# Using strongly typed genetic programming to combine technical and sentiment analysis for algorithmic trading.

Eva Christodoulaki, Michael Kampouridis
*School of Computer Science and Electronic Engineering*
*University of Essex*
*United Kingdom*
{ec19888, mkampo}@essex.ac.uk

***Abstract*— Algorithmic trading has become an increasingly thriving research area and a lot of focus has been given on indicators from technical and sentiment analysis. In this paper, we examine the advantages of combining features from both technical and sentiment analysis. To do this, we use two different genetic programming algorithms (GP). The first algorithm allows trees to contain technical and/or sentiment analysis indicators without any constraints. The second algorithm introduces technical and sentiment analysis types through a strongly typed GP, whereby one branch of a given tree contains only technical analysis indicators and another branch of the same tree contains only sentiment analysis features. This allows for better exploration and exploitation of the search space of the indicators. We perform experiments on 10 international stocks and compare the above two GPs' performances. Our goal is to demonstrate that the combination of the indicators leads to improved financial performance. Our results show that the strongly typed GP is able to rank first in terms of Sharpe ratio and statistically outperform all other algorithms in terms of rate of return.**

***Index Terms*—Technical Analysis, Sentiment Analysis, Genetic Programming, Algorithmic Trading**

## I. Introduction

With the surge of services and research papers regarding financial forecasting and algorithmic trading, we reach the conclusion that the topics are two of the most popular ones when implementing machine learning to. So far, the published papers have focused on the indicators that derive from technical analysis (TA), with sentiment analysis (SA) being used, mostly, in the last decade as a means to financial forecasting, with a rapid increase of papers after 2010 onward. Little research has been done in combining the indicators of TA and SA. We believe that since both of these individual analyses have been shown to be effective in financial forecasting and algorithmic trading, creating models that contain both TA and SA indicators has the potential to create more powerful trading strategies.

To achieve the above, in this paper we use a genetic programming (GP) algorithm. GP algorithms have been successfully used in many financial applications, including financial forecasting [1]. In this paper, we use a GP that allows terminals from both TA and SA indicators. The first GP variant includes in its terminal sets both TA and SA indicators, thus enabling GP trees to create trading strategies that contain both indicator types. The second GP variant also includes both TA and SA indicators in its terminal set, but assigns different types to these two indicator categories. As a result, only TA-type terminals are allowed in one side of GP trees, while only SA-type terminals are allowed in another side of the GP trees. This type constraint has the advantage of allowing the GP search to focus on the search space of each individual indicator type, thus perform better exploration and exploitation. We also introduce type constraints during crossover and mutation, to ensure legality of all trees.

Our aim is to demonstrate that combining the TA and SA indicators under the above strongly typed GP leads to trading strategies that have improved financial performance. We compare the performance of the above GPs with GPs that only contain one type of indicators, either TA or SA. We use 6 years' worth of data from 10 international companies.

The rest of this paper is organised as follows: firstly, we introduce related research work in Section II, and we also provide background information on the topic of financial forecasting in Section III. Later, we set forth the methodology as seen in Section IV and the experimental setup in Section V. Lastly, we present the experimental results and their analysis in Section VI, followed by the conclusion and further experiments we plan to include in our research in Section VII.

## II. Literature Review

In the literature review Section we dive into the works of previous researchers on the topic of financial forecasting and algorithmic trading, in papers using TA or SA features, as well as, the combination of those two analyses; including works that have implemented GP in their studies.

### A. Technical analysis

In this Section we will mention works that use technical analysis as indicators in machine learning algorithms, presenting based on their topic. Since the 1980s, many researchers have studied Artificial neural networks for financial forecasting. More recent studies are those of [2] using linear models, [3] utilizing TA indicators into a long short-term memory

(LSTM) model to forecast future trends of stock prices. One of the first papers to dive into GP for financial forecasting is that of [4], where the algorithm was able to outperform commonly used, non-adaptive, individual technical rules. Similar findings can be seen in other papers, too (e.g., see [5]–[9]). Generally, we observe from [10] and [1], that GP can evolve trading strategies, generating solutions that endure extreme market conditions, as well as, it can generate new solutions and optimize the solution parameters.

### B. Sentiment analysis

As previous publications did, we, also, aim to study the importance of events in financial forecasting. In this Section, we discuss notable works on sentiment analysis (SA) in order of publication date. One of the most significant papers on SA is that of [11], who using neural networks researched how to increase the predictive power of multivariate models using prior knowledge from newspaper headlines. An important addition is that of [12], who used support vector machines (SVM) with tree kernels and semantic frame parses in order to generalize from sentences to scenarios. Continuing, [13] produced an event-driven stock model, feeding news into a deep convolutional neural network (CNN). [14], also, considered the source of the sentiment, trying to understand the quality of the news, thus their impact to the stock movement.

