A novel strongly-typed Genetic Programming algorithm for combining sentiment and technical analysis for algorithmic trading

Eva Christodoulaki[a,], Michael Kampouridis[b], Maria Kyropoulou[b]

[a]*School of Engineering Mathematics and Technology, University of Bristol, Ada Lovelace Building, Bristol, BS8 1TW, United Kingdom*
[b]*School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom*

Abstract

The use of algorithms in finance and trading has become an increasingly thriving research area, with researchers creating automated and pre programmed trading instructions utilising indicators from technical and sentiment analysis. The indicators of the two analyses have been used mostly individually, despite evidence that their combination can be profitable and financially advantageous. In this paper, we examine the advantages of combining indicators from both technical and sentiment analysis through a novel genetic programming algorithm, named STGP-SATA. Our algorithm introduces technical and sentiment analysis types, through a strongly-typed architecture, whereby the associated tree contains one branch with only technical indicators and another branch with only sentiment analysis indicators. This approach allows for better exploration and exploitation of the search space of the indicators. To evaluate the performance of STGP-SATA we compare it with three other GP variants on three financial metrics, namely Sharpe ratio, rate of return and risk. We furthermore compare STGP-SATA against two financial and four algorithmic benchmarks, namely, multilayer perceptron, support vector machine, extreme gradient boosting, and long short term memory network. Our study shows that the combination of technical and sentiment analysis indicators through STGP-SATA improves the finan-

Email addresses: eva.christodoulaki@britol.ac.uk (Eva Christodoulaki), mkampo@essex.ac.uk (Michael Kampouridis), maria.kyropoulou@essex.ac.uk (Maria Kyropoulou)

cial performance of the trading strategies and statistically and significantly outperforms the other benchmarks across the three financial metrics.

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

1. Introduction

Algorithmic trading involves the use of pre-programmed trading strategies to execute orders and generate profits. This practice has been employed in trading for many years and continues to gain popularity, particularly as more services and companies become available for trading. The topic of algorithmic trading is of interest to researchers who are exploring the potential of Machine Learning (ML) implementations to maximise returns and minimise risk. ML algorithms examine historical information of the stock market and identify patterns, learning how certain indicators are associated with certain trends. Then, when they recognise such a pattern, the algorithms generate signals indicating an upcoming change in trend, which can be used to generate profit.

Technical analysis is a financial technique that uses price trends and patterns to identify trading opportunities. Sentiment analysis corresponds to recognising events relevant to stocks, identifying their importance towards influencing their price and using that for predicting stock prices. Researchers have mainly utilised Technical Analysis (TA) indicators, such as volatility and moving average, for algorithmic trading, but sentiment analysis (SA) indicators, such as sentiment polarity, have also been successfully considered in the more recent years. The benefits observed by the two individual analyses have now brought about the promise of achieving an improved performance by their combination. Indeed, [1] and [2], very recently provided initial evidence supporting this promise, by creating financially advantageous trading strategies utilising both analysis types.

In our approach, we aim to integrate both TA and SA indicators within genetic programming (GP) algorithms. By combining these two types of indicators, we seek to improve the accuracy and effectiveness of our trading strategies. The reason for using GP algorithms is due to the immense number of potential trading strategies that can be created. GP algorithms have been shown to be effective in evolving profitable trading rules that can adapt to changing market conditions [3]. Another advantage of GP algorithms is their ability to efficiently search the vast solution space and generate domain-

specific solutions/strategies. By incorporating both TA and SA indicators into them, we can create more sophisticated and robust trading strategies that can adjust to market conditions in real-time. As a result, we believe this approach has the potential to lead to improved trading outcomes and higher profits for investors.

Our proposed algorithm STGP-SATA, which was first introduced in [2], uses a strongly-typed GP architecture, where TA and SA indicators are handled in separate parts of the model (subtrees/branches of the tree). This has several advantages. Firstly, it allows the algorithm to focus on the search space of each individual indicator type and encourages better exploration and exploitation of the solution space of each indicator. This is achieved by combining the two types of indicators at the root of the tree with an `AND` function. The primary motivation behind the design of the STGP-SATA algorithm is to ensure that both technical and sentiment analysis indicators are given due consideration, effectively creating diverse and effective trading strategies. In addition, STGP-SATA enables the creation of more adaptable strategies that can adjust to changing market conditions by deciding whether to give more weight on TA or SA indicators, and subsequently maximise profits for investors. While a non-strongly-typed GP with both TA and SA indicators allows for complex interactions, STGP-SATA is more advantageous. It ensures a balanced representation, preventing one indicator type from dominating and enabling focused exploration of solutions, leading to more effective combinations.

Our current article extends our previous work in the following six ways: (i) we present a more in-depth presentation of the STGP-SATA algorithm, (ii) we increase the number of companies we use in our experiments from 10 to 60, (iii) we increase the number of benchmarks from 3 to 9, as we include both financial and machine learning benchmarks, (iv) we discuss not only average results, but also results of the best trading strategy, which offers a realistic case study, as in the real-world a single (the best) trading strategy is used, and finally, (v) we analyse the results depending on the type of market (e.g. uptrend vs downtrend market).

The main objective of our research is to demonstrate the effectiveness of combining TA and SA indicators and incorporating them into the terminal set of a strongly-typed GP algorithm for creating financially advantageous trading strategies. Ultimately, our research aims to contribute to the development of more effective algorithmic trading strategies that can generate profits and minimise risk for investors in the financial market.

The remainder of this paper is structured as follows. Section 2 provides an overview of previous research work related to the use of TA and SA indicators in algorithmic trading. In Section 3, we describe the methodology used in our research, including the data preparation, indicator engineering, and the implementation of the GP algorithms. Section 4 presents the details of the experimental setup, including the dataset used, performance metrics, and the different benchmarks. The results and analysis of the study are presented in Section 5, where we compare the performance of the different GP algorithms and evaluate the effectiveness of combining TA and SA indicators. Finally, in Section 6, we conclude the paper by summarising the main findings of our research and discussing potential avenues for future work.

2. Literature Review

This section aims to examine prior research on financial forecasting and algorithmic trading, with particular emphasis on studies utilising technical and sentiment analysis indicators. Several of these papers also incorporate both technical and sentiment analysis indicators.

2.1. Technical analysis

Technical analysis is a financial method that employs price patterns and trends to identify trading opportunities. The indicators derived from technical analysis have been utilised as inputs to machine learning algorithms for many years. Since the 1980*s*, numerous studies have used artificial neural networks for financial forecasting, and subsequently for algorithmic trading.

Some studies include [4], which used technical analysis indicators with linear models, and [5], which utilised a long short-term memory (LSTM) model to predict future trends of stock prices. In [6] the authors created a hybrid deep learning model and used TA for financial forecasting, using two stocks in their experiments. [7] demonstrated the use of meta-synthesis techniques to identify optimisation components within financial systems, highlighting the importance of blending traditional financial metrics with AI methods to optimise decision-making. Moreover, the study of [8] explored the role of technical indicators in improving the accuracy of option price predictions using deep learning models, showcasing improvements in predictive accuracy.

One of the first papers to incorporate technical analysis indicators for financial forecasting using genetic programming (GP) is [9], where the algorithm outperformed commonly used, non-adaptive technical rules. Over the

last decade, several studies have reported similar outcomes, such as [10], [11], and [12]. Following, [13] used Genetic Algorithms (GA) to optimise technical trading strategies, highlighting the ability of evolutionary algorithms to identify market inefficiencies and improve profitability. Continuing, [14] proposed self-adaptive Evolutionary algorithms for stock prediction and portfolio composition, demonstrating higher Sharpe ratios and reduced risk. Similarly, [15] applied genetic programming combined with directional change and technical analysis indicators, using a multi-objective optimisation approach. The authors achieved to improve returns while balancing risk, showcasing the effectiveness of evolutionary methods in trading strategy development.

As demonstrated in [16] and [3], GP algorithms can develop trading strategies, generate solutions that survive extreme market conditions, and create new solutions while optimising the solution parameters.

2.2. Sentiment analysis

Algorithmic trading is a complex topic with numerous variables to consider, and further research may be required to include additional indicators. One approach is to determine the significance of events and how they affect the stock market. One of the most influential studies on sentiment analysis is Kohara et al.'s [17] research, which used neural networks to investigate how prior knowledge from newspaper headlines could enhance the accuracy of prediction in multivariate models. Similarly, researchers have investigated the importance of sentiment analysis for financial forecasting, investment decisions and trading, i.e. [18], [19], [20], [21].

A substantial contribution to the literature is the work of Xie et al. [22], who used support vector machines (SVM) with tree kernels and semantic frame parses to generalise from sentences to scenarios. Ding et al. [23] created an event-driven stock model by feeding news into a deep convolutional neural network (CNN), and Day et al. [24] considered the source of the sentiment by attempting to assess the quality of the news and its impact on stock movement. [25] performed sentiment analysis on tweets and [26] segregated tweets on non-fungible tokens (NFTs) using Pearson Product-Moment Correlation Coefficient (PPMCC) and studied 8-scale emotions, along with Positive and Negative sentiments. Following, [27] presented a framework combining sentiment analysis with graph neural networks (GNNs) for predicting stock price movements, using relational data between stocks and social media sentiment to model the interdependence of market dynamics and

investor sentiment. The study of [28] demonstrated the benefits of integrating technical analysis, fundamental indicators, and market sentiment into a multilayer perceptron neural network for stock market forecasting. The authors concluding that the inclusion of sentiment analysis improves model accuracy for a significant portion of the S&P 500 companies. Furthermore, [29] introduced neutrosophic logic for sentiment analysis to address uncertainty in social media data, combining the improved sentiment results with LSTM for stock movement prediction. Similarly, [30] analysed news sentiment and combined it with LSTM for stock price prediction, demonstrating better performance compared to traditional models.

Finally, [31], [32], and [1] and [33] are the only known studies thus far to use sentiment analysis indicators as inputs to a GP algorithm for algorithmic trading. Although the studies differ in their implementation and trading strategies, both were successful in demonstrating the financial profitability of sentiment analysis.

2.3. Technical and Sentiment analysis combination

Researchers have been exploring the combination of technical and sentiment analysis for financial forecasting, with promising results in terms of profitability. In [34], an external knowledge base was used to detect events based on reasoning, combining event knowledge and standard information of companies. Similarly, [35] used deep neural networks (DNNs) to predict stock price movements by combining historical prices and online financial news. [36] utilised text mining on news from Reuters regarding the S&P500 index in a hybrid model of recurrent neural networks (RNNs) and convolutional neural networks (CNNs), incorporating financial news articles and technical indicators as inputs. The model outperformed CNN in the same implementation and demonstrated the usefulness of both technical and sentiment analysis in financial forecasting. Nan et al. [37] developed a reinforcement learning approach that utilised traditional time series stock price data and news headline sentiments while leveraging knowledge graphs to exploit news about implicit relationships. Their study showed that the trained reinforcement learning agent resulted in better profits when the additional information on headline sentiments was included. The combination of technical and sentiment analysis has shown promising results in improving the accuracy and profitability of financial forecasting models, indicating the potential for further research in this area. Moreover, [38] used Genetic Algorithm hybrid models for portfolio optimisation, using tweets and the United States stock,

achieving better performance in terms of common measures of portfolio performance including Sharpe ratio, cumulative returns, and value-at-risk. In their work, the authors of [39] used stock historical data, technical indicators and financial news to calculate the investors' sentiment index, while they combined the data types and adopted a LSTM network for predicting the China Shanghai A-share market. Similarly, the study of [40] combined LSTM with investor sentiment from social media and technical indicators, demonstrating that sentiment analysis improves predictive accuracy. [41] showcased that GA for feature selection with LSTM improves prediction accuracy using sentiment and technical indicators. Moreover, [42] combined sentiment and technical analysis using the LASSO algorithm to eliminate multicollinearity among variables, achieving an 8.53% improvement compared to standard LSTM methods. In addition, [43] demonstrated the importance of combining indicators from different analysis types, but this time using Deep Reinforcement Learning, optimising trading strategies effectively. A weighted linear equation is used by [44] to integrate both sentiment and technical analysis, highlighting the effectiveness of their combination in market predictions. Similarly, the authors of [45] achieved to show the same by using deep learning to predict stock prices. They combined economic and sentimental data, demonstrating their advantages in predicting performance of stock market strategies. [46] used ensemble learning approaches showcased that the combination of sentiment and technical analysis and its important in identifying stocks with growth potential.

