An in-depth investigation of genetic programming and nine other machine learning algorithms in a financial forecasting problem

Xinpeng Long, Michael Kampouridis, Delaram Jarchi
School of Computer Science and Electronic Engineering
University of Essex
Wivenhoe Park, United Kingdom
{xl19586, mkampo, delaram.jarchi}@essex.ac.uk

***Abstract*—Machine learning (ML) techniques have shown to be useful in the field of financial forecasting. In particular, genetic programming has been a popular ML algorithm with proven success in improving financial forecasting. Meanwhile, the performance of such ML algorithms depends on a number of factors including data analysis from different markets, data periods, forecasting days ahead, and the transaction cost which have been neglected in most previous studies. Therefore, the focus of this paper is on investigating the effect of such factors. We perform an extensive evaluation of a financial genetic programming-based approach and compare its performance against 9 popular machine learning algorithms and the buy and hold trading strategy. Experiments take place over daily data from 220 datasets from 10 international markets. Results show that genetic programming not only provides profitable results but also outperforms the 9 machine learning algorithms in terms of risk and Sharpe ratio.**

***Index Terms*—Genetic programming, Machine learning, Financial forecasting, Algorithmic trading**

I. Introduction

Financial forecasting has always played a vital role in the world. In order to obtain the greatest trading returns with the least risk, financial traders hope to predict market trends and reversal points. A lot of research has been done on financial forecasting. Although most researchers claimed that their models can beat the market at certain times in certain markets, there is no universal algorithm that can consistently beat the market. In fact, many studies have recognized that simple models can be as good as or even better than complex models such as artificial neural networks [1]. For example, [2] confirmed that a simple model which was not based on strong statistical theory, achieved the highest performance in their work. Nevertheless, there is constantly research for new profitable trading algorithms.

One traditional method used in financial forecasting is technical analysis. Traders, who believe in technical analysis, believe that there is a potential relationship or pattern between historical data and the future trend of market prices. Market trends always repeat in the same pattern. Therefore, a difficult problem for traders is about finding patterns.

Recently, machine learning (ML) algorithms have been successfully adopted in many applications including financial forecasting [3]–[6]. ML is favored by traders because it can effectively discover patterns that apply to historical price data over days, weeks, and even years. Genetic programming (GP) is one of the most common machine learning algorithms applied to financial forecasting. As an evolutionary technology, GP applies the Darwinian principle of evolution to improve its models.

To the best of our knowledge, most published works tend to examine a few ML algorithms. In this paper, we have decided to compare a financial GP's performance against 9 popular ML algorithms, namely gradient boost (GB), stochastic gradient descent (SGD), random forest (RF), multi-layer perceptron (MLP), extra tree (ET), passive active classifier (PAC), C-support vector classification (SVC), k-nearest neighbors (KNN), and decision tree (DT). The experiments take place over 110 datasets. In addition, we investigate the effect of using longer period data; to do this, we train and test the ML algorithms over a 5-year period and also over a 10-year period. So in the end, we perform these experiments over 220 datasets. The reason behind this investigation is because there are many references in the ML literature that indicate that more data can allow an ML algorithm to better generalise; but there are also other works that suggest that old data might be irrelevant in financial problems [7]. We also summarise our results in terms of different stock market and country, in an attempt to identify stronger performing markets. Finally, we compared GP with a traditional financial benchmark (i.e., buy and hold). Our aim is to conduct an in-depth analysis on the performance of the different ML algorithms, and also report on how different factors (such as the period length and financial market) can affect the algorithms' financial performance.

The rest of this paper is organized as follows. In Section II, we provide a brief background and literature review on technical analysis and genetic programming for financial forecasting. Our proposed methodology is provided in Section III, where we first present the details of the GP and then we discuss how the GP models are used as part of a trading strategy. Section IV provides a description of our experimental setup, and presents the datasets used in our experiments, as long as the benchmarks and the parameter tuning process. Then, in

Section V we discuss the results of our experiments. Finally, Section VI concludes the paper by highlighting the major contributions of the paper and discussing future potential directions.