### C. Technical and Sentiment analysis combination

There have, also, been papers that have combined the technical and sentiment analysis, presented in a chronological order. Although, less researchers have studied the combination of the two analyses, we observe that it can be financially profitable. [15] used event knowledge and standard information of companies to create a specific scenario. The authors proposed an external knowledge base to provide information on the events, so these events can be detected based on reasoning. An important addition to the publications has been this of [16], who used DNNs to predict stock price movements, through historical prices and online financial news, showing that adding financial news into a financial data set can improve the accuracy of the model. Another important study is that of [17], who used text mining on news from Reuters regarding the S&P500 index in a RNN and CNN hybrid model, to predict the price and intraday directional movement. Using financial news articles and a set of technical indicators into their hybrid model, they showed it performed better than the CNN in the same implementation and how useful TA and SA are in financial forecasting.

As we can see in Sections II-A - II-C, TA and SA have been used with various machine learning models so far, although there is a lack of papers that use SA into GP algorithms. Furthermore, we notice a lack of papers conducting research on the combination of TA and SA, as well as, using GP models to achieve that. Research in that area could generate financially profitable results and it seems to be worth searching into, due to the advantages of GP on producing white-box models, effective global search, good exploration and exploitation.

## III. Background Information

In this Section we present the background information related to technical and sentiment analysis, in order to give the reader a deeper understanding of the research.

### A. Technical analysis

By analyzing technical indicators, recognizing trends and patterns, we are able to estimate the stock market with higher accuracy, as well as, generate higher profits. Technical analysis has been used in various ways and it continues to be developed, as researchers and technical analysts rely on past prices and other metrics to generate useful indicators to understand the status of a company and the overall financial market.

In this research, we chose indicators that have been widely used for Technical analysis. More specifically, we use 6 different indicators in 2 different time periods, lasting for 5 and 10 days, respectively. The set consists of the *Moving Average*, the *Momentum*, the *Rate of Change (ROC)*, the *Williams' %R*, the *Midprice* and the *Volatility*; a total of 12 features. The mathematical definitions of these 6 indicators can be found below, from Equation 1-6. For more information, let $n \in \{5, 10\}$ be the size of the lookup window. $P_i$ defines the adjusted closing price at the $i$ day of this period, with the convention that the most recent adjusted closing price in the look back period is $P_n$, the first adjusted closing price in the same period is $P_1$, and the last date of the previous set of prices is $P_0$, which is important in order to find the price change for the *Volatility* indicator. We denote by $\mathsf{Close}$ the most recent closing price, and by $H_h$ and $L_l$ the highest high and the lowest low price over all days in the lookup window. Finally, we define by $\mathsf{Var}$ the sample variance over a dataset.

$$\mathsf{MovingAverage} = \frac{\sum_{i=1}^{n} P_i}{n} \tag{1}$$

$$\mathsf{Momentum} = P_n - P_1 \tag{2}$$

$$\mathsf{ROC} = \left(\frac{P_n}{P_1} - 1\right) \cdot 100 \tag{3}$$

$$\mathsf{Williams'\%R} = -100 \cdot \frac{H_h - \mathsf{Close}}{H_h - L_l} \tag{4}$$

$$\mathsf{Volatility} = \sqrt{\mathsf{Var}\left(\left\{\frac{P_i}{P_{i-1}} - 1\right\}_{i \in \{1,\ldots,n\}}\right)} \tag{5}$$

$$\mathsf{Midprice} = \frac{H_h - L_l}{2} \tag{6}$$

$\mathsf{MovingAverage}$ is used to smooth the data and helps with noise elimination and to identify trends. $\mathsf{Momentum}$ and $\mathsf{ROC}$ show the difference between the most recent adjusted closing price and the one $n$ days ago; $\mathsf{ROC}$, also, normalizes the price. Williams' $\%\mathsf{R}$ is an indicator that takes values between $0$ and $100$ and measures overbought and oversold

levels. Historical volatility measures past performance and is a statistical measure of the dispersion of returns over a given period of time; the higher the historical volatility value, the riskier the security is. The last indicator, Midprice, returns the midpoint value from two different input fields.

### B. Sentiment analysis

Sentiment analysis is the procedure of extracting the meaning out of sentences, articles, online comments, in order to find useful information. In the past decade, as the financial markets get influenced by events and stocks will increase/decrease along the available information and people's decisions, SA has gotten greater recognition for its contribution in increasing the accuracy in financial forecasting.

The events can be classified as positive, negative or neutral and can be used with various machine learning algorithms. CNNs are one of the most used algorithms for text classification, but they require great amount of classified data to perform with a high accuracy. Due to the lack of such data for news articles in the financial sector, we used popular specialized SA programs, as used in the relevant literature, i.e., TextBlob [18], SentiWordNet [19] and AFINN sentiment [20].

*TextBlob* is a Python library, offering a simple API assisting in calculating the polarity and subjectivity. *WordNet* is a lexical taxonomy of the English language, in which *SentiWordNet* is based upon. *SentiWordNet 3.0* is an enhanced lexical resource explicitly devised to support sentiment classification and opinion mining, containing a list of words classified as positive, negative, or neutral. Then, we use the weighted average of the classified words in the text and assign an overall percentage of the sentiment. *AFINN* sentiment is a popular lexicon for sentiment analysis developed by Finn Årup Nielsen, containing more than 3300 words with a polarity score associated with each word and we use the in-built function for this lexicon, which is available in Python.

