Combining Technical and Sentiment Analysis under a Genetic Programming algorithm

Eva Christodoulaki and Michael Kampouridis

School of Computer Science and Electronic Engineering,
University of Essex, Wivenhoe Park, UK
{ec19888,mkampo}@essex.ac.uk

Abstract. Throughout the years, a lot of interest has been given to algorithmic trading, due to development of the stock market and provided securities. In the field of algorithmic trading, genetic programming (GP) is a very popular algorithm, due to its ability to produce white-box models, effective global search, and good exploration and exploitation. In this paper, we propose a novel GP algorithm to combine the features of two financial techniques. Firstly, technical analysis that studies the financial market action by looking into past market data. Secondly, sentiment analysis, which is used to determine the sentiment strength from a text in order to decide its implication in the stock market. Both techniques create indicators that are used as inputs in machine learning algorithms, with both showing in past studies the ability to return profitable trading strategies. However, these techniques are rarely used together. Thus, we examine the advantages when combining technical and sentiment analysis indicators under a GP, allowing trees to contain technical and/or sentiment analysis features in the same branch. We run experiments on 60 different stocks and compare the proposed algorithm's performance to two other GP algorithms, namely a GP that uses only technical analysis features (GP-TA), and a GP that uses only sentiment analysis features (GP-SA). Results show that the GP using the combined features statistically outperforms GP-TA and GP-SA under several different financial metrics, as well as the financial benchmark of buy and hold.

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

1 Introduction

Algorithmic trading has gotten much attention in the most recent years with a surge of services in the sector, and relevant research using machine learning. Genetic programming (GP) has been very popular in the field of finance and in particular in algorithmic trading [3], due to its ability to generate white-box models and perform good exploration and exploitation of the search space. It has been particularly popular with technical analysis (TA), which looks into historical data and creates features for future trend prediction in the market.

Another technique that has recently been receiving popularity in financial markets is sentiment analysis (SA), which uses its own indicators that determine the sentiment strength of market participants towards a particular security.

While both TA and SA have been used individually, there are very few studies that have combined these two techniques. Since both TA and SA can perform well on their own, there is a high possibility of creating powerful algorithmic trading models by combining these two techniques. Thus, in this paper, we propose using a novel GP algorithm that takes into account features from both TA and SA. As mentioned above, GP algorithms have been popular with TA; on the other hand, they have not been used with SA, with the exception of [5]. Allowing the GP to combine features from both techniques (GP-SATA) will not only consider past price trends, but also the current sentiment in the market. Our aim is to showcase that GP-SATA can effectively search the combined search space and return solutions that outperform GP-SA and GP-TA. To achieve this, we run experiments on 5-year data from 60 international stocks, and we compare the performance of the GP algorithms.

The rest of the paper is organised as follows: firstly, we discuss related work in Section 2. Then we present the proposed methodology in Section 3, along with the experimental setup in Section 4. Lastly, we present and discuss the results of our experiments in Section 5, followed by the conclusion in Section 6.

2 Related Work

Studies in algorithmic trading with TA features include the work in [15], using neural network architectures. In [16] the authors imported TA features into a long-short term memory (LSTM) model, to predict future stock price trends. For hybrid models, the authors in [11] created a LSTM-CNN model which outperformed the single models in predicting stock prices and in [6] they suggested a systematic method to find the time window size and topology for the LSTM network using EA. For algorithmic trading with GP, one of the first papers is [13], where the GP outperformed commonly used, non-adaptive technical rules. Likewise, same findings were produced in later papers, too, e.g., see [8–10]. As we see in [2] and [3], GP has the ability to evolve the trading strategies, create solutions enduring extreme market conditions and optimise solution parameters.

For SA, a significant paper is written by [12], who utilised neural networks and improved the predictive ability of multivariate models using newspaper headlines. In addition, [22] used support vector machines (SVM) with tree kernels and semantic frame parses to generalise from sentences to scenarios. Moreover, [7] used news into a deep CNN model and managed to generate an event-driven stock model. In [5] the authors have compared TA and SA features in the same GP structure; showing the financial profitability of SA features.

