**2.2122**	0	-0.0318
KERING	0.5836	0.7536	0.9268	0.7333	**1.1841**	0.3960	0
KODAK	1.1804	1.4972	**1.6641**	0.4966	1.1869	1.0455	1.2845
MCDON	-0.0502	**0.9143**	-4.3046	0	-0.0126	0	-1.2986
NESTLE	-0.0649	2.3210	-0.0132	2.6653	**9.8132**	0.1730	5.5454
NFLX	8.0428	**17.8119**	0.9170	6.6686	3.1125	5.0024	9.3962
NINTENDO	0.1078	1.1302	-0.1277	0.4118	**3.8835**	0.7198	0.6481
NYT	0.3141	**3.2256**	-0.7486	0.1152	-0.0254	-2.3197	0.1232
PANASONIC	0.1002	2.5060	3.9634	2.3647	1.3352	3.6128	**7.4402**
SAINSBURY	14.9238	2.2750	2.0274	5.3955	1.0712	**15.2586**	2.3479
SHISEIDO	-0.6942	**4.9143**	0.5602	-2.4178	-1.0828	1.8925	-0.6254
SUBARU	0.6109	-0.1824	**7.7342**	0	-0.5199	0.0901	2.0504
SUZUKI	5.5550	-1.0981	2.6771	0.6488	0	6.3275	**9.1027**
TENCENT	0.3319	0	-0.5544	**15.8820**	4.7378	0.3920	3.7368
TESCO	0.1186	0.9316	2.2420	-0.2571	**15.6728**	0	-14.5138
TESLA	0.7089	**3.5859**	2.1825	2.0695	0.9962	1.6463	2.7183
TOYOTA	**3.0496**	1.1699	0	-0.3441	3.0060	0	0
UBISOFT	1.0275	0	1.5272	4.0745	**5.7834**	-0.4325	0
WALMART	**2.8774**	1.1366	1.9536	0.2454	1.5732	0.1634	1.5442
XEROX	**6.4406**	0.1473	-0.8290	2.3401	1.4751	0	0.3188
YAMAHACO	0.5634	1.3774	0.4464	0.7569	1.7156	2.1979	**2.2089**
YAMAHAMO	1.6966	0	0.6509	0	**6.6631**	-0.1101	0.3586

Table 7 Summary statistics of Sharpe ratio across all 50 GP runs and all 42 companies. Boldface is used to denote the best value for each statistic.

Algorithm	***Average***	***Median***	***StDev***	***Maximum***	***Minimum***	**Trades**
GP-FA	2.18	0.67	10.37	83.59	-52.54	100
GP-SA	2.65	0.57	10.75	173.13	-14.16	5
GP-TA	2.32	0.70	5.62	31.47	-20.9	15
GP-FASATA	2.99	0.58	12.1	**206.25**	-9.21	27
STGP-FASATA	3	0.64	9.9	144.32	**-8.81**	21
STGP-FASATA-S	3.78	1.07	11.57	83.59	-27.80	17
ActTrade	**3.9**	**1.318**	8.22	64.35	-14.51	18

ActTrade's standard deviation is also third best (GP-TA actually has the lowest values), indicating a higher concentration around the mean. The last column of Table 7 also presents the average number of trades per algorithm. On average, the ActTrade model executed 18 trades per company, compared to 17 for STGP-FASATA-S, 21 for STGP-FASATA, and 25 for GP-FASATA, all of which performed competitively among the benchmark algorithms. We observe a clear trend in which the number of trades decreases as the algorithm becomes more structured and selective — from 100 trades for the simplest GP-FA model down to 18 for ActTrade. At the same time, performance, as measured by average Sharpe ratio, improves consistently, rising from 2.18 to 3.9. Importantly, the more selective algorithms are not merely under-trading, but rather executing higher-quality trades. This points to increased efficiency and better risk-adjusted decision-making as algorithmic

sophistication increases.

Since each trade incurs the fee once, a uniform per-trade fee reduces a strategy's cumulative return roughly in proportion to how frequently it trades, so strategies with different turnover are not affected equally by a single fee assumption. We therefore report average trade counts alongside performance metrics (Table 7), allowing readers to judge how sensitive the reported rankings are to this cost assumption.

The algorithms fall into two broad groups: a low-turnover cluster (ActTrade: 18, STGP-FASATA-S: 17, STGP-FASATA: 21) and a higher-turnover cluster (GP-FASATA: 27, GP-TA: 15, GP-FA: 100). Because ActTrade's closest competitors in Sharpe ratio and rate of return fall within the low-turnover cluster, its advantage over them is relatively insensitive to the exact fee level, since both incur similar cost penalties as fees rise. Its advantage over markedly higher-turnover strategies such as GP-FA is more sensitive to the cost assumption, as GP-FA's much larger trade count means its returns erode faster under higher fees.

To better interpret the results, we conducted pairwise Kolmogorov–Smirnov (KS) tests comparing the ActTrade algorithm (chosen as the control due to its superior average and median performance) against each of the six GP benchmarks. The KS test evaluates whether the distributions of performance metrics (Sharpe ratio, rate of return, and risk), in this case only showing the Sharpe ratio results, for two algorithms originate from the same population. A low p-value indicates a statistically significant difference in their empirical distributions.To control for the multiple pairwise comparisons across algorithms and metrics, we applied the Benjamini–Hochberg False Discovery

Rate (FDR) correction with $q = 0.05$. This approach balances statistical power across the 18 simultaneous tests. Furthermore, for the sensitivity test, we conducted two analyses, one using using only valid runs (non-zero-return strategies) and the other using all runs, including zero-return cases, to assess any potential bias.

The results in Table 8 show that ActTrade exhibits statistically significant differences from nearly all other algorithms across Sharpe ratio under the FDR-adjusted significance level. When including zero-return runs (sensitivity test), the results remain robust, with most comparisons remaining significant, with only a few (notably versus GP-FA and STGP-FASATA) losing significance due to the inclusion of inactive runs.

Table 8 Kolmogorov–Smirnov (KS) test p-values comparing ActTrade against GP benchmarks under Benjamini-Hochberg (FDR) correction at $q = 0.05$ for Sharpe ratio. Bold values denote statistically significant differences. Results are shown for valid runs and for all runs (sensitivity analysis).

Dataset	**Comparison**	**p-value**	**FDR-adjusted p**
not incl. zero-return runs	ActTrade vs GP-FA	**1.41E-03**	**2.45E-03**
	ActTrade vs GP-SA	**2.49E-05**	**6.40E-05**
	ActTrade vs GP-TA	**9.78E-101**	**8.80E-100**
	ActTrade vs GP-FASATA	**8.87E-06**	**2.66E-05**
	ActTrade vs STGP-FASATA	**2.21E-03**	**3.32E-03**
	ActTrade vs STGP-FASATA-S	2.33E-01	2.33E-01
incl. zero-return runs	ActTrade vs GP-FA	1.58E-01	1.67E-01
	ActTrade vs GP-SA	**3.26E-34**	**2.93E-33**
	ActTrade vs GP-TA	**1.21E-02**	**1.67E-02**
	ActTrade vs GP-FASATA	**5.88E-10**	**2.60E-09**
	ActTrade vs STGP-FASATA	**4.66E-07**	**1.40E-06**
	ActTrade vs STGP-FASATA-S	6.41E-02	8.24E-02

5.2. Rate of Return

Table 9 presents the average rate of return values for all runs of the 42 companies. The algorithm that has the most highest values is ActTrade (9 companies), followed by GP-SA and GP-FA with 8 and 7 companies, respectively. GP-TA, STGP-FASATA and STGP-FASATA-S all have the highest value in 6 companies, while GP-FASATA in none of the companies.

Table 10 summarises the results, by presenting the mean, median, stan-

Table 9 Average values for rate of return per company. Boldface is used to denote the best value for the particular dataset.

Company	*GP-FA*	*GP-SA*	*GP-TA*	*GP-FASATA*	*STGP-FASATA*	*STGP-FASATA-S*	*ActTrade*
AAPL	0.0116	0.0135	0.0237	0.0243	0.0227	0.0277	**0.0242**
ADIDAS	0.0014	0.0014	0.0050	0.0023	**0.0082**	0	0
ALIBABA	0.0200	0.0304	0.0146	0.0229	0.0210	0.0244	**0.0326**
AMAZON	**0.0295**	-0.0013	0.0205	0.0223	0.0187	0	0
ASUS	-0.0339	0.0009	**0.0191**	-0.0107	-0.0022	0.0009	0.0119
ATVI	0.0103	0.0099	0.0157	0.0080	0.0124	0.0079	**0.0259**
BLACKB	0.0062	-0.0089	**0.0344**	0.0146	0.0281	0.0144	0.0022
COKE	**0.0163**	-0.0403	0.0085	0.0095	0.0131	0.0111	0.0119
EBAY	0.0094	0.0016	0	-0.0295	-0.0059	**0.0102**	0.0026
ESTEE	0.0028	**0.0095**	0.0091	0.0054	0.0094	0.0092	0.0053
FORD	-0.0038	0.0140	-0.0285	0.0006	0.0102	0	**0.0182**
FUJIFILM	**0.0317**	0.0308	-0.0257	0.0005	0.0131	-0.0321	0.0071
FUJITSU	0.0355	0.0646	0.0932	0.0894	0.0689	**0.1016**	0.0833
GM	-0.0007	**0.0237**	-0.0894	-0.0125	-0.0354	0	0
GOOGL	0.0015	0.0060	-0.0071	0.0019	0.0099	-0.0033	**0.0103**
HITACHI	-0.0027	**0.0125**	0	0.0054	-0.0184	0	0
HONDA	0.0111	0.0085	0.0012	0.0128	**0.0141**	0.0139	0
HSBC	-0.0026	**0.0099**	0.0023	0.0095	0.0052	0	0.0005
HYUNDAI	0.0130	0.0124	-0.0205	0.0268	0.0191	0.0344	**0.0356**
IBM	0.0150	0.0100	0.0315	0.0041	0.0134	**0.0467**	0.0250
INTC	**0.0251**	-0.0093	0.0016	-0.0092	0.0185	0	-0.0026
KERING	0.0217	0.0179	0.0132	0.0056	**0.0273**	0.0192	0
KODAK	0.0115	**0.0535**	0.0293	0.0113	0.0158	0.0228	0.0390
MCDON	-0.0019	**0.0123**	-0.0003	0	-0.0126	0	-0.0563
NESTLE	-0.0025	**0.0082**	-0.0024	0.0004	0.0052	-0.0021	0.0039
NFLX	**0.0608**	0.0033	-0.0105	0.0598	0.0349	0.0470	0.0525
NINTENDO	0.0036	0.0143	-0.0124	0.0147	0.0061	0.0069	**0.0168**
NYT	0.0127	**0.0326**	-0.0015	0.0059	0.0053	0.0113	0.0087
PANASONIC	0.0011	0.0260	0.0160	0.0147	0.0068	0.0061	**0.0262**
SAINSBURY	**0.0348**	0.0167	0.0154	0.0257	0.0118	0.0270	0.0217
SHISEIDO	-0.0480	-0.0106	0.0208	-0.0399	-0.0292	**0.0283**	-0.0468
SUBARU	-0.0057	-0.0155	**0.0338**	0	-0.0256	0.0036	0.0292
SUZUKI	0.0355	-0.0365	0.0007	0.0140	0	**0.0646**	0.0403
TENCENT	0.0069	0	-0.0543	0.0163	-0.0013	0.0099	**0.0187**
TESCO	0.0039	0.0164	0.0088	-0.0097	**0.0419**	0	-0.0302
TESLA	0.0505	0.0550	**0.0751**	0.0433	0.0413	0.0596	0.0601
TOYOTA	0.0172	0.0087	0	-0.0029	**0.0333**	0	0
UBISOFT	-0.0044	0	**0.0281**	0.0267	-0.0118	-0.0443	0
WALMART	0.0034	-0.0028	**0.0047**	-0.0004	0.0038	0.0029	0.0026
XEROX	0.0315	0.0058	-0.0175	0.0223	**0.0405**	0	0.0174
YAMAHACO	-0.0001	0.0049	-0.0069	0.0024	0.0076	**0.0140**	0.0113
YAMAHAMO	**0.0160**	0	0.0063	0	-0.0091	-0.0025	0.0042