We use these three programs on the full texts of the articles, their titles and their summaries, generating a total of 12 SA features, which can be found in in Table III in Section IV-B.

## IV. Methodology

For this Section, we first present the financial part, where we introduce the TA and SA methodology. Then, we showcase the GP algorithms and their fitness function, the GP operators and the trading algorithm, common in all four GP models.

### A. Financial analysis processes

*1) Technical analysis process:* We first downloaded the historical prices of the enterprises we are interested in via Yahoo! Finance. We utilize the information from the Adjusted Close, Close, High and Low columns available on the datasets to create the indicators and to use along when run in the GP algorithms, normalizing the values of the features to be between $[-1, 1]$. More information on the indicators is available in the Section III-A, regarding the background information of TA.

*2) Sentiment analysis process:* For SA, we needed to download the articles related to the companies via a made scrapper from the first to the twentieth pages of Google search engine results, searching for each company's name and during the same dates as in Section IV-A1. We used the GoogleNews library, which is offered in Python.

Due to the plethora of information that can be found online, some of the downloaded articles were not related to the companies we wanted to observe, thus we kept the articles that we could only find the name of the company and its stock market name in. Furthermore, in order to ensure that the articles have been downloaded correctly, we shortlisted them based on their length and only if they have at least 500 characters. Moreover, we saved articles in an ascending order, from the oldest to the most recent one.

We obtained the sentiment of the articles, their titles and summaries using the TextBlob polarity and subjectivity tool, the SentiWordNet and AFINN sentiment. Since more than one articles appeared in the same date, we used the average sentiment value of the features of the articles published at the same day. Continuing with normalizing the sentiment values to be between $[-1, 1]$.

Finally, we matched the dates of the sentiment features with their stock market dates, since we wanted to link the sentiment of the specific date with its stock market value. Furthermore, since the stock price is not open in the weekends but we still get articles at that time, we had to include the sentiment of the weekend days to that of the Fridays, as we expect that the sentiment of these three days will influence the stock price of Monday. Finally, for the days that do have a stock price value, but no articles, we kept them in the datasets, so we do not have any breaks in between the days, and we set their sentiment to be equal to $0$, indicating neutrality and/or no movement.

### B. Genetic programming

In this Section we call GP-TA the GP algorithm run with the TA features, and GP-SA, the one running on the SA features. GP-SATA is the name of the GP running with the combination of TA and SA features, and STGP-SATA (ST stands for *strongly typed*), indicating that some of its branches allow only specific TA or SA type.

*1) Model representation:* The models are formed with a tree structure, with the function nodes created based on the logical functions AND, OR, Greater than (GT) and Less than (LT). There is also a single If-Then-Else function, which acts as the root of the trees. The AND, OR, GT and LT functions are used by the following GPs: GP-TA, GP-SA, and GP-SATA. All these algorithms do not differentiate between TA and SA types, and thus allow any combination of functions and terminals. On the other hand, as GP-SATA-TB is a strongly typed GP that differentiates between SA and TA types, it has functions that only allow specific types. Hence, instead of a generalised GT, and LT function, there are two variants, one that takes TA indicators, and one that takes SA indicators. These variants are LT_SA, GT_SA, and LT_TA, GT_TA. The AND and OR functions also enforce types, where their first

branch always takes SA types of functions (e.g. LT_SA) and terminals (e.g. TEXTpol). The function set for all GP variants is summarised in Table I.

The terminal set also varies depending on the GP variant. For GP-TA, it includes only technical analysis indicators, summarised in Table II. The TA indicators are MovingAverage, Momentum, ROC, Williams' %R, Volatility and Midprice, for the two different periods of 5 and 10 days. For GP-SA, it includes only sentiment analysis indicators, which are summarised in Table III. These terminals consist of the text, title and summary of the polarity and subjectivity of TextBlob the polarity of SentiWordNet, and the AFINN sentiment.

TABLE I
FUNCTION SET

| Function set | |
|---|---|
| Function set (GP-TA, GP-SA, GP-SATA) | AND, OR, LT, GT |
| Function set (GP-SATA-TB) | AND, OR, LT_SA, GT_SA, LT_TA, GT_TA |

TABLE II
TERMINAL SET -TA

| Terminal set | |
|---|---|
| TA (for 5 and 10 days) | Moving Average<br>Momentum<br>ROC<br>Williams' %R<br>Volatility<br>Midprice<br>ERC |

TABLE III
TERMINAL SET -SA

| Terminal set | |
|---|---|
| SA-textBlob | TEXTpol, TEXTsub<br>TITLEpol,TITLEsub<br>SUMMpol,SUMMsub |
| SA-SentiWordNet | TEXTsenti, TITLEsenti, SUMMsenti |
| SA-AFINN | TEXTafinn, TITLEafinn, SUMMafinn<br>ERC |

As observed from Tables II and III, there is also a variable named Ephemeral Random Constant (ERC), which takes random values from $-1$ to 1, and acts as a threshold value to the indicators.