For the combination of the analyses, [20] employed event knowledge and information of companies to create a specific scenario. [18] used DNNs for stock movement forecasting, deploying historical prices and financial news, improving their model's accuracy. [21], also, produced an important study, by using news

from Reuters on the S&P500 index in a RNN and CNN hybrid model; predicting the price and intraday directional movement. However, it should be mentioned that the above papers tend to form the exception rather than the rule; in other words, it is rare to find published articles that combine TA and SA.

To sum up, in the literature we observe the lack of papers using GP algorithms with SA features for the means of algorithmic trading, as well as, a lack of studies combining these features with TA indicators. Furthermore, the lack of research in the combination of these two features in other machine learning algorithms, is, also, noticeable, although they have been widely used individually.

3 Methodology

This section is divided into three main parts: Section 3.1 holds the methodology for obtaining the TA and SA indicators, Section 3.2 for the GP methodology, Section 3.3 for the trading algorithm that uses the GP recommendations.

3.1 Financial Analysis Processes

Technical Analysis In order to be able to achieve good trading results, it is essential to analyse the indicators derived from TA that assist with recognizing the patterns. TA can be used in many ways and the features showcase the condition of a company and of the financial market as a whole. Studying previous works on TA, we chose *features that have been used in the past by researchers and technical analysts*. We downloaded the historical prices of the companies via Yahoo! Finance. We use the prices showcased at the Adjusted Close, Close, High and Low columns available on the datasets. We used this information to create 6 indicators in 2 different chronological periods, namely 5 and 10 days. These features are Moving Average, the Momentum, the Rate of Change (ROC), the Volatility, the Williams' %R and the Midprice, adding up to 12 features. Below, we introduce the mathematical equations of the indicators.

From Equation (1-6), let $n \in \{5, 10\}$ be the size of the lookup window. $Price_i$ is the adjusted closing price at the i day of this window, with the convention that $Price_n$ is the most recent price, $Price_1$ is the first price and, $Price_0$ is the last price of the previous set of prices, needed to find the price change for Volatility. Moreover, we denote as $Close$, the most recent closing price, and by $Highest_h$ and $Lowest_l$, the highest high and the lowest low price over all days in the lookup window. Lastly, Var is the sample variance over a dataset.

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

$$\mathsf{Momentum} = Price_n - Price_1 \tag{2}$$

$$\mathsf{ROC} = \left(\frac{Price_n}{Price_1} - 1\right) \cdot 100 \tag{3}$$

$$\mathsf{Volatility} = \sqrt{\mathsf{Var}\left(\left\{\frac{Price_i}{Price_{i-1}} - 1\right\}_{i\in\{1,\dots,n\}}\right)} \tag{4}$$

$$\mathsf{Williams'\%R} = -100 \cdot \frac{Highest_h - \mathsf{Close}}{Highest_h - Lowest_l} \tag{5}$$

$$\mathsf{Midprice} = \frac{Highest_h - Lowest_l}{2} \tag{6}$$

Moving Average assists with noise elimination and trend identification. Momentum and ROC, indicate the difference between the most recent price and the price n days ago, with ROC normalizing the price. Continuing, Volatility computes the past performance of the stock and it is a measure of the dispersion of returns of the n days period. Williams' R calculates the overbought and oversold levels and Midprice shows the midpoint value from two different input fields.

Sentiment Analysis SA is the method of obtaining useful information from news and it is common for researchers to classify their data as positive, negative and neutral. Due to the lack of already classified financial news, we needed to download the articles via a scraper, which was made using the Google Search Console API available in Python. We extracted information from the first to the twentieth page of Google search engine results, searching for each company's name. We discarded the articles where we could not find the name of the company and its stock market name inside the text, we shortlisted them based on their length, keeping those with more than 500 characters and we saved them in an ascending order, from the oldest to the most recent artcile.

