Investment Opportunities Forecasting: Extending the Grammar of a GP-based Tool

Michael Kampouridis[1] Edward Tsang[2]

[1]*School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, CO4 3SQ, UK*
Email: mkampo@essex.ac.uk

[2]*Centre of Computational Finance and Economic Agents, University of Essex, Wivenhoe Park, CO4 3SQ, UK*
Email: edward@essex.ac.uk

Abstract: In this paper we present a new version of a GP financial forecasting tool, called EDDIE 8. The novelty of this version is that it allows the GP to search in the space of indicators, instead of using pre-specified ones. We compare EDDIE 8 with its predecessor, EDDIE 7, and find that new and improved solutions can be found. Analysis also shows that, on average, EDDIE 8's best tree performs better than the one of EDDIE 7. The above allows us to characterize EDDIE 8 as a valuable forecasting tool.

Keywords: Financial Forecasting, Genetic Programming, Genetic Decision Tree, Decision Support Tool

1 Introduction

The forecasting of time series is an important area in computational finance. There are numerous works that attempt to forecast the future price movements of a stock; several examples can be found in the literature.[1,2] Some more recent works on time series predictions are Ref. 3-7, which describe applications in both low and high frequency data. Furthermore, several different methods have been used for financial forecasting. Some examples of such methods are Support Vector Machines,[8–10] Artificial Neural Networks,[11–14] and Genetic Programming.[15,16] Furthermore, some of these methods have been combined to produce hybrid systems. For instance, Huang et al.[17] combined support vector machines with neural networks to investigate credit rating, and Kim[18] combined neural networks with evolutionary strategies for financial forecasting.

EDDIE (Evolutionary Dynamic Data Investment Evaluator),[19–21] is a decision support tool that uses Genetic Programming (GP),[22,23] for financial forecasting. In this paper we present EDDIE 8, which is the newest version. The novelty of this algorithm is in its rich, extended grammar. Instead of using a fixed number of pre-specified indicators from technical analysis,[24] like the previous versions do, EDDIE 8 allows the GP to search in the space of these technical indicators and use the ones that it considers to be optimal. Thanks to its extended grammar, EDDIE 8 is considered to be an improvement. This is because it has the potential, through the learning process, to discover better solutions that its predecessors cannot. A similar approach to ours, where there is an attempt to address the problem of fixed number of pre-specified strategies, can be found in Ref. 25,26, where Grammatical Evolution was used in place of the traditional GP.

In a previous work,[27] in order to present the value of EDDIE 8, we compared it with EDDIE 7, which is a re-implementation of Jin Li's EDDIE 4 (a.k.a. FGP-2)[20,28], with the addition of some indicators that Martinez-Jaramillo[29] found helpful and used in his own version of EDDIE. Those experiments took place under an artificial dataset framework. This work serves as an important extension, because we test the performance of EDDIE 7 and EDDIE 8 under 10 *empirical datasets* and then compare these results with the existing ones from the artificial framework. The rest of this paper is organized as follows: Section 2 explains how EDDIE functions; it also presents the two versions discussed in this paper, EDDIE 7 and EDDIE 8. Section 3 presents the experimental parameters for our tests, and Sect. 4 discusses the results of the comparison of the two versions, on 10 different empirical datasets. Section 5 then briefly discusses the performance results and conclusions for the tests that took place under the artificial datasets.[27] We then extend the conclusions drawn from the artificial datasets experiments to the empirical datasets experiments, in Sect. 6. Finally, Sect. 7 concludes this paper and discusses future work.

2 How EDDIE works

In this section we present the two versions, EDDIE 7 and EDDIE 8, and explain their differences. We first start by presenting EDDIE 7 and the way it works.

2.1 EDDIE 7

EDDIE is a forecasting tool, which learns and extracts knowledge from a set of data. As we said in the previous section, EDDIE 7 is a re-implementation of Jin Li's FGP-2 with the only difference being that EDDIE 7 uses some additional indicators that Martinez-Jaramillo used in his version of EDDIE.[29]

The way that EDDIE 7, and all versions of EDDIE, work is as follows. The user first feeds the system with a set of past data; EDDIE then uses this data and, through a GP process, it produces and evolves Genetic Decision Trees (GDTs), which make recommendations to buy (1) or not-to-buy (0). It then evaluates the performance of these GDTs, on a training set, for each generation. The GDT with the highest fitness at the last generation is finally applied to a testing set.