Figure 1 presents a sample tree for GP-SATA-TB. As we can observe, the root is an If-Then-Else statement. The second and third branch indicate a decision, 1 (buy) and 0 (hold). These parts of the GP trees (If-Then-Else, second branch being 1, and third branch being a 0) *are not evolved*. The only part of the trees that gets evolved is the first branch, which allows us to focus on the search on different ways of combining the TA and/or SA indicators. As we can observe in Figure 1, the first branch of this sample tree contains the AND statement. Given that this is the GP-SATA-TB variant, we have enforced the first child of the AND function to be SA-related (thus LT_SA, and TEXTpol), and the second child to be TA-related (thus GT_TA, and Moving Average). So this trees checks if the SA TEXTpol indicator is less than 0.7 and if the TA Moving Average indicator is greater than 0.4; if this is true, then it recommends to buy (1), otherwise to hold (0). The generated signals are then used by the trading strategy, found in Section IV-C.

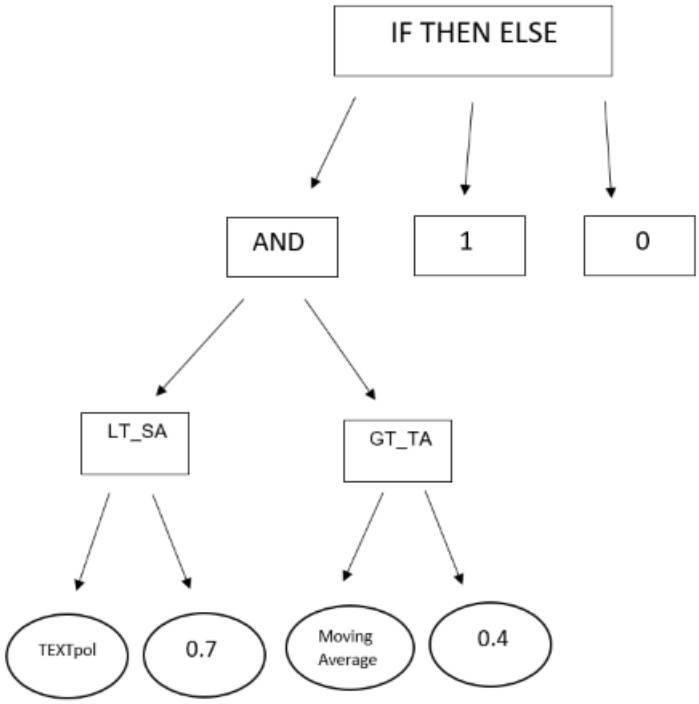


Fig. 1. Sample tree for GP-SATA-TB. The first branch of the AND statement is enforced to take SA-related nodes, and the second branch of the AND statement is enforced to take TA-related nodes. The 0.7 and 0.4 terminal values have been calculated through an Ephemeral Random Constant (ERC).

The above GP tree representation applies to all four GP variants. Thus, all of them use an If-Then-Else function as the root, and the second and third branch are 1 and 0, respectively. Only the first branch of the If-Then-Else function can vary, depending on the GP variant. The relevant functions and terminals are the ones already presented in Tables I - III.

*2) Fitness function:* In this research, we train the GP algorithms by maximising the *Sharpe ratio*. This holds as our fitness function. In Equation 7, we define the *Sharpe ratio* as:

$$\mathsf{SharpeRatio} = \frac{\mathsf{E}(R) - R_f}{\sqrt{\mathsf{Var}(R)}}, \tag{7}$$

E and Var denote the sample mean value and the sample variance, respectively. $R$ symbolises the *rate of return* and $R_f$ is the risk-free rate.The denominator of Equation 7 reflects the *risk*, as the standard deviation of the returns. The trading

algorithm that computed the *rate of return* and the *risk* can be found in Section IV-C.

*3) Genetic programming operators:* For the evolution of the trees, we used point mutation and subtree crossover. The crossover ratio is denoted as $p$, while the mutation probability is $1-p$. This is a common scheme, as seen in [21]. Lastly, we use elitism to ensure that the best individual of each generation is the one being copied to the next generation.

It should be noted that for GP-SA, GP-TA, and GP-SATA, the above operators can be applied to all GP individuals and nodes with no type constraints. On the other hand, for GP-SATA-TB, when a crossover is selected, it acts in two stages: first it takes place in the SA part of the tree, and then it takes place in the TA part of the tree. This is to ensure that both the SA and TA search spaces are being searched. With regards to mutation, we also ensure types are preserved; thus, a node of type SA can only be mutated to an SA-type, and similarly, a node of type TA can only be mutated to a TA-type.

### C. Trading algorithm

As introduced in Section IV-B, we get signal 1 if the feature is greater/lesser than the ERC, or 0 as a signal if otherwise. If the signal is 1, the model buys a stock and it later performs a trade by selling that stock depending on the number of days the evaluation took place (*n*) and the increase rate of reference (*r*). Assuming that $n = 10$ and $r = 0.05$, the above would be stated as "If in the next 10 days the price increases more than $5\%$ of what it was bought, we sell the stock, generating a trade. If otherwise, then the stock is sold at the final price of these 10 days."

$$R = \left\{ \frac{P - P_b}{P_b} \right\} \tag{8}$$

$$\mathsf{Risk} = \sqrt{\mathsf{Var}(R)} \tag{9}$$

The return from the trade relies on the closing price $P_b$, which is the one we bought the stock and the $P$ closing price, in which we sold the stock to form the trade. The returns from all the trades in a dataset are being saved in a list, and at the end we calculate the sample mean to find the overall *rate of return*. The *risk* is the standard deviation of the *returns* list. These can be seen in more detail at Equations 8 and 9. The *R*, in Equation 8, is used in Equation 7, to find the Sharpe ratio.