To assign sentiment to the articles, we used popular specialised SA programs, *as used in the relevant literature*, i.e., TextBlob [14], SentiWordNet [1] and AFINN sentiment [17]. As a Python library, TextBlob is offering an API supporting with computing the polarity and subjectivity of the texts. SentiWordNet 3.0 is an enhanced lexical resource, and it includes a list of words classified as positive, negative, or neutral. We use the weighted average of the classified words and assign an overall percentage of the sentiment for the text. AFINN sentiment is a popular lexicon for SA developed by Finn Årup Nielsen, using more than 3300 words with a polarity score associated with each word. In the paper, we use these three programs on the full texts of the articles, their titles and their summaries, producing 12 SA features, available in Table 1.

After acquiring the sentiment of the articles, we took more factors into consideration to ensure we use the features as efficiently as possible when we run the GP. Some important tasks are to use the average sentiment value of the features for the articles with the same date, since more than one article will appear for some dates. Furthermore, since the stock market is not open on the weekends, but we still get articles at that time, we included the sentiment from the weekend days to that of Fridays; expecting the sentiment of these days to effect the stock price of Monday. Lastly, for the dates that we did not manage to

download any articles for, thus the dates that have no sentiment, but still have a stock price assigned to them, we kept the dates in the dataset and assigned 0 to its sentiment; indicating neutrality and/or no movement. This decision was made so we do not have any break points in between dates. It is worth to note, that to ensure the effective use of the features inside the GP, we normalised all features' values (i.e., both TA and SA features) to be between $[-1, 1]$.

3.2 Genetic Programming

Model representation The function set consists of the following logical functions: AND, OR, greater than ($>$) and less than ($<$). The terminal set includes the indicators of both TA and SA. In addition, the terminal set includes an Ephemeral Random Constant (ERC), which takes random values from -1 to 1, used as a threshold value for the features. Both sets are summarised in Table 1.

Table 1. Function and Terminal sets for GP-TA, GP-SA, GP-SATA.

Function and Terminal sets	
Function set	AND, OR, $<$, $>$
TA (for 5 and 10 days)	Moving Average Momentum ROC Williams' %R Volatility Midprice ERC
SA-textBlob	TEXTpol, TEXTsub TITLEpol,TITLEsub SUMMpol,SUMMsub
SA-SentiWordNet	TEXTsenti, TITLEsenti, SUMMsenti
SA-AFINN	TEXTafinn, TITLEafinn, SUMMafinn ERC

We present a sample tree of GP-SATA in Part 1 of Figure 1. 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 ITE statement is the evolved GP tree (Part 1). The second and third branches of the ITE are buy (1) and hold (0) actions. 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 tree uses one SA indicator ($TEXTpol$) and one TA indicator ($Volatility$), thus utilizing features from both techniques. It is worth noting that the generated signals in Part 2 are always 1 and 0, thus there is no sell action. We will discuss how a sell action is performed in Section 3.3.

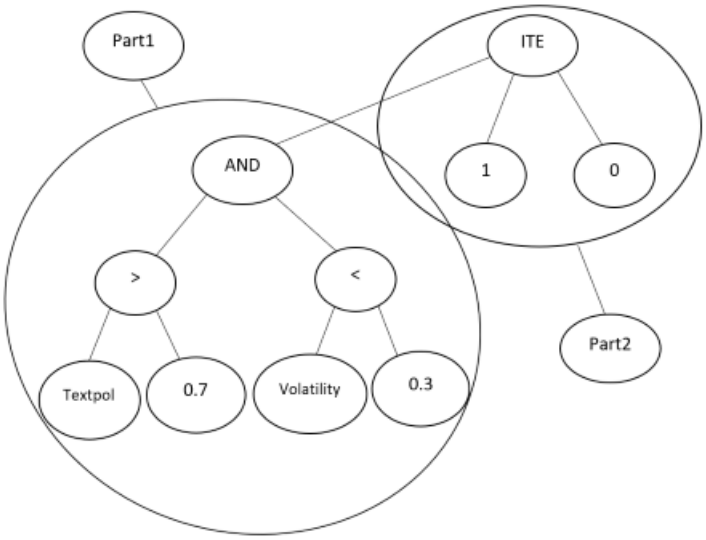

Fig. 1. Example tree for GP-SATA. The root is an AND statement, uniting SA and TA features under it in the its two branches, respectively.