In their work, the authors of [47] proposed a hybrid learning-based model for sentiment and technical analysis, which utilises a three-stage method to determine the final trend prediction based on two intermediate predictions. Following, in [48], the authors studied the correlation between news sentiment indices, technical analysis, and the US stock market using econometric models, revealing significant linkages. Similarly, the authors of [49] proposed the creation of TA's sentiment in the stock market index by aggregating signals from over $2,000$ trading rules and demonstrates its strong correlation with traditional sentiment measures. The TA sentiment index effectively predicts short-term market momentum and long-term reversals. In their study, [50] used machine learning classifiers, highlighting the integration of both sentiment and technical analysis indicators, as well as macroeconomic data for predictions. Unlike strongly-typed genetic programming (STGP) algorithm approaches, the above studies do not ensure logical consistency between TA and SA, as they mainly focus on optimising neural network architectures.

While effective, these methods may not be as easy to interpret, and mostly used to predict future values, rather than optimise specific financial metrics.

In Genetic Programming, there are studies that combine technical and sentiment analysis indicators for algorithmic trading, such as [32] and [2]. In [32], the authors found that using news and Twitter for sentiment analysis is more financially profitable than performing technical analysis alone or combining TA and SA indicators while considering a simple GP architecture. However, [2] showed that the combination of technical and sentiment analysis indicators under a strongly-typed GP is more financially profitable.

While significant progress has been made in integrating TA and SA indicators for stock market prediction and algorithmic trading, several limitations persist. One such limitation is interpretability, many machine learning models, such as LSTMs, can achieve high predictive accuracy but lack interpretability, making it difficult to understand how sentiment and technical indicators contribute to predictions. Another limitation is the imbalanced use of indicator types, where one type may dominate the analysis, reducing the effectiveness of a truly hybrid approach. Furthermore, scalability and real-time application is something not easily addressed, with the latter being underexplored. This is because it is not easy to gather and use clean high-frequency textual data, as opposed to historical stock market data. Moreover, many studies are tailored to specific markets or asset types, therefore these methods may not be well applied across different financial environments. Differences in sentiment across regions or industries may introduce biases that are challenging to generalise or adapt to without extensive preprocessing and normalisation. Finally, the dynamic nature of financial markets poses challenges for model longevity. Models trained on specific market conditions may struggle to adapt to sudden shifts, such as economic crises, geopolitical events, or rapid technological changes. Such limitations motivate the need for novel methodologies that address the trade-offs between interpretability, the combination of different data types, and the scalability of such applications.

Genetic programming has several advantages to address these limitations, including white-box models, effective global search, and good exploration and exploitation, which make it a promising approach for combining TA and SA indicators. The proposed approach has the potential to generate more effective and profitable algorithmic trading strategies that can take advantage of both TA and SA indicators. By using a strongly-typed GP, we can ensure that each individual always contains dedicated TA and SA nodes, which can prevent the search from focusing on one type only, and improve the accuracy

and robustness of our trading strategies, leading to better financial returns.

3. Methodology

Our research methodology consists of four parts. Section 3.1 provides an overview of the GP methodology, which includes the model representation and GP operators. In Section 3.2, we introduce the two types of analysis, namely technical analysis and sentiment analysis, and discuss the relevant indicators that we will consider in our study. Section 3.3 discusses the trading signals and trading strategy we implement, while Section 3.4 presents the fitness function and metrics that will be considered and how we evaluate the performance of the algorithm. The experiments were conducted on a high-performance computing cluster consisting of 30 compute nodes and 13 GPU nodes, equipped with between 1 and 4 NVIDIA GPUs per node. This enabled efficient handling of the computational demands of processing multiple datasets, each undergoing 50 generations of evolutionary runs.

3.1. Genetic programming and the STGP-SATA algorithm

Genetic programming (GP) algorithms are evolutionary algorithms inspired by natural selection, where in this study, candidate solutions are represented as tree structures. These trees evolve over generations through genetic operators, such as subtree crossover and point mutation that we utilise in our paper. Subtree crossover swaps subtrees between parent individuals, combining their characteristics to create offspring. Point mutation alters specific nodes within a tree, introducing variability and ensuring a diverse exploration of the solution space. These operations, combined with selection mechanisms, drive the evolution of increasingly optimal solutions.

In a strongly-typed GP architecture, each node in the tree structure is explicitly typed, ensuring that operations only occur between compatible data types. This prevents logical inconsistencies, such as mixing indicators of unrelated types, thereby improving the reliability of the evolved strategies.

The balanced strongly-typed approach allows the algorithm to not rely on one type of analysis too much, thus, reducing the possibility of creating strategies that can underperform by using an indicator type more than other. This can lead to more consistent financial returns and better performing trading strategies.

3.1.1. Model representation

Part 1 of Figure 1 shows a sample tree that the STGP-SATA algorithm can create. The strongly-typed architecture of our algorithm enforces that the root will have two children, one allowing only SA indicators and the other allowing only TA indicators. The root is always an `AND` function that unites the two branches; the first branch of the `AND` function is forced to be SA-related and the second branch is forced to be TA-related.

The function nodes are based on the logical functions `AND`, `OR`, Greater than (`GT`) and Less than (`LT`), with different variants allowing for different indicators. In particular, our algorithm uses $\texttt{AND}_{\texttt{SA}}$, $\texttt{OR}_{\texttt{SA}}$, $\texttt{GT}_{\texttt{SA}}$, $\texttt{LT}_{\texttt{SA}}$ 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 function set is summarised in Table 1. This ensures that the algorithm generates models that fully utilise both types of indicators, enforcing type consistency, which helps prevent errors and enhances the exploration of the search space.

Table 1: Function set for the STGP-SATA algorithm.

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

With respect to the terminal sets, different sets are allowed at different branches of the corresponding trees. The terminal sets for the TA (Table 2) and SA branches (Table 3) of the tree refer to the specific indicators or variables that are allowed to be used in each branch, and in addition to those specific indicators, both terminal sets also include a random variable called Ephemeral Random Constant (ERC) that acts as a threshold value to the indicators and consists of random values between -1 and 1.

Compared to a non-strongly-typed GP, our proposed algorithm can fully take advantage of the search space of each individual indicator type. The two branches corresponding to the two indicator types are united using the

AND function at the root of the tree, and this creates the foundation for better exploration and exploitation. Thus, the model can create more diverse, effective and adaptable trading strategies.

3.1.2. GP operators

We incorporate the *subtree crossover* and *point mutation* operators in our research. When performing subtree crossover in the case of strongly-typed GP algorithms, we exchange corresponding parts from both the left (SA) and the right (TA) subtree of the model. The nodes being exchanged must be of the same type (e.g., a terminal node with another terminal node) and data type (e.g., SA branch with SA branch) to maintain the tree's validity. To ensure the legality of the tree exchange, we first exchange the SA branches of the two selected parents, and once that process is complete, we exchange the TA branches of the two trees.

With respect to point mutation in a strongly-typed setting, there again are certain limitations that must be observed. For example, function node $\texttt{OR}_{\texttt{SA}}$ can only be changed to $\texttt{AND}_{\texttt{SA}}$, function node $\texttt{GT}_{\texttt{TA}}$ can be replaced only with $\texttt{LT}_{\texttt{TA}}$(similarly for other function nodes), an ERC can be only replaced with another ERC and a terminal variable can only be replaced with another variable of the same indicator type. The algorithm thus ensures that valid data types replaced the mutated nodes. Point mutation happens in one of the two branches per tree.

The individuals who will act as parents of those operators are selected through tournament selection. A selected individual will undergo crossover with probability p and will undergo mutation with the remaining probability, $1-p$. Elitism is, also, in place to ensure the best individual of each generation is being copied to the next one.

3.2. Financial analysis processes

With our framework established, we move on by incorporating financial indicators derived from both technical and sentiment analysis to generate robust trading strategies. In this section, we will discuss the processes of technical analysis and sentiment analysis in two separate subsections. To ensure a fair comparison between the algorithms, especially the GP-SA and GP-TA algorithms, we use the same number of indicators. This prevents any inherent bias arising from the algorithm using the indicators of one analysis type more than the other, which could disproportionately influence the performance of the algorithms.

3.2.1. Technical analysis

Technical analysis (TA) is a widely used tool in financial forecasting and algorithmic trading. It involves analysing financial metrics to create technical indicators that help identify trends in the stock market, understand the financial status of companies, and generate higher profits. Researchers often develop and refine TA indicators from financial data to derive actionable insights into market trends and trader behaviour. These indicators utilise financial data such as stock prices and trading volumes, enabling traders to make data-driven decisions about buying and selling assets based on historical price movements.

In our study, we consider six widely used technical analysis (TA) indicators (or indicators), namely Moving Average, Momentum, Rate of Change, Williams %R, Midprice, and Volatility. We chose these indicators as they are widely recognised and used in the bibliography to understand diverse market dynamics, so by focusing on them, our research can be easily contrasted with other studies in the literature, such as [51], [52], [53], [54], [55], [56]. The indicators are defined in Equations (1) - (6) below. These are calculated based on historical data on (adjusted) close prices, highest and lowest daily prices of selected companies, available on Yahoo! Finance (more details on our datasets are presented in Section 4.1). Each indicator is considered with respect to look-up windows of $n = 5$ and $n = 10$ days, giving rise to 12 TA indicators summarised in Table 2. The computation of indicators was automated using Python, using the pandas [57] library for rolling window calculations and the NumPy library [58] for mathematical operations.

Table 2: Technical Analysis indicators. Each indicator is considered for two different lookup windows (n).

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

The Moving Average is defined as follows and is used to smooth out stock

price data and helps with noise elimination towards identifying trends by filtering out short-term price fluctuations. It is one of the most fundamental TA tools, widely used to measure general trend directions and reduce noise. p_j denotes the adjusted closing price of the j-th day in our dataset for a corresponding stock.

$$\text{Moving Average}(n, j) = \frac{\sum_{i=j-n}^{j} p_i}{n}, \text{ for } j \geq n. \tag{1}$$

Momentum tracks short-term fluctuations in price relative to a baseline, highlighting the velocity of price changes and potential turning points. In this way, Momentum measures how quickly the market is moving up or down. The Momentum captures the difference between the most recent adjusted closing price and the adjusted closing price n days ago, as follows.

$$\text{Momentum}(n, j) = p_j - p_{j-n}, \tag{2}$$

In addition, the Rate of Change (ROC) normalises the price difference by dividing it by the price n days ago. It measures the percentage shift in price over a given lookback window, providing a proportional change as opposed to the absolute change from Momentum. This way, ROC enables the comparison between assets with different price levels, even if they have the same Momentum, because it highlights the relative significance of their price movements.

$$\text{ROC}(n, j) = \left(\frac{p_j}{p_{j-n}} - 1 \right) \cdot 100. \tag{3}$$

Volatility is a statistical measure of the dispersion of returns over a given period of time. It captures the variability of the returns over a specific period and it is important in understanding uncertainty. A higher volatility implies greater potential price changes, making it a critical component in risk management and trading strategy decisions. We calculate the following relevant indicator.

$$\text{Volatility}(n, j) = \sqrt{\text{Var}\left(\left\{ \frac{p_{j-i}}{p_{j-n}} - 1 \right\}_{i \in \{0, \ldots, n-1\}} \right)}, \tag{4}$$

where Var defines the sample variance over a dataset.