## V. Experimental Setup

### A. Data

We used 10 companies' data, news articles and historical prices, from 1st of January 2015 to 31st of December 2020, counting to 6 years. The companies we chose to use in this research are AMAZON, Activision Blizzard, Berkshire Hathaway, BlackBerry Limited, General Motors, IBM, Kodak, Tesla, Ubisoft, Xerox. We chose these 10 companies due to their different market movements and wanting to include companies with dissimilar services and products; aiming to search the advantages of combining TA and SA features into a GP algorithm, when maximising the *Sharpe ratio*.

For TA, we collected the price data for our technical indicators on, from Yahoo! Finance. SA data were articles, their titles and summaries that we downloaded by using a hand-made scrapper utilizing the Google Search Console API available in Python.

After gathering the data, we first compute the six TA features for the two periods each, i.e. 5 and 10 days, creating 12 indicators in total. For the SA features, we used TextBlob, SentiWordNet and AFINN sentiment, more information can be found in Section IV-B, to find the sentiment of the articles, their titles and summaries, generating another 12 features.

We separated all datasets into $60\%$ training, $20\%$ validation and $20\%$ for testing. In this research, we use the 12 features from the two analyses individually, along with the closing prices of the companies' stock, as inputs to GP-TA and GP-SA; and the 24 features in total, when combining the datasets for GP-SATA and GP-SATA-TB.

### B. Parameter tuning

The parameter tuning was completed in two steps. The first step was to perform a grid search on the validation set to find the possible GP parameters, regarding the population size, crossover probability (*p*), number of generations, tournament size and, lastly, maximum depth of the trees; keeping the trading parameters $n$ and $r$ constant. We decided upon keeping $n$ to be 30 and $r$ equal to $0.05$ as the time of the tuning would increase if adding two more variables. Moreover, the $n$ and $r$ should be different for each company, while the parameters in Table IV can be the same for the algorithms. The mutation probability is *1-p*, thus we did not include the parameter in the grid search. In Table IV, we can see the best set of parameters on the validation set for GP-TA, GP-SA, GP-SATA.

TABLE IV
GP Parameters

| **GP Parameters** | |
|---|---|
| Population size | 1000 |
| Crossover probability | 0.9 |
| Mutation probability | 0.1 |
| Generations | 30 |
| Tournament size | 4 |
| Maximum tree depth | 4 |

The same two steps apply for the parameter tuning of GP-SATA-TB. In Table V, we can see the parameters for GP-SATA-TB, after we performed a different grid search on the validation set for it, since the GP algorithm behaves in a different way, due to the type constraints.

When the parameter tuning for the GP algorithms was over, we continued to the second step, to select the trading parameters, i.e. number of days $n$ and increase rate $r$, as described in Section IV-C. The $n$ and $r$ are different for each company, and the tuning was set on the validation set, again.

## VI. Results and Analysis

In this Section we present the results and analysis derived from the execution of the four GP algorithms, along with a brief discussion at the end.

TABLE V
GP PARAMETERS GP-SATA-TB

| GP Parameters | |
|---|---|
| Population size | 500 |
| Crossover probability | 0.95 |
| Mutation probability | 0.1 |
| Generations | 30 |
| Tournament size | 4 |
| Maximum tree depth | 7 |

### A. Results and Analysis

Each GP algorithm was run 50 independent runs on the test set for each one of the 10 datasets. We united the runs of the 10 companies' results and presented the mean values of these results, for each algorithm. The means are calculated on results with non-zero values, which means we did not include the runs that did not have any trades. This is because we have observed that in certain cases, the GP would decide to not take any trading action throughout the test set, as this would otherwise have resulted in a loss. So, when this happens and the GP does not trade, the *Sharpe ratio*, *rate of return* and *risk* are still recorded as 0 in our results file, which can distort our summary statistics. Thus when there is no trading, we do not include those results into our calculations. Because we observed outliers in many datasets' runs, which were affecting the mean values of the metrics. Additionally, we are presenting the mean results that are within one, two, and three standard deviations from the mean. Thus, this allows us to exclude some extreme values from our results, which could be affecting the comparisons among the different GP algorithms. Typically within three standard deviations all four GP algorithms include around 97-99% of the complete distributions' observations.

*1) Sharpe ratio:* Table VI presents the mean *Sharpe ratio* values for each algorithm over 50 GP runs. The mean results for the overall (complete) distribution are presented first, and then we present the mean results for within one, two, and three standard deviations. The picture here is very consistent: GP-SATA-TB has the highest *mean Sharpe ratio* for the complete distribution, as well as within one, two, and three standard deviations from the mean. The second ranking algorithm is GP-TA, but its difference from GP-SATA-TB is at least 30% in all sets of data.

