Fundamental, Technical and Sentiment Analysis for Algorithmic Trading with Genetic Programming

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

***Abstract*— Algorithmic trading is a topic with major developments in the last years. Investors rely mostly on indicators derived from fundamental (FA) or technical analysis (TA), while sentiment analysis (SA) has also received attention in the last decade. This has led to great financial advantages with algorithms being the main tool to create pre-programmed trading strategies. Although the three analysis types have been mainly considered individually, their combination has not been studied as much. Given the ability of each individual analysis type in identifying profitable trading strategies, we are motivated to investigate if we can increase the profitability of such strategies by combining their indicators. Thus, in this paper we propose a novel Genetic Programming (GP) algorithm that combines the three analysis types and we showcase the advantages of their combination in terms of three financial metrics, namely Sharpe ratio, rate of return and risk. We conduct experiments on 30 companies and based on the results, the combination of the three analysis types statistically and significantly outperforms their individual results, as well as their pairwise combinations. More specifically, the proposed GP algorithm has the highest mean and median values for Sharpe ratio and rate of return, and the lowest (best) mean value for risk. Moreover, we benchmark our GP algorithm against multilayer perceptron and support vector machine, and show that it statistically outperforms both algorithms in terms of Sharpe ratio and risk.**

***Index Terms*—Algorithmic Trading, Genetic Programming**

I. Introduction

Algorithmic trading uses pre-programmed strategies to generate profits and has gained popularity as more companies enter the industry. Researchers use Machine Learning (ML) to analyse historical stock market data and identify patterns, generating signals for upcoming trend changes to maximise profits.

There are three well-known analysis types used in financial forecasting or algorithmic trading: fundamental analysis (FA), which evaluates economic and financial factors of companies; technical analysis (TA), which studies price trends and patterns to identify trading opportunities; and sentiment analysis (SA), which uses macroeconomic events to predict future stock prices. Historically, researchers have focused on fundamental and technical analysis indicators like earns per share (EPS), volatility and moving average, but sentiment analysis indicators, such as sentiment polarity, have become increasingly popular in recent years.

There are numerous examples in the literature showcasing the ability of each analysis type to provide profitable and low-risk trading strategies. However, due to the fact that each one represents a different school of thought when it comes to approaching and analysing market behaviour, there have been limited studies in creating trading strategies that combine indicators from the above three analysis types. The above can be considered as a 'missed opportunity' in the field of algorithmic trading, given the excellent trading performance of each individual types.

In this work, we propose a novel genetic programming (GP) algorithm that uses indicators from all analysis types, i.e. FA, SA and TA. We aim to show that the GP is able to create novel and profitable trading strategies that are able to consider information across all three analysis types, but, also, that the derived trading strategies are able to significantly outperform the trading performance of strategies derived only by FA, TA, or SA.

Genetic programming is chosen as it is able to perform effective search in large search spaces; in addition, it has been shown to be successful in different financial applications [1]. To compare the trading performance of the proposed algorithm, which we call GP-FASATA, we also create GP algorithms that use the three analysis types individually (FA, SA, TA) and in pairwise combinations (FATA, FASA, SATA). All algorithms are evaluated using five years' worth of data on 30 international companies, based on three financial metrics, namely Sharpe ratio, rate of return, and risk.

The rest of this paper is organised as follows: First, we present the related work in the Literature Review (Section II),while the background information and the methodology can be found in Methodology (Section III). Finally, we present the experimental setup (Section IV), the results and the analysis (Section V), and the conclusion and future works (Section VI).

II. Literature Review

In this section we look into related studies on algorithmic trading, using indicators from fundamental, technical and/or sentiment analysis.

Fundamental analysis (FA) evaluates a security's intrinsic value by studying financial statements, industry trends, and management quality, along with other financial and economic factors. Financial analysis indicators have been used with Deep Neural Networks [2], as well as hybrid models, a fusion of linear regression and XGBoost model [3].

Technical analysis (TA) finds trading opportunities by analysing past prices and trends in the stock market, while the analysis' indicators are a popular research topic. [4] used a long short-term memory (LSTM) model for future trends prediction, while [5] showed that a GP algorithm could outperform commonly used technical techniques. More studies like [6]–[8], achieved similar results.

The stock market movement can be influenced by global events, so previous related works on sentiment analysis (SA) have studied the importance of events and news, as in [9] with neural networks, [10] who proposed an event-driven stock model, and [11] utilising GP algorithms.