Fitness Function and Metrics The GP algorithm is trained by maximising the the Sharpe ratio, which is the fitness function. The metric was chosen as it is used to identify the risk-adjusted return of investments. In Equation (7), we define the Sharpe ratio as:

$$\mathsf{SharpeRatio} = \frac{\mathsf{E}(Return) - Risk_f}{Risk}, \tag{7}$$

$Risk_f$ denotes the risk free rate. $\mathsf{E}(Return)$ is the sample mean return, where return is defined in Equation 8:

$$Return = \left\{ \frac{Price_s - Price_b}{Price_b} \right\} \tag{8}$$

The returns from each trade is based on the price we bought the stock, $Price_b$, and the price we sold the stock, $Price_s$. Discussion on when a sell action is performed will take place in Section 3.3. The returns of all the formed trades in a dataset are reserved as a list, and at the end, we compute the sample mean of the returns, finding the rate of return. Risk, is defined in Equation (9), as the standard deviation of the returns list, where Var denotes the sample variance:

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

Following, in Equation (10), we present the equation for cumulative returns as the sum value of the returns' list. All of the Equations (7-10) show the metrics we save as results during the GP runs and we base our analysis on.

$$C = \Sigma(Return) \tag{10}$$

Genetic Programming Operators For evolving the trees, we used point mutation and subtree crossover. The crossover ratio is symbolised as p and the mutation probability as $1-p$. As seen in [19], is a common scheme used by many researchers. We also use elitism to ensure that the best individual of a generation is copied to the next generation. Moreover, all GP algorithms are programmed to prefer the trees with the best results, as well as, the tress with the least depth; to ensure that they will be easy to interpret.

3.3 Trading Algorithm

As explained in Section 3.2, the second branch of the If-Then-Else tree always returns 1, denoting a buy action, and the third branch always returns 0, denoting a hold action. When the GP tree returns 1, we buy one amount of stock that later GP sells considering the price increase $r\%$ within n days. If the price increase within n days is greater than $r\%$ we sell on that day, otherwise we sell the stock on the n^{th} day. This is always considered after a buy action has taken place.

We record the rate of return (Equation 8) of each trade into a list, from which we then calculate the Sharpe ratio and risk (Equations 7 and 9). Additionally, we calculate the cumulative return (Equation 10). These metrics allow us to evaluate the performance of the trading algorithm, and their results are presented and discussed in detail in Section 5. The transaction cost for each trade is 0.025%.

4 Experimental Setup

4.1 Data

In our research, we used datasets from 60 different companies, including news articles and historical prices. The dates are from a total of 5 years, the 1st of January 2015 to 31st of January 2020 and the companies were chosen based on their popularity. For TA, we downloaded the daily closing price data from Yahoo! Finance and for SA the data was articles downloaded by a scraper. After we gathered all the data, we generated the 12 TA indicators and the 12 SA indicators, as already discussed in Section 3.

4.2 Benchmarks

The proposed GP-SATA is benchmarked against two other GP algorithms, one that only uses TA features (GP-TA), and one that only uses SA features (GP-SA). Both benchmarks follow the same representation, fitness function, and operators as GP-SATA. Essentially, the only difference amongst the algorithms is their terminal sets, as GP-SATA combines the terminal sets of SA and TA, while GP-TA includes only TA features and GP-SA only SA indicators. The rationale

behind this comparison is to investigate if there are added benefits in introducing the combined feature set to the GP and whether GP-SATA will have difficulty handling its diverse search space. We, also, compare the GP performance against a common financial benchmark, namely, the Buy and Hold (BnH) technique. In BnH, investors buy the stock at the beginning of the period they are interested in and they sell it at the end, forming one trade only across the whole dataset.