The Williams %R indicator, defined in Equation (5), reflects the level of most recent closing price, cl_j (at day j), to the highest high price, $hh_{n,j}$, of

all values in the lookup window ending at day j. $ll_{n,j}$ denotes the lowest low price over all days in the lookup window ending at day j. It finds overbought/oversold market conditions by comparing the current closing price with the high–low range of the lookback window. It is a popular measure for short-term decision-making and helps traders identify potential price reversals.

$$\text{Williams } \%\text{R}(n, j) = -100 \cdot \frac{hh_{n,j} - cl_j}{hh_{n,j} - ll_{n,j}} \tag{5}$$

Midprice, defined in Equation (6), returns the midpoint value of the highest high price, $hh_{n,j}$, and the lowest low price, $ll_{n,j}$, over all days in the lookup window ending at day j. It determines the central value between the highest high and lowest low in the lookback window, serving as a measure of central tendency within a recent high and low range. By highlighting this midpoint, traders can detect potential equilibrium levels or pivot points in price action over the lookback period.

$$\text{Midprice}(n, j) = \frac{hh_{n,j} - ll_{n,j}}{2} \tag{6}$$

Overall, by incorporating these indicators—covering trend detection (Moving Average), market speed (Momentum and ROC), overbought/oversold signals (Williams $\%R$), price reference points (Midprice), and market risk (Volatility) - our analysis captures a broad spectrum of market behaviours. All TA indicators were normalised between $[-1, 1]$.

3.2.2. Sentiment analysis

As financial markets get influenced by events and stocks' prices increase / decrease along with people's decisions on online information, there is a surge of studies using sentiment analysis indicators in the areas of financial forecasting and algorithmic trading. Sentiment analysis (SA) is the process of extracting the sentiment out of articles and online comments and utilising into increasing the accuracy of stock estimation and trading strategies' profits.

Two widely adopted sentiment analysis indicators are the sentiment polarity and subjectivity of given texts. The former, captures the inclination of sentiment, and the relative text is classified as positive, negative or neutral. The latter captures the extent to which the respective text expresses a personal opinion rather than a fact. In our analysis we use indicators based

on the polarity and subjectivity, while distinguishing between the method of calculating them (definitions of respective methods appear below).

Sentiment analysis classification research commonly uses specialised SA programs to calculate the polarity and subjectivity of text. Three popular tools are TextBlob [59], SentiWordNet [60], and AFINN sentiment [61]. *TextBlob* is a Python library that provides a straightforward API for determining the polarity and subjectivity of text. *SentiWordNet 3.0* is a lexical resource based on the English language's lexical taxonomy, *WordNet*, that is specifically designed to support sentiment classification and opinion mining. It contains a list of words classified as positive, negative, or neutral, and the overall sentiment of a given text is calculated as a weighted average of these words. *AFINN* sentiment is a widely used sentiment lexicon that includes over 3300 words, each with a polarity score, developed by Finn Årup Nielsen. In our research, we utilise the built-in function for the lexicon, which is available in Python. The selection of these tools was based on their popularity within the academic community, as demonstrated in [62], [63], [64], [65], [66], [67], and [68].

In particular, we consider 12 SA indicators summarised in Table 3. The sentiment analysis indicators we use involve the polarity and subjectivity levels extracted by TextBlob, as well as the sentiment polarity extracted by SentiWordNet and AFINN. We analyse the relevant articles, titles, and summaries separately, resulting in a total of 12 sentiment analysis indicators.

Table 3: 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.

TextBlob		**SentiWordNet**	**AFINN**
TEXTpol,	TEXTsub	TEXTsenti	TEXTafinn
TITLEpol,	TITLEsub	TITLEsenti	TITLEafinn
SUMMpol,	SUMMsub	SUMMsenti	SUMMafinn

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

In our analysis, we downloaded articles related to the selected companies and linked their sentiment to the corresponding dates and price changes in the stock market. Firstly, we gathered articles using a custom web scraper that utilised the Google Search Console API in Python. This API was chosen

for its ability to provide automated and efficient access to large volumes of search data, including near real-time and historical results. For each company, the scraper queried the first twenty pages of daily Google search results, using the company's name as the primary keyword. The articles were collected for the same timeframe as the technical analysis indicators to ensure consistency. Furthermore, along with the full-text content of each article, the scraper extracted the title and summary, allowing for the creation of sentiment indicators not only from the body of the article but also from its title and summary.

Afterwards, to ensure only relevant articles were included in the analysis, we applied two filtering criteria: 1. Articles must be at least 500 characters long to avoid overly brief or irrelevant content; 2. Articles must explicitly mention the company's name and stock ticker symbol, allowing us to filter out articles that were not related to the companies we were interested in.

Following, to match the sentiment of articles with corresponding stock price data, we synchronised the publication dates of the articles with relevant stock prices. For articles published on weekends, when the stock market is closed, the sentiment scores were assigned to the preceding Friday to capture their potential influence on stock prices the following Monday.

Finally, for days when multiple articles were published for the same company, we calculated the average sentiment value across all relevant articles for that day. For days with no articles, we assigned a sentiment value of zero (0) to denote neutrality or lack of action, ensuring continuity and avoiding gaps in the data.

3.3. Trading signals and trading strategy

Specifically, the result of the root `AND` function, which is a True/False value, is utilised by the algorithms to determine whether to issue a 'buy' signal or maintain a 'hold' position for a given stock. The recommendation is made as follows. Each evolving GP model is incorporated into another tree architecture with an If-Then-Else (ITE) node as the root. The second and third branches of this ITE statement are *fixed* and represent buy (1) and hold (0) decisions, respectively, as illustrated in Figure 1. It is important to note that only Part 1 of Figure 1 undergoes evolution through GP operations. The overall tree in Figure 1 starts with the root node `AND` (i.e. Part 1), while the SA branch starts with the root node $\texttt{GT}_{\texttt{SA}}$ (blue-coloured nodes), and the TA branch corresponds to the root node $\texttt{LT}_{\texttt{TA}}$ (yellow-coloured nodes).

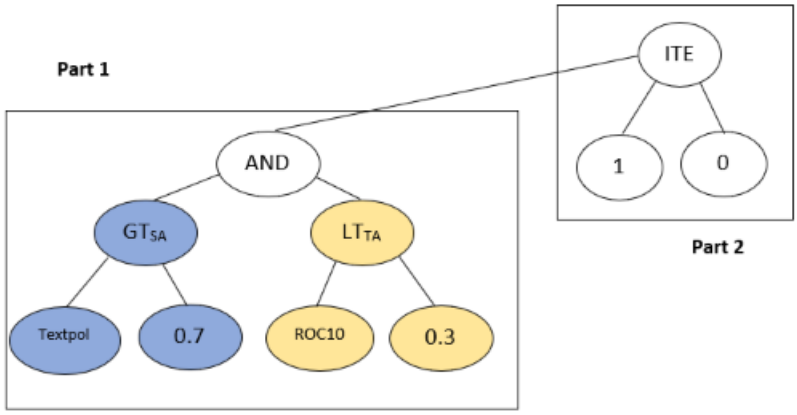

Figure 1: Sample tree of STGP-SATA. The first child of the AND function is enforced to be SA-related and the second child to be TA-related. This tree checks if the TEXTpol indicator is greater than the ERC 0.7 and if the ROC10 indicator is less than the ERC 0.3. If both of them are true, the recommendation will be to buy (1), otherwise it will be to hold (0).

If the GP tree (Part 1) evaluates to True, the ITE tree directs the process to the second branch (1). This triggers the GP algorithm to issue a 'buy' signal, initiating a trade. The stock is then sold based on the following rule: For a given holding period, d, and a target return rate, r, if the stock price increases by more than r within d days, the stock is sold on that day. Otherwise, it is sold at the end of the d days. For instance, if $d = 30$ and $r = 0.05$, this implies that the stock is sold either when its price exceeds a 5% increase within 30 days or at the end of the 30-day period, whichever comes first. Conversely, if the GP tree evaluates to False (signal: 0), the ITE tree leads to a 'hold' action, and no trade is executed.

Parameters d and r are optimised during the validation phase and are the same for all GP algorithms considered, but different across different companies (see Section 4.3).

3.4. Fitness function and Metrics

STGP-SATA is trained to optimise the Sharpe ratio, a widely used metric that evaluates the trade-off between return and risk. The Sharpe ratio was selected as the fitness function due to its ability to balance profitability against risk, making it particularly suitable for evaluating trading strategies. It is a comprehensive measure of performance, suitable for risk-averse in-

vestors. By maximising the Sharpe ratio, STGP-SATA is able to prioritise strategies that achieve high returns while minimising the risk.

Calculating the Sharpe ratio requires determining the returns (R) from individual trades, defined as the profit expressed as a percentage of the initial investment. The calculation of the profit takes into account the transaction costs, $c_t = 0.025\%$, as well, as has been similarly seen in [69], [70], and [15]. This cost is applied consistently across all algorithms and all trades, ensuring fairness in performance comparisons. Incorporating transaction costs ensures that the Sharpe ratio reflects realistic market conditions. R is found as shown in Equation (7), where V_f denotes the final value, or the price the stock was sold, and V_i denotes the initial value, or the price the stock was bought.

$$R = \frac{(1 - c_t) \cdot V_f - V_i}{V_i} \tag{7}$$

The rate of return, RoR, denotes the sample mean of the returns of all trades in a corresponding period of time in question.

The *risk* is captured as the standard deviation of the returns, that is $\sqrt{var[R]}$.

The *Sharpe ratio*, which is the metric STGP-SATA maximises is calculated as seen bellow. S_r, is defined as the ratio of the expected value of the excess return compared to the risk free return, R_f, over the risk. Formally,

$$S_r = \frac{\mathrm{E}[R - R_f]}{\sqrt{var[R]}} \tag{8}$$

where R_f is the risk free return, a given value of 0.022%, consistent across all algorithms, an average value as portrayed in Fernandez et. al. [71] and as seen in Long et. al. [72].

The Sharpe ratio is a widely recognised concept in finance that gauges the return on an investment strategy relative to the risk it entails. This metric facilitates comparisons of investment strategies with varying levels of risk, helping investors to make informed decisions. By assessing the risk of a stock or company and evaluating whether its potential return justifies that risk, the Sharpe ratio empowers investors to make sound investment choices. Our STGP-SATA algorithm prioritises the return-risk tradeoff by placing the Sharpe ratio at the heart of its strategy.

4. Experimental Setup

4.1. Data

In this study, we utilised datasets from 60 different companies, which are listed in Table 4. These companies were chosen due to their popularity, ensuring we would be able to collect enough sentiment analysis data for each one of them. Furthermore, we wanted to represent a broad range of industries. This diversity ensures that the findings are not biased toward a particular sector and can be generalised across different market conditions. We collected news articles and historical prices for each company. The selection of companies was based on their popularity to ensure that we could collect a sufficient amount of articles. The research period covered 5 years, from January 1st, 2015, to January 31st, 2020, which excludes the pandemic of COVID-19, since that would make the train/validation sets too different from the test set, and the parameter tuning would not be reliable.