II. Background and Literature Review

A. Technical analysis and indicators

Technical analysis as one of the most traditional methods of evaluating stocks is used by approximately 90% of the traders [8]. The main idea of technical analysis is the use of charts and graphs integrated with various statistical methods to predict the market trend and stock price [9]. In textbooks of technical analysis, there are three rules to guide traders [10]. The first one is that market price's action reflects all the information. In other words, the price movement is derived from all relevant financial information. The second rule is about the trend of price movement. The final goal of technical analysts is to find the trend and the time that the trend starts to reverse. It enables traders to get profit from selling stocks in the downtrend and buying stocks in the uptrend. The last rule is that history repeats itself. A big assumption behind technical analysis is that the same event happens under the same condition. Thus, traders tend to make the same decisions when all conditions are similar. Since there is no exit for similar conditions, the prediction of technical analysis is not 100% accurate. There was doubt whether the technical analysis could return unusual profit based on the past price or not. This is likely to be against the efficient market's hypothesis that no one can gain exceeding profit unless taking a corresponding risk. Therefore, more and more studies were developed to prove whether the technical analysis is profitable or not. As an example, a study in [11] found no evidence to confirm technical analysis can earn exceeding profit in the stock market at the very early stages. However, having further research development, newer and more studies provided strong evidence for the profitability of technical analysis [12] [13].

There are two main types of technical analysis. The first one is charting or chart pattern, which is a subjective form of technical analysis. It allows traders to look at the specific period past the target price and figure out patterns by skills and experience [10]. To reduce subjective factors during trading, the technical indicators were applied. By mathematical calculation, the original data is converted into a value that measures the data's different characteristics. Based on the indicators, traders could ignore the noise on the data and find out when to buy and when to sell. The most common indicators used by traders is the moving average, which smooths the historical price to help traders spot trends easier. In real trading, various indicators are adopted to reduce noise at the same time. We will present the indicators used in this paper in Section III.

B. Genetic programming

GP was first developed by [14] and has been used on financial forecasting problems for over 20 years [15]. For example, [16] built a 1-day-ahead trading system based on GP which allows them to investigate stocks by groups of artificial traders. The result showed the GP-based system dominated several benchmark models in short-term prediction. Besides, [17] presented the GP-based technical trading rules that were able to outperform the buy-and-hold trading strategy on S&P500 when taking into account the transaction costs. GP was also compared with the neural network, another popular ML model, and showed its effectiveness [18]. More recently, [19] created an automated system that combines multi-objective optimization, GP, technical analysis, and feature selection. They evaluated the performance of the system in six BOVESPA shares for two periods, from 2013 to 2015 and 2016 to 2016. The system obtained profit even when the asset is devalued. Moreover, [1] provided evidence that the GP system is competitive with the traditional algorithms, in some cases even statistically better. Another common and successful application of GP is under an alternative way with traditional time series prediction, namely Directional changes [20]. [21] approved that the new approach combined with GP and Directional changes was able to find the profitable trading strategy. The GP application under Directional changes as a new approach is one of the popular area. More related studies could be found in [22] and [23]. Lastly, another recent GP application is [24], which combined indicators from technical and sentiment analysis.

From the above, we can see that GP has been used successfully in financial forecasting in many cases. But often experiments take palce over a limited number of markets (usually 1-2), e.g. [17]. In general, GP was only compared with few benchmark algorithms, usually less than 4 (e.g. [18] and [1]). It is also worth noting that some works took transaction costs into account but others did not. Therefore, our goal in this paper is to fill the gap by considering all the factors that we discussed above and make an in-depth investigation of a financial GP algorithm. To achieve this goal, we apply a GP-based trading strategy and compare it with the other 9 selected machine learning algorithms, as well as with buy and hold strategy. Datasets are from 10 international financial markets. The transaction costs are taken into account and the periods ahead for prediction are tuned for each dataset. Furthermore, we analyze the algorithms' financial performance in terms of the stock market the data comes from, data length, and country. We discuss all this in detail in Sections III and IV.

III. Methodology

A. Genetic programming model

1) Terminal set: We use 146 different technical indicators as terminals in our GP trees. All the technical indicators that are used in this work are listed in the Table I. We selected 5, 10, 15, and 30 days as the periods for most indicators. For those indicators that requires two periods, we have selected pairs of periods as: [5,10], [5,15], [10,15], [10,30], and [15,30]. It is worth noting that few indicators do not need periods, e.g., On Balance Volume, Ease of Movement, and Market Facilitation Index.