4.3 Parameter Tuning

We performed the parameter tuning in two steps. Firstly, we performed a grid search using the validation set to find the GP parameters. Namely, the population size, crossover probability (p), number of generations, tournament size and maximum depth of the trees; while keeping the trading strategy parameters n and r constant. For all algorithms no parameter combination was statistically different than another and we chose to keep the same parameters for all in order to be able to compare them with one another. Thus, we concluded with the GP parameters [1] of Table 2 that performed well in the validation sets of the models.

Table 2. GP Parameters for GP-TA, GP-SA, GP-SATA.

GP Parameters	
Population size	1000
Crossover probability	0.99
Mutation probability	0.01
Generations	50
Tournament size	4
Maximum tree depth	6

Once the first step was complete, we moved to the second step, which was tuning the trading parameters n and r. We chose to have tailored values for these two parameters for each company, enabling better trading performance.

5 Results and Analysis

5.1 Summary Statistics

For each company we applied the three GP algorithms for 50 independent runs on the training set. The resulting trading strategies were then run on the test set on which we evaluated the performance of the GPs. We found the averages, the standard deviation, the maximum and minimum values of these results of Sharpe ratio, rate of return, cumulative return and risk of each enterprise; as

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

shown in Table 3. The metrics are calculated on results with non-zero values, meaning we did not include the runs with no 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. However, when there is one trade in a run, we do get values for the metrics, but not for risk, which cannot be computed with only one trade. Few companies do showcase these results, thus, for all algorithms, the minimum value of risk is 0.

Furthermore, we perform a two-sample Kolmogorov-Smirnov (KS) test on all of the 50 runs of the individual algorithms for all 60 companies, excluding the values of 0, again. We chose the KS-test due to its sensitivity to differences of two samples. Since we perform multiple comparisons (GP-SATA vs GP-TA and GP-SATA vs GP-SA), we apply the Bonferroni Correction. Thus, the p-value for a 5% significance level is equal to $\alpha = \frac{0.05}{8} = 0.00625$. The denominator is the number of the multiple comparisons taking place. The null hypothesis is that the two distributions that we compare come from the same continuous distribution.

Table 3. Metric results for GP-TA, GP-SA, GP-SATA.

Metric	***Algorithm***	***Mean***	***StDev***	***Max***	***Min***
Sharpe	GP-TA	3.363717	8.96	59.086	-5.12
	GP-SA	1.018225	13.81	14.895	-101.74
	GP-SATA	**8.02535**	**47.41**	**315.84**	**-4.0682**
Rate of return	GP-TA	0.01015	0.021	0.0797	**-0.0375**
	GP-SA	0.00813	0.0187	0.0499	-0.053
	GP-SATA	**0.0123**	**0.029**	**0.104**	-0.049
CumulativeRet	GP-TA	0.034964	0.2252	0.5076	-1.19
	GP-SA	0.0219	0.047	0.14	**-0.147**
	GP-SATA	**0.05614**	**0.594**	**3.13**	-0.47
Risk	GP-TA	**0.0218**	0.0178	**0.092**	0
	GP-SA	0.03123	0.024	0.0986	0
	GP-SATA	0.02614	**0.52**	2.922	0

Starting with the Sharpe ratio results in Table 3, GP-SATA performs higher than GP-TA and GP-SA. The KS-test of GP-SATA to GP-TA is 3.50E-08, while GP-SATA to GP-SA is 4.33E-15. The differences in mean Sharpe ratio presented in Table 3 are thus statistically significant and GP-SATA statistically outperforms GP-TA and GP-SA. Moreover, the best standard deviation, maximum and minimum values, belong to GP-SATA. The high standard deviation of GP-SATA, which indicates outliers. However, we, also, also performed a KS-test excluding the outliers from the distributions, and the tests still showed strong statistical significance, hence confirming the statistical difference of the distributions.