dard deviation, maximum and minimum values across all 50 runs for each company. As we can observe, the proposed ActTrade has again the best mean value, and also has a median value very close to the best observed (0.0145 vs 0.015). Its standard deviation (0.031) is comparable to the lowest value of 0.028, showing again that it has relatively low volatility. The maximum rate of return value is the same for all algorithms but four algorithms and ActTrade has the lowest minimum value for rate of return.

Table 10 Summary statistics of rate of return across all 50 GP runs and all 42 companies. Boldface is used to denote the best value for each statistic.

Algorithm	***Average***	***Median***	***StDev***	***Maximum***	***Minimum***
GP-FA	0.014	**0.015**	0.038	0.069	-0.30
GP-SA	0.009	0.01	**0.028**	**0.11**	-0.12
GP-TA	0.013	0.014	0.033	0.10	-0.089
GP-FASATA	0.015	0.014	0.029	0.10	-0.10
STGP-FASATA	0.0135	0.0125	**0.028**	**0.11**	-0.18
STGP-FASATA-S	0.021	**0.015**	0.035	**0.11**	-0.087
ActTrade	**0.0225**	0.0145	0.031	**0.11**	**-0.06**

Kolmogorov–Smirnov (KS) test p-values comparing ActTrade against GP benchmarks under Benjamini-Hochberg (FDR) correction at $q = 0.05$ for rate of return. Bold values denote statistically significant differences. Results are shown for valid runs and for all runs (sensitivity analysis).

Table 11 Kolmogorov–Smirnov (KS) test p-values comparing ActTrade against GP benchmarks on the Rate of Return. Values are shown for both valid runs and all runs (sensitivity analysis). Reported p-values are raw and Benjamini–Hochberg FDR-adjusted ($q = 0.05$); significant differences are shown in bold.

Dataset	**Comparison**	**p-value**	**FDR-adjusted p**
not incl. zero-return runs	ActTrade vs GP-FA	**1.50E-03**	**2.45E-03**
	ActTrade vs GP-SA	**3.78E-06**	**1.36E-05**
	ActTrade vs GP-TA	**6.85E-90**	**4.11E-89**
	ActTrade vs GP-FASATA	**3.29E-03**	**4.23E-03**
	ActTrade vs STGP-FASATA	**2.73E-03**	**3.77E-03**
	ActTrade vs STGP-FASATA-S	1.98E-01	2.09E-01
incl. zero-return runs	ActTrade vs GP-FA	9.17E-01	9.17E-01
	ActTrade vs GP-SA	**4.61E-35**	**8.30E-34**
	ActTrade vs GP-TA	**7.73E-03**	**1.16E-02**
	ActTrade vs GP-FASATA	**4.79E-10**	**2.60E-09**
	ActTrade vs STGP-FASATA	**1.50E-06**	**3.39E-06**
	ActTrade vs STGP-FASATA-S	1.02E-01	1.23E-01

5.3. Risk

Table 12 presents the average risk values for all runs of the 42 companies. ActTrade has again the highest number of best (lowest) values, along with

GP-TA (10 each)[12]. GP-SA and STGP-FASATA-S have 8 and 7 companies with the lowest risk values, respectively. Lastly, the algorithms with the least companies are STGP-FASATA (4 companies), GP-FASATA (2 companies) and GP-FA (1 company).

[12]Zero (0) risk values, which indicate that no trading was performed by an algorithm, are not taken into account when comparing values to identify the best performing algorithm.

Table 12 Average values for risk per company. Boldface is used to denote the best value for the particular dataset.

Company	*GP-FA*	*GP-SA*	*GP-TA*	*GP-FASATA*	*STGP-FASATA*	*STGP-FASATA-S*	*ActTrade*
AAPL	0.0076	0.0236	0.0082	0.0089	0.0174	**0.0059**	0.0110
ADIDAS	0.0270	**0.0148**	0.0264	0.0276	0.0261	0	0
ALIBABA	0.0280	0.0040	0.0308	**0.0239**	0.0264	0.0256	0.0027
AMAZON	0.0248	0.0311	**0.0022**	0.0179	0.0196	0	0
ASUS	0.0248	0.0219	0.0362	0.0137	0.0131	0.0162	**0.0087**
ATVI	0.0212	**0.0043**	0.0083	0.0239	0.0168	0.0201	0.0094
BLACKB	0.0429	0.0869	**0.0280**	0.0387	0.0352	0.0411	0.0593
COKE	0.0070	0.0701	0.0125	0.0084	0.0091	**0.0060**	0.0100
EBAY	0.0185	0.0288	0	0.0523	0.0455	0.0234	**0.0176**
ESTEE	0.0230	0.0044	0.0168	0.0211	0.0139	**0.0028**	0.0151
FORD	0.0295	0.0057	0.0482	0.0334	0.0147	0	**0.0055**
FUJIFILM	0.0302	0.0554	0.0291	0.0235	0.0237	**0.0193**	0.0296
FUJITSU	0.0515	0.0173	0.0170	0.0147	0.0324	**0.0079**	0.0241
GM	0.0418	0.0166	**0.0043**	0.0532	0.0530	0	0
GOOGL	0.0253	0.0146	0.0466	0.0198	0.0203	0.0279	**0.0104**
HITACHI	0.0488	0.0565	0	0.0534	**0.0145**	0	0
HONDA	0.0100	**0.0080**	0.0082	0.0175	0.0176	0.0127	0
HSBC	0.0179	**0.0054**	0.0088	0.0064	0.0078	0	0.0066
HYUNDAI	0.0231	0.0216	0.0446	0.0163	0.0320	0.0304	**0.0150**
IBM	0.0326	0.0674	**0.0038**	0.0488	0.0483	0.0145	0.0144
INTC	0.0300	0.0446	0.0317	0.0704	**0.0083**	0	0.0902
KERING	0.0378	**0.0258**	0.0343	0.0411	0.0294	0.0478	0
KODAK	0.0463	0.0356	0.0361	0.0753	0.0615	0.0347	**0.0345**
MCDON	0.0432	0.0397	**0.0299**	0	0	0	0.0435
NESTLE	0.0237	**0.0034**	0.0173	0.0139	0.0101	0.0160	0.0102
NFLX	**0.0075**	0.0622	0.0898	0.0113	0.0271	0.0310	0.0123
NINTENDO	0.0310	0.0357	0.0491	0.0353	**0.0206**	0.0505	0.0314
NYT	0.0413	**0.0190**	0.0406	0.0399	0.0405	0.0297	0.0385
PANASONIC	0.0426	**0.0109**	0.0204	0.0259	0.0262	0.0302	0.0263
SAINSBURY	0.0196	0.0272	0.0229	0.0304	0.0344	0.0198	**0.0115**
SHISEIDO	0.0699	0.0864	0.0585	**0.0417**	0.0487	0.0625	0.0758
SUBARU	0.0349	0.0559	0.0170	0	0.0515	0.0380	**0.0141**
SUZUKI	0.0324	0.0686	0.0561	0.0455	0	**0.0102**	0.0314
TENCENT	0.0260	0	0.0987	0.0088	0.0227	0.0246	**0.0049**
TESCO	0.0361	0.0189	0.0224	0.0415	0.0027	0	**0.0021**
TESLA	0.0407	0.0474	**0.0346**	0.0550	0.0665	0.0590	0.0426
TOYOTA	0.0127	0.0184	0	0.0276	**0.0110**	0	0
UBISOFT	0.0466	0	**0.0277**	0.0297	0.0460	0.0817	0
WALMART	0.0116	0.0176	**0.0074**	0.0244	0.0153	0.0225	0.0127
XEROX	0.0300	0.0612	**0.0213**	0.0340	0.0272	0	0.0539
YAMAHACO	0.0314	0.0242	0.0374	0.0263	0.0241	**0.0142**	0.0156
YAMAHAMO	0.0243	0	**0.0175**	0	0.0206	0.0229	0.0276

Similarly to Section 5.1 and 5.2, Table 13 displays the summary statistics for the values of risk of all 50 runs for each company across the eight GP algorithms. As we can observe, we get a very similar picture to what we have seen so far: the proposed ActTrade algorithm has the lowest mean and median values, as well as the lowest standard deviation value. The lowest maximum risk value is observed by STGP-FASATA-S, while the lowest minimum risk by STGP-FASATA. The KS statistical test in Table **??** confirms the above results, as the proposed ActTrade algorithm is able to statistically and significantly outperform all other GP algorithms at the 5% level.