TABLE I
TECHNICAL INDICATORS AND THE CORRESPONDING PERIODS BE USED IN THIS PAPER

Categories	Indicators	Periods (days)
Market Strength Indicators	Money Flow Index (MFI)	5,10,15,30
	Accumulation/Distribution (A/D)	
	On Balance Volume (OBV)	
Momentum Indicators	Momentum (MTM)	(5,10,15,30)
	Relative Difference in Percentage(RDP)	(5,10,15,30)
	Rate of Change (RoC)	(5,10,15,30)
	Disparity index	(5,10,15,30)
	Percentage Price Oscillator (PPO)	([5,10],[5,15],[10,15],[10,30],[15,30])
	Ease of Movement (EOM)	
	Stochastic Momentum Index (SMI)	(5,10,15,30)
	Vertical Horizontal Filter (VHF)	(5,10,15,30)
Volatility Indicators	Average True Range (ATR)	(5,10,15,30)
	Relative Volatility Index (RVI)	(5,10,15,30)
Oscillating Indicators	Relative Strength Index (RSI)	(5,10,15,30)
	Relative Momentum Index	([5,10],[5,15],[10,15],[10,30],[15,30])
	Stochastic Oscillator (K% and D%)	(5,10,15,30)
	Commodity Channel Index (CCI)	(5,10,15,30)
	Williams' %R	(5,10,15,30)
	Chande Momentum Oscillator (CMO)	(5,10,15,30)
	Detrended Price Oscillator (DPO)	(5,10,15,30)
	Klinger Oscillator (KO)	
	Mass Index	
	Percentage Volume Oscillator	([5,10],[5,15],[10,15],[10,30],[15,30])
Trend Indicators	Moving average (MA)	(5,10,15,30)
	Exponential Moving Average (EMA)	(5,10,15,30)
	Double Exponential Moving Average	(5,10,15,30)
	triple exponential average	(5,10,15,30)
	Volume Adjusted Moving Average	(5,10,15,30)
	Moving Average Convergence/Divergence (MACD)	([5,10],[5,15],[10,15],[10,30],[15,30])
	Aroon Indicator (up and down)	(5,10,15,30)
	Donchian Channels	(5,10,15,30)
	Directional Movement Index	(5,10,15,30)
Other indicators	Market Facilitation Index	
	Negative Volume Index	
	Parabolic SAR	
	Polarized Fractal Efficiency indicator	([5,10],[5,15],[10,15],[10,30],[15,30])
	Random Walk Index (High and low)	(5,10,15,30)

In addition to the indicators, another terminal we use is an Ephemeral Random Constant (ERC). The ERC picks a uniformly distributed random number between -1 and 1.

2) Function set: The function set includes two logical operators, namely AND and OR. It also includes two logical expressions, namely less than ($<$) and greater than ($>$).

3) Model representation: Given the above function and terminal set, the GP evolves different logical expressions, either with AND/OR as the root or with less/greater than. This tree is then integrated into the first branch of an If-Then-Else (ITE) statement. The second branch of the ITE statement (the 'Then' branch) always returns a leaf node with a value of 1, representing a buy action. The third branch of the ITE statement (the 'Else' branch) always returns a node with a value of 0, representing a hold action. Note that there is no leaf for a sell action. We discuss how a sell action is implemented in Section III-B. A sample tree is presented in Figure 1. The evolved GP tree is at the left hand-side of the Figure (Part 1), while the buy/hold actions are at the right hand-side of the Figure (Part 2). The reason we did not make Part 2 a part of the GP is because its values always remain constant. This allowed the GP to focus its search on the technical indicators space.

4) Fitness function: In the literature, there are different ways to calculate fitness such as accuracy, error, and risk. For our experiment, the fitness is determined a financial metric that takes into account returns and risk, namely the Sharpe ratio (S_p), which is presented in Equation 1.

$$S_p = \frac{r_p - r_f}{\sigma_p} \tag{1}$$

Where r_p is the rate of return (RoR), r_f is the risk-free rate and σ_p the standard deviation of the RoR.

5) Selection method: We use the tournament selection to select individuals for crossover and mutation.

6) Operators: We use elitism, sub-tree crossover and point mutation.

A summary of the GP configuration is presented in Table II.

B. Trading strategy

As mentioned earlier, the GP tree is embedded into an ITE tree, which is the trading strategy. The action branches of the tree always have two actions: buy and hold. So when the first