To perform technical analysis, we collected the daily closing price data from Yahoo! Finance. For sentiment analysis, we gathered articles, their titles, and their summaries using a scraper and the Google Search Console API in Python. The choice of data sources was based on their reliability and accessibility. Yahoo! Finance provided an accurate and standardised source of historical price data, while the Google Search Console API enabled the retrieval of comprehensive textual data for sentiment analysis. After collecting all the necessary data, we generated the 12 SA indicators and the 12 TA indicators presented in Section 3. Then, we split the datasets of the 60 companies in sequence into three parts : 60% for training, 20% for validation, and 20% for testing.

For each of the 60 companies, we conducted 50 independent runs on the training set, evolving distinct trading strategies for each run. This approach ensures the robustness of our findings by accounting for variability across multiple runs.

4.2. Benchmarks

The proposed STGP-SATA is benchmarked against three other GP algorithms in this study:

- **GP-TA** is a GP algorithm that only includes technical analysis indicators in its terminal set.

Table 4: Companies in our experiment.

Sector	***Companies***
Technology	Apple, Adobe, Asus, BlackBerry, Facebook, Fujifilm, Fujitsu, Google, IBM, Intel, Kodak, Microsoft, Nikon, Nokia, Nvda, Panasonic, Sony, Tencent, Xerox
Design-Cosmetics	Adidas, Asics, Asus, Dior, Estee, Fila, Kering, Nike, Shiseido
Drinks-Food	Coca Cola, McDonalds, Nestle, Sainsbury, Starbucks, Tesco, Walmart
E-commerce	Alibaba, Amazon, Ebay
Vehicles	BMW, Ford, General Motors, Honda, Hyundai, Kia, Mitsubishi, Nissan, Subaru, Suzuki, Tesla, Toyota, Yamaha Motor
Conglomerate-Finance-Pharmaceutical	Berkshire Hathaway, Johnson Johnson, Hitachi, HSBC, Yamaha Corp
Gaming-Production-Media	Activision Blizzard, Netflix, Nintendo, New York Times Co, Ubisoft

- **GP-SA** is a GP algorithm that only includes sentiment analysis indicators in its terminal set.
- **GP-SATA** is a (non-strongly-typed) GP algorithm that combines indicators of technical and sentiment analysis.

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)**

The four algorithmic benchmarks have been extensively used in related literature, and in this study, the scikit-learn library's built-in models in Python were employed. We use these algorithms to tackle the following binary classification problem: "Will the stock price increase by $r\%$ within the next d days?". Class 1 represents a buy action, while Class 0 represents a hold action. As previously stated in Section 3.3, the sell action is carried out as part of the trading strategy.

Finally, STGP-SATA is evaluated against the following financial benchmarks:

- **Buy and Hold (BnH)**: A very popular investment strategy dependent on historical prices, where we buy one unit at the very beginning of the trading period and sell at the very end. This strategy relies on the fact that, over time, the value of investments will generally increase.

- **Trading-Strategy$_{d,r}$ (TS$_{d,r}$)**: Buy at the beginning of every trading period. Sell when the price increases by more than the rate of reference r, or after d days have passed, whichever happens sooner.

The TS$_{d,r}$ benchmark follows the same trading strategy that the GP algorithms are using, but without the learning element of generating buy and hold signals. This allows us to examine the added value of our GP algorithm when it is separated from the pure trading element of the strategy.

4.3. Parameter Tuning

To determine the optimal GP parameters, a grid search was performed on the validation set and it was completed in two steps. The parameter tuning process was designed to ensure optimal performance and fairness across all algorithms.

A grid search was conducted on the validation set in two phases: (1) tuning genetic programming (GP) parameters; and (2) tuning trading strategy parameters (d and r) specific to each company.

In the first phase, GP parameters such as population size, crossover probability (p) [1], number of generations, tournament size, and maximum tree depth were tuned. A grid search was performed to identify a parameter combination that optimised performance across all GP variants, ensuring fair

[1]The mutation probability is 1-p, thus it was not necessary to include it in the parameter tuning process.

comparison. During this phase, trading strategy parameters ($d = 30$ days and $r = 0.05$) were kept constant to reduce tuning complexity and runtime. The optimal GP parameters identified by this process were used in all runs for all algorithms and companies and are presented in Table 5.

Table 5: GP Parameters for GP-TA, GP-SA, GP-SATA. STGP-SATA.

GP Parameters	
Population size	1000
Crossover probability	0.95
Mutation probability	0.05
Generations	50
Tournament size	4
Maximum tree depth	6

In the second phase, the trading strategy parameters, d and r, were tuned independently for each company and algorithm. This independent tuning approach ensures that the trading performance is optimised while maintaining consistency in the GP algorithms' application. The parameters were selected based on their overall performance on the validation set.

The parameter tuning for MLP, SVM, LSTM, and XGBoost is performed separately using binary classification, where one class corresponds to a price increase of a certain percentage for the next day and the other to a different price update (see Section 4.2). Later, the model with the best predictive ability on the validation set is chosen. The predicted class is used at the testing dataset and it serves as a signal, fed into the trading strategy. The trading strategy parameters are set to be the same d (days) and r (percentage increase) values as in the GP-variants. The tuning process for these two machine learning algorithms for trading purposes is based on [73].

5. Results and Analysis

This section presents the results of our experiments comparing STGP-SATA with the benchmarks presented in Section 4.2. We run 50 independent runs on the training set of each of the 60 companies, for each algorithm. Each of these runs corresponds to a different trading strategy. The derived trading strategies were subsequently applied to the test set, which formed

the basis of the analysis. Conducting 50 independent runs for each of the 60 companies increases statistical consistency, making our results more robust and delivering a more reliable evaluation of the algorithm's performance.

During some of the 50 independent runs performed, the GP algorithms did not execute any trading action due to the likelihood of incurring losses. Based on Table 6, this occurred in 3 companies for STGP-SATA, followed by GP-SATA and GP-TA (4 companies each) and GP-SA (7 companies). These runs were reported as 0 for Sharpe ratio, rate of return, and risk. Moreover, when only one trade was made, the risk was uncomputable and reported as 0, again. To prevent distortions in statistical analysis, the mean values presented in Tables 6 to 12 were calculated based on independent runs that involved more than one trade, since two or more trades are needed to measure the standard deviation of returns, i.e. the risk. The tables contain the mean, median, standard deviation, maximum, and minimum values of the distribution for each algorithm across 60 companies. However, some algorithms did not produce any results for certain companies (i.e. they did not execute any trades at all for any of the 50 individual GP runs), leading to rows with mean values of 0.

To validate the statistical significance of our findings, we conducted a two-sample Kolmogorov-Smirnov (KS) test to compare the results across all runs and companies for each algorithm,while we excluded values of 0. The KS test was chosen because it is sensitive to differences in the shape of the empirical cumulative distribution functions of two samples and identifies the maximum difference between their distributions. This makes it particularly suitable for detecting variations in the performance distributions of the algorithms. The test was applied separately for each financial metric.

To address the issue of multiple comparisons, we applied the Holm-Bonferroni correction to adjust the significance threshold for statistical tests. This correction ensures that the overall Type I error rate remains within the desired significance level ($\alpha = 0.05$). The minimum acceptable p-value for a given rank, is calculated using the formula $\alpha(rank) = \frac{0.05}{3-rank+1}$, where $rank \in \{1, 2, 3\}$. The term $rank$ corresponds to p-value magnitude order, with 1 being the smallest and 3 being the largest. For example, if the p-value between STGP-SATA and GP-SATA is 0.01, the p-value between STGP-SATA and GP-SA is 0.02, and the p-value between STGP-SATA and GP-TA is 0.03, that means the first p-value is the smallest so it is assigned a $rank$ of 1, the second p-value has a $rank$ of 2, adn the third p-value, being the largest, has a $rank$ of 3.

Since we compared the STGP-SATA to the 3 GP benchmarks, resulting in 3 different comparisons for each financial metric, the denominator 3 corresponds to the number of different comparisons being made. To determine if two distributions are statistically different at 5% significance level, the p-value for each comparison is compared to the corresponding minimum acceptable p-value for its rank. Specifically, the first-ranked p-value should be less than 0.0166, the second-ranked p-value should be less than 0.025, and the third-ranked p-value should be less than 0.05. This approach ensures that the probability of a false positive result is kept below a predetermined threshold, thereby providing more reliable statistical results.

5.1. Sharpe ratio

Table 6 displays the average Sharpe ratio values of 50 runs for each company across the three GP algorithms. GP-TA was the best performing algorithm in 17 out of 60 companies, while GP-SA was the best in 21 companies. On the other hand, GP-SATA had the highest Sharpe ratio in 10 companies, while STGP-SATA, which has the highest mean Sharpe ratio (see Table 7), was the best performing algorithm in 11 companies.

Even though STGA-SATA does not have the highest number of best performing occurrences, it is important to also look at the descriptive statistics of each algorithm. Table 7 presents the mean and median Sharpe ratio for each algorithm, as well as the standard deviation and the maximum and minimum values. When looking at Sharpe ratio results, it is important to keep in mind that this metric can be sensitive to the number of trades and can experience large values (either positive or negative) if very few trades are performed. This is because the risk, which is calculated as the standard deviation of returns experienced in a given period, can end up being extremely small if an algorithm happens to perform very few trades (e.g., 2-4 trades throughout the test set). Given that risk is the denominator of the Sharpe ratio, a very small decimal number of risk can lead to a very high value of the Sharpe ratio.

As we can observe in Table 7, the proposed STGP-SATA algorithm has the highest mean value (10.79), which is approximately three times the value of GP-SATA (3.61). However, as we said above, we need to take into account the variability of the Sharpe ratio results. As we can observe, STGP-SATA has the largest maximum value (426), as well as the lowest minimum value (-9.7). Its standard deviation is also the highest. The median can therefore be a more appropriate metric in this case. As we can observe, STGP-SATA

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

Company	***GP-SATA***	***GP-TA***	***GP-SA***	***STGP-SATA***	***Company***	***GP-SATA***	***GP-TA***	***GP-SA***	***STGP-SATA***
AAPL	2.36	**6.24**	2.99	3.99	KIA	5.04	0	1.06	**45.5**
ADBE	3.44	4.94	**17.36**	7.60	KODAK	0.97	**1.66**	1.49	1.23
ADID	**1.12**	0.10	-0.88	0.73	MCDON	0	-4.3	**0.91**	0
ALIB	2.64	2.77	**12.87**	1.72	MITSU	-38.3	**5.4**	0	1.83
AMAZ	3.5	**14.7**	1.79	3.51	MSFT	**44.8**	6.1	0	0.017
ASICS	1.16	**4.81**	1.59	0.11	NESTLE	3.49	-0.01	2.32	**3.56**
ASUS	**15.63**	0.5	3.43	9.97	NFLX	-0.34	0.91	**17.8**	8.39
ATVI	-0.12	3.5	2.56	**4.96**	NIKE	5.77	4.87	**6.05**	1.82
BERK	2.8	**19.15**	4.22	0.52	NIKON	6.19	**10.83**	-0.6	4.35
BLACB	**45.42**	2.8	-0.03	1.71	NINT	0.31	-0.12	**1.13**	0.73
BMW	11.22	**17.6**	6.01	7.70	NISS	-1.09	-0.69	-188	**-0.20**
COKE	0.61	**0.68**	-0.58	0.61	NOKIA	-5.14	-2.49	**1.56**	-2.73
DIOR	-0.17	0.49	**24.2**	-0.01	NVDA	9.65	**35**	6.7	7.87
EBAY	0.11	0	6.36	**426**	NYT	0.3	-0.74	**3.22**	0.41
ESTEE	1.56	1.23	**2.09**	2.05	PANA	2.77	**3.9**	2.5	-0.31
FB	-0.74	**3.04**	0	-0.68	SAINS	1.99	2.02	**2.27**	2.10
FILA	0.13	0.27	0	**0.68**	SHIS	1.65	0.56	**4.91**	0.64
FORD	**39.89**	-0.59	8.3	20.5	SONY	0.83	1.76	13.44	**19.9**
FJFILM	-1.16	-1.59	**0.55**	-1.30	STARB	-0.3	-2.7	**2.76**	-0.04
FJTSU	15.21	13.7	3.7	**18**	SUBA	0.58	**7.73**	-0.18	5.43
GM	-0.22	-20.9	**4.01**	0	SZK	5.2	2.67	-1.09	**13.95**
GOOG	**3.52**	2.3	2.47	2.85	TENC	0	-0.55	0	0
HITA	0	0	**0.24**	0	TESCO	-1.1	**2.24**	0.93	-1.20
HONDA	6.34	2.5	6.63	**6.99**	TESLA	2.02	2.18	**3.5**	2.72
HSBC	0.03	0.17	**2.92**	-0.6	TOYO	0.25	0	**1.16**	0.91
HYUND	**5.71**	-0.64	0.71	-9.7	UBIS	0.69	**1.5**	0	1.1
IBM	0.73	**10.57**	0.93	7.90	WALM	0.69	1.95	1.13	**3.65**
INTC	0.70	1.46	1.97	**2.62**	XEROX	**1.36**	-0.82	0.14	-0.10
JNJ	4.17	2.53	**10.44**	3.05	YACO	0.72	0.44	**1.37**	1.23
KERI	-1.23	**0.92**	0.75	0.22	YAMO	**3.16**	0.65	0	2.33

Table 7: Summary statistics of Sharpe ratio. Boldface is used to denote the best value for each statistic.