Table 13 Summary statistics of risk across all 50 GP runs and all 42 companies. Boldface is used to denote the best value for each statistic.

Algorithm	***Average***	***Median***	***StDev***	***Maximum***	***Minimum***
GP-FA	0.029	0.026	0.024	0.21	0.0007
GP-SA	0.034	0.022	0.04	0.25	0.0003
GP-TA	0.027	0.018	0.025	0.18	0.0006
GP-FASATA	0.030	0.023	0.027	0.17	0.00007
STGP-FASATA	0.028	0.022	0.026	0.19	**0.000067**
STGP-FASATA-S	0.026	0.0177	0.026	**0.126**	0.00046
ActTrade	**0.024**	**0.016**	**0.023**	0.18	0.0004

Table 14 reports the pairwise Kolmogorov–Smirnov (KS) test results for risk, comparing ActTrade against the GP benchmarks. Following the Benjamini–Hochberg FDR correction, ActTrade remains statistically distinct from nearly all benchmarks in both valid-run and all-run datasets. These

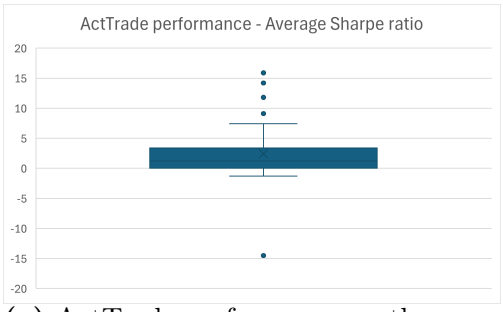
outcomes confirm that ActTrade's superior performance is not sensitive to data filtering and is robust across evaluation metrics.

Table 14 Kolmogorov–Smirnov (KS) test p-values comparing ActTrade against GP benchmarks under Benjamini-Hochberg (FDR) correction at $q = 0.05$ for risk. Bold values denote statistically significant differences. Results are shown for valid runs and for all runs (sensitivity analysis).

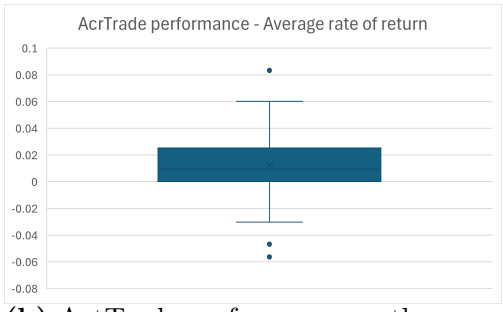
Dataset	**Comparison**	**p-value**	**FDR-adjusted p**
not incl. zero-return runs	ActTrade vs GP-FA	**2.05E-06**	**9.22E-06**
	ActTrade vs GP-SA	**6.52E-05**	**1.47E-04**
	ActTrade vs GP-TA	**5.24E-152**	**9.43E-151**
	ActTrade vs GP-FASATA	**6.61E-04**	**1.32E-03**
	ActTrade vs STGP-FASATA	**5.94E-03**	**7.13E-03**
	ActTrade vs STGP-FASATA-S	1.85E-01	2.09E-01
incl. zero-return runs	ActTrade vs GP-FA	**6.34E-06**	**1.27E-05**
	ActTrade vs GP-SA	**1.48E-03**	**2.43E-03**
	ActTrade vs GP-TA	**4.42E-05**	**7.95E-05**
	ActTrade vs GP-FASATA	**7.21E-10**	**2.60E-09**
	ActTrade vs STGP-FASATA	**9.16E-07**	**2.35E-06**
	ActTrade vs STGP-FASATA-S	1.37E-01	1.54E-01

For completeness, we also present Figure 2, which shows box plots of the ActTrade algorithm for the three metrics of Sharpe ratio, rate of return, and risk.

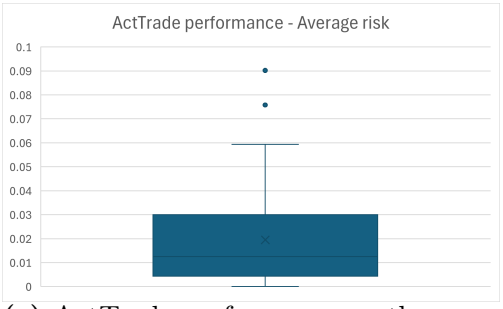
Figure 2 ActTrade performance in Sharpe ratio, rate of return, and risk.



(a) ActTrade performance on the average Sharpe ratios per company.



(b) ActTrade performance on the average rate of return per company.



(c) ActTrade performance on the average risk per company.

5.4. The role of the active trading operator

Given the above positive results for the ActTrade algorithm, we are interested in understanding the effects of the proposed genetic operator during the learning (training) phase of the models. To do this, we track the fitness value (Sharpe ratio) of the population for each generation.

As previously mentioned in Sections 3.2.4 and 3.2.5, the novel genetic operator yields offspring through two distinct mechanisms. First, it generates offspring directly, by incorporating the three newly created individuals into the next generation. These individuals are identified as follows: Offspring-

FA (where the FA subtree maximises the Sharpe ratio, while the SA and TA subtrees maximise the number of trades), Offspring-SA (with the SA subtree maximising the Sharpe ratio, and the FA and TA subtrees maximising trades), and Offspring-TA (where the TA subtree maximises the Sharpe ratio, while the FA and SA subtrees maximise trades). Second, the operator indirectly influences offspring generation by using Offspring-FA, Offspring-SA, and Offspring-TA as parents for crossover and mutation. Consequently, at each generation, we encounter not only Offspring-FA, Offspring-SA, and Offspring-TA but also several individuals generated by having these offspring as parents. Furthermore, traditional tournament selection also contributes to the creation of individuals in each generation.[13]

To organise our analysis effectively, we categorise the offspring in each generation in the following categories:

- Offspring-FA
- Offspring-SA
- Offspring-TA
- Tournament Selection Group (Group-Tourn): Individuals in this group are the result of conventional tournament selection.

[13]There is also an elitist individual that exists at each generation, but for the purpose of this section's analysis, we focus solely on understanding the performance of individuals derived through the proposed genetic operator and how they compare to those generated by tournament selection.

- FA Group (Group-FA): This group comprises offspring whose first parent is Offspring-FA. Given that this is a selection method, several offspring are generated in this way, hence the word 'group' in the naming.
- SA Group (Group-SA): The SA group includes offspring whose first parent is Offspring-SA.
- TA Group (Group-TA): Offspring in this group have parents whose first parent is Offspring-TA.

Table 15 presents the average Sharpe ratio values for each offspring group, across all generations, over the 50 GP runs. Our primary objective is to rigorously compare the performance of offspring generated through our proposed operator with those generated via traditional tournament selection. Note that we exclude duplicate models (i.e. models/trees that exist more than once in a given group). As we can observe, all categories of offspring show high average Sharpe ratio values. More importantly, there are many cases that offspring generated through tournament selection (Group-Tourn) have on average lower fitness values that the other categories. This is an important finding, because it indicates that offspring generated by the proposed genetic operator (either directly or indirectly through acting as a parent for crossover and mutation) has the ability to create competitive models that often outperform models derived from tournament selection.

Table 15 Average fitness function (Sharpe ratio) values of the different offspring categories, across all generations, over 50 GP runs. Boldface is used to denote the best value for the particular company.

Company	*Offspring-FA*	*Offspring-SA*	*Offspring-TA*	*Group-Tourn*	*Group-FA*	*Group-SA*	*Group-TA*
AAPL	5.33	**39.28**	11.63	15.20	15.67	16.85	11.63
ADIDAS	**19.43**	2.73	6.06	13.62	15.89	13.86	15.29
ALIBABA	14.22	13.08	**16.29**	11.39	13.26	12.39	15.81
AMAZON	5.32	32.30	**42.95**	24.27	25.43	27.13	26.56
ASUS	12.13	5.60	**23.25**	11.83	12.63	12.35	12.07
ATVI	6.95	10.54	**13.31**	8.54	8.50	8.63	8.37
BLACKB	5.68	16.89	**28.64**	14.96	15.33	16.98	16.83
COKE	8.61	9.48	**21.29**	10.62	11.13	10.65	13.24
EBAY	8.68	20.11	**34.07**	16.18	16.81	17.33	18.46
ESTEE	**8.52**	7.74	6.80	6.49	7.78	7.67	6.99
FORD	6.60	10.07	**18.90**	9.54	10.98	11.05	8.92
FUJIFILM	11.03	13.54	**15.00**	12.00	14.53	14.52	10.32
FUJITSU	17.42	10.06	**50.89**	20.25	24.53	24.21	19.32
GM	27.42	4.34	**63.08**	28.01	31.12	28.40	33.35
GOOGL	10.20	**39.83**	11.62	24.43	25.25	27.69	25.90
HITACHI	**206.79**	3.81	72.04	141.38	173.94	159.22	169.06
HONDA	14.99	16.94	**20.11**	16.33	18.42	18.98	17.17
HSBC	3.18	4.28	**11.64**	5.55	6.14	6.39	6.23
HYUNDAI	5.23	6.31	**19.49**	11.54	12.28	12.34	12.62
IBM	13.74	**106.57**	26.19	46.14	41.85	50.43	26.60
INTC	14.62	**45.51**	7.78	19.78	21.86	23.82	17.26
KERING	11.65	**21.62**	17.77	12.16	13.69	14.58	15.18
KODAK	4.41	**10.41**	6.39	5.16	5.93	6.38	5.21
MCDON	19.96	4.43	**51.78**	31.28	33.16	31.55	35.05
NESTLE	5.44	12.37	**23.92**	11.30	12.55	13.05	10.94
NFLX	17.24	12.92	**24.38**	16.66	17.17	16.11	19.79
NINTENDO	5.04	4.72	**6.98**	4.39	4.86	4.79	4.24
NYT	57.03	8.04	**597.05**	225.10	241.75	233.01	246.84
PANASONIC	8.07	8.03	**15.21**	9.73	9.93	10.31	9.34
SAINSBURY	25.82	29.24	**35.56**	18.64	21.78	22.03	17.83
SHISEIDO	11.54	10.35	**28.96**	16.75	18.76	17.69	17.14
SUBARU	12.11	5.39	**13.89**	10.63	11.39	10.29	9.91
SUZUKI	10.66	17.85	**19.50**	15.66	15.36	15.74	14.34
TENCENT	5.69	**12.82**	12.11	9.30	7.11	8.42	9.47
TESCO	**64.87**	16.83	13.56	36.99	45.71	41.62	50.50
TESLA	13.41	**19.12**	8.58	11.73	13.05	13.78	12.26
TOYOTA	3.61	12.95	**279.70**	161.56	98.91	103.87	258.83
UBISOFT	15.82	10.47	11.49	15.51	17.07	17.02	**18.72**
WALMART	3.26	**11.28**	7.49	6.51	6.23	7.10	6.73
XEROX	19.79	7.84	**49.99**	27.40	29.48	27.81	34.08
YAMAHACO	3.47	5.31	**11.01**	6.18	6.96	6.70	6.30
YAMAHAMO	12.98	8.41	**15.83**	11.16	12.63	12.31	13.53

5.5. Ablation study integrating the three branches in the ActTrade algorithm

For each company, we first exclude runs with $\text{Sharpe}_{\text{Total}} = 0$ (to avoid degenerate cases), then compute per-company averages over all remaining runs. We report the mean of these per-company averages across the 42 companies.