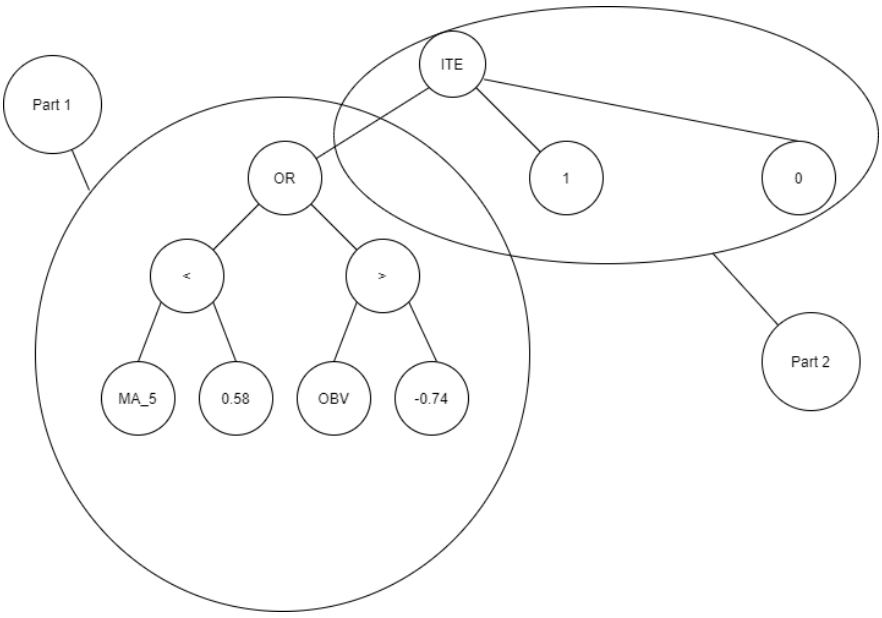

Fig. 1. Example of the GP tree structure and the If-then-else structure

TABLE II
CONFIGURATION OF THE GP ALGORITHM

Configuration	Value
Function set	AND, OR, $>$, $<$
Terminal set	146 technical indicators and ERC
Genetic operators	Elitism, subtree crossover and point mutation
Selection	Tournament

branch of the ITE statement evaluates to True, we buy one amount of stock, as long as we do not already hold a stock. If the first branch evaluates to False, then we take no action, i.e. we hold. To determine when to sell, we look at the following trading question: "Is the stock price going increase by $r\%$ within the next n days?". If we already hold a stock, and if the price indeed increases by $r\%$ within the next n days, then we sell the stock on the given day this happened. If the price does not increase within the given period, then we still sell the stock on the n^{th} day. The algorithm does not allow short-selling, so if we do not hold a stock, then we cannot sell. At the end of a sell action, we calculate and record the profit. All positions take transaction costs into account. The transaction cost is 0.025% per trade. A pseudocode is shown in Algorithm 1.

Algorithm 1 Pseudocode for our trading strategy given threshold r and days limit n

Require: Initialise variables (O represents the prediction of model,index indicates whether the stock is held)

if O = 1 and index =0 **then**
  Buy one amount of stock
  $index \leftarrow 1$
  $N \leftarrow i$ //Start time for trade
  $K \leftarrow p$ //Stock price when buying
else
  if index = 1 and (p$>$(1+r)N OR (i-K) $>$ n) **then**
    Sell the stock
    $index \leftarrow 0$
  end if
end if

IV. EXPERIMENTAL SET UP

A. Data

As we discussed in Section I, our goals in this paper can be summarised as follows: (i) Compare the performance of the GP algorithm against 9 popular ML algorithms, (ii) Investigate the effects of using a longer period (10 years) against a shorter period (5 years), (iii) Identify markets and countries that perform better than others, and (iv) Compare the GP performance against the buy and hold benchmark.

In order to achieve the above, we used daily data from 110 stocks derived from 10 markets in 6 countries. These markets are listed as follows: the Dow Jones Industrial Average (DJIA), the Nasdaq Stock Market (NASDAQ), the New York Stock Exchange (NYSE), the Russell 2000 Index, and the Standard and Poor's 500 (S&P500) in the United States, the Nifty Fifty (NIFTY 50) in India, the Taiwan Stock Exchange Corporation (TSEC) in China (Taiwan), the DAX performance index in German, Nikkei 225 in Japan, and the Financial Times Stock Exchange 100 Index in the United Kingdom. To investigate the effects of using a longer training and test period, we use two sets of periods: 5 and 10 years. Thus the above 110 stock are first examined in the period 2015-2020 (5 years), and then in the period 2010-2020 (10 years). Therefore since we use two different periods to train and test the algorithms, we end up with 220 datasets. We divide each period into a training,

TABLE III
PARAMETERS OF THE GP ALGORITHM

Parameters	Value
Max depth	6
Population size	500
Crossover probability	0.95
Tournament size	2
Numbers of generation	50

validation and test set in the following way: 60%:20%:20%. The validation set is used for parameter tuning.

Before conducting the experiments, we cleaned the data, including removing incorrect, missing, and null values from each dataset. Furthermore, these data were converted into technical indicators presented in the previous Sections. Data normalization was also performed to set indicator values between -1 to 1.