In addition, there has been limited works that combine pairs of the three analysis types. For example, [12], [13] combined FA and TA indicators and demonstrated that this resulted into trading strategies with more explanatory power. Similarly, there's been some limited works combining FA and SA, focusing on the sentiment of 10-K filings, such as [14]. Combining TA and SA indicators has been more popular in the literature, e.g. [15] used a RNN and CNN hybrid model, [16] used reinforcement learning with stock price data and news headline sentiments, and [17], [18] using genetic programming and achieved competitive results. Lastly, it is worth noting that there have been limited works that combine all three analysis types, such as [19]–[21].

As we can observe from the above review, existing studies show algorithmic trading with FA, TA, and SA indicators can be profitable, however little research combines these analysis types. In addition, there has been no work, to the best of our knowledge, which uses genetic programming to combine information from all three analysis types. Given the fact that in our previous works [17], [18] we have demonstrated that a GP that combines TA and SA indicators is able to statistically and significantly outperform algorithms that only use TA or SA, we are motivated in proposing in this paper a novel GP that also includes fundamental analysis, thus combining FA, SA, and TA. We believe that this can lead to further improvements in the profitability and risk of the trading strategies.

III. Methodology

In this section, we first introduce the indicators that we will use from each financial analysis type, and then we will present how they are used under the GP algorithm.

A. Financial analysis processes

This section covers the processes of fundamental, technical and sentiment analysis in three different subsections.

1) Fundamental analysis: Fundamental analysis (FA) is used by investors/researchers for profit generation and has been used ever seen to understand the financial status of companies, whether they are over/under-priced and to generate higher profits. In this work, we use 12 different indicators, namely Net Profit ratio, Return on Equity, Quick ratio, Debt to Equity, Price-Earnings ratio, Price to Book ratio, Price-Sales ratio, Total Revenues, levered free Cash Flow, Diluted earnings per share (EPS), earnings before interest, taxes, depreciation and amortization (EBITDA), and research and development (R&D) expenses.

This data was found in the 10-K filings of the financial content service company *Seeking Alpha* and they are on the companies used to conduct this research. More specifically, the five indicators of Diluted EPS, Total Revenues, EBITDA, levered free Cash Flow and R&D Expenses are already given in the 10K-filings. The remaining seven can be found in the equations below. In the equations below, y denotes the financial year, price denotes the price of the stock and j a specific day during the 5 year period.

The Net Profit ratio (NetProf) is a measure of profitability and is calculated by dividing the Net Income with the Revenue.

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

Return on Equity (RoE), also, measures the profitability and it is normally understood as the return on assets minus the liabilities. RoE is calculated by dividing the Net Income with the Shareholders' Equity.

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

Quick ratio (QuickR) is a liquidity ratio, which quantifies the ability of a business to utilise its cash or quick assets to pay off its liabilities immediately. It is found by dividing the Current Asses minus the Inventory of a company with the Current Liabilities.

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

Debt to Equity (DebtEq), measures how much of the shareholders' equity and debt used to finance a company's assets. It is calculated when dividing the Total Debt with the Shareholders' Equity.

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

Price-Earnings ratio (P/E) is found when we divide the Share Price (p) with the Earnings per Share (EPS) and it is used to value companies, finding out whether they are overvalued or undervalued.

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

Price to Book ratio (P/B) is found when we divide the Share Price (p) with the Book value per share (BVPS) and it is used to identify potential investments.

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

Price-Sales (P/S) is calculated by dividing the Share Price (p) with the Revenue per share (Revenue) and it is meant to find the value that financial markets have placed on each dollar of business's revenues.

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

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

2) Technical analysis: Technical analysis (TA) is an common analysis type in algorithmic trading and researchers utilise technical analysis indicators daily, in order to recognise trends in the stock market. This study uses six widely-adopted technical analysis indicators, namely the Moving Average, the Momentum, the Rate of Change (RoC), the Williams %R, the Midprice and the Volatility, as shown in Equations (8) - (13). To calculate the indicators, historical data on (adjusted) closing prices, highest and lowest daily prices of selected companies, available on Yahoo! Finance were used. Every indicator is considered based on two look-up windows, $n = 5$ and $n = 10$ days.

The Moving Average is defined as follows, and is used to smooth out the data, helping with noise elimination towards identifying trends. p_j is the adjusted closing price of the j-th day for a corresponding stock.