For rate of return, GP-SATA performs higher than GP-TA and GP-SA. What is also interesting is that *GP-SATA's rate of return is approximately equal to the sum of the rate of return of both GP-TA and GP-SA*. This clearly demonstrates

the added value of the combination of features, where essentially both TA and SA's returns have been added up together to form SATA's returns. The KS-tests showed a statistically significant difference in the distributions of GP-SATA and GP-TA (p-value was 0.00066), while there was no statistical difference between GP-SATA and GP-SA (p-value 0.08926). In addition, the standard deviations among the three algorithms are all around 0.02-0.03, while GP-SATA has the highest maximum value of rate of return and GP-TA the lowest minimum value.

When looking at the average cumulative returns of the 60 companies, GP-SATA again outperforms GP-TA and GP-SA. We also again observe that *GP-SATA's cumulative returns are approximately equal the sum of the cumulative returns of both GP-TA and GP-SA*. The KS-test p-value for GP-SATA with GP-TA is 0.002016 and for GP-SA is 5.09E-11, confirming that GP-SATA statistically outperforms GP-TA and GP-SA. Moreover, GP-SATA continues having the highest standard deviation among the algorithms, but also the best (maximum) value; while GP-SA shows the lowest minimum value in the results.

For risk, the lowest values is that of GP-TA, followed by GP-SATA and GP-SA. However, it should be noted that the risk values of GP-SATA and GP-TA are relatively close (0.02614 and 0.0218, respectively); a KS-test between these two distributions confirms that there are no statistical differences (p-value=0.35023). On the other hand, GP-SATA outperforms statistically GP-SA at risk, with the KS-test p-value being 0.00024. Lastly, GP-SATA has again the highest standard deviation, while the best maximum value is that of GP-TA (lowest value). With regards to the minimum values, risk is 0 for all algorithms, which as we explained at the beginning of this section is because all algorithms have some cases with a single trade, where the standard deviation of risk cannot be calculated.

5.2 GP-SATA compared to Buy and Hold Method

Since the companies we have used come from a predominately bull market, the buy and hold (BnH) method performs extremely well, as the last price where the selling action is performed is significantly higher than the initial price where the buy action took place. In addition, because GP-SATA uses the Sharpe ratio as the fitness function, instead of the cumulative return, it is disadvantaged in this particular comparison and is thus outperformed by BnH.

To enable a fairer comparison, we re-run GP-SATA with cumulative return as the fitness function. In this scenario, the mean results are still in favour of BnH, with an average performance of 1.26 vs 0.88 of GP-SATA. However, given that we observed several extreme values in the BnH distribution, we also looked at the median values. In this case, BnH has median cumulative returns of 0.62, while GP-SATA has a higher value of 1.084. We also performed the non-parametric Friedman test, which returned an average rank of 1.33 for GP-SATA and 1.66 for BnH, meaning GP-SATA ranked higher than BnH. The p-value of the test was 0.0098, confirming that GP-SATA's better rank statistically outperforms BnH.

5.3 Computational Times

Since all GPs follow the same structure, their computational times are very similar. A single run for GP-TA was on average 1.39 minutes, while GP-SA lasted 1.49. As GP-SATA used more features than the other two algorithms, it lasted slightly longer, around 2 minutes. The differences are small, and all algorithms run relatively fast. Besides, computational times can be further cut down by parallelisation, as it has been previously demonstrated in [4], where speed ups by 21 times were achieved.