Statistic	***GP-SATA***	***GP-TA***	***GP-SA***	***STGP-SATA***
Average	3.61	2.8	0.23	**10.79**
Median	1.1	1.4	1.7	**1.8**
StDev	11.1	**7**	25	54.6
Max	45.42	35	24	**426**
Min	-38.4	-20	-188	**-9.7**

still has the highest median value (1.8) among the four GP algorithms. The companies that perform with a Sharpe ratio less than 1 are 26, and in 15 companies the STGP-SATA trees used slightly more TA than SA indicators (HYUND, NISSAN, NOKIA). On the other hand, the number of companies performing with a Sharpe ratio more than 1 were 34, of which 22 companies were using as many or slightly SA than the TA indicators (HONDA, KIA, SZK).

The null hypothesis of the KS tests is that each pair of distributions being compared come from the same continuous distribution. A p-value below the corresponding significance level indicates that the null hypothesis is rejected, and the two distributions are considered statistically different. As STGP-SATA has both the highest average and median Sharpe ratio values, it is used as the control algorithm for the statistical test and is thus compared pairwise with the other GP variants. The results show that STGP-SATA statistically outperforms all algorithms, with p-values (second column) significantly lower than the corresponding significance level values (fourth column). More specifically, STGP-SATA statistically and significantly outperforms GP-SATA with a p-value of 0.0016, GP-SA with a p-value of $4.2E-7$, and GP-TA with a p-value of $6.44E-06$.

Concluding, STGP-SATA has the highest average and median values of Sharpe ratio, and it statistically outperforms the other three GP variants when performing the Kolmogorov-Smirnov statistical test.

5.2. Rate of Return

Similarly to Table 6, Table 10 presents the mean rate of return per trade over 50 runs per algorithm. Based on the average values of the 50 runs

Table 8: Pairwise Kolmogorov-Smirnov test p-values on Sharpe ratio of the proposed STGP-SATA algorithm against the 3 GP benchmarks. Statistical significance changes based on the Holm-Bonferroni correction. Statistically significant differences at the 5% level are indicated in boldface.

Algorithm	***STGP-SATA p-values***	**Rank**	***Significance level***
GP-SATA	**0.0016**	3	0.05
GP-SA	**4.2E-7**	1	0.016
GP-TA	**6.44E-06**	2	0.025

per company, STGP-SATA has the highest number (along with GP-SA) of best performances, as each algorithm has the highest rate of return in 20 companies. It is also worth noting that all algorithms have yielded negative returns for certain companies. More specifically, GP-SATA, GP-TA and STGP-SATA have 13, 12 and 11 companies with negative rate of return, while GP-SA has only 7. Furthermore, when looking at the summary statistics in Table 9, we can observe that the proposed STGP-SATA has again the best average and median values. Furthermore, it has the lowest standard deviation and the highest minimum value, indicating that even when it is not performing well, its losses are less than the other GP algorithms.

Table 9: Summary statistics of rate of return. The best value per metric is presented in boldface.

Statistic	***GP-SATA***	***GP-TA***	***GP-SA***	***STGP-SATA***
Average	0.0081	0.0074	0.0079	**0.0105**
Median	0.005	0.006	0.007	**0.008**
StDev	0.021	0.027	**0.020**	**0.020**
Max	**0.095**	0.093	0.065	0.09
Min	-0.037	-0.09	-0.05	**-0.033**

Table 10: Averages for rate of returns per company. Boldface is used to denote the best value for the particular dataset.

Company	*GP-SATA*	*GP-TA*	*GP-SA*	*STGP-SATA*	*Company*	*GP-SATA*	*GP-TA*	*GP-SA*	*STGP-SATA*
AAPL	0.018	**0.023**	0.013	0.02	KIA	0.04	0	0.016	**0.05**
ADBE	0.009	0.011	**0.016**	0.005	KODA	0.014	0.02	**0.05**	0.03
ADID	0.019	0.004	0.0014	**0.02**	MCD	0	-0.0003	**0.012**	0
ALIB	0.012	0.014	**0.03**	0.002	MIT	-0.006	**0.024**	0	0.008
AMAZ	-0.004	**0.02**	-0.001	0.004	MSFT	0.028	**0.03**	0	0.0012
ASIC	0.008	**0.025**	-0.009	0.015	NEST	0.0016	-0.002	**0.008**	0.0005
ASUS	0.02	0.019	0.0008	**0.03**	NFLX	0.027	-0.01	0.003	**0.05**
ATVI	-0.0026	**0.015**	0.009	**0.015**	NIKE	-0.011	**0.029**	0.028	0.011
BERK	0.005	**0.008**	0.005	0.004	NIK	0.03	**0.045**	-0.03	0.021
BLB	0.027	**0.03**	-0.008	**0.03**	NINT	0.002	-0.012	0.014	**0.027**
BMW	**0.041**	0.037	0.030	0.002	NISS	-0.037	-0.027	-0.05	**-0.013**
COKE	**0.009**	0.008	-0.04	0.0082	NOK	-0.032	-0.029	**0.004**	-0.033
DIOR	-0.0008	**0.001**	-0.009	-0.003	NVDA	-0.009	0.042	**0.048**	-0.025
EBAY	-0.013	0	0.001	**0.015**	NYT	0.007	-0.0015	**0.03**	0.010
ESTEE	0.006	0.009	0.009	**0.010**	PANA	0.0044	0.015	**0.02**	-0.009
FB	-0.012	**0.015**	0	-0.009	SAIS	**0.02**	0.015	0.016	**0.02**
FILA	0.004	-0.0007	0	**0.013**	SHIS	**0.03**	0.02	-0.01	0.012
FORD	0.012	-0.02	0.014	**0.02**	SONY	**0.003**	-0.0015	0.0003	-0.010
FJF	-0.020	-0.02	**0.03**	-0.01	STB	-0.015	-0.03	**0.007**	-0.002
FJT	**0.09**	**0.09**	0.064	**0.09**	SUBA	0.0008	**0.03**	-0.015	-0.006
GM	-0.020	-0.09	**0.023**	0	SZK	-0.013	0.0007	-0.03	**0.03**
GOOG	0.003	-0.007	**0.005**	**0.005**	TENC	0	-0.05	0	0
HITA	0	0	**0.012**	0	TESC	-0.006	0.008	**0.016**	0.015
HONDA	**0.014**	0.001	0.008	0.008	TESL	0.047	**0.07**	0.054	0.06
HSBC	0.0004	0.002	**0.009**	-0.013	TOYO	0.006	0	0.008	**0.015**
HYU	0.002	-0.02	**0.012**	0	UBIS	**0.04**	0.02	0	0.02
IBM	**0.03**	**0.03**	0.009	**0.03**	WAL	0.0006	0.004	-0.002	**0.008**
INTC	0.0006	0.001	-0.009	**0.02**	XERO	-0.003	-0.006	**0.004**	0.0015
JNJ	0.008	0.007	**0.015**	0.007	YACO	**0.0086**	-0.007	0.005	0.0016
KERI	-0.018	0.013	**0.017**	0.004	YAMO	**0.019**	0.006	0	0.003

When performing the KS-tests, STGP-SATA is again selected as the control algorithm, since it has the highest average and median values for rate of return. We can observe from Table 11 that STGP-SATA statistically outperforms GP-SATA, since the p-value is 0.0068, as well as GP-TA with a p-value of $3.59E-6$ and GP-SA with a p-value of $1.46E-6$.

To sum up, STGP-SATA has the highest average and median values in rate of return, and it statistically and significantly outperforms at the 5% level the other 3 GP algorithms.

5.3. Risk

Table 12 presents the mean results for risk per trade over 50 runs for each one of the GP algorithms. We note that GP-SA performs the best in 24 companies, STGA-SATA in 20, GP-TA in 18, and GP-SATA in 13.

Table 11: Pairwise Kolmogorov-Smirnov test p-values on rate of return of the proposed STGP-SATA algorithm against the 3 GP benchmarks. Statistical significance changes based on the Holm-Bonferroni correction. Statistically significant differences at the 5% level are indicated in boldface.

Algorithm	***STGP-SATA p-values***	**Rank**	***Significance level***
GP-SATA	**0.0068**	3	0.05
GP-SA	**1.46E-6**	2	0.025
GP-TA	**3.59E-6**	1	0.016

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

Company	***GP-SATA***	***GP-TA***	***GP-SA***	***STGP-SATA***	***Set***	***GP-SATA***	***GP-TA***	***GP-SA***	***STGP-SATA***
AAPL	0.019	**0.008**	0.023	0.02	KIA	0.036	0	0.051	**0.001**
ADBE	0.014	0.0095	**0.004**	0.02	KODA	0.062	**0.03**	**0.03**	0.046
ADID	0.022	0.026	**0.014**	0.03	MCDN	0	**0.029**	0.03	0
ALIB	0.037	0.03	**0.004**	0.045	MITS	0.021	**0.02**	0	0.037
AMAZ	0.041	**0.002**	0.031	0.03	MSFT	**0.02**	**0.02**	0	0.05
ASIC	0.058	0.045	**0.039**	0.06	NEST	**0.014**	0.017	0.003	**0.014**
ASUS	0.017	0.036	0.021	**0.005**	NFLX	**0.006**	0.089	0.062	0.016
ATVI	0.031	0.008	**0.004**	0.02	NIKE	0.048	0.023	**0.021**	0.033
BERK	0.012	**0.011**	0.012	0.015	NIKO	**0.01**	**0.01**	0.048	0.034
BLB	0.035	**0.03**	0.08	**0.03**	NINT	0.045	0.049	**0.035**	0.042
BMW	**0.005**	0.011	0.021	0.065	NISS	0.053	0.057	**0.023**	0.05
COKE	**0.01**	**0.01**	0.07	**0.01**	NOKI	**0.02**	**0.02**	0.043	**0.02**
DIOR	0.029	0.024	**0.019**	0.03	NVDA	0.083	0.008	**0.01**	0.1
EBAY	0.048	0	**0.02**	**0.02**	NYT	0.041	0.04	**0.019**	0.03
EST	0.013	0.016	**0.004**	0.008	PANA	**0.008**	0.02	0.01	0.034
FB	0.04	0.041	0	**0.03**	SAIS	**0.02**	0.022	0.027	**0.02**
FILA	0.03	0.048	0	**0.02**	SHIS	0.064	**0.058**	0.086	0.079
FORD	**0.001**	0.048	**0.001**	0	SONY	**0.015**	0.016	0.032	0.025
FJF	0.027	0.029	0.055	**0.015**	STB	0.039	0.036	**0.029**	0.04
FJT	0.01	0.017	0.017	**0.008**	SUBA	0.037	**0.016**	0.055	0.042
GM	0.09	0.004	**0.016**	0	SZK	0.079	0.056	0.068	**0.041**
GOOG	0.016	0.04	**0.01**	**0.01**	TENC	0	**0.098**	0	0
HITA	0	0	**0.056**	0	TESC	0.013	0.022	0.018	**0.004**
HOND	**0.003**	0.008	0.007	0.012	TESL	0.076	**0.034**	0.047	0.05
HSBC	0.006	0.008	**0.005**	0.015	TOYO	0.022	0	**0.018**	0.02
HYUN	0.022	0.044	0.021	**0.0022**	UBIS	0.051	**0.027**	0	0.04
IBM	0.045	0.003	0.067	**0.025**	WAL	0.015	**0.007**	0.017	0.01
INTC	0.073	0.03	0.044	**0.006**	XERO	**0.021**	**0.021**	0.06	0.05
JNJ	0.003	0.004	**0.001**	0.003	YACO	0.028	0.037	**0.02**	**0.02**
KERI	0.033	0.034	**0.02**	**0.02**	YAMO	**0.006**	0.017	0	0.035