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

The Momentum captures the difference between the most recent adjusted closing price and the adjusted closing price d days ago.

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

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

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

Volatility is a statistical measure of the dispersion of returns over a given period of time, with Var defining the sample variance over a dataset.

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

Williams $\%$R in Equation (12), reflects the level of most recent closing price, cl_j (at day j), to the highest high price, $hh_{j,n}$, of all values in the lookup window ending at day j. $ll_{j,n}$ denotes the lowest low price over all days in the lookup window ending at day j.

$$\text{Williams }\%\text{R}(j) = -100 \cdot \frac{hh_j - cl_j}{hh_j - ll_j} \tag{12}$$

$Midprice$ returns the midpoint value of the highest high price, $hh_{j,n}$, and the lowest low price, $ll_{j,n}$, over all days in the lookup window ending at day j.

$$\text{Midprice}(j) = \frac{hh_j - ll_j}{2} \tag{13}$$

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

3) Sentiment analysis: Sentiment analysis (SA) extracts sentiment from online articles of fluctuating global events with the aim to improve trading strategies and increase profits. Two commonly used SA 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 this study we use indicators of we consider 12 SA indicators, all normalised between $[-1, 1]$.

In sentiment analysis research, it is widely adopted to use specialised SA programs, namely TextBlob [22], SentiWordNet [23] and AFINN sentiment [24] for calculating the polarity and/or subjectivity of texts. *TextBlob* is a Python library which offers a simple API to calculate the polarity and subjectivity of the text. *SentiWordNet 3.0* is an enhanced lexical resource, based on lexical taxonomy of the English language, made for sentiment classification. *AFINN* sentiment is a popular lexicon for sentiment analysis, having more than 3300 words with a polarity score to each one of them, developed by Finn Årup Nielsen. In our research, all programs are available and used in Python. Our sentiment analysis indicators consider the polarity and subjectivity levels extracted by TextBlob, as well as the sentiment polarity extracted by *SentiWordNet* and *AFINN*. The online articles, their titles and summaries are examined individually, giving a total of 12 SA indicators, namely: TEXTpol (TextBlob), TEXTsub (TextBlob), TITLEpop (TextBlob), TITLEsub (TextBlob), SUMMpol (TextBlob), SUMMsub (TextBlob), TEXTsenti (SentiWordNet), TITLEsenti (SentiWordNet), SUMMsenti (SentiWordNet), TEXTafinn (AFINN), TITLEafinn (AFINN), and SUMMafinn (AFINN).

The analysis of the SA indicators included downloading articles relevant to the companies in question and associating their sentiment with the corresponding date and price changes. The development of a web scrapper using the Google Search Console API in Python was needed to download the first twenty pages of daily Google Search results, using the name of each company as a keyword. After downloading all the articles, the $cleaning$ process started by narrowing down to the articles that contain at least 500 characters and that they included both the name of the corresponding company and its stock market ticker. This helped ensure that only relevant to the research articles were included.

One of the last steps was to match the dates of the articles' appearance with the relevant stock price data. For articles appearing on weekends while the stock market is closed, the sentiment was included to that of Friday's, to capture their influence on the stock price of the following day (Monday). In cases where more than one articles were appearing for the same company on the same date, the average sentiment value for that day was calculated. For the days with no articles, a sentiment of 0 was assigned to indicate neutrality and/or no action, to ensure continuity of our data points.

B. Genetic programming

We present the proposed GP-FASATA algorithm, which uses indicators from fundamental, sentiment, and technical analysis, under a genetic programming algorithm.

1) Model representation: When creating the individuals (trees), the inner nodes are composed of the logical functions AND, OR, Greater than (GT) and Less than (LT), while the terminal set includes all indicators presented earlier in Section

III-A, along with an Ephemeral Random Constant (ERC), which takes random real values from -1 to 1, and it acts as a threshold for the indicators, i.e., the algorithm checks whether the value of the indicator is greater than (or less than) this random value, as part of maximising the *Sharpe ratio*. Table I lists all terminal set indicators.