B. Benchmarks

In order to evaluate the performance of GP, we need to compare it with some benchmarks. As it was noted earlier, we have selected nine machine learning algorithms: GB, SGD, RF, MLP, ET, PAC, SVC, KNN, and DT. We use the above algorithms to tackle a binary classification problem in the form of "Is the stock price going to increase by $r\%$ within the next n days?". Class 1 denotes a buy action, and Class 0 denotes a hold action. The sell action takes again place as a part of the trading strategy that was described earlier in Section III-B.

C. Parameter tuning for GP

We performed grid search to decide the optimal GP parameters. Tuning took place in the validation set. Based on [25], we adopted the most common values for each parameter, namely 4, 6, 8 (max depth); 100, 300, 500 (population size); 0.75, 0.85, 0.95 (crossover probability); 2, 4, 6 (tournament size); and 25, 35, 50 (number of generations). Mutation probability is equal to (1-crossover probability), so we did not need to separately tune this parameter. Table III shows the selected parameters and their value after tuning.

D. Parameter tuning for Trading strategy

As we have already explained before, there are 2 parameters on our trading strategy derived for the question "whether the stock price will increase by r% on next n days ?". In addition, to 'encourage' the GP models to perform more actions (as we had noted that some models were doing very few trades), we added a 'minimum number of trades' parameter. Rather than tuning the above parameters and then selecting the best set across all datasets (which is what we did for the GP), we decided to allow for tailored values for each dataset. The configuration space for these three parameters is presented in Table IV.

As a reminder, we also adopt a transaction cost of 0.025% for each trading action.

TABLE IV
CONFIGURATION SPACE FOR THE TRADING STRATEGY

Parameters	Configuration space
n (days-ahead of prediction)	1,5,15
r (percentage of price movement)	0.01,0.05,0.1,0.2
Minimum number of trades	1,10,26,50,100,200

TABLE V
AVERAGE ROR, RISK, AND SHARPE RATIO RESULTS OF GP AND OTHERS ML ALGORITHMS. BEST VALUE PER METRIC IS SHOWN IN BOLDFACE.

Algorithms	Rate of return	Risk	Sharpe ratio (S_p)
GP	0.2376%	0.0395	**1.2296**
DT	0.7601%	0.1126	0.0687
ET	1.3826%	0.0663	0.8890
GB	1.2284%	0.1054	-0.0950
KNN	1.0568%	0.0771	0.1082
MLP	1.0889%	0.0942	0.4250
PAC	1.9016%	0.0974	0.2029
RF	1.6348%	0.0768	0.2227
SGD	**1.9761%**	0.0864	0.2535
SVC	1.5736%	**0.0275**	0.4299

V. RESULT AND ANALYSIS

This Section is divided in four parts. In the first part, Section V-A, we present the GP results and compare them against the 9 ML algorithms. In Section V-B, we study the GP's performance across different financial markets and countries. In Section V-C, we present the results from the comparison between 5 and 10 years' worth of data. Lastly, in Section V-D, we present the buy-and-hold results and compare them against the GP's results. All Sections' results are presented in terms of three financial metrics, namely rate of return (RoR) for each trade, risk, and Sharpe ratio. To examine the statistical significance of each Section's results, we performed the non-parametric Kolmogorov-Smirnov (KS) test.

A. GP vs ML algorithms

The analysis starts with comparing the performance of GP with 9 common classification models. Table V presents the average RoR, risk, and Sharpe ratio (S_p) on GP and 9 classification algorithms. From Table V, we can observe that in terms of rate of return, all algorithms have yielded positive returns, with SGD having the highest return at 1.97621% per trade. The lowest return comes by the GP at 0.2376% per trade. In terms of risk, the lowest risk comes from SVC (0.0275), while the GP has the second lowest risk at 0.0395. Lastly, in terms of Sharpe ratio, it is the GP that ranks first with a S_p of 1.2296. This is because there is a trade-off between return and risk, e.g. SGD that had the highest return also experienced a relatively high risk value. On the other hand, GP appears to have done a better job in optimizing both return and Sharpe ratio, as it is evident by its S_p value.

Additionally, we performed the non-parametric KS test for GP and other ML algorithms. Given that GP ranked first in terms of the aggregate Sharpe ratio metric, we use it as the control algorithm, and each KS test compares the GP's distribution against a different ML algorithm's distribution. The null hypothesis is that the two distributions come from

TABLE VI
P-VALUES OF KS TEST BETWEEN GP WITH OTHERS ML ALGORITHMS. STATISTICAL SIGNIFICANCE AT A 5% LEVEL IS WHEN THE P-VALUE IS BELOW 0.0056.