To support the above analysis, we perform a two-sample Kolmogorov-Smirnov (KS) test on all the runs of the four different sets of data, excluding the values of 0, which means excluding the runs that did not produce any trading action. The test was chosen since it is sensitive to differences in the location and the shape of the empirical cumulative distribution functions of the two samples. It reports the maximum difference between the samples' cumulative distributions and then computes a p-value. Furthermore, it works well with unequal sample size data, such as ours. To account for multiple comparisons among the four different algorithms, we perform the Bonferroni Correction where the p-value for a $5\%$ significance level is equal to $\alpha = \frac{0.05}{3} = 0.0167$. The denominator value represents the number of multiple comparisons among the GP algorithms, namely GP-SATA-TB vs GP-SATA, GP-SATA-TB vs GP-SA, and GP-SATA-TB vs GP-TA.

TABLE VI
MEAN SHARPE RATIO VALUES FOR THE COMPLETE DISTRIBUTION, AS WELL AS FOR WITHIN ONE, TWO, AND THREE STANDARD DEVIATIONS FROM THE MEAN. BEST VALUE DENOTED IN BOLDFACE.

| *Distribution* | *Algorithm* | *Mean* |
|---|---|---|
| Complete | GP-SATA | 0.761 |
| | GP-SA | 0.057 |
| | GP-TA | 1.493 |
| | GP-SATA-TB | **2.337** |
| One STDEV | GP-SATA | 0.3189 |
| | GP-SA | 0.297 |
| | GP-TA | 0.7321 |
| | GP-SATA-TB | **1.037** |
| Two STDEV | GP-SATA | 0.531 |
| | GP-SA | 0.651 |
| | GP-TA | 0.919 |
| | GP-SATA-TB | **1.327** |
| Three STDEV | GP-SATA | 0.5813 |
| | GP-SA | 0.7199 |
| | GP-TA | 1.01 |
| | GP-SATA-TB | **1.327** |

Table VII presents the KS test p-values for the comparisons against the best ranking algorithm (i.e. GP-SATA-TB). Results are again presented for the complete distribution, as well as for the one, two, and three standard deviation distributions. The null hypothesis is that the respective two distributions being compared come from the same continuous distribution. As we can observe, GP-SATA-TB statistically outperforms GP-SATA and GP-SA, while its difference with GP-TA is not significant at the 5% level. Nevertheless, the fact that it showed a higher mean value in Table VI means that GP-SATA-TB would be the preferred algorithm between them.

TABLE VII
KOLMOGOROV TEST P-VALUES ON SHARPE RATIO. STATISTICAL SIGNIFICANCE AT 5% LEVEL IS WHEN THE P-VALUE IS LESS THAN 0.0167, AS IT HAS BEEN ADJUSTED BY THE BONFERRONI CORRECTION TO ACCOUNT FOR MULTIPLE COMPARISONS.

| *Distribution* | *Algorithm* | *GP-SATA* | *GP-SA* | *GP-TA* |
|---|---|---|---|---|
| Complete | GP-SATA-TB | **4.17E-04** | **1.54E-05** | 0.0951 |
| One STDEV | GP-SATA-TB | **7.23E-06** | **1.67E-08** | 0.0537 |
| Two STDEV | GP-SATA-TB | **3.59E-04** | **3.48E-06** | 0.0903 |
| Three STDEV | GP-SATA-TB | **0.0011** | **7.61E-06** | 0.0681 |

*2) Rate of Return:* Table VIII presents the mean *rate of return* per trade over 50 runs per algorithm. We, again present the mean results for the complete distribution, as well as the mean results for within one, two, and three standard deviations. As we can observe, in term of the complete distribution, GP-SA has the highest return value (0.0279), while GP-SATA-TB comes second with the value of 0.0211. However, our analysis showed that there were a few outliers in the GP-SA results, which had a very positive effect on its mean values. When we exclude these outliers (one, two, and three standard deviations from the mean), GP-SATA-TB ranks first.

TABLE VIII
MEAN RATE OF RETURN VALUES FOR THE COMPLETE DISTRIBUTION, AS WELL AS FOR WITHIN ONE, TWO, AND THREE STANDARD DEVIATIONS FROM THE MEAN. BEST VALUE DENOTED IN BOLDFACE.

| *Distribution* | *Algorithm* | *Mean* |
|---|---|---|
| Complete | GP-SATA | 0.014 |
| | GP-SA | **0.0279** |
| | GP-TA | 0.00826 |
| | GP-SATA-TB | 0.0211 |
| One STDEV | GP-SATA | 0.0226 |
| | GP-SA | 0.0254 |
| | GP-TA | 0.0324 |
| | GP-SATA-TB | **0.0366** |
| Two STDEV | GP-SATA | 0.0126 |
| | GP-SA | 0.0291 |
| | GP-TA | 0.0335 |
| | GP-SATA-TB | **0.04123** |
| Three STDEV | GP-SATA | 0.0124 |
| | GP-SA | 0.0305 |
| | GP-TA | 0.023 |
| | GP-SATA-TB | **0.0328** |

The Kolmogorov-Smirnov test (p-values presented in Table IX) confirm the above results. As we can observe, there are statistically significant differences across the different comparisons. This means that while the GP-SA statistically outperforms all other algorithms for the complete distribution, this can be explained because of some extreme positive values that this algorithm has returned. However, when we exclude these extreme values, GP-SATA-TB ranks first and statistically outperforms all the other algorithms.