TABLE I
TERMINAL SET INDICATORS FOR FUNDAMENTAL, SENTIMENT, AND TECHNICAL ANALYSIS

Analysis type	**Indicator**
Fundamental Analysis	Net Profit ratio, Return on Equity, Quick ratio, Price-Earnings ratio, Price to Book ratio, Price-Sales ratio, Debt to Equity, Total Revenues, levered free Cash Flow, Diluted EPS, EBITDA, R&D Expenses ERC
Sentiment Analysis(TextBlob)	TEXTpol, TEXTsub TITLEpol, TITLEsub SUMMpol, SUMMsub
Sentiment Analysis(SentiWordNet)	TEXTsenti, TITLEsenti, SUMM-senti
Sentiment Analysis(AFINN)	TEXTafinn, TITLEafinn, SUM-Mafinn ERC
TA (for 5 and 10 days)	Moving Average Momentum ROC Williams' %R Volatility Midprice ERC

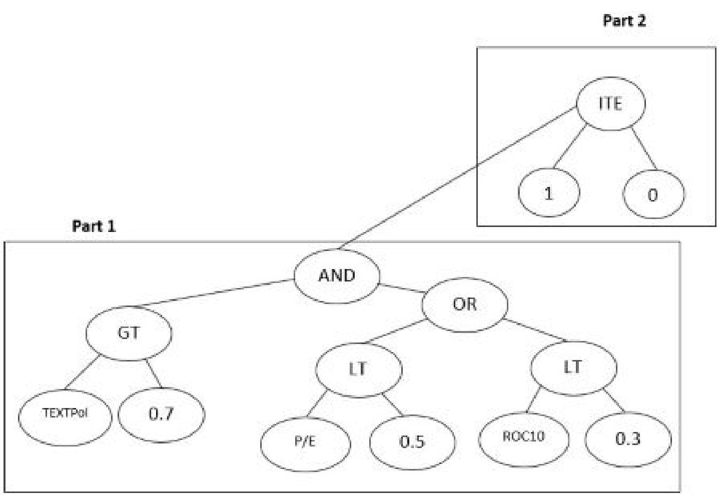

Fig. 1. This individual includes FA, SA and TA indicators and it is an example tree the GP-FASATA algorithm can produce. In particular, the indicators represented are the Sentiment polarity for the article's summaries as a SA indicators, the P/E indicator of FA and and ROC of 10 days as the TA indicator. Random numbers from -1 to 1 would be in the place of the Ephemeral Random Constant (ERC).

Part 1 of Figure 1 presents a sample image of a GP individual. Every GP individual that is being evolved is embedded into another tree, which has an If- Then-Else (ITE) statement as its root. The first branch of the tree is the evolved GP tree (Part 1). The second and third branches are the buy (1) and hold (0) actions, which we will further discuss in Section III-C. Part 2 is not evolved, since their values remain always the same, so there was no need to include them in the GP algorithm. As we can observe, this sample tree uses one SA indicator (TestPol), one FA indicator (P/E) and one TA indicator (Volatility), thus utilising indicators from all analysis types. We will discuss how a sell action is performed in Section III-C.

2) GP operators: Genetic Programming algorithms are evolving a population of candidate solutions through the genetic operations of crossover and mutation. 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 mutation with the remaining probability, $1 - p$. Our study uses the operators of sub-tree crossover and point mutation. We also use elitism to ensure the best individual of a generation is copied to the next generation.

C. Trading algorithm

As explained earlier, each evolving GP model is incorporated into another tree structure 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, see Figure 1. When the GP tree returns 1, one amount of stock is bought. To sell, the trading strategy considers whether the price increases by $r\%$ within the next d days; in this case, the strategy sells one amount of stock. Alternatively, the strategy sells the stock on the d^{th} day. This is always considered after a buy action has taken place.

We record the return (Equation 14) of each trade into a list, from which we calculate the Sharpe ratio (Equation 15), rate of return and risk. We evaluate the performance of the trading algorithm based on these metrics and we compare their results.

D. Fitness function and Metrics

In these sections we introduce the metrics of *rate of return*, *risk* and *Sharpe ratio*, which are defined as follows.

The *return*, R, of a trade captures the profit made as a percentage of the amount invested. The calculation of the profit takes into account the transaction cost of 0.025% of the selling price (c_t). In particular, the *return* is calculated as shown in Equation (14), 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.

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

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, which is $\sqrt{var[R]}$. Lastly, the *Sharpe ratio*, S_a, 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,

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

The *fitness function* of the proposed GP algorithm, GP-FASATA, is defined as the maximisation of the Sharpe ratio. The Sharpe ratio is a financial concept that evaluates the return on an investment strategy relative to its risk, aiding investors in making informed decisions by assessing the risk of a stock or company and determining if the potential return justifies it.