We evaluate interpretability and robustness of the proposed ActTrade based trading strategies by performing an internal ablation analysis of the individual branches in the algorithm. For each company and each branch ($x \in \{\text{FA}, \text{SA}, \text{TA}, \text{Total}\}$), we compute the following metrics:

Delta Sharpe Ratio Contribution (ΔSRC).

$$\Delta\text{SRC}_{\text{Max}} = \text{Sharpe}_{\text{Total}} - \max\left(\text{Sharpe}_{\text{FA}}, \text{Sharpe}_{\text{SA}}, \text{Sharpe}_{\text{TA}}\right) \tag{17}$$

$$\Delta\text{SRC}_{\text{Avg}} = \text{Sharpe}_{\text{Total}} - \frac{\text{Sharpe}_{\text{FA}} + \text{Sharpe}_{\text{SA}} + \text{Sharpe}_{\text{TA}}}{3} \tag{18}$$

Trade share.

$$\text{TradeShare}_x = \frac{\text{Trades}_x}{\text{Trades}_{\text{Total}}}, \qquad x \in \{\text{FA}, \text{SA}, \text{TA}\} \tag{19}$$

Table 16 reports the results averaged across 42 companies. On average, the integrated strategy attains a higher Sharpe ratio (3.9) than any individual branch (FA 0.84, SA 0.87, TA 1.3), and a positive $\Delta\text{SRC}_{\text{Avg}}$ (2.2), indicating that the full tree improves over the average performance of the three separate branches. However, the relatively small $\Delta\text{SRC}_{\text{Max}}$ (0.3) and the non-significant Wilcoxon result show that the integrated tree does not systematically outperform the best single branch for a given company. Large

standard deviations reflect substantial cross-company heterogeneity, consistent with the market-dependent effectiveness of FA, SA, and TA signals.

Table 16 Internal ablation results (average across companies).

Metric	Mean
$\text{Sharpe}_{\text{FA}}$	0.84
$\text{Sharpe}_{\text{SA}}$	0.87
$\text{Sharpe}_{\text{TA}}$	1.3
$\text{Sharpe}_{\text{Total}}$	3.9
$\Delta\text{SRC}_{\text{Max}}$	0.3
$\Delta\text{SRC}_{\text{Avg}}$	2.2
$\text{TradeShare}_{\text{FA}}$	71.71%
$\text{TradeShare}_{\text{SA}}$	44.74%
$\text{TradeShare}_{\text{TA}}$	44.57%

Overall, these patterns suggest that ActTrade gains from integrating all three branches in terms of typical performance and stability, but the effect is incremental rather than a strong synergy. The integrated tree delivers Sharpe ratios that are comparable to, and in many cases higher than, the best individual FA, SA, or TA branch, while avoiding strong dependence on any single signal.This suggests that ActTrade's strength lies in providing robust performance rather than systematically outperforming the optimal branch.

To isolate the contribution of each of the three novel components introduced in this study, we map the evidence above explicitly onto each component in turn.

For the active trading operator, Section 5.4 and Table 15 show that offspring generated directly or indirectly by this operator achieve, on average, higher fitness (Sharpe ratio) than offspring generated by standard tournament selection across the majority of companies, indicating that this operator's contribution to performance is isolated from the effect of the strongly-typed architecture itself.

For the third analysis branch, the ΔSRC_{Max} and ΔSRC_{Avg} results reported above indicate that the integrated tree improves over average single-branch performance, though it does not systematically exceed the single best-performing branch for every company. This suggests the benefit of integration is incremental and stability-driven rather than a uniform synergy across all cases.

For the fitness function, the comparison between STGP-FASATA and STGP-FASATA-S (Tables 7, 10, 13) isolates its contribution, where both share the same strongly-typed architecture and lack the active-trading operator, differing only in the fitness function used. STGP-FASATA-S's improvement over STGP-FASATA across Sharpe ratio, rate of return, and risk indicates a positive, isolated contribution from the proposed fitness function, independent of the active-trading operator.

5.6. Results of each market

Although all 42 companies were observed over the same time period, their price trajectories vary significantly, with some exhibiting an overall upward trend, while others trend downward. We therefore consider it important to examine the performance of the GP algorithms across these different market profiles to understand their adaptability under varying conditions.

To do this, we looked at the first and last price of the test set for each company, and calculated the return value. We then created three groups:

- Group 1, which includes those companies whose price experienced a long-term increase of at least 20%.
- Group 2, which includes those companies whose price experienced a long-term increase between 0% and 19.99%.
- Group 3, which includes those companies whose price experienced a long-term decrease, i.e. had a negative return.

After defining the above groups, 19 companies were placed in Group 1, 14 in Group 2, and 9 in Group 3.

We then report the average value of each metric (Sharpe ratio, rate of return, and risk) for each GP algorithm, across the datasets of each group. As we can observe from Table 17, the proposed ActTrade performs strongly in terms of the aggregate metric of Sharpe ratio for Group 1, and it has a high value for Group 3, while in Groups 2 and 3 the winner is GP-FASATA. With regards to rate of return, ActTrade has the best value for Group 1 and Group 3, while in Group 2 the best performance is that of STGP-FASATA-S. Lastly, in terms of risk ActTrade performs way lower than the rest of the algorithms for Group 2 and Group3, while STGP-FASATA has the lowest risk for Group 1.

In summary, ActTrade achieves the best rate of return in two of the groups and consistently delivers low risk values across all groups. Given the variability of results across groups and metrics, the Sharpe ratio is particularly useful as an aggregate measure of both return and risk. As discussed, ActTrade

demonstrates strong performance in datasets with significant positive price movements (Group 1) and performs best in two metrics for datasets with negative price movements (Group 3). It also performs notably well, particularly in terms of risk, for datasets with average price movements (Group 2). These results suggest that our proposed algorithm excels in both strongly uptrend and downtrend markets—an important finding that highlights its versatility across different market conditions. Furthermore, the solid performance of the strongly-typed GP algorithms in terms of the Sharpe ratio emphasises the value of the strongly-typed GP architecture in effectively combining different types of analysis indicators.

These three groups of companies also provide an indirect view of the model's behaviour under varying market conditions — with Group 3 (declining prices) and Group 2 (moderate price changes) likely encompassing periods of higher volatility or lower trading volume. The model's ability to maintain favourable Sharpe ratios and minimise risk in these groups demonstrates its robustness under such challenging conditions.

Table 17 Separated average results per metric per trend group.

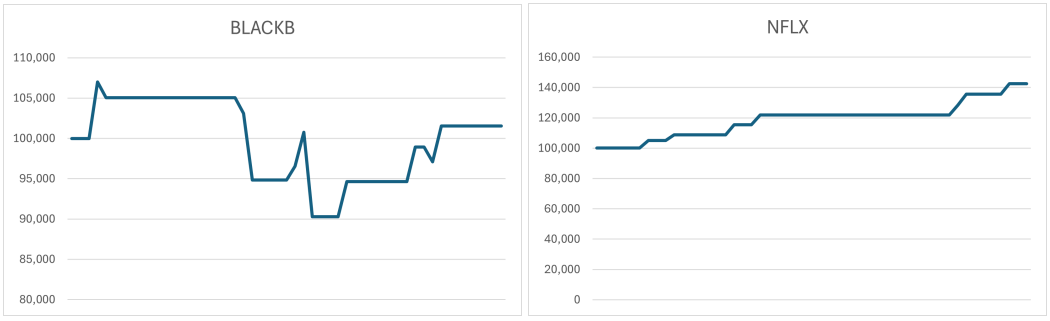
Market	***Algorithm***	***Sharpe ratio***	***Rate of return***	***Risk***
Group 1 (>20%)	GP-FA	1.63	0.016	0.028
	GP-SA	2.2	0.013	0.029
	GP-TA	2.7	0.012	0.029
	GP-FASATA	2.49	0.014	0.029
	STGP-FASATA	2.55	0.017	**0.023**
	STGP-FASATA-S	3.3	0.017	0.024
	ActTrade	**4.17**	**0.020**	0.027
Group 2 (0% - 19.99%)	GP-FA	1.09	0.003	0.031
	GP-SA	3.8	0.01	0.034
	GP-TA	1.89	0.0035	0.039
	GP-FASATA	**4.86**	0.006	0.030
	STGP-FASATA	3.7	0.008	0.029
	STGP-FASATA-S	1.8	**0.02**	0.029
	ActTrade	1.8	0.011	**0.021**
Group 3 (<0%)	GP-FA	2.99	0.010	0.030
	GP-SA	2.26	0.001	0.036
	GP-TA	-1.32	-0.001	0.025
	GP-FASATA	1.43	0.006	0.035
	STGP-FASATA	2.26	-0.001	0.033
	STGP-FASATA-S	**3.5**	0.011	0.030
	ActTrade	2.18	**0.012**	**0.024**

Lastly, Figure 3 presents the average equity curves over 50 GP runs for BlackBerry and Netflix during the test period from February 2019 to January 2020. These two companies exhibited volatile price trajectories: BlackBerry's

share price declined by nearly 25%, while Netflix saw a more modest average daily volatility of around 2%, a maximum drawdown of almost 34%, and ended the period with a 1.3% increase. We include these equity curves to demonstrate the algorithm's ability to adapt to challenging and uncertain market conditions—preserving capital during sharp downturns (as with BlackBerry) and capturing gains even in mildly declining markets (as seen with Netflix).