TABLE IX
KOLMOGOROV TEST P-VALUES ON RATE OF RETURN. STATISTICAL SIGNIFICANCE AT 5% LEVEL IS WHEN THE P-VALUE IS LESS THAN 0.0167, AS IT HAS BEEN ADJUSTED BY THE BONFERRONI CORRECTION TO ACCOUNT FOR MULTIPLE COMPARISONS.

| *Distribution* | *Algorithm* | *GP-SATA* | *GP-SA* | *GP-TA* |
|---|---|---|---|---|
| Complete | GP-SATA-TB | **0.0012** | **0.0092** | **0.0065** |
| One STDEV | GP-SATA-TB | **0.0011** | **2.63E-05** | **0.0098** |
| Two STDEV | GP-SATA-TB | **2.53E-05** | **5.81E-04** | **0.014** |
| Three STDEV | GP-SATA-TB | **5.77E-04** | **0.0036** | **0.0111** |

*3) Risk:* Table X presents the mean results for *risk* per trade over 50 runs for each of the four algorithms. We can observe here that GP-SA is the least risky strategy, as it ranks first even when we account for outliers. GP-SATA-TB ranks second for the complete and within one, two, and three standard deviation distributions, making it the second least risky trading strategy among the four algorithms discussed in this paper.

Table XI presents the p-values for the K-S tests. As GP-SA was the best ranking algorithm, we are now using it as the control algorithm, instead of GP-SATA-TB that we previously did in Tables VII and IX. What we can observe here is that GP-SA statistically outperforms GP-SATA and GP-TA, but it only outperforms GP-SATA-TB under one standard deviation. For the complete distribution, as well as for within two and three standard deviations, the differences in the mean risk values are not statistically significant at the 5% level.

TABLE X
MEAN RISK VALUES FOR THE COMPLETE DISTRIBUTION, AS WELL AS FOR WITHIN ONE, TWO, AND THREE STANDARD DEVIATIONS FROM THE MEAN. BEST VALUE DENOTED IN BOLDFACE.

| *Distribution* | *Algorithm* | *Mean* |
|---|---|---|
| Complete | GP-SATA | 0.1036 |
| | GP-SA | **0.065** |
| | GP-TA | 0.08607 |
| | GP-SATA-TB | 0.07961 |
| One STDEV | GP-SATA | 0.0651 |
| | GP-SA | **0.0292** |
| | GP-TA | 0.0565 |
| | GP-SATA-TB | 0.0415 |
| Two STDEV | GP-SATA | 0.07678 |
| | GP-SA | **0.0379** |
| | GP-TA | 0.067 |
| | GP-SATA-TB | 0.055 |
| Three STDEV | GP-SATA | 0.079 |
| | GP-SA | **0.051** |
| | GP-TA | 0.0728 |
| | GP-SATA-TB | 0.0596 |

TABLE XI
KOLMOGOROV TEST P-VALUES ON RISK. STATISTICAL SIGNIFICANCE AT 5% LEVEL IS WHEN THE P-VALUE IS LESS THAN 0.0167, AS IT HAS BEEN ADJUSTED BY THE BONFERRONI CORRECTION TO ACCOUNT FOR MULTIPLE COMPARISONS.

| *Distribution* | *Algorithm* | *GP-SATA-TB* | *GP-SATA* | *GP-TA* |
|---|---|---|---|---|
| Complete | GP-SA | 0.3189 | **4.26E-05** | **6.50E-04** |
| One STDEV | GP-SA | **0.0055** | **1.62E-12** | **2.06E-09** |
| Two STDEV | GP-SA | 0.0375 | **1.10E-08** | **3.20E-06** |
| Three STDEV | GP-SA | 0.315 | **5.24E-06** | **1.64E-04** |

### *B. Discussion*

Summing up our findings of the Tables VI - XI and focusing on the results of GP-SATA-TB, we observe that it does statistically outperform all the other algorithms in the *rate of return* results, while its *risk* is the second lowest and higher than that of GP-SA; in addition, the above return and risk performance has led GP-SATA-TB to have a statistically higher *Sharpe ratio* than GP-SATA and GP-SA.

On the other hand, while GP-TA has the second highest *Sharpe ratio* and *rate of return*, its *risk* is too high. On the contrary, *GP-SA* has the lowest *risk*, but with a trade-off in terms of *Sharpe ratio* and *rate of return*, which resulted in GP-SA ranking last for these two metrics.

We can thus conclude that GP-SATA-TB is the most robust algorithm out of the four, having higher *rate of return* compared to the other algorithms, and the second lowest *risk* value among the four different datasets.