IV. Experimental Setup

A. Data

In our research, we analysed 30 international companies, from different stock exchanges, using 10K-filings, historical stock prices, and relevant news articles. The research period was 5 years from January 1st, 2015, to January 31st, 2020.

For the fundamental analysis data, 10K-filings were downloaded from Seeking Alpha. Daily closing price data for technical analysis was retrieved from Yahoo! Finance. Sentiment analysis was performed by scraping articles, titles, and summaries using a Python scraper and the Google Search Console API. Once all the required data was collected, we generated 36 relevant indicators, popular in the financial domain, 12 for each analysis indicator as outlined in Section III. Subsequently, we divided the datasets of 30 companies into three parts in sequence: 60% for training, 20% for validation, and 20% for testing.

B. Benchmarks

The proposed GP-FASATA algorithm is benchmarked against six other GP algorithms. GP-FA, a GP algorithm with only fundamental analysis indicators on its terminal set, GP-TA and GP-SA, having technical and sentiment analysis indicators, respectively, and their pairwise combinations, i.e. GP-FASA, GP-FATA, and GP-SATA. Furthermore, two widely-used benchmarks, implemented using scikit-learn library in Python, the Multilayer perceptron (MLP) and Support vector machine (SVM) algorithms were considered as benchmarks too. They were used to solve a binary classification problem of predicting whether the stock price will increase within the next d days. Class 1 indicates a buy action, while Class 0 represents a hold action. These comparisons are made to understand the financial advantages of the combination of the three data types, as well as, to understand the possible implications of their unison and the disadvantages of the individual analysis types.

C. Parameter tuning

A two-step grid search was done on the validation set to determine the optimal GP parameters. The grid search involved adjusting the population size, number of generations, tournament size, maximum depth of the trees and the crossover probability (p)[1]. The trading parameters d and r (presented in Section III-C), were kept constant at 30 and 0.05, respectively, to reduce the time for parameter tuning. We identified a set of parameters that worked equally well and without any statistical differences for all GP variants on the validation set.

[1]Since the mutation probability can be calculated as 1-p, it was not required to include it as a separate parameter during the tuning process.

The parameters in Table II were used in all runs for all GP algorithms and companies.

TABLE II
GP Parameters for GP-FASATA

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

To improve the trading performance, the parameters d and r of the trading strategy were adjusted independently for each company. The selection of d and r was based on their overall performance across the GP algorithms, and this process was carried out using the validation set.

MLP and SVM are tuned separately using binary classification, with the best model being chosen based on its predictive ability on the validation set. This model's predicted class is used as signals for the trading strategy, with the same d and r parameters as the GP-variants. The tuning process for these machine learning algorithms for trading is based on [25].

V. Results and Discussion

This section presents the results of 50 independent runs for each of the GP algorithms on 30 companies, comparing GP-FASATA with the benchmarks from Section IV-B. Each run generated a tree/model to maximise the Sharpe ratio and was evaluated on the test set.

In our analysis, we present only the runs where the GP algorithm performs at least two trades, since the runs with zero or one trades would distort the statistical analysis as the risk (and RoR and Sharpe ratio, for zero trades) would be 0.

We performed a Kolmogorov-Smirnov test on the pairwise comparisons of GP-FASATA with the GP-benchmarks under the null hypothesis that the two samples come from the same population. Furthermore, we performed the Holm-Bonferroni correction to account for the multiple comparisons. In the Holm-Bonferroni correction the minimum acceptable p-value for a statistical significance at a 5% significance level is equal to $\alpha(rank) = \frac{0.05}{6-rank+1}$, where $rank \in \{1, 2, 3, 4, 5, 6\}$ and it differs for the different ranks of the p-values found.

We compared GP-FASATA to 6 GP benchmarks, resulting in 6 different comparisons for each financial metric. Rank indicates p-value magnitude order, with 1 as smallest and 6 as largest. The ranked p-values show a significant difference between the samples: the first p-value should be less than 0.0083, second less than 0.01, third less than 0.0125, fourth less than 0.0166, fifth less than 0.025, and sixth less than 0.05.

A. Summary statistics on financial metrics

1) GP algorithms - Sharpe ratio: Table III presents the mean, median and standard deviation (StDev) for the Sharpe ratio values for each algorithm over the 50 independent GP runs on the 30 companies. As we can observe, GP-FASATA has the highest mean and median Sharpe ratio values, while the standard deviation is the fourth highest out of the 7 algorithms, but the lowest in the 4 algorithms that combine the data types.