As shown, ActTrade effectively managed risk in the face of a steep decline in BlackBerry's price, limiting drawdowns to the low 90k range and ultimately recovering to finish slightly above its starting equity. This contrasts sharply with the substantial losses that would have occurred under a passive buy-and-hold strategy. For Netflix, the strategy steadily compounded gains, rising from 100k to over 140k despite the stock's overall volatility. These results highlight ActTrade's dual strengths in volatile conditions: strong drawdown control and the ability to capitalise on trading signals to generate consistent returns, even when market direction is unfavourable.

Figure 3 ActTrade equity curves for BlackBerry (BLACKB) and Netflix (NFLX).



5.7. Best tree results

Although the average performance metrics discussed in previous sections provide a general overview of how each algorithm performs across multiple runs, this section focuses on the best-performing outcome achieved by each genetic programming (GP) algorithm. Specifically, for each of the 50 independent runs conducted per company, we identify the tree with the highest fitness score on the training set. This top-performing tree is then evaluated on the corresponding unseen test set to assess its generalisation ability. Thus, the results in this section represent the single best model produced across the multiple runs, per dataset.

This type of analysis holds particular relevance in financial applications. In practice, investors and practitioners who use GP-based trading strategies often execute multiple runs and select the most promising model, typically the one with the highest fitness or best financial performance, for deployment in live trading. Therefore, an algorithm's capacity to generate high-quality individual solutions, even if only occasionally, is a key strength in real-world financial scenarios.

Table 18 reports the average test-set performance of these best-performing trees across all 42 datasets for each GP algorithm. To ensure completeness, this table includes all GP variants introduced earlier, even those not previously discussed in detail with respect to best-tree performance, in order to avoid redundancy in earlier sections.

For consistency with prior analyses, any company where all runs produced a performance value of 0 has been excluded from this summary. Lastly, note that results are presented separately for each financial performance metric

(Sharpe ratio, rate of return, and risk) to facilitate a clearer comparison of the algorithms' strengths.

Table 18 Best trees average performance across the 42 datasets

Algorithm	***Sharpe ratio***	***Rate of return***	***Risk***
GP-FA	-3.92	-0.003	0.025
GP-SA	1.77	0.018	0.024
GP-TA	0.22	0.003	0.05
GP-FASATA	0.95	0.020	0.025
STGP-FASATA	3.23	-0.004	0.027
STGP-FASATA-S	1.76	0.020	0.04
ActTrade	**3.51**	**0.039**	**0.021**

As we can observe, the proposed ActTrade algorithm has the highest Sharpe ratio and rate of return values while achieving the lowest risk. STGP-FASATA-S, STGP-FASATA and GP-SA algorithms have the highest Sharpe ratio values after ActTrade, although STGP-FASATA does have the lowest rate of return. As the Sharpe ratio is an aggregate metric that considers both return and risk, the fact that the best tree of ActTrade has the best value makes it a very positive result. It is also worth noting that practitioners pay particular attention to such aggregate metrics [54], thus ActTrade best tree's performance is of particular importance.

To further contextualise the best tree results, we compared them against the average performance across all 50 runs. For the ActTrade algorithm, the best tree achieved a Sharpe ratio of 3.51, compared to an average of 3.9; a

rate of return of 0.039, versus an average of 0.0225; and a risk value of 0.021, versus an average of 0.023. These relatively small differences indicate that the best-performing model is not an extreme outlier, but rather representative of the upper end of the distribution. This reinforces the relevance of the best tree analysis in practical financial settings, where users may reasonably select and deploy the strongest model from a batch of independent runs.

5.8. Algorithmic Complexity

The computational complexity of the proposed ActTrade GP algorithm is driven by (i) the cost of fitness evaluation, (ii) the cost of GP operators under the strongly-typed constraints, and (iii) the overhead of the active-trading operator. Let N denote the population size, G the number of generations, T the number of trading days, and D the maximum tree depth.

For each individual, the fitness function evaluates four Sharpe ratios, the global strategy and the three FA, SA, and TA subtrees. Each Sharpe computation requires iterating through the entire price series, executing trading logic, and recording the return distribution. Since all three subtrees reuse the same daily data and trading simulation, the asymptotic cost remains linear in T. Thus, the per-individual fitness cost is $O(T)$.

The crossover operator exchanges compatible subtrees between parents and the strong typing constrains the exchange to subtrees of the same analysis type (FA, SA, TA). Since tree depth is limited to D, crossover requires time $O(D)$.

Point mutation involves replacing a function or terminal node with another node of the same type. This also requires at most a subtree regeneration step bounded by depth D, resulting in $O(D)$ for mutation.

Across the entire population, the cumulative operator cost per generation is therefore $O(ND)$, which is dominated by the $O(NT)$ fitness term for realistic values where $T \gg D$.

For the active trading operator, we identify the high-activity, risk-adjusted individuals and inject modified variants of their subtrees into the population. Thus, we have two main computations, candidate selection and tree construction. The first ranks individuals by activity and Sharpe ratio uses statistics already computed during fitness evaluation and therefore requires only $O(N)$. The second recombines FA, SA, and TA subtrees which produce new individuals with a time bounded by $O(D)$.

Thus, the overall overhead introduced by the active trading operator per generation is $O(N + D)$.

In addition to these operators, elitism selects the best individuals from the current population and copies them into the next generation. This requires ranking or partially sorting individuals by fitness. Even if we assume that the algorithm is sorting the entire population, the computational cost is at most $O(N \log N)$ per generation. In practice, since the elitist set is small, the effective cost is closer to $O(N)$, but in either case it is dominated by the fitness-evaluation cost when $T \gg \log N$. Thus, elitism does not change the overall complexity of the algorithm.

For the overall complexity, we aggregate the contributions of all components, yielding a computational cost of $O(NGT) + O(NGD) + O(NG)$.

Since $T \gg D$ in typical financial datasets and T dominates all constant or depth-bounded terms, the complexity simplifies to the standard GP form $O(NGT)$, representing linear scaling in the number of trading days, popula-

tion size, and generations.

5.9. Indicators' usage and model interpretability

In the indicator usage analysis (see Table 19), we focused on the best-performing tree from each algorithm for each of the 42 companies. To provide a comprehensive overview, we computed the total number of times each indicator was used across all datasets. This analysis reveals a well-distributed utilisation of indicators across the GP algorithms.

Notably, the GP-SA and STGP-FASATA-S algorithms tend to produce trees with higher complexity. For GP-SA, this is due to the consistently high use of indicators across companies, while for STGP-FASATA-S, it stems from outlier cases in certain datasets. In contrast, the GP-FASATA algorithm shows less indicator overuse. Among the indicator groups, technical analysis (TA) features are more frequently selected by GP-FASATA, while the strongly-typed GP variants show balanced usage across all three analysis types. As expected, STGP-FASATA-S –guided by its fitness function– consistently selects a broad range of features from FA, SA, and TA. Meanwhile, ActTrade demonstrated the ability to select from a wide range of indicators while relying on a smaller subset, leading to more compact trees and improved predictive performance.

This ability to generate smaller trees marks an important step towards model interpretability. As shown in Table 20, the average number of nodes in the final generation for the ActTrade algorithm is 24, while the average number for best-performing trees is 22. Many of these trees are simple and easy to interpret, as illustrated in Figure 4, which shows a representative example from the Apple dataset. The tree consists of 14 nodes and encodes a

straightforward trading strategy. The root node is a `3-AND`, combining fundamental analysis (LT_{FA}), sentiment analysis (AND_{SA}) and technical analysis (LT_{TA}) sub-branches, of them two using one indicator and associated thresholds and the third going further, using two indicators and the corresponding thresholds.

Beyond indicator frequencies, the interpretability of the evolved models can be assessed by examining the composition of representative trees. Figure 4 shows such an example for Apple (AAPL), where the ActTrade algorithm produces a compact 14-node tree. The tree structure demonstrates three important interpretability characteristics:

1. **Balanced feature integration:** The root node is a 3-AND operator, requiring agreement between one FA indicator (`Quick Ratio`), one SA composite node (`SUMsubj`, `SUMsen`), and one TA indicator (`Midprice-5`). This reflects the algorithm's design to encourage the combination of difference indicator types.
2. **Threshold-based transparency:** The tree encodes explicit thresholds (e.g., Quick Ratio < 0.98, Midprice-5 < 0.45), making the decision rules interpretable and actionable for human analysts. Unlike black box models, this allows the user to understand the economic meaning behind each branch.
3. **Selective complexity:** Although ActTrade is capable of producing large trees (see Table 20), many of the best-performing individuals are compact.

When we combine the above, we observe a synergy between frequency analysis and structural interpretability. While some indicators (e.g., Price-to-

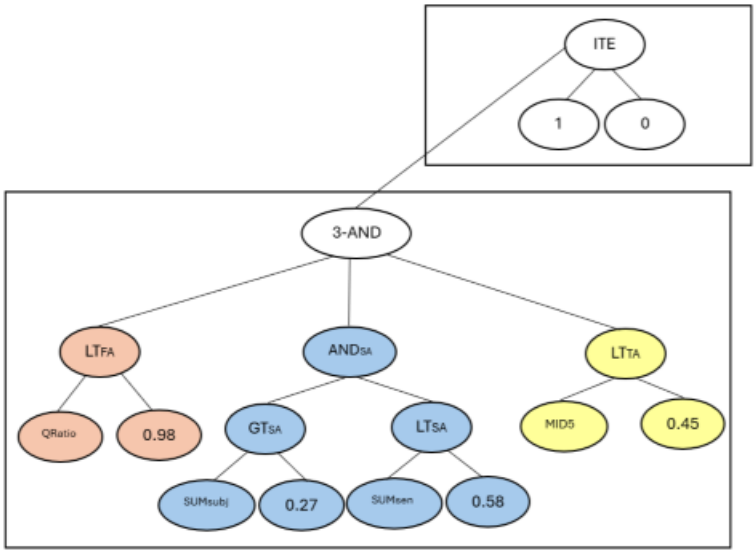
Table 19 Indicators from best models of 42 companies for strongly-typed GP algorithms.