Algorithms	Rate of return	Risk	Sharpe ratio (S_p)
DT	0.0125	**1.54E-31**	**6.78E-05**
ET	0.2557	**6.78E-05**	**0.0017**
GB	0.0302	**4.56E-21**	**0.0024**
KNN	0.8260	**1.86E-08**	**2.42E-04**
MLP	0.0227	**1.16E-20**	**0.0017**
PAC	0.0397	**1.77E-21**	**0.0048**
RF	0.3696	**1.04E-07**	**0.0048**
SGD	0.1087	**3.29E-11**	**0.0034**
SVC	**6.75E-06**	**3.29E-11**	**5.91E-08**

TABLE VII
SINGLE RUNNING TIME BETWEEN GP WITH OTHERS ML ALGORITHMS

Algorithms	Time
GP	160.6260s
DT	0.4605s
ET	0.5117s
GB	7.9554
KNN	0.4986
MLP	23.0963
PAC	0.7459s
RF	1.4871s
SGD	0.2825s
SVC	0.3663s

the same continuous distribution. To account for the 9 multiple comparisons (9 ML algorithms that were compared with GP), we performed the Bonferroni correction for a 5% significance level and as a result, the null hypothesis is rejected when the p-value is below 0.0056 (0.05/9).

Table VI presents the results of the KS test. When a difference is statistically significant at the 5% level, this is indicated by putting the relevant p-value in boldface. As we can observe, in terms of rate of return, there are no statistically significant differences apart from the pair of GP and SVC. So even though the mean rate of return of the GP was lower when compared to the other ML algorithms' mean rate of return, *this difference was not statistically significant at a 5% level*. This can be explained by the fact that there were outliers with extremely high rate of return for the majority of the 9 ML algorithms, which had the effect of inflating the mean value.

In terms of risk, we can observe that all comparisons of the distribution pairs are statistically significant. Given that the GP ranked second best, this indicates that the GP statistically outperformed DT, ET, GB, KNN, MLP, PAC, RF, and SGD, while it was statistically outperformed by SVC.

In terms of Sharpe ratio, we can again observe that the null hypothesis is rejected for all KS tests. Given that GP ranks first in terms of mean S_p value, this indicates that GP statistically outperforms all 9 ML algorithms.

Lastly, Table VII presents the computational times for all algorithms. As we can observe, GP is taking significantly longer time to run, but this is not surprising, given that GP is a multi-generation population-based algorithm. However, the lengthy training process takes place offline; once it is complete, the best model is applied in real time to the (unseen) test set, which only takes 1-2 seconds to run. We believe that the significant improvements we have observed in Sharpe ratio and risk justify the slower execution time. Besides, the GP's execution time can be reduced by parallelization, as it has previously been shown in the literature (e.g, [26]).

TABLE VIII
GP'S AVERAGE PERFORMANCE UNDER DIFFERENT STOCK MARKETS. BEST VALUE PER FINANCIAL METRIC IS SHOWN IN BOLDFACE.

Indexes	Rate of return	Risk	Sharpe ratio (S_p)
DAX	-0.6112%	0.0317	0.5519
DJIA	0.3523%	0.0391	0.7282
FTSE100	1.5427%	0.0350	0.3524
NASDAQ	**1.6302%**	0.0823	1.1960
NIFTY 50	1.2593%	0.0368	1.2531
NIKKEI 225	-1.5516%	**0.0210**	0.0548
NYSE	-0.4442%	0.0494	0.1809
RUSSELL 2000	-0.8550%	0.0476	2.2676
S&P500	0.6264%	0.0233	0.8613
TSEC	0.4271%	0.0290	**4.8494**

B. Market and countries

In this section, we analyze the results in terms of different financial indices and countries to investigate if there are any particular markets with stronger performance. We look into the GP results, given its competitive performance from the previous section.

Table VIII shows that there are 6 indices which have positive RoR and also 4 indices have S_p higher than 1. In addition, the S_p on TSEC and RUSSELL 2000 went above the value of 2, denoting a profitable yet not risky performance. The best average RoR is 1.6302% on NASDAQ and the best average S_p is 4.8494 on TSEC. Risk values range from around 0.03 to around 0.05, with NASDAQ being the only exception with a higher risk of 0.0823. In general, GP achieved good and stable performance on each index.

In addition, we split the results into 6 countries based on where each market is located. Table IX shows the average result for each countries' market. Again we can see good performances in terms of risk across all countries especially for China, showing that the GP performs stable regardless of the dataset. In terms of RoR, results are less uniform, with the lowest RoR being observed for Japan (-1.5516) and Germany (-0.6112). In terms of S_p, the US, China, and India markets have the highest ratios, while the performance in the remaining markets (Japan, Germany, UK) is less impressive.