In terms of summary statistics, which are presented in Table 13, we can observe that all GP algorithms have very similar risk values. In terms of

Table 13: Summary statistics for risk. Since we need the risk to be as small as possible, The lowest value per metric is presented in boldface.

Statistic	***GP-SATA***	***GP-TA***	***GP-SA***	***STGP-SATA***
Average	0.029	0.027	**0.026**	0.027
Median	**0.022**	**0.022**	**0.022**	0.025
StDev	0.022	**0.020**	0.023	0.021
Max	0.09	0.098	**0.086**	0.11
Min	**0**	**0**	**0**	**0**

average risk, GP-SA has the lowest average risk value (0.026), with STGP-SATA experiencing only a slightly higher risk value (0.027). Similarly, in terms of median risk, the best value is 0.022, with STGP-SATA having a slightly higher value of 0.025. The standard deviation is also very similar across all four GP algorithms.

Given that GP-SA is the algorithm that shows the lowest average and median risk values, it is used as the control algorithm and it is compared pairwise with the rest of the GP variants. As we can observe from Table 14, although GP-SA stastically outperforms GP-SATA and GP-TA, it does not statistically outperform our proposed algorithm, since the test's p-value is 0.129418.

Table 14: Pairwise Kolmogorov-Smirnov test p-values on risk of the GP-SA algorithm against the 3 GP benchmarks. Statistical significance changes based on the Holm-Bonferroni correction. Statistically significant differences at the 5% level are indicated in boldface.

Algorithm	***GP-SA p-values***	**Rank**	***Significance level***
GP-SATA	**1.73E-16**	2	0.025
GP-TA	**1.05E-30**	1	0.016
STGP-SATA	0.129418	3	0.05

Lastly, it is worth noting that STGP-SATA is able to show low risk values in situations where it is yielding negative returns. Looking back at Table 10, we had identified that GP-SATA was yielding negative returns in 13 companies, GP-TA in 12, STGP-SATA in 11, and GP-SA in 7. When taking

into consideration only these cases of negative returns, GP-SATA and GP-TA have an average risk value of 0.03, while STGP-SATA and GP-SA have an average risk value of 0.023. Consequently, the average Share ratio for the companies yielding negative returns is -2.6 (GP-SATA), -0.8 (STGP-SATA), -1 (GP-TA), and -4.6 (GP-SA). This shows that the STGP-SATA algorithm is able to perform at lower risk even in non-profitable trading strategies. This is particularly important, as all trading strategies underperform from time to time. The key factor lies in maintaining minimal volatility during periods of turmoil, as it enables traders to minimise their losses effectively. Therefore, the noteworthy aspect of STGP-SATA is its ability to exhibit the lowest risk and highest Sharpe ratio values in such situations, signifying the algorithm's crucial ability to mitigate losses efficiently.

In conclusion, although GP-SA performs with the least risk, it is not statistically and significantly different than STGP-SATA, which ranks second. Furthermore, STGP-SATA has the least risk in the runs that perform with a negative rate of return and Sharpe ratio.

5.4. Results of each market

The robustness of the algorithms' performance was assessed by categorising companies into market trend groups and evaluating performance within these categories. Regarding the time period, we chose to exclude the pandemic of COVID-19, because that would make the train/validation sets too different from the test set, making the parameter tuning not reliable. Although all 60 companies' data comes from the same time period, there is a lot of variation among their price series. Some of them tend to experience a positive price movement, while others experience a negative overall movement. We thus believe it is important to examine the GP algorithms' performance across different market profiles.

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, 27 companies were placed in Group 1, 17 in Group 2, and 16 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 15, STGP-SATA demonstrates advantages in terms of Sharpe ratio, particularly in Groups 1 and 3. Its average Sharpe ratios of 4.9 and 28.7, respectively, performing better than the rest of the GP-variants. Even when examining the medians, STGP-SATA maintains competitive results, indicating robust performance across most datasets. For instance, in Group 1, the median Sharpe ratio of 2.05—while lower than the average—is still higher than the median values of GP-SATA, GP-SA, and GP-TA. This confirms its ability to perform consistently well. Similarly, in Group 3, despite the exceptional average Sharpe ratio of 28.7, the median value of 1.47 highlights its good performance. For the rate of return and risk results, the proposed algorithm achieves the best average rate of return in Groups 2 and 3, alongside strong median values, demonstrating its capacity in neutral and downtrend markets. Additionally, STGP-SATA achieves the lowest average and second lowest median risk in Group 2. These results suggest that the algorithm balances profitability and risk exceptionally well.

Given the variation of results across groups and metrics, it is useful to look at the Sharpe ratio, which as an aggregate metric takes into account both return and risk. As mentioned, STGP-SATA shows strong performance for datasets that either have very strong positive price movements (Group 1) and negative price movements (Group 3). This indicates that our proposed algorithm is able to perform very well on strongly uptrend markets, as well as on downtrend markets. This is an important finding, as it demonstrates that our algorithm can perform well on opposite types of markets. Lastly, the fact that GP-SATA is also performing very well in terms of Sharpe ratio (best value in Group 2) indicates the importance of combining SA and TA indicators, even when this happens in a non-strongly-typed manner. Furthermore, the better performance of STGP-SATA in rate of return and risk indicates its effectiveness in achieving stable returns even in neutral markets where price movements are less pronounced.

This better performance of STGP-SATA in Groups 1 and 3, indicating its robustness in capturing trends in both highly optimistic and pessimistic market conditions, is likely due to its ability to balance sentiment and technical indicators to mitigate losses and capitalise on short-term opportunities. The strongly-typed GP structure ensures logical consistency, allowing the algo-

rithm to better adapt to datasets with pronounced trends, whether strongly positive (Group 1) or negative (Group 3). By maintaining separate branches for TA and SA, the algorithm effectively balances insights from both data types, reducing the risk of dominating the decision-making process.

Table 15: Separated average and median results per metric per market group.

Market	***Algorithm***	***Sharpe Ratio (Avg/Median)***	***Rate of Return (Avg/Median)***	***Risk (Avg/Median)***
Group 1 (>20%)	GP-SATA	3.7 / 1.12	0.0092 / 0.004	0.031 / 0.027
	GP-SA	4 / 2.1	**0.012 / 0.0095**	0.025 / **0.02**
	GP-TA	3.6 / 1.46	0.010 / 0.005	**0.021 / 0.02**
	STGP-SATA	**4.9 / 2.05**	0.010 / 0.006	0.028 / 0.025
Group 2 (0% - 19.99%)	GP-SATA	**5** / 1.65	0.010 / 0.006	**0.02 / 0.013**
	GP-SA	4.1 / **2.5**	0.011 / 0.01	0.028 / 0.021
	GP-TA	3 / 1.66	0.0053 / 0.008	0.032 / 0.023
	STGP-SATA	3.3 / 1.24	**0.012 / 0.011**	**0.02** / 0.015
Group 3 (<0%)	GP-SATA	2 / 0.71	0.0042 / 0.003	0.036 / 0.031
	GP-SA	-10 / 0.68	-0.0031 / 0	0.027 / 0.024
	GP-TA	1.6 / 1.09	0.0041 / 0.0043	**0.026 / 0.022**
	STGP-SATA	**28.7 / 1.47**	**0.0074 / 0.011**	0.033 / 0.03

5.5. Best tree results

Building upon the insights gained from the analysis of market trend groups and aggregated results, this section evaluates the best-performing models generated during the 50 independent training runs for each dataset. While the average results presented in previous sections offer a broad understanding of the algorithms' expected performance, it is equally important to consider the best outcomes achieved. In this context, the *best tree* refers to the model with the highest fitness score from the training set, selected from the 50 independent runs, and subsequently tested on the unseen test set. This focus on the best-performing model is particularly significant in the financial sector, where practitioners aim to maximise profitability by identifying and deploying the most effective trading strategy. If an investor was using a GP algorithm in the stock market, they would first run the algorithm multiple times and then select the best performing tree (model) for trading. By focusing on the best trees, we further explore the robustness of STGP-SATA in achieving better performance under practical conditions. Having an algorithm with very good performance in terms of *best tree* is an

important aspect in the financial sector. Table 16, thus, presents the average performance of the best trees across the 60 datasets for each GP algorithm.

Table 16: Best trees average performance across the 60 datasets

Algorithm	***Sharpe ratio***	***Return***	***Risk***
GP-SATA	0.45	0.003	0.005
GP-SA	0.47	**0.008**	**0.003**
GP-TA	0.063	0.003	0.006
STGP-SATA	**0.53**	0.007	0.004

As we can observe, GP-SA has the best return and risk values across the four GP algorithms. However, the proposed STGP-SATA comes second in both metrics, with only a small difference from the values of GP-SA. More importantly, STGP-SATA has the best Sharpe ratio with a value of 0.53, while the second-best algorithm (GP-SATA) has a value of 0.45. As the Sharpe ratio is an aggregate metric that takes into account both return and risk, the fact that the best tree of STGP-SATA has the best value makes it a very positive result. It is also worth noting that practitioners pay particular attention to such aggregate metrics [74], thus STGP-SATA best tree's performance is of particular importance. When it comes to the algorithm's interpretability, the average indicators used by the strategies are 14 for both tree branches (between the 24 combined sentiment and technical analysis indicators), as opposed to 50 for GP-SATA, 134 for GP-SA, and 61 for GP-TA.

5.6. Algorithmic Complexity

Understanding the computational complexity of STGP-SATA is needed to evaluate its scalability and practical feasibility. This section provides an analysis of the computational cost associated with our proposed STGP-SATA algorithm, considering its key parameters and evolutionary operations. The main parameters of STGP-SATA, as well as the rest of the GP variants (GP-SATA, GP-SA and GP-TA), are the population size (p), the number of generations (g), the maximum tree size (n), the training set size (m), and the elitism chosen children (e). The maximum depth a tree can achieve is k, and each function node produces a binary outcome connecting two terminal nodes. Thus, the maximum size a tree can potentially achieve is $n = 2k$. The complexity is broken into the following three parts:

(i) Population initialisation:
The initialisation of an individual has computational complexity $O(n)$, as its maximum size is n. This is repeated p times to make the initial population so the complexity of the initialisation step of the algorithm is $O(pn)$.