Indicator	*GP-FA*	*GP-SA*	*GP-TA*	*GP-FASATA*	*STGP-FASATA*	*STGP-FASATA-S*	*ActTrade*
Net Profit Ratio	47	-	-	22	7	35	1
Return on Equity	27	-	-	17	6	16	4
Quick Ratio	42	-	-	24	8	43	2
Price-Earnings ratio	26	-	-	10	10	33	7
Price to Book ratio	55	-	-	29	4	73	12
Price-Sales ratio	89	-	-	40	7	88	13
Debt to Equity	104	-	-	15	10	62	13
Total Revenues	40	-	-	7	6	30	1
Levered free Cash Flow	50	-	-	17	3	12	4
Diluted EPS	30	-	-	13	7	11	2
EBITDA	39	-	-	7	5	8	3
R&D Expenses	37	-	-	28	0	48	6
TEXTpol	-	90	-	15	6	21	11
TEXTsubj	-	52	-	15	9	20	3
TTITLEpol	-	88	-	53	6	36	4
TTITLEsubj	-	91	-	28	10	67	6
SUMMpol	-	69	-	14	8	29	6
SUMMsubj	-	63	-	30	7	72	2
TEXTsenti	-	41	-	17	1	26	6
TITLEsenti	-	44	-	14	2	23	4
SUMMsenti	-	62	-	26	11	17	8
TEXTafinn	-	121	-	20	4	93	9
TITLEafinn	-	94	-	23	11	43	5
SUMMafinn	-	169	-	14	11	165	10
Moving Average - 5 days	-	-	40	34	8	102	11
Moving Aeverage - 10 days	-	-	62	64	12	89	9
Momentum - 5 days	-	-	46	17	9	18	6
Momentum - 10 days	-	-	38	18	4	21	10
ROC - 5 days	-	-	60	17	9	24	5
ROC - 10 days	-	-	40	12	6	18	13
William %R - 5 days	-	-	45	14	9	52	16
Williams %R - 10 days	-	-	39	38	14	38	9
Midprice - 5 days	-	-	53	24	11	39	7
Midprice - 10 days	-	-	45	51	7	33	6
Volatility - 5 days	-	-	53	19	9	59	8
Volalitily - 10 days	-	-	51	64	7	50	15

Table 20 Final and Best Tree Total Nodes per Company

Company	Final Tree Nodes	Best Tree Nodes
AAPL	14	14
ADIDAS	14	14
ALIBABA	26	30
AMAZON	10	10
ASUS	26	14
ATVI	22	14
BLACKB	10	22
COKE	18	14
EBAY	18	14
ESTEE	10	14
FORD	18	10
FUJIFILM	26	14
FUJITSU	22	18
GM	18	18
GOOGL	10	30
HITACHI	18	18
HONDA	14	30
HSBC	14	10
HYUNDAI	42	22
IBM	10	22
INTC	26	14
KERING	18	14
KODAK	30	18
MCDON	14	26
NESTLE	22	14
NFLX	26	30
NINTENDO	62	14
NYT	18	42
PANASONIC	46	42
SAINSBURY	10	14
SHISEIDO	18	22
SUBARU	10	22
SUZUKI	22	66
TENCENT	46	30
TESCO	30	10
TESLA	30	42
TOYOTA	46	46
UBISOFT	26	22
WALMART	42	30
XEROX	18	18
YAMAHACO	22	14
YAMAHAMO	38	10

Figure 4 Tree of ActTrade for AAPL.



Sales ratio, Debt-to-Equity, SUMMafinn) appear frequently across datasets, their actual role within final trees varies, since they may be used in simple one-node tests or in deeper architectures. This suggests that indicator frequency alone does not dictate model complexity; rather, the combination of the FA, SA, and TA indicators contributes to robust and generalisable strategies.

5.10. Non-GP Benchmarks

5.10.1. ActTrade compared to the Machine Learning benchmarks

For further algorithmic benchmarks, aside of the GP variants, the built-in models of scikit-learn library are used in Python for the machine learning algorithms.

As shown in Table 21, the average values for MLP across the 42 companies were 0.30 for the Sharpe ratio, 0.009 for the rate of return, and 0.041 for risk.

These results contrast with the significantly higher values achieved by the GP algorithms, particularly in Sharpe ratio and risk. The KS statistical test revealed that MLP was statistically different from ActTrade, with p-values of $7.45e-10$ for the Sharpe ratio, 0.004 for the rate of return, and 0.0018 for risk, all well below the 5% significance level.

Similar trends were observed with the SVM algorithm, which had an average Sharpe ratio of 0.37, a rate of return of 0.01, and a risk value of 0.038. Again, significant differences were noted in the Sharpe ratio and risk. The KS tests indicated that SVM was statistically different from ActTrade, with p-values of $7.45e-10$ for the Sharpe ratio, 0.0018 for the rate of return, and 0.018 for risk, all falling below the 5% significance threshold.

For the XGBoost algorithm, the average values were 0.36 for the Sharpe ratio, 0.010 for the rate of return, and 0.041 for risk. ActTrade outperformed XGBoost in the financial metrics, and the KS tests confirmed that the novel GP algorithm was statistically different from XGBoost. The p-values were $3.89e-05$ for the Sharpe ratio, 0.0001 for the rate of return, and 0.041 for risk.

Finally, the LSTM algorithm had average values of 0.23 for the Sharpe ratio, 0.006 for the rate of return, and 0.043 for risk. ActTrade performed better across all three financial metrics, and the KS tests showed statistical differences between the novel GP algorithm and LSTM, with p-values of $3e-05$ for the Sharpe ratio, $1.13e-06$ for the rate of return, and 0.0040 for risk.

These results highlight the limitations of these four algorithms compared to GP algorithms in algorithmic trading, particularly when creating strate-

gies that account for both returns and risk.

Table 21 Average values for ActTrade, MLP, SVM, XGBoost, and LSTM

Algorithm	**Sharpe ratio**	**Rate of return**	**Risk**
MLP	0.30	0.009	0.041
SVM	0.37	0.010	0.038
XGBoost	0.36	0.010	0.041
LSTM	0.23	0.006	0.043
ActTrade	**3.9**	**0.0225**	**0.024**

5.10.2. ActTrade compared to financial benchmarks

In this section, we compare the ActTrade algorithm to the Buy and Hold (BnH) strategy. To ensure a fair comparison, we run ActTrade using maximisation of thecumulative returns as its fitness function, which measures the total profit or loss over a given period. We focus on cumulative returns rather than the Sharpe ratio because BnH involves a single trade—buying one unit of stock on the first day and selling it on the last day, without considering the rate of return and risk metrics that factor into the Sharpe ratio calculation.

When comparing ActTrade and BnH, the GP algorithm achieves an average cumulative return of 0.75 with a median of 0.91, whereas BnH records an average of 0.16 and a median of 0.169. The Kolmogorov-Smirnov test confirms that these results are statistically different at the 5% level, with a p-value of $3.93e-06$.

Furthermore, we compare ActTrade to the same trading strategy ActTrade uses when training.As noted in Section 3.2.2, the $TS_{d,r}$ strategy buys

one unit of stock on the first day of trading and sells it when the price increases by $r\%$ within d days or at the end of the d-day period.

When comparing ActTrade and TS$d, r$, we first observe that TS$d, r$ performs significantly more trades on average (240) compared to ActTrade, which averages only 19 trades. This is expected, as ActTrade focuses on the most profitable and low-risk opportunities, choosing not to trade in less favourable conditions. As a result, ActTrade shows superior performance across all three financial metrics, as seen in Table 22. The KS test confirms that the differences in Sharpe ratio and risk are statistically different at the 5% level, with p-values of $4.3e-13$ for the Sharpe ratio, $1.13e-6$ for the rate of return, and 0.00011 for risk.

Table 22 Average values of ActTrade and TS$d, r$.

Algorithm/Metric	***Sharpe ratio***	***Return***	***Risk***
TS$d, r$	0.10	0.006	0.067
ActTrade	**3.9**	**0.0225**	**0.024**

5.11. Real-world deployment and scalability

Real-time scalability and practical deployment considerations are crucial for the applicability of any algorithmic trading system. In this section, we discuss the computational resources and time requirements for both training and live execution of the proposed ActTrade GP algorithm.

The training phase of ActTrade is computationally intensive, involving the evolutionary process across multiple generations and numerous candidate solutions (GP individuals). This process typically takes several hours to

complete on standard computational hardware, depending on dataset size and complexity.

Table 23 presents the average training time per run, per company, in minutes for all algorithms tested in this paper. Note that the GP-based models are trained over 50 generations in one run to evolve full trading strategies, while our Machine Learning benchmarks (MLP, SVM, XGBoost, LSTM) are trained to perform binary classification. As such, the reported times reflect different computational objectives and are not directly comparable in terms of task complexity. Nevertheless, they provide a rough estimate of training cost across methods. Also, it needs to be noted that the computational times per algorithm were very close across all companies.

Table 23 Computational times per algorithm

Algorithm	**Value (in minutes)**
GP-FA	1.51
GP-SA	1.55
GP-TA	1.40
GP-FASATA	1.53
STGP-FASATA	1.36
STGP-FASATA-S	7.10
ActTrade	7.48
MLP	0.05
SVM	0.05
XGBoost	0.08
LSTM	0.07

Although GP-based models require longer training times compared to traditional machine learning models, they tackle a fundamentally more complex task: directly evolving complete trading strategies rather than classifying short-term stock movements. This increased complexity naturally results in longer training durations. However, in algorithmic trading practice, this training process typically occurs offline, while only the final evolved models are deployed in real-time. Once trained, the evolved GP trading rules consist of simple mathematical and logical expressions that can be executed extremely efficiently, requiring just milliseconds per decision. Therefore, despite their offline computational demands, GP-based models like ActTrade remain highly suitable for practical real-time trading scenarios.