The set of data EDDIE uses comprises three parts: daily closing price of a stock, attributes, and signals. Stocks' daily closing prices can be obtained online at websites such as `http://finance.yahoo.com` and also from financial statistics databases like *Datastream*. The attributes are indicators commonly used in technical analysis.[24] The choice of indicators de-

Table 1 Technical Indicators used by EDDIE 7. Each indicator uses 2 different periods, 12 and 50, in order to take into account a short-term and a long-term period. For completeness, we provide formulas of our interpretation for these indicators in the Appendix.

Technical Indicators (Abbreviation)	Period
Moving Average (MA)	12 & 50 days
Trade Break Out (TBR)	12 & 50 days
Filter (FLR)	12 & 50 days
Volatility (Vol)	12 & 50 days
Momentum (Mom)	12 & 50 days
Momentum Moving Average (MomMA)	12 & 50 days

<Tree> ::= If-then-else <Condition> <Tree> <Tree> | Decision
<Condition> ::= <Condition> "And" <Condition> |
<Condition> "Or" <Condition> |
"Not" <Condition> |
Variable <RelationOperation> Threshold
<Variable> ::= MA_12 | MA_50 | TBR_12 | TBR_50 | FLR_12 |
FLR_50 | Vol_12 | Vol_50 | Mom_12 | Mom_50 |
MomMA_12 | MomMA_50
<RelationOperation> ::= ">" | "<" | "="
Decision is an integer, Positive or Negative implemented
Threshold is a real number

Figure 1 The Backus Naur Form of the EDDIE 7.

pends on the user and his belief in their relevance to the prediction. Table 1 presents the technical indicators that EDDIE 7 uses.[1]

The signals are calculated by looking ahead of the closing price for a time horizon of n days, trying to detect if there is an increase of the price by $r\%$.[19] Thus, if such an increase occurs, we denote it by 1; otherwise, by 0. A positive signal (1) means that there is a buy opportunity in the market, because the price is going up. Therefore if someone could predict this, he would make profit. The more opportunities EDDIE can correctly predict, the more successful it is. The values of n and r are discussed later in this paper, in Sect. 3.

After we feed the data to the system, EDDIE creates and evolves a population of GDTs. Figure 1 presents the Backus Naur Form (BNF)[31] (grammar) of EDDIE 7. As we can see, the root of the tree is an If-Then-Else statement. Then the first branch is either a Boolean (testing whether a technical indicator is greater than/less than/equal to a value), or a logic operator (and, or, not), which can hold multiple boolean conditions. The 'Then' and 'Else' branches can be a new Genetic Decision Tree (GDT), or a decision, to buy or not-to-buy (denoted by 1 and 0).

We would also like to draw the reader's attention at the *Variable* symbol of Fig. 1; here are the 12 indicators which we mentioned earlier in Table 1 that EDDIE 7 uses. They are pre-specified and should thus be considered as constants of the system. As we will see later, EDDIE 8 does not use these constants, but a function instead.

Each GDT's performance is evaluated by a fitness function, presented here. If the prediction of the GDT is positive (buy-1), and also the signal in the data for this specific entry is also positive (buy), then this is classified as True Positive (TP). If the prediction is positive, but the signal is negative (not-buy), then this is False Positive (FP). On the other hand, if the prediction is negative, and the signal is positive, then this is False Negative (FN), and if the prediction of the GDT is negative and the signal is also negative, then this is classified as True Negative (TN). These four together give the familiar confusion matrix, which is presented in Table 2.

Table 2 Confusion Matrix

	Actual Positive	Actual Negative
Positive Prediction	True Positive (TP)	False Positive (FP)
Negative Prediction	False Negative (FN)	True Negative (TN)

As a result, we can use the metrics presented in (1), (2) and (3).

Rate of Correctness

$$RC = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Rate of Missing Chances

$$RMC = \frac{FN}{FN+TP} \tag{2}$$

Rate of Failure

$$RF = \frac{FP}{FP+TP} \tag{3}$$

We can then combine the above metrics and define the following fitness function, which is presented in (4):

$$ff = w_1 * RC - w_2 * RMC - w_3 * RF \tag{4}$$

where w_1, w_2 and w_3 are the weights for RC, RMC and RF respectively. Li[28] states that these weights are given in order to reflect the preferences of investors. For instance, a conservative investor would want to avoid failure; thus a higher weight for RF should be used. However, Li also states that tuning these parameters does not seem to affect the performance of the GP. For our experiments, we chose to include strategies that mainly focus on correctness and reduced failure. Thus these weights have been set to 0.6, 0.1 and 0.3 respectively, and are given in this way in order to reflect the importance of each measure to our predictions.