In summary, for RoR, GP's performance varies widely across countries and markets from -1.5516% to 1.6302%. For risk, GP's performance is relatively close to each country and market and is very good at around 0.03. Also for S_p, GP shows a good performance in 4 markets out of 10 with a value greater than 1.

C. Periods

From time periods view, we run GP on two different periods: 5 years (from 2015 to 2020) and 10 years (from 2010 to 2020). Our goal was to investigate whether longer data is beneficial, or if it adds unnecessary noise, given that values from so long

TABLE IX
GP'S AVERAGE PERFORMANCE UNDER DIFFERENT COUNTRIES

Country	Rate of return	Risk	Sharpe ratio (S_p)
US	0.4282%	0.0464	1.0812
China	0.4271%	0.0290	**4.8494**
Germany	-0.6112%	0.0317	0.5519
Japan	-1.5516%	**0.0210**	0.0548
UK	**1.5427%**	0.0350	0.3524
India	1.2593%	0.0368	1.2531

TABLE X
AVERAGE RESULT FOR GP ON 5 YEARS VERSUS 10 YEARS

GP	Rate of return	Risk	Sharpe ratio (S_p)
5 years	-0.5632%	0.0442	0.6006
10 years	1.0384%	0.0349	1.8585

ago might contain information that is not relevant any more to the current state of the market [7]. Table X shows the average result of GP on 5 years and 10 years. From Table X, it can be observed that RoR on 10 years GP is higher than 5 years GP. But if we look at the S_p, it is easy to observe that 10 years GP has a very high S_p (1.8585) and 5 years GP only has a value of 0.6006. Risk is at similar levels for both time periods, with the 10 year period risk being slighly lower. To conclude, the information from Table X shows that 10 years' worth of data is more beneficial across all 3 metrics of rate of return, risk, and Sharpe ratio.

D. Buy and hold

So far, we have evaluated the GP's forecasting performance by comparing it with ML algorithms, and investigating its performance under different markets and data periods. In this Section, we will look at the difference between the GP and the most traditional method buy and hold strategy.

Before proceeding with the comparison, we noticed that in many occasions the GP model was deciding not to perform any trades. In fact, out of the 220 datasets, the GP traded in only 155 of them. On the other hand, buy-and-hold always performs a trade, given that it buys one amount of stock on the first day of the data, and then sells it on the last. To make the comparison between the two algorithms fairer, we used the 155 datasets for the comparison, instead of the 220. In addition, we observed that both the GP and buy-and-hold contained outliers, which could significantly skew the distribution results. To deal with this issue, we removed these outliers, by only using all results that were within three standard deviations of the median. In the end, we removed four outliers from the GP and three outliers from the buy-and-hold.

Our results showed a cumulative return of around 6.11% for the GP, and around 14.05% for buy and hold. Other ML algorithms had similar to the GP cumulative return performance. This could be explained by two factors: first of all, the data period we have used is predominately a bull market, especially when we take into account the first and last day of each stock. This thus puts the buy and hold strategy in a very advantageous position. In addition, the GP algorithm was trained by having the Sharpe ratio as its fitness function. However, as we cannot calculate the Sharpe ratio for buy and hold[1], we can only compare the cumulative return for GP and buy and hold. If, on the other hand, were to use the rate of return as GP's fitness function, then the mean value of the cumulative return would be close to 15%, and thus outperform the buy and hold's cumulative return.

VI. CONCLUSION

To conclude, the main contribution of this work is the in-depth comparison of a genetic programming algorithm against different machine learning algorithms. Experiments took place over 220 datasets from 10 international markets. We have shown that GP was able to statistically outperform all other algorithms in terms of Sharpe ratio and most algorithms in terms of risk, while it also returned profitable results. This is an important finding, because until now published works tend to focus on fewer ML algorithms and/or fewer datasets. Further analysis also showed the differences in terms of international indices and markets performance. Furthermore, GP was also competitive against the buy and hold benchmark, when using a rate of return fitness function.

In terms of future work, we would like to create a new GP system that combines indicators from physical time and event-based time. Until now, the majority of the literature (including this work) focuses on indicators from physical time (such as daily closing prices), e.g. technical analysis indicators. Recent literature has also provided indicators from event-based systems, such as the directional changes summaries. Such event-based systems are able to focus on important events in the market, rather than the artificially created points in time, such as daily price. Our goal is to use a GP algorithm to trade with DC-based indicators and compare their performance against technical analysis. Eventually, we also aim to combine DC and technical analysis indicators to build better performing trading algorithms.

REFERENCES

[1] Mario Graff, Hugo Jair Escalante, Fernando Ornelas-Tellez, and Eric S Tellez. Time series forecasting with genetic programming. *Natural Computing*, 16(1):165–174, 2017.