Evaluating the GP strategies is computationally negligible, but efficient storage and streaming of real-time market and sentiment data are necessary to ensure timely decisions. The modular design of ActTrade makes it compatible with existing parallel data pipelines, allowing it to scale from a handful of assets to large portfolios in practical trading environments. We recognise the need for periodic retraining of the GP models to maintain high performance as market conditions evolve. In a live deployment, retraining could be conducted at intervals aligned with data availability and market dynamics, such as monthly or quarterly, to incorporate recent market behaviour and ensure continued adaptability and performance. This a common practice, aligning with previous research [55, 56], where it is found that constant retraining is not necessary and periodic retraining maintains accuracy while cutting costs and energy usage.

Furthermore, integrating ActTrade into a live trading pipeline would in-

volve continuously updating the input features (fundamental, sentiment, and technical indicators) as new data becomes available. For instance, fundamental and sentiment indicators can be refreshed daily or whenever new financial reports or news articles are published. Technical indicators can be updated at even higher frequencies if intraday data is utilised. Given the low latency of evaluating the GP-generated rules, trading decisions can be generated almost instantaneously upon receipt of updated input data.

From an engineering perspective, scaling ActTrade for real-time operation could be effectively managed using parallel computing. The inherent parallelism in GP evaluation (where multiple GP individuals or trading rules can be evaluated simultaneously) can be leveraged to ensure scalability across large datasets or multiple trading assets (e.g., see [57]).

In addition to computational scalability, practical deployment also requires consideration of execution latency, transaction costs, and market constraints. The latency requirements of ActTrade depend on the trading style as mentioned above. Since the evolved rules consist of lightweight mathematical expressions, ActTrade can satisfy these low-latency constraints with ease, since its usage is to already create trading strategies and does not require daily training.

Overall, ActTrade is designed to accommodate real-world trading requirements efficiently. With proper engineering and system design, our approach is well-suited for practical deployment in real-time algorithmic trading environments.

5.12. Summary of findings

To sum up, as seen in Tables 7 - 22 and focusing on the findings from ActTrade, the results have been summarised in 2 categories: first on GP variants' results and, second, on the results of other benchmarks.

When comparing the GP variants with each other, it is evident that:

- The proposed ActTrade algorithm demonstrates strong performance across all three financial metrics—Sharpe ratio, rate of return, and risk—while also statistically and significantly outperforming other GP variants.

- Developing trading strategies that incorporate information from all three indicator types (as in the strongly-typed GP variants) results in more profitable and lower-risk strategies compared to those that rely on a single indicator type (e.g., GP-FA, GP-SA, GP-TA).

- The strongly-typed architecture of the proposed algorithm enables the GP to more effectively explore the space of FA, SA, and TA indicators, compared to a GP that includes all indicator types in its terminal set without the strongly-typed structure (GP-FASATA). This enables a more structured and focused exploration of the indicator space, balancing exploration with exploitation.

- The proposed genetic operator has the ability to create strong offspring (in terms of Sharpe ratio), which often outperform offspring derived by tournament selection. This novel operator supports optimisation by actively generating high-quality solutions that maintain a favourable risk-return balance.

Regarding the comparison of ActTrade to the non-GP benchmarks, we observe the following:

- ActTrade outperforms the four algorithmic benchmarks, with statistically significant results. The higher risk associated with the machine learning benchmarks is reflected in their low Sharpe ratio values, all below 0.4, while the novel GP algorithm achieves a rate of return that is twice that of the four algorithmic benchmarks.

- The ActTrade algorithm is more financially advantageous compared to both the Buy and Hold strategy and the TS$d, r$ trading strategy. It outperforms the Buy and Hold strategy in cumulative returns and surpasses TS$d, r$ across all financial metrics. These results underscore the robustness of the ActTrade algorithm, particularly in its ability to manage uncertainty and adapt to the dynamic conditions of financial markets.

6. Conclusion

In conclusion, this article introduced a novel strongly-typed GP algorithm that integrates fundamental, sentiment, and technical analysis indicators. The algorithm also featured an innovative fitness function that rewards both individual subtrees and the overall tree. Additionally, a new genetic operator was proposed to promote active trading. This novel approach aligns with key soft computing principles, offering a flexible, adaptive solution to financial trading challenges by managing uncertainty and exploiting the search space more effectively.

The importance of combining the three analysis indicator types, which is not usually occurring in the previous studies, is evident from our findings. Combining the indicators enhances the models' knowledge and create financially more advantageous trading strategies. Moreover, based on our analysis, it is not profitable enough to simply combine the different types of indicators, as GP-FASATA does, and a strongly-typed architecture is essential towards achieving an improved performance. Lastly, the use of the novel fitness function and the novel genetic operator leads to significantly improved results across all three metrics of Sharpe ratio, rate of return, and risk.

While our findings are promising, it is important to acknowledge the limitations of this study. Our experiments were conducted on historical data and a fixed set of international stocks, and as such, real-world performance may vary under different market conditions or asset classes. Additionally, the model's flexibility and customisability come with added structural complexity, which can pose challenges for interpretability, although we have taken steps to address this through explicit design choices and indicator usage analysis.As a result, there are several research avenues for future work. For example, one possible direction is the incorporation of more advanced sentiment analysis techniques. While we employed established and interpretable tools in this study, more sophisticated approaches – including neuromorphic models (e.g., spiking neural networks) and state-of-the-art pre-trained language models such as BERT [58] or VADER – may offer richer and more human-like sentiment interpretation. Integrating these methods into our framework could potentially improve the expressiveness and nuance of the sentiment signals used in trading decisions.Furthermore, we wish to explore the use of

alternative sentiment data sources, such as specialist news sentiment APIs. While our current approach relies on Google News for their broad coverage and consistency across companies, incorporating more domain-specific or real-time sentiment sources may further enhance the model's effectiveness and robustness, as well as adjusting our current daily trading framework to high-frequency intraday time frames.Moreover, incorporating explicit modelling of market microstructure effects, including liquidity constraints and dynamic adaptation of strategies over time, could further validate robustness in realistic deployment scenarios.As the current study is deliberately designed at the single-stock level to isolate and evaluate the effect of integrating FA, SA, and TA on individual strategy performance we plan to extend this to a portfolio-level analysis.To further strengthen model robustness, future work could explore the use of ensemble techniques, such as combining multiple top-performing GP individuals from different runs. This may help reduce variance in outcomes, albeit at the cost of model simplicity. Another potential enhancement is the application of time-based cross-validation strategies, such as rolling window evaluation, to assess generalisation across different market regimes. While these approaches were beyond the scope of the present work due to computational complexity, they represent promising directions to improve stability and reliability in real-world trading scenarios.

Another limitation of our work is that our current methodology relies on a single, chronologically ordered train/validation/test split. This ordering prevents the model from training on future information, but it leaves two distinct issues unaddressed. First, a single split yields one performance estimate rather than a distribution, so it lacks the statistical guarantees that

purged/embargoed cross-validation would provide against overfitting to a particular split. Second, while we apply Benjamini–Hochberg FDR correction to control for multiplicity across our pairwise KS tests, this does not correct for the prior selection of ActTrade as the control on the basis of its superior average/median performance — a form of selection bias that a fixed set of pairwise comparisons cannot address, and one that grows more severe as more strategies are compared. Addressing the first would require combinatorial purged cross-validation with train-only preprocessing pipelines; addressing the second would require selection-aware significance testing such as Model Confidence Set or Deflated Sharpe Ratio, which are designed specifically to correct for the bias introduced when a "best" strategy is chosen from a larger candidate set. We treat both as necessary next steps before claiming statistically robust outperformance.

Lastly, it is worth noting our proposed algorithm is not limited to equity-specific features. Hence, it could be applied to other equity types, such as commodities and crypto currencies, provided that appropriate indicators are available or defined.

References

[1] M. H. Eng, Y. Li, Q.-G. Wang, T. H. Lee, Forecast forex with artificial neural networks using fundamental data, in: 2008 International Conference on Information Management, Innovation Management and Industrial Engineering, Vol. 1, IEEE, 2008, pp. 279–282.

[2] M. Abe, H. Nakayama, Deep learning for forecasting stock returns in

the cross-section, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2018, pp. 273–284.

[3] Q. He, Z. Cai, W. Liu, Prediction of listed companies' revenue based on model-fused, in: 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), IEEE, 2019, pp. 257–260.

[4] A. Herawati, A. S. Putra, et al., The influence of fundamental analysis on stock prices: The case of food and beverage industries, European Research Studies Journal 21 (3) (2018) 316–326.

[5] C. Ajekwe, A. Ibiamke, Financial statement analysis to predict stock returns of listed consumer goods firms in nigeria, Available at SSRN 3247805 (2018).

[6] K. Kohara, T. Ishikawa, Y. Fukuhara, Y. Nakamura, Stock price prediction using prior knowledge and neural networks, Intelligent Systems in Accounting, Finance & Management 6 (1) (1997) 11–22.

[7] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, in: Twenty-fourth international joint conference on artificial intelligence, 2015.

[8] B. Xie, R. Passonneau, L. Wu, G. G. Creamer, Semantic frames to predict stock price movement, in: Proceedings of the 51st annual meeting of the association for computational linguistics, 2013, pp. 873–883.

[9] M.-Y. Day, C.-C. Lee, Deep learning for financial sentiment analysis

on finance news providers, in: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2016, pp. 1127–1134.

[10] R. Gupta, M. Chen, Sentiment analysis for stock price prediction, in: 2020 IEEE conference on multimedia information processing and retrieval (MIPR), IEEE, 2020, pp. 213–218.

[11] C. Qian, N. Mathur, N. H. Zakaria, R. Arora, V. Gupta, M. Ali, Understanding public opinions on social media for financial sentiment analysis using ai-based techniques, Information Processing & Management 59 (6) (2022) 103098.

[12] H. Yun, G. Sim, J. Seok, Stock prices prediction using the title of newspaper articles with korean natural language processing, in: 2019 international conference on artificial intelligence in information and communication (ICAIIC), IEEE, 2019, pp. 019–021.