(ii) Fitness calculation:
The fitness calculation of each individual has to pass through all the data points of the training set (m). As there are p individuals in the population, the combined complexity is $O(pm)$.

(iii) Operators application:
The operators used by STGP-SATA are mutation, crossover, and elitism. Mutation has a constant complexity of $O(1)$ for randomly changing a node in the tree. Crossover has a complexity of $O(2n) = O(n)$ for STGP-SATA, since it is applied to both branches separately, and it includes extracting and replacing each of the two subtrees with selected subtrees that would not violate the validity of the overall tree. These operators are applied repeatedly until a new generation of less than p individuals is created (elitism will complete the new generation), thus, the overall complexity of applying the mutation operator is at most $O(p)$, while the overall complexity for applying the crossover operator is $O(pn)$. We note that for the GP-SATA, GP-SA, and GP-TA the crossover complexity is again $O(pn)$.

For elitism we implement an initial sorting of the fitness values among the individuals of the population, and select the one with the highest fitness between them. The complexity of this step is $O(eloge)$, however that can be $O(plogp)$ in the worst case scenario where the algorithm sorts the fitness functions of all children and then through a tournament selects the highest. This brings the overall complexity of the operators application to $O(p+pn+plogp) = O(pn)$.

Steps (ii) and (iii) are repeated for each generation, i.e. g times.

We can conclude that the total complexity of STGP-SATA is $O(pn + g(pm + p + 2pn + plogp))$, which is equivalent to $O(pn + g * p(m + n))$.

5.7. Interpretability

Genetic programming algorithms are white-box models, offering a transparency in the decision-making process. In our proposed algorithm, this transparency is achieved through the explicit structure of the generated trees, which outlines how inputs are processed to produce solutions. Such interpretability allows researchers to understand the reasoning behind each decision, making it easier to identify and correct errors. This transparency is

particularly important in algorithmic trading, where interpretable decision-making processes is important for risk management. This can help traders assess whether the algorithm's logic aligns with financial principles and market conditions. Furthermore, the algorithms' interpretability enables real-time adjustments to changing market trends, making them adaptable and robust in dynamic financial environments. These properties not only improve trust in the model but also improve their performance in creating profitable trading strategies.

When it comes to the proposed algorithm's interpretability, the average indicators used by the strategies are 14 for both tree branches (selected out of the 24 combined sentiment and technical analysis indicators), as opposed to 50 for GP-SATA, 134 for GP-SA, and 61 for GP-TA. That means, that for each function node we can calculate two terminal nodes, one is the SA/TA indicator and the other is the numeric threshold. Then we include the root node of each tree, too. Thus, for STGP-SATA the average total number of nodes is 28, GP-SATA and GP-TA is 82, and for GP-SA is 142. The significantly lower number of nodes/indicators used leads to small tree sizes for STGP-SATA, which makes it easier to be read and interpreted by humans. To illustrate this, we provide a tree image in Figure 2, coming from the NETFLIX dataset. As we can observe, the tree consists of 15 nodes, showing a straightforward and simple to implement trading strategy. In this specific tree, the root node (AND) combines the sentiment analysis (AND_SA) and technical analysis (AND_TA) branches, with each branch using two indicators and two thresholds. The tree then is connected to the "If then else" (ITE) statement to give the signal of 1 if TRUE or 0 if FALSE. This separation of data types enhances explainability by preventing type imbalance and allowing for actionable insights.

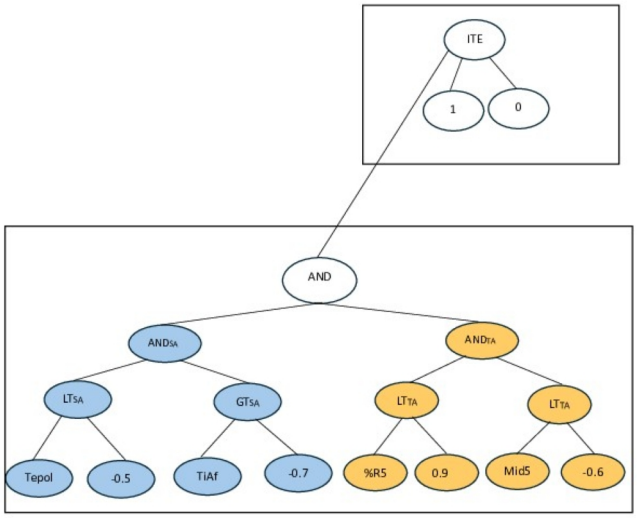

Figure 2: Tree of STGP-SATA for NETFLIX.

5.8. Non-GP Benchmarks

5.8.1. STGP-SATA compared to the algorithmic benchmarks

For further algorithmic benchmarks aside of the GP variants, the built-in models of scikit-learn library are used in Python for the MLP, SVM, XGBoost, and LSTM algorithms.

As we can observe from Table 17, the average values of MLP on the 60 companies for Sharpe ratio was 0.26, for rate of return 0.009 and for risk 0.044. This comes in contrast with the values of the GP, which are significantly higher for Sharpe ratio and lower for risk. When performing the KS statistical test, MLP was statistically different from STGP-SATA with a p-value of $1e-08$ for Sharpe ratio and 0.00011 for risk. On the other hand, the distribution of the returns for the two algorithms was not statistically different with a p-value of 0.26 at the $a = 0.05$ statistical level.

The same results continued on to the SVM, where the average value for Sharpe ratio was 0.25, for rate of return 0.009 and for risk 0.045, again seeing a big differences between rate of Sharpe ratio and risk. The KS tests, also, showed SVM being statistically different at a p-value of $3.65e-08$ for Sharpe

ratio and $4.75e-05$ for risk; while it was not statistically different regarding rate of return, since the p-value was 0.5.

For XGBoost, the average value for Sharpe ratio was higher, at 0.26, for rate of return 0.009 and for risk 0.044. The KS tests showed XGBoost being statistically different at a p-value of $3.15e-9$ for Sharpe ratio and $4.75e-0.5$ for risk, while the difference is not statistically different for rate of return with a p-value of 0.37.

Finally, for LSTM, the average value for Sharpe ratio was 0.45, for rate of return 0.0035 and for risk 0.044. The KS tests showed LSTM being statistically different at a p-value of $1.3e-11$ for Sharpe ratio and 0.07 for rate of return, and $4.75e-5$ for risk.

Table 17: Comparison of average values for STGP-SATA, MLP, SVM, XGBoost, and LSTM

Algorithm	**Sharpe ratio**	**Rate of return**	**Risk**
MLP	0.26	0.009	0.044
SVM	0.25	0.009	0.045
XGBoost	0.26	0.009	0.044
LSTM	0.45	0.0035	0.044
STGP-SATA	**10.79**	**0.0105**	**0.027**

Based on the metrics and the statistical tests, the algorithmic benchmarks are outperformed from STGP-SATA in Sharpe ratio and risk. This shows the disadvantage of these two algorithms compared to GP algorithms when it comes to algorithmic trading and creating strategies using metrics that consider both the returns and the risk.

5.8.2. STGP-SATA compared to Buy and Hold

In this section, we will compare the STGP-SATA algorithm to the Buy and Hold (BnH) strategy. To make a fair comparison between the BnH financial strategy and the GP algorithm we run the STGP-SATA using the cumulative returns as its fitness function, which measure the total profit or loss over a given period. The reason we focus on the cumulative returns and not using the Sharpe ratio as fitness function is because BnH involves just one trade, meaning we buy one unit of stock on the first day and we sell it on the last day. Thus, it does not take into account the rate of return and risk metrics involved in the calculation of the Sharpe ratio.

When comparing STGP-SATA and BnH, the GP algorithm has an average cumulative returns of 0.40 and a median of 0.26, while BnH has an average of 0.16 and a median of 0.169. The Kolmogorov-Smirnov test confirms that the differences in their values are statistically significant at the 5% level, with a p-value of 0.04.

5.8.3. STGP-SATA compared to a financial trading strategy

Wanting to evaluate the results in the three main financial metrics, and not only the cumulative returns, we compare the STGP-SATA algorithm to the $\text{TS}_{d,r}$ financial strategy.

As mentioned in Sections 4.2 and 3.3, through the $\text{TS}_{d,r}$ we buy on the first day of every trading period and we sell when the price increases by more than r, or after d days have passed. The variables r and d vary from company to company and they are the same for all algorithms, as explained in Section 4.3.

When comparing STGP-SATA and $\text{TS}_{d,r}$, the first thing we observe is that the latter does on average many more trades (230), while the former performs only 10. This is expected, because STGP-SATA is able to focus on the most profitable and low-risk opportunities, while choosing not to trade at all in all other cases. As a result, STGP-SATA has better values across all three metrics, as it can be seen from Table 18. The Kolmogorov-Smirnov test confirms that the differences in Sharpe ratio and return are statistically significant, with p-values of $2.25e-10$ (Sharpe ratio), and $1.02e-05$ (return). The p-value for risk is 0.051, thus marginally non-significant at the 5% level.

Table 18: Average values of STGP-SATA and TS$d, r$.

Algorithm/Metric	***Sharpe ratio***	***Return***	***Risk***
TS$d, r$	0.15	0.004	0.048
STGP-SATA	**10.8**	**0.01**	**0.027**

5.9. Summary of findings

In conclusion, as seen in Tables 6 - 18 and focusing on the findings from STGP-SATA, 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 strongly-typed GP algorithm STGP-SATA statistically outperforms the remaining GP variants in Sharpe ratio results, while it has

the highest rate of return and similar low risk to GP-TA and GP-SA. Furthermore, it has the highest median and maximum values, while it produces the lowest minimum value.

- On the other hand, the simple combination algorithm GP-SATA comes second after STGP-SATA in Sharpe ratio and rate of return, while it is being statistically outperformed for the first financial metric. Also, although GP-SATA has a similarly low risk to the other algorithms, it comes last.

- The algorithm that uses only technical analysis indicators in its terminal set, GP-TA, is being outperformed by STGP-SATA in Sharpe ratio and rate of return, while it outperforms the other algorithms in terms of risk.

- GP-SA, the GP algorithm with sentiment analysis data in its terminal set, has the lowest Sharpe ratio and rate of return, while its risk is similar to STGP-SATA and GP-TA. The median values of GP-SA come second, denoting its financial advantages in the data types combinations.

- Concluding, STGP-SATA is the most robust of the four GP variants. It is essential to stress that combining the TA and SA indicators under a strongly-typed GP, which ensures that effective search takes place in both the TA and SA search space, is essential in creating financially more advantageous trading strategies; in contrast with the simple combination of the two data types under GP-SATA.

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

- STGP-SATA performs better than the four algorithmic benchmarks. Although the algorithms have similar rate of returns, the produced risk of STGP-SATA is significantly less than that of MLP, SVM, XGBoost, and LSTM. The higher risk of the four algorithmic benchmarks is also reflected by their low Sharpe ratio values, which fall below 0.45.

- Algorithm STGP-SATA is more financially advantageous compared to the trading strategies BnH and $TS_{d,r}$, and it is able to statistically outperform the former in terms of cumulative returns, and the later in terms of Share ratio and rate of return.

6. Conclusion

In conclusion, the aim of our research was to investigate and compare the performance of trading strategies created by different GP algorithms that combine technical and sentiment analysis indicators. To achieve that, a novel strongly-typed GP was introduced, which ensured the produced trees utilise both analysis types in different tree branches. Our algorithm is compared to three other GP algorithms and other non-GP benchmarks against three different metrics, i.e. Sharpe ratio, returns and risk. As observed from the results, our proposed GP is competitive and statistically outperforms the other algorithms in many cases.