TABLE III
STATISTICAL ANALYSIS ON SHARPE RATIO VALUES. BEST VALUES DENOTED IN BOLDFACE.

Algorithm	*Mean*	*Median*	*StDev*
GP-FA	2.32	1.22	3.32
GP-SA	3.16	1.46	4.84
GP-TA	2.69	1.74	4.31
GP-FASA	3.67	1.25	9.42
GP-FATA	1.82	1.43	**9.96**
GP-SATA	3.44	1.46	8.25
GP-FASATA	**4.15**	**2.35**	6.52

Table IV presents the Kolmogorov-Smirnov test p-values for the comparisons against GP-FASATA, which is the control algorithm. We can observe that, with the exception of GP-TA's p-value, the GP-FASATA p-values (second column) are lower than the corresponding significance level values (fourth column) for all other comparisons. Thus, this indicates that GP-FASATA's Sharpe Ratio results are significantly better than those of GP-FA, GP-SA, GP-FASA, GP-FATA, and GP-SATA.

TABLE IV
PAIRWISE KOLMOGOROV-SMIRNOV TEST P-VALUES ON SHARPE RATIO OF THE PROPOSED GP-FASATA ALGORITHM AGAINST THE 6 GP BENCHMARKS. STATISTICAL SIGNIFICANCE CHANGES BASED ON THE HOLM-BONFERRONI CORRECTION. STATISTICALLY SIGNIFICANT DIFFERENCES AT THE 5% LEVEL ARE DENOTED IN BOLDFACE.

Algorithm	*GP-FASATA p-values*	**Rank**	*Significance level*
GP-FA	**6.39E-08**	2	0.01
GP-SA	**1.30E-17**	1	0.0083
GP-TA	0.31	6	0.05
GP-FASA	**4.10E-08**	3	0.0125
GP-FATA	**0.0008**	5	0.025
GP-SATA	**0.00025**	4	0.0166

With regards to MLP and SVM's performance, they achieved a mean Sharpe ratio of 0.31 and 0.33, respectively, which is much lower than GP-FASATA's (4.15). This result is also confirmed by the Kolmogorov-Smirnov tests, which return a p-value of $2.37e-05$ for both comparisons, thus indicating a very strong statistically significant difference.

2) GP algorithms - Rate of Return: Table V presents the mean, median and standard deviation values on the *rate of return (RoR)* per algorithm. GP-FASATA has, again, the highest mean and median, while GP-TA is the one with the highest standard deviation value. The advantage of the indicators' combination is evident in the median RoR since all algorithms, except GP-FASATA, perform with a median value for RoR below 0.10.

The Kolmogorov-Smirnov test p-values in Table VI, show that the mean RoR results of GP-FASATA to be statistically significant and to statistically outperform the mean values of all the benchmarks, including GP-TA that has the highest p-value and ranks last in the Holm-Bonferroni correction.

TABLE V
STATISTICAL ANALYSIS ON RATE OF RETURN VALUES. BEST VALUES DENOTED IN BOLDFACE.

Algorithm	*Mean*	*Median*	*StDev*
GP-FA	0.012	0.010	0.020
GP-SA	0.0089	0.0097	0.020
GP-TA	0.011	0.0089	**0.028**
GP-FASA	0.011	0.009	0.021
GP-FATA	0.012	0.007	0.022
GP-SATA	0.013	0.007	0.021
GP-FASATA	**0.016**	**0.014**	0.022

TABLE VI
PAIRWISE KOLMOGOROV-SMIRNOV TEST P-VALUES ON RATE OF RETURN OF THE PROPOSED GP-FASATA ALGORITHM AGAINST THE 6 GP BENCHMARKS. STATISTICAL SIGNIFICANCE CHANGES BASED ON THE HOLM-BONFERRONI CORRECTION. STATISTICALLY SIGNIFICANT DIFFERENCES AT THE 5% LEVEL ARE DENOTED IN BOLDFACE.

Algorithm	*GP-FASATA p-values*	**Rank**	*Significance level*
GP-FA	**1.74E-05**	3	0.0125
GP-SA	**2.56E-17**	1	0.0083
GP-TA	**0.012**	6	0.05
GP-FASA	**6.39E-08**	2	0.01
GP-FATA	**0.0076**	5	0.025
GP-SATA	**0.00087**	4	0.0166

Regarding rate of return, MLP and SVM yielded a mean return of 0.009 and 0.015, respectively. The statistical test did not show any statistical difference between them and GP-FASATA with a p-value of 0.24 and 0.59, respectively. However, it is important to note that the proposed GP-FASATA still yields higher mean return (0.016).