[13] T. Marty, B. Vanstone, T. Hahn, News media analytics in finance: a survey, Accounting & Finance 60 (2) (2020) 1385–1434.

[14] B. Hasselgren, C. Chrysoulas, N. Pitropakis, W. J. Buchanan, Using social media & sentiment analysis to make investment decisions, Future Internet 15 (1) (2023) 5.

[15] M. Costola, O. Hinz, M. Nofer, L. Pelizzon, Machine learning sentiment analysis, covid-19 news and stock market reactions, Research in International Business and Finance (2023) 101881.

[16] R. Hochreiter, Computing trading strategies based on financial sentiment data using evolutionary optimization, in: Mendel 2015: Recent Advances in Soft Computing, Springer, 2015, pp. 181–191.

[17] S. Y. Yang, S. Y. K. Mo, A. Liu, A. A. Kirilenko, Genetic programming optimization for a sentiment feedback strength based trading strategy, Neurocomputing 264 (2017) 29–41.

[18] E. Christodoulaki, M. Kampouridis, P. Kanellopoulos, Technical and sentiment analysis in financial forecasting with genetic programming, in: IEEE Symposium on Computational Intelligence for Financial Engineering & Economics (CIFEr), 2022.

[19] M. M. Mostafa, Forecasting stock exchange movements using neural networks: Empirical evidence from kuwait, Expert Systems with Applications 37 (9) (2010) 6302–6309.

[20] D. M. Nelson, A. C. Pereira, R. A. de Oliveira, Stock market's price movement prediction with long short-term memory neural networks, in: 2017 International joint conference on neural networks (IJCNN), IEEE, 2017, pp. 1419–1426.

[21] A. F. Kamara, E. Chen, Z. Pan, An ensemble of a boosted hybrid of deep learning models and technical analysis for forecasting stock prices, Information Sciences 594 (2022) 1–19.

[22] J. Li, E. P. Tsang, Improving technical analysis predictions: An application of genetic programming., in: flairs Conference, 1999, pp. 108–112.

[23] M. Kampouridis, E. Tsang, Investment opportunities forecasting: Extending the grammar of a genetic programming based tool, International Journal of Computational Intelligence Systems 5 (3) (2012) 530–541.

[24] M. Kampouridis, A. Alsheddy, E. Tsang, On the investigation of hyper-heuristics on a financial forecasting problem, Annals of Mathematics and Artificial Intelligence 68 (2013) 225–246.

[25] M. Kampouridis, F. E. Otero, Heuristic procedures for improving the predictability of a genetic programming financial forecasting algorithm, Soft Computing 21 (2) (2017) 295–310.

[26] J. L. Bettman, S. J. Sault, E. L. Schultz, Fundamental and technical analysis: substitutes or complements?, Accounting & Finance 49 (1) (2009) 21–36.

[27] E. Beyaz, F. Tekiner, X.-j. Zeng, J. Keane, Comparing technical and fundamental indicators in stock price forecasting, in: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2018, pp. 1607–1613.

[28] A. Pornwattanavichai, S. Maneeroj, S. Boonsiri, Bertforex: Cascading model for forex market forecasting using fundamental and technical indicator data based on bert, IEEE Access 10 (2022) 23425–23437.

[29] M. Azimi, A. Agrawal, Is positive sentiment in corporate annual reports

informative? evidence from deep learning, Evidence from Deep Learning (July 2019) (2019).

[30] M. R. Vargas, B. S. De Lima, A. G. Evsukoff, Deep learning for stock market prediction from financial news articles, in: 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), IEEE, 2017, pp. 60–65.

[31] A. Nan, A. Perumal, O. R. Zaiane, Sentiment and knowledge based algorithmic trading with deep reinforcement learning, arXiv preprint arXiv:2001.09403 (2020).

[32] E. Christodoulaki, M. Kampouridis, Combining technical and sentiment analysis under a genetic programming algorithm, in: UK Workshop of Computational Intelligence (UKCI), 2022.

[33] E. Christodoulaki, M. J. Kampouridis, Using strongly typed genetic programming to combine technical and sentiment analysis for algorithmic trading, in: 2022 IEEE Congress on Evolutionary Computation (CEC), 2022, pp. 1–8.

[34] B. Weng, M. A. Ahmed, F. M. Megahed, Stock market one-day ahead movement prediction using disparate data sources, Expert Systems with Applications 79 (2017) 153–163.

[35] A. Picasso, S. Merello, Y. Ma, L. Oneto, E. Cambria, Technical analysis and sentiment embeddings for market trend prediction, Expert Systems with Applications 135 (2019) 60–70.

[36] J. Min, Financial market trend forecasting and performance analysis using long short-term memory, arXiv preprint arXiv:2004.01502 (2020).

[37] G. Mu, N. Gao, Y. Wang, L. Dai, A stock price prediction model based on investor sentiment and optimized deep learning, Ieee Access 11 (2023) 51353–51367.

[38] J.-Y. Huang, C.-L. Tung, W.-Z. Lin, Using social network sentiment analysis and genetic algorithm to improve the stock prediction accuracy of the deep learning-based approach, International Journal of Computational Intelligence Systems 16 (1) (2023) 93.

[39] A. L. Awad, S. M. Elkaffas, M. W. Fakhr, Stock market prediction using deep reinforcement learning, Applied System Innovation 6 (6) (2023) 106.

[40] G.-M. Chatziloizos, D. Gunopulos, K. Konstantinou, Deep learning for stock market prediction using sentiment and technical analysis, SN Computer Science 5 (5) (2024) 446.

[41] E. Christodoulaki, M. Kampouridis, M. Kyropoulou, A novel strongly-typed genetic programming algorithm for combining sentiment and technical analysis for algorithmic trading, Knowledge-Based Systems (2025) 113054.

[42] E. Christodoulaki, M. Kampouridis, Fundamental, technical and sentiment analysis for algorithmic trading with genetic programming, in: 2023 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2023, pp. 83–89.

[43] F. Dakalbab, A. Kumar, M. A. Talib, Q. Nasir, Advancing forex prediction through multimodal text-driven model and attention mechanisms, Intelligent Systems with Applications 26 (2025) 200518.

[44] P. Gangwani, V. Panthi, Leveraging multimodal data and deep learning for enhanced stock market prediction, AI-Based Advanced Optimization Techniques for Edge Computing (2025) 93–127.

[45] M. A. Arauco Ballesteros, E. A. Martínez Miranda, Stock market forecasting using a neural network through fundamental indicators, technical indicators and market sentiment analysis, Computational Economics 66 (2) (2025) 1715–1745.

[46] A. Damodaran, The little book of valuation: how to value a company, pick a stock and profit, Vol. 34, John Wiley & Sons, 2011.

[47] J. Deveikyte, H. Geman, C. Piccari, A. Provetti, A sentiment analysis approach to the prediction of market volatility, Frontiers in Artificial Intelligence 5 (2022) 836809.

[48] M. S. Azuero Melo, Topic modeling and sentiment analysis for predicting the direction of movement of the s&p 500 index based on financial news/author miguel santiago azuero melo, bsc (2024).

[49] A. Brabazon, M. Kampouridis, M. O'Neill, Applications of genetic programming to finance and economics: past, present, future, Genetic Programming and Evolvable Machines 21 (1) (2020) 33–53.

[50] P. Fernandez, E. de Apellániz, J. F Acín, Survey: Market risk premium and risk-free rate used for 81 countries in 2020 (2020).

[51] X. Long, M. Kampouridis, P. Kanellopoulos, Genetic programming for combining directional changes indicators in international stock markets, in: International Conference on Parallel Problem Solving from Nature, Springer, 2022, pp. 33–47.

[52] S. Abolmakarem, F. Abdi, K. Khalili-Damghani, H. Didehkhani, A multi-stage machine learning approach for stock price prediction: Engineered and derivative indices, Intelligent Systems with Applications 24 (2024) 200449.

[53] R. Menoita, S. Silva, Evolving financial trading strategies with vectorial genetic programming, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2025, pp. 875–878.

[54] W. F. Sharpe, The Sharpe ratio, Streetwise–the Best of the Journal of Portfolio Management 3 (1998) 169–185.

[55] B. Bieganowski, R. Ślepaczuk, Supervised autoencoder mlp for financial time series forecasting, Journal of Big Data 12 (1) (2025) 207.

[56] M. Zanotti, Do global forecasting models require frequent retraining?, arXiv preprint arXiv:2505.00356 (2025).

[57] J. Brookhouse, F. E. Otero, M. Kampouridis, Working with opencl to speed up a genetic programming financial forecasting algorithm: Initial results., in: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, 2014, pp. 1117–1124.

[58] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

Appendix A. Hyperparameter search spaces

Table A.24 reports the search space explored during the two-step grid search described in Section 4.3 for the GP-based algorithms. Table A.25 reports the corresponding search space for the machine learning benchmarks (MLP, SVM, XGBoost, LSTM). In both cases, the final selected values used in the reported experiments are given in Table 5 (GP) and Section 4.3 (ML benchmarks); the ranges below indicate the full space considered during tuning, not only the selected configuration.

Table A.24 Hyperparameter search space for the GP algorithms.

Parameter	**Search space**
Population size	[100, 300, 500, 1000, 2000]
Crossover probability (p_c)	[0.7, 0.9, 0.95]
Number of generations	[20, 30, 50]
Tournament size	[2 – 5]
Maximum tree depth	[4 – 7]
Tournament probability (p_1)	0.3, 0.4, 0.5
Active-trading offspring probabilities (p_2, p_3, p_4)	(0.10, 0.15, 0.20)

Table A.25 Hyperparameter search space for the machine learning benchmarks.

Algorithm	**Parameter**	**Search space**
MLP	Hidden layer size(s)	[1–4 layers]
	Learning rate	[0.0001–0.01]
	Activation function	[ReLU, tanh, logistic]
SVM	Kernel	[linear, RBF, polynomial]
	C (regularisation)	[0.01–100]
	γ (RBF kernel)	[0.0001–1, 'auto']
XGBoost	Max tree depth	[3–10]
	Learning rate	[0.01–0.3]
	Number of estimators	[50–500]
LSTM	Layers / units	[1–4 layers]
	Learning rate	[0.0001–0.01]
	Batch size	[32–128]