The fitness function is a constrained one, which allows EDDIE to achieve lower RF. The effectiveness of this constrained fitness function has been discussed in Ref. 21,28. The constraint is denoted by R, which consists of two elements represented by percentage, given by

$$R = [Cmin, Cmax],$$

where $C_{min} = \frac{P_{min}}{N_{tr}} \times 100\%$, $C_{max} = \frac{P_{max}}{N_{tr}} \times 100\%$, and $0 \leq C_{min} \leq C_{max} \leq 100\%$. N_{tr} is the total number of training data cases, P_{min} is the minimum number of positive position predictions required, and P_{max} is the maximum number of positive position predictions required.

Therefore, a constraint of $R = [50, 65]$ would mean that the percentage of positive signals that a GDT predicts[2] should fall into this range. When this happens, then w_1 remains as it is (i.e. 0.6 in

[1] We use these indicators because they have been proved to be quite useful in developing GDTs in previous works like Ref. 29,30, . Of course, there is no reason why not use other information like fundamentals or limit order book information. However, the aim of this work is not to find the ultimate indicators for financial forecasting.

[2] As we have mentioned, each GDT makes recommendations of buy (1) or not-to-buy (0). The former denotes a positive signal and the latter a negative. Thus, within the range of the training period, which is t days, a GDT will have returned a number of positive signals

```
Procedure EDDIE ( )
Begin
Partition whole data into training data and testing data; /* While training
data is employed to train EDDIE to find the best-so-far-rule, the test data is
used to determine the performance of predictability of the best-so-far-rule */
Pop <- InitializePopulation (Pop); /* randomly create a population of
GDTs.*/
Evaluation (Pop);/* calculate fitness of each GDT in Pop */
Repeat
  Pop <- Reproduction (Pop) + Crossover (Pop); /*new population is created
after genetic operators of reproduction (which reproduces M*Pr individuals)
and crossover (which creates M*(1-Pr) individuals). Pr denotes the reproduc-
tion probability and M is the population size */
  Pop <- Mutation (Pop); /*Apply mutation to population */
  Evaluation (Pop); /* Calculate the fitness of each GDT in Pop */
Until (TerminationCondition( )) /* determine if we have reached the last gen-
eration */
Apply the best-so-far rule to the test data;
End
```

Figure 2 Pseudo code for the procedure that EDDIE follows. (Based on Li's work[28], p.76)

our experiments). Otherwise, w_1 takes the value of zero.

During the evolutionary procedure, we allow three operators: crossover, mutation and reproduction. After reaching the last generation, the best-so-far GDT, in terms of fitness, is applied to the testing data.

Figure 2 summarizes what we have said so far, by presenting the pseudo code that the EDDIE algorithms use for their experiments.

This concludes this short presentation of EDDIE 7. However, EDDIE 7 and its previous versions are considered to have a drawback: nobody can guarantee that the periods chosen for the indicators are the appropriate ones. Why is 12 days MA the right period for a short term period and not 10, or 14? As we mentioned earlier, choosing an indicator and, as a consequence, a period for this indicator, depends on the user of EDDIE and his belief in how helpful this specific indicator can be for the prediction. However, it can be argued that this is subjective and different experts could pick a different period for their indicators. In addition, this choice of indicators limits the patterns that EDDIE 7 can discover. This is hence the focus of our research. We believe that allowing EDDIE to search in the space of the periods of the indicators would be advantageous and eliminate any possible weaknesses of the human decision process. For these purposes, we implemented a new version, EDDIE 8, which allows the GP to search in the search space of the periods of the indicators. The following section explains how EDDIE 8 manages this.

2.2 EDDIE 8

Let us consider a function $y = f(x)$, where y is the output, and x is the input. In our case, the input is the indicators and the output is the prediction made by our GP. The function f is unknown to the user and is the GDTs that the algorithm generates, in order to make its prediction. As we just said in the previous section, the input is fixed in EDDIE 7; it uses 6 indicators, with 2 different pre-specified periods (12 and 50 days). This limits EDDIE 7's capability to find patterns that cannot be expressed in its vocabulary. EDDIE 8 uses another function $y = f(g(z))$, where $x = g(z)$; in other words, g is a function that generates indicators and periods for EDDIE to use. EDDIE 8 is not only searching in the space of GDTs, but also in the space of indicators. It can thus return Genetic Decision Trees (GDTs) that are using any period within a range that is defined by the user.