3) GP algorithms - Risk: Table VII summarises the mean, median and standard deviation results on risk for each of the algorithms. Although the differences in risk are smaller than in the other metrics, GP-FASATA has the lowest mean, while GP-SATA has the lowest median risk and GP-FATA has the lowest standard deviation out of all the 7 algorithms.

TABLE VII
STATISTICAL ANALYSIS ON RISK VALUES. BEST VALUES DENOTED IN BOLDFACE.

Algorithm	*Mean*	*Median*	*StDev*
GP-FA	0.027	0.026	0.016
GP-SA	0.026	0.020	0.024
GP-TA	0.027	0.020	0.022
GP-FASA	0.025	0.023	0.018
GP-FATA	0.023	0.021	**0.013**
GP-SATA	0.024	**0.018**	0.019
GP-FASATA	**0.022**	0.021	0.015

For risk, GP-FASATA again statistically outperforms all benchmark algorithms, as in Table VIII.

When it comes to risk, MLP and SVM performed at 0.042 and 0.0418, respectively, while GP-FASATA is statistically different from them with a p-value of 0.006 and 0.015 at the 5% significance level.

TABLE VIII
PAIRWISE KOLMOGOROV-SMIRNOV TEST P-VALUES ON RISK OF THE PROPOSED GP-FASATA ALGORITHM AGAINST THE 6 GP BENCHMARKS. STATISTICAL SIGNIFICANCE CHANGES BASED ON THE HOLM-BONFERRONI CORRECTION. STATISTICALLY SIGNIFICANT DIFFERENCES AT THE 5% LEVEL ARE DENOTED IN BOLDFACE.

Algorithm	***GP-FASATA p-values***	**Rank**	***Significance level***
GP-FA	**2.40E-05**	3	0.0125
GP-SA	**2.35E-21**	1	0.0083
GP-TA	**0.0044**	6	0.05
GP-FASA	**9.15E-10**	2	0.01
GP-FATA	**0.0012**	5	0.025
GP-SATA	**0.0004**	4	0.0166

VI. CONCLUSION AND FUTURE WORK

Our paper explores the performance of a genetic programming algorithm that incorporates fundamental, sentiment, and technical analysis indicators. Through experiments on 30 companies, we found that combining the three data types yielded competitive results compared to individual analysis or algorithms that combine only two analysis types. More specifically, GP-FASATA outperformed the benchmarks in all three metrics of Sharpe ratio, rate of return and risk.

The above demonstrates the significant advantages of combining indicators from FA, SA, and TA. While each analysis type is able to provide good trading results individually, these results can be further improved by combining each terminal set. It is also worth noting that in the majority of results, particularly for Sharpe ratio and Rate of Return, we also observed that the pairwise combinations (i.e. GP-FASA, GP-FATA, GP-SATA) was not always better than the individual analysis algorithms (i.e. GP-FA, GP-SA, GP-TA). This indicates that while there can be advantages by using the above analysis types individually, the real advantages come when we combine all three analysis types, which appear to be complimentary to each other.

Future work includes creating a strongly typed GP, which will assign a different branch for each one of FA, SA, and TA terminals. This will allow the search to focus on each indicator type separately with the goal of improving the quality of the search and return even better trading 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. Abe and H. Nakayama, "Deep learning for forecasting stock returns in the cross-section," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 273–284.

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

[4] D. M. Nelson, A. C. Pereira, and 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] J. L. Bettman, S. J. Sault, and E. L. Schultz, "Fundamental and technical analysis: substitutes or complements?" *Accounting & Finance*, vol. 49, no. 1, pp. 21–36, 2009.

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

[14] M. Azimi and A. Agrawal, "Is positive sentiment in corporate annual reports informative? evidence from deep learning," *Evidence from Deep Learning (July 2019)*, 2019.

[15] 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.

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

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

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

[19] B. Weng, M. A. Ahmed, and F. M. Megahed, "Stock market one-day ahead movement prediction using disparate data sources," *Expert Systems with Applications*, vol. 79, pp. 153–163, 2017.

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

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

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

[23] 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

[24] 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

[25] 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.