[2] Spyros Makridakis and Michele Hibon. The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.

[3] SP Nitsure, SN Londhe, and KC Khare. Wave forecasts using wind information and genetic programming. *Ocean Engineering*, 54:61–69, 2012.

[4] K Kwong. Financial forecasting using neural network or machine learning techniques. *University of Queensland*, 13:221–228, 2001.

[5] Adesola Adegboye and Michael Kampouridis. Machine learning classification and regression models for predicting directional changes trend reversal in fx markets. *Expert Systems with Applications*, 173:114645, 2021.

[6] Adesola Adegboye, Michael Kampouridis, and Fernando Otero. Improving trend reversal estimation in forex markets under a directional changes paradigm with classification algorithms. *International Journal of Intelligent Systems*, 36(12):7609–7640, 2021.

[1]There's only two trades, one at the beginning and one at the end, and thus the standard deviation, which is the denominator in the Sharpe ratio formula, cannot be calculated

[7] Michael Kampouridis, Shu-Heng Chen, and Edward Tsang. Microstructure dynamics and agent-based financial markets: Can dinosaurs return? *Advances in Complex Systems*, 15(supp02):1250060, 2012.

[8] JG Agrawal, V Chourasia, and A Mittra. State-of-the-art in stock prediction techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(4):1360–1366, 2013.

[9] Raymond Hon-fu Chan, Alan Wing-keung Wong, and Spike Tsz-ho Lee. *Technical analysis and financial asset forecasting: From simple tools to advanced techniques*. World Scientific Publishing Company, 2014.

[10] Christopher J Neely and Paul A Weller. Technical analysis in the foreign exchange market. *Federal Reserve Bank of St. Louis Working Paper No*, 2011.

[11] Eugene F Fama and Marshall E Blume. Filter rules and stock-market trading. *The Journal of Business*, 39(1):226–241, 1966.

[12] Panagiotis Rousis and Spyros Papathanasiou. Is technical analysis profitable on athens stock exchange? *Mega Journal of Business Research*, 2018, 2018.

[13] Cheol-Ho Park and Scott H Irwin. What do we know about the profitability of technical analysis? *Journal of Economic surveys*, 21(4):786–826, 2007.

[14] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

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

[16] Viktor Manahov, Robert Hudson, and Hafiz Hoque. Return predictability and the 'wisdom of crowds': Genetic programming trading algorithms, the marginal trader hypothesis and the hayek hypothesis. *Journal of International Financial Markets, Institutions and Money*, 37:85–98, 2015.

[17] Lee A Becker and Mukund Seshadri. Gp-evolved technical trading rules can outperform buy and hold. In *3rd international workshop on computational intelligence in economics and finance*. Citeseer, 2003.

[18] Hitoshi Iba and Takashi Sasaki. Using genetic programming to predict financial data. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 1, pages 244–251. IEEE, 1999.

[19] Alexandre Pimenta, Ciniro AL Nametala, Frederico G Guimarães, and Eduardo G Carrano. An automated investing method for stock market based on multiobjective genetic programming. *Computational Economics*, 52(1):125–144, 2018.

[20] Edward Tsang. Directional changes, definitions. *Working Paper WP050-10 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex Revised 1, Tech. Rep.*, 2010.

[21] Jeremie Gypteau, Fernando EB Otero, and Michael Kampouridis. Generating directional change based trading strategies with genetic programming. In *European Conference on the Applications of Evolutionary Computation*, pages 267–278. Springer, 2015.

[22] Michael Kampouridis and Fernando EB Otero. Evolving trading strategies using directional changes. *Expert Systems with Applications*, 73:145–160, 2017.

[23] Michael Kampouridis, Adesola Adegboye, and Colin Johnson. Evolving directional changes trading strategies with a new event-based indicator. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 727–738. Springer, 2017.

[24] Eva Christodoulaki, Michael Kampouridis, and Panagiotis Kanellopoulos. Technical and sentiment analysis in financial forecasting with genetic programming. In *Proceedings of the IEEE Computational Intelligence for Financial Engineering and Economics (CIFEr)*. IEEE, 2022.

[25] Riccardo Poli, William B Langdon, and Nicholas Freitag McPhee. A field guide to genetic programming. *Published via http://lulu. com and freely available at http://www. gp-field-guide. org. uk.(With contributions by JR Koza)*, 2008.

[26] James Brookhouse, Fernando E.B. Otero, and Michael 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*, GECCO Comp '14, page 1117–1124, New York, NY, USA, 2014. Association for Computing Machinery.