6 Conclusion

To conclude, this paper presented a novel GP that combined features from both TA and SA; allowing for new trading strategies to be created, considering historical patterns and the current market sentiment for a given stock. Results on 60 datasets showed that the proposed GP-SATA was able to statistically outperform GP-TA and GP-SA in terms of Sharpe ratio and cumulative returns and it was very competitive in terms of rate of return and risk and was never outperformed by the other GP algorithms. What we found interesting was that for rate of return and cumulative returns, GP-SATA's results were approximately the sum of of the GP-TA and GP-SA performances. Thus, we can assume that the proposed algorithm was fully taking advantage of the benefits of each technique, and combining them under its novel framework. Lastly, GP-SATA was also able to statistically outperform buy and hold. The results are very encouraging and show the potential of GP algorithms that combine heterogeneous features. Future work will focus on marking further improvements in the GP-SATA.

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: LREC, European Language Resources Association (2010)
2. Berutich, J.M., López, F., Luna, F., Quintana, D.: Robust technical trading strategies using gp for algorithmic portfolio selection. Expert Systems with Applications **46** (2016) 307–315
3. Brabazon, A., Kampouridis, M., O'Neill, M.: Applications of genetic programming to finance and economics: past, present, future. Genetic Programming and Evolvable Machines **21**(1) (2020) 33–53
4. Brookhouse, J., Otero, F.E., Kampouridis, M.: 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) 1117–1124
5. Christodoulaki, E., Kampouridis, M., Kanellopoulos, P.: Technical and sentiment analysis in financial forecasting with genetic programming. In: IEEE Computational Intelligence for Financial Engineering and Economics (CIFEr). (2022)

6. Chung, H., Shin, K.s.: Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability **10**(10) (2018) 3765
7. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: Twenty-fourth international joint conference on artificial intelligence. (2015)
8. Kampouridis, M., Alsheddy, A., Tsang, E.: On the investigation of hyper-heuristics on a financial forecasting problem. Annals of Mathematics and Artificial Intelligence **68** (2013) 225–246
9. Kampouridis, M., Otero, F.: Heuristic procedures for improving the predictability of a genetic programming financial forecasting algorithm. Soft Computing **21** (2017) 295–310
10. Kampouridis, M., Tsang, E.: Investment opportunities forecasting: Extending the grammar of a gp-based tool. International Journal of Computational Intelligence Systems **5**(3) (2012) 530–541
11. Kim, T., Kim, H.Y.: Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. PloS one **14**(2) (2019) e0212320
12. Kohara, K., Ishikawa, T., Fukuhara, Y., Nakamura, Y.: Stock price prediction using prior knowledge and neural networks. Intelligent Systems in Accounting, Finance & Management **6**(1) (1997) 11–22
13. Li, J., Tsang, E.P.: Improving technical analysis predictions: An application of genetic programming. (1999) 108–112
14. Loria, S.: textblob documentation. Release 0.15 **2** (2018)
15. Mostafa, M.M.: Forecasting stock exchange movements using neural networks: Empirical evidence from kuwait. Expert Systems with Applications **37**(9) (2010) 6302–6309
16. Nelson, D.M., Pereira, A.C., de Oliveira, R.A.: Stock market's price movement prediction with lstm neural networks. In: 2017 International joint conference on neural networks (IJCNN), IEEE (2017) 1419–1426
17. Nielsen, F.Å.: A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In Rowe, M., Stankovic, M., Dadzie, A.S., Hardey, M., eds.: Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages. Volume 718 of CEUR Workshop Proceedings. (May 2011) 93–98
18. Peng, Y., Jiang, H.: Leverage financial news to predict stock price movements using word embeddings and deep neural networks. arXiv preprint arXiv:1506.07220 (2015)
19. Poli, R., Langdon, W., McPhee, N.: A field guide to genetic programming. (2009)
20. Teymourian, K., Rohde, M., Paschke, A.: Knowledge-based processing of complex stock market events. In: Proceedings of the 15th International Conference on Extending Database Technology. (2012) 594–597
21. Vargas, M.R., De Lima, B.S., Evsukoff, A.G.: 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) 60–65
22. Xie, B., Passonneau, R., Wu, L., Creamer, G.G.: Semantic frames to predict stock price movement. In: Proceedings of the 51st annual meeting of the association for computational linguistics. (2013) 873–883