It is also worth noting that simply combining SA and TA indicators (as GP-SATA did) is not enough to yield improved performance over the individual analysis indicators (GP-TA, GP-SA). In fact, GP-SATA often ranks last across all three financial metrics (Sharpe ratio, rate of return, risk). This thus demonstrates the importance of combining the TA and SA indicators under a typed GP, which ensures that effective search takes place in both the TA and SA search space.

### C. Computational times

In this section we discuss the computational times of each GP algorithm for a single run, under the GP parameters presented earlier in Section V-B. The time for all four models appears in minutes.

TABLE XII
COMPUTATIONAL TIMES (IN MINUTES) PER GP ALGORITHM

| Computational times per analysis | |
|---|---|
| GP-SATA | 1.1 |
| GP-SA | 1.17 |
| GP-TA | 1.12 |
| GP-SATA-TB | 0.24 |

As we can see in Table XII, all models except GP-SATA-TB have similar computational efficiency. This may be due to the type constraints we have introduced for GP-SATA-TB that assists the model into optimising the algorithm more efficiently and faster. Generally, the computational costs can be shortened by parallelizing the execution of the models, since candidate solution is produced individually from the rest in the GP algorithm's population. This has been tested in [22], where the authors managed to speed up the algorithms by up to 21 times.

## VII. CONCLUSION

To sum up, the goal of this paper was to investigate the performance of trading algorithms that combine technical and sentiment analysis indicators. We presented a novel GP, which introduced type constraints to ensure that its trees combine information from both analysis types. We compared this GP's performance to three other GP algorithms across 10 different sets of data. Our findings showed that our proposed GP is competitive and statistically outperforms the other algorithms in terms of *Sharpe ratio* and *rate of return*, while it ranks second in terms of *risk*.

The above finding is important for two reasons: firstly, because it demonstrates the significance of combining indicators information from both technical and sentiment analysis, which is not something that often happens in the literature. Combining this information can enhance the models' knowledge and lead to better performing trading strategies. However, what is also important is how this combination takes place. Our analysis showed that simply combining these indicators is not enough, as the relevant algorithm (GP-SATA) tended to rank in the last places for *Sharpe ratio*, *rate of return*, and *risk*. It is thus crucial to allow the GP to effectively search the spaces of both technical and sentiment analysis indicators, and identify the most promising features. Our proposed algorithm, GP-SATA-TB, did exactly this, resulting in improved performance when compared to the combination of SA and TA, as well as when compared to the performance of the individual analyses (GP-TA and GP-SA).

Future work will focus on investigating for ways to improve the risk performance of GP-SATA-TB, as well as increase the amount of datasets we examine to be able to further generalise our results.

## REFERENCES


[1] A. Brabazon, M. Kampouridis, and M. O'Neill, "Applications of genetic programming to finance and economics: past, present, future," *Genetic Programming and Evolvable Machines*, vol. 21, no. 1, pp. 33–53, 2020.

[2] M. M. Mostafa, "Forecasting stock exchange movements using neural networks: Empirical evidence from kuwait," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6302–6309, 2010.

[3] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1419–1426.

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

[5] M. Kampouridis and E. Tsang, "Investment opportunities forecasting: Extending the grammar of a gp-based tool," *International Journal of Computational Intelligence Systems*, vol. 5, no. 3, pp. 530–541, 2012.

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

[7] M. Kampouridis and F. E. Otero, "Heuristic procedures for improving the predictability of a genetic programming financial forecasting algorithm," *Soft Computing*, vol. 21, no. 2, pp. 295–310, 2017.

[8] X. Long, M. Kampouridis, and 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.

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

[10] J. M. Berutich, F. López, F. Luna, and D. Quintana, "Robust technical trading strategies using gp for algorithmic portfolio selection," *Expert Systems with Applications*, vol. 46, pp. 307–315, 2016.

[11] K. Kohara, T. Ishikawa, Y. Fukuhara, and Y. Nakamura, "Stock price prediction using prior knowledge and neural networks," *Intelligent Systems in Accounting, Finance & Management*, vol. 6, no. 1, pp. 11–22, 1997.

[12] B. Xie, R. Passonneau, L. Wu, and 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.

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

[14] M.-Y. Day and 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.

[15] K. Teymourian, M. Rohde, and 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.

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

[17] M. R. Vargas, B. S. De Lima, and 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.

[18] S. Loria, "textblob documentation," *Release 0.15*, vol. 2, p. 269, 2018.

[19] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *LREC*, N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, Eds. European Language Resources Association, 2010. [Online]. Available: http://nmis.isti.cnr.it/sebastiani/Publications/LREC10.pdf

[20] F. Å. Nielsen, "A new ANEW: evaluation of a word list for sentiment analysis in microblogs," in *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, ser. CEUR Workshop Proceedings, M. Rowe, M. Stankovic, A.-S. Dadzie, and M. Hardey, Eds., vol. 718, May 2011, pp. 93–98. [Online]. Available: http://ceur-ws.org/Vol-718/paper_16.pdf

[21] R. Poli, W. Langdon, and N. McPhee, *A Field Guide to Genetic Programming*. Lulu Enterprises, UK Ltd, 2008.

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