The importance of combining technical and sentiment indicators, which is not usually occurring in the previous studies, is evident from our findings. Combining the indicators can enhance the models' knowledge and create financially more advantageous trading strategies. However, it is essential to note that the way the two indicators' types are combined is, also, important. Based on our analysis, it is not profitable enough to simply combine the different types of indicators, as GP-SATA does, and a strongly-typed architecture is essential towards achieving an improved performance. Due to its design, STGP-SATA can create more diverse and efficient trading strategies, leading to better financial performance. This is because it ensures a balanced and integrated use of both technical and sentiment indicators, thus, balancing and taking advantage of both indicator types' advantages. Technical indicators can identify price trends and patterns, while sentiment indicators can capture market reactions to news and events. Furthermore, by enforcing type constraints, STGP ensures semantically valid and interpretable solutions, it improves trading performance but also enhances the practical applicability of the generated strategies. By combining the two, we can create more comprehensive models that are better in complex/extreme market conditions (as seen from Table 15). Furthermore, the algorithm is allowed a wider variety of combinations within each type and a greater diversity of trading strategies, as the algorithm is not constrained to focus on one indicator type.

While our current methodology shows promising results, we recognise the importance of continuous improvement. By incorporating more pretrained language models, such as BERT [75], we believe we can further enhance the robustness of our approach. Meanwhile, we plan to expand our data sources to include articles from financial forums and columns, to offer a more diverse and abundant information. While this study focused on price-based

technical indicators due to their popularity in the literature, we acknowledge the importance of including volume and liquidity-based indicators such as Volume Weighted Average Price (VWAP), On-Balance Volume (OBV), and Money Flow Index (MFI). These indicators will allow for a more comprehensive analysis of market trends and trading behaviour, which matches with our plans for future expansion of sentiment and technical indicators.

Running the algorithm live introduces complexities such as execution speed, slippage, and interactions with other market participants' algorithms, which can unpredictably impact performance. Addressing these challenges could involve solutions like co-location, dynamic order routing, and robustness testing. Moreover, an adaptive algorithm capable of dynamic retraining in response to changing market conditions presents a promising direction. The algorithm will be able to periodically update its models based on more indicators that will be available to it if each strategy fails to produce solutions that meet/exceed predefined performance thresholds, such as achieving a Sharpe ratio above 2 or a minimum annual return of 0.10, while maintaining low levels of risk. Furthermore, further research will include extracting more information from the individual indicator types by utilising the individual best models of each data type and creating a brand new tree which will be added in the population, again. Moreover, future research will include data derived from fundamental analysis, to incorporate more information in the models. The above are meant to extend the abilities of the STGP-SATA algorithm to create more practical, scalable, and explainable solutions in algorithmic trading.

References

[1] 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.

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

[3] A. Brabazon, M. Kampouridis, M. O'Neill, Applications of genetic pro-

gramming to finance and economics: past, present, future, Genetic Programming and Evolvable Machines 21 (1) (2020) 33–53.

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

[5] 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.

[6] 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.

[7] E. Ghasemzadeh, M. A. Keramati, S. Mehrinejad, A. Mehrani, Applying meta-synthesis techniques in identifying optimization components of fintech based on artificial intelligence indicators in the financial market, International Journal of Innovation Management and Organizational Behavior (IJIMOB) 4 (3) (2024) 173–179.

[8] A. Sharma, C. Verma, Investigating the impact of technical indicators on option price prediction through deep learning models, in: 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), IEEE, 2024, pp. 1–5.

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

[10] 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.

[11] 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.

[12] 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.

[13] L. L. Macedo, P. Godinho, M. J. Alves, A comparative study of technical trading strategies using a genetic algorithm, Computational Economics 55 (1) (2020) 349–381.

[14] R. L. de Almeida, R. F. Neves, Stock market prediction and portfolio composition using a hybrid approach combined with self-adaptive evolutionary algorithm, Expert Systems with Applications 204 (2022) 117478.

[15] X. Long, M. Kampouridis, P. Kanellopoulos, Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators, in: 2023 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2023, pp. 1–8.

[16] J. M. Berutich, F. López, F. Luna, D. Quintana, Robust technical trading strategies using genetic programming for algorithmic portfolio selection, Expert Systems with Applications 46 (2016) 307–315.

[17] 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.

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] N. Das, B. Sadhukhan, R. Chatterjee, S. Chakrabarti, Integrating sentiment analysis with graph neural networks for enhanced stock prediction: A comprehensive survey, Decision Analytics Journal (2024) 100417.

[28] 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 (2024) 1–31.

[29] B. A. Abdelfattah, S. M. Darwish, S. M. Elkaffas, Enhancing the prediction of stock market movement using neutrosophic-logic-based sentiment analysis, Journal of Theoretical and Applied Electronic Commerce Research 19 (1) (2024) 116–134.

[30] W. jun Gu, Y. hao Zhong, S. zun Li, C. song Wei, L. ting Dong, Z. yue Wang, C. Yan, Predicting stock prices with finbert-lstm: Integrating news sentiment analysis, in: Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing, 2024, pp. 67–72.

[31] 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.

[32] 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.

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

[34] K. Teymourian, M. Rohde, A. Paschke, Knowledge-based processing of complex stock market events, in: Proceedings of the 15th International Conference on Extending Database Technology, 2012, pp. 594–597.

[35] Y. Peng, H. Jiang, Leverage financial news to predict stock price movements using word embeddings and deep neural networks, arXiv preprint arXiv:1506.07220 (2015).

[36] 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.

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

[38] E. K. W. Leow, B. P. Nguyen, M. C. H. Chua, Robo-advisor using genetic algorithm and bert sentiments from tweets for hybrid portfolio optimisation, Expert Systems with Applications 179 (2021) 115060.

[39] S. Wu, Y. Liu, Z. Zou, T.-H. Weng, S_i_lstm: stock price prediction based on multiple data sources and sentiment analysis, Connection Science 34 (1) (2022) 44–62.

[40] 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.

[41] 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.

[42] J. Yang, Y. Wang, X. Li, Prediction of stock price direction using the lasso-lstm model combines technical indicators and financial sentiment analysis, PeerJ Computer Science 8 (2022) e1148.

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

[44] S. Agrawal, N. Kumar, G. Rathee, C. A. Kerrache, C. T. Calafate, M. Bilal, Improving stock market prediction accuracy using sentiment and technical analysis, Electronic Commerce Research (2024) 1–24.

[45] 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.

[46] G. Gaharwar, S. Pandya, The ensemble model for long-term stock investment is based on sentiment analysis and technical analysis.

[47] Z. Wang, Z. Hu, F. Li, S.-B. Ho, E. Cambria, Learning-based stock trending prediction by incorporating technical indicators and social media sentiment, Cognitive Computation 15 (3) (2023) 1092–1102.

[48] F. C. Dumiter, F. Turcaș, A. Nicoară, C. Bențe, M. Boiță, The impact of sentiment indices on the stock exchange—the connections between quantitative sentiment indicators, technical analysis, and stock market, Mathematics 11 (14) (2023) 3128.

[49] W. Ding, K. Mazouz, O. Ap Gwilym, Q. Wang, Technical analysis as a sentiment barometer and the cross-section of stock returns, Quantitative Finance 23 (11) (2023) 1617–1636.

[50] M. S. Amin, E. H. Ayon, B. P. Ghosh, M. S. C. MD, M. S. Bhuiyan, R. M. Jewel, A. A. Linkon, Harmonizing macro-financial factors and twitter sentiment analysis in forecasting stock market trends, Journal of Computer Science and Technology Studies 6 (1) (2024) 58–67.

[51] A. Grimes, The art and science of technical analysis: market structure, price action, and trading strategies, Vol. 544, John Wiley & Sons, 2012.

[52] M. Agrawal, A. U. Khan, P. K. Shukla, Stock indices price prediction based on technical indicators using deep learning model, International Journal on Emerging Technologies 10 (2) (2019) 186–194.

[53] Y. Peng, P. H. M. Albuquerque, H. Kimura, C. A. P. B. Saavedra, Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators, Machine Learning with Applications 5 (2021) 100060.

[54] M. Agrawal, P. K. Shukla, R. Nair, A. Nayyar, M. Masud, Stock prediction based on technical indicators using deep learning model., Computers, Materials & Continua 70 (1) (2022).

[55] N. Nazareth, Y. V. R. Reddy, Financial applications of machine learning: A literature review, Expert Systems with Applications 219 (2023) 119640.

[56] Y. Han, Y. Liu, G. Zhou, Y. Zhu, Technical analysis in the stock market: A review, Handbook of Investment Analysis, Portfolio Management, and Financial Derivatives: In 4 Volumes (2024) 1893–1928.

[57] W. McKinney, et al., pandas: a foundational python library for data analysis and statistics, Python for high performance and scientific computing 14 (9) (2011) 1–9.

[58] T. E. Oliphant, et al., Guide to numpy, Vol. 1, Trelgol Publishing USA, 2006.

[59] S. Loria, Textblob documentation, Release 0.15 2 (2018) 269.

[60] S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining (2010). URL `http://nmis.isti.cnr.it/sebastiani/Publications/LREC10.pdf`

[61] F. Å. Nielsen, A new ANEW: evaluation of a word list for sentiment analysis in microblogs, in: M. Rowe, M. Stankovic, A.-S. Dadzie, M. Hardey (Eds.), Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, Vol. 718 of CEUR

Workshop Proceedings, 2011, pp. 93–98.
URL http://ceur-ws.org/Vol-718/paper_16.pdf

[62] S. W. Chan, M. W. Chong, Sentiment analysis in financial texts, Decision Support Systems 94 (2017) 53–64.

[63] X. Li, P. Wu, W. Wang, Incorporating stock prices and news sentiments for stock market prediction: A case of hong kong, Information Processing & Management 57 (5) (2020) 102212.

[64] C. Nousi, C. Tjortjis, A methodology for stock movement prediction using sentiment analysis on twitter and stocktwits data, in: 2021 6th South-East Europe design automation, computer engineering, computer networks and social media conference (SEEDA-CECNSM), IEEE, 2021, pp. 1–7.

[65] S. Albahli, A. Irtaza, T. Nazir, A. Mehmood, A. Alkhalifah, W. Albattah, A machine learning method for prediction of stock market using real-time twitter data, Electronics 11 (20) (2022) 3414.

[66] S. R. Velu, V. Ravi, K. Tabianan, Multi-lexicon classification and valence-based sentiment analysis as features for deep neural stock price prediction, Sci 5 (1) (2023) 8.

[67] V. Khandelwal, H. Varshney, G. Munjal, Sentiment analysis based stock price prediction using machine learning, in: 2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT), IEEE, 2024, pp. 182–187.

[68] J. J. Sonia, A. Goenka, A. Jain, Stock price analysis using sentiment analysis of twitter data, in: 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), IEEE, 2024, pp. 01–07.

[69] X.-Y. Liu, H. Yang, Q. Chen, R. Zhang, L. Yang, B. Xiao, C. D. Wang, Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance, arXiv preprint arXiv:2011.09607 (2020).

[70] H. Yang, X.-Y. Liu, S. Zhong, A. Walid, Deep reinforcement learning for automated stock trading: An ensemble strategy, in: Proceedings of the first ACM international conference on AI in finance, 2020, pp. 1–8.

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

[72] 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.

[73] X. Long, M. Kampouridis, D. Jarchi, An in-depth investigation of genetic programming and nine other machine learning algorithms in a financial forecasting problem, in: IEEE Congress on Evolutionary Computation (CEC), 2022.

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

[75] 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).