As we can see from the new syntax at Fig. 3, there is no such thing as a *Variable* symbol in EDDIE 8. Instead, there is the *VarConstructor* function, which takes two children. The first one is the indicator, and the second one is the Period. Period is an integer within the parameterized range [MinP, MaxP] that the user specifies.

```
<Tree> ::= If-then-else <Condition> <Tree> <Tree> | Decision
<Condition> ::= <Condition> "And" <Condition> |
                <Condition> "Or" <Condition> |
                "Not" <Condition> |
                VarConstructor <RelationOperation> Threshold
<VarConstructor> ::= MA period | TBR period | FLR period |
                     Vol period | Mom period | MomMA period
<RelationOperation> ::= ">" | "<" | "="
Terminals:
   MA, TBR, FLR, Vol, Mom, MomMA are function symbols
   Period is an integer within a parameterized range, [MinP, MaxP]
   Decision is an integer, Positive or Negative implemented
   Threshold is a real number
```

Figure 3 The Backus Naur Form of EDDIE 8

Table 3 GP Parameters.

GP Parameters	
Max Initial Depth	6
Max Depth	17
Generations	50
Population size	500
Tournament size	6
Reproduction probability	0.1
Crossover probability	0.9
Mutation probability	0.01
Period (EDDIE 8)	[2,65]

As a result, EDDIE 8 can return decision trees with indicators like 15 days Moving Average, 17 days Volatility, and so on. The period is not an issue anymore, and it is up to EDDIE 8, and as a consequence up to the GP and the evolutionary process, to decide which lengths are more valuable to the prediction.

The immediate consequence of this is that now EDDIE 8 is not restricted only to the 12 indicators that EDDIE 7 uses (which are still part of EDDIE 8's search space); on the contrary, it now has many more options available, thanks to this new grammar.

3 Experimental Parameters

As we said in Sect. 2, the data we feed to EDDIE consist of daily closing prices. These closing prices are from 10 arbitrary stocks from FTSE100. These stocks are: British American Tobacco (BAT), British Petroleum (BP), Cadbury, Carnival, Hammerson, Imperial Tobacco, Next, Schroders, Tesco, and Unilever. The training period is 1000 days and the testing period 300. The GP parameters are presented in Table 3. The values of these parameters are the ones used by Koza.[22] The results seem to be insensitive to these parameters. For statistical purposes, we run the GP 50 times for both EDDIE 7 and EDDIE 8.

Thus, the process is as follows. We create a population of 500 GDTs, which are evolved for 50 generations, over a training period of 1000 days. At the last generation, the best performing GDT in terms of fitness is saved and applied to the testing period. As we have already said, this procedure is done for 50 individual runs.

In addition, we should emphasize that we want the datasets to have a satisfactory number of actual positive signals. By this we mean that we are neither interested in datasets with a very low

number of signals, nor those with an extremely high one. Such cases would be categorized as chance discovery, where people are interested in predicting rare events such as a stock market crash. Clearly this is not the case in our current work, where we use EDDIE for investment opportunities forecasting. We are thus interested in datasets that have opportunities around 50-70% (i.e. 50-70% of actual positive signals). Therefore, we need to calibrate the values of r and n accordingly, so that we can obtain the above percentage from our data. For our experiments, the value of n is set to 20 days. The value of r varies, depending on the dataset. This is because one dataset might reach a percentage of 50-70% with $r = 4\%$, whereas another one might need a higher or lower r value. Accordingly, we need to calibrate the value of the R constraint, so that EDDIE produces GDTs that forecast positive signals in a range which includes the percentage of the *actual* positive signals of the dataset we are experimenting with. R thus takes values in the range of $[-5\%, +5\%]$ of the number of positive signals that the dataset has. For instance, if under $r = 4\%$ and $n = 20$ days, a dataset has 60% of actual positive signals, then R would be set to [55,65].

Finally, we should mention that a single run of either version does not last for more than a few minutes. EDDIE 8 is slightly slower than EDDIE 7 of course, due to its large search space, but this fact does not seem to significantly affect its runtime.

4 Test Results

This section presents the experimental results after having tested the 10 datasets under EDDIE 7 and EDDIE 8. We first start by observing how EDDIE 8 affects the fitness of the population during the training period. We are interested in seeing whether the extended grammar is giving EDDIE 8 an advantage, and if this is the case, how fast this happens during the evolutionary procedure. We then continue by presenting a summary statistics comparison between the two versions, under the data of the testing period. At this point we should mention that all fitness results have been normalized to a scale of [0,1]. The other measures (RC, RMC, RF) are already in this scale and thus no normalization took place.

4.1 Training performance comparison

In this section, we compare the training fitness of the two algorithms. As we have said, we are interested in examining the behaviour of the GP, now that it searches in a much bigger search space. Does it find very good solutions from the beginning of the evolutionary procedure, because now it has more options to look into? Or does it start with low performance due to these many options and later manages to focus on the promising ones? These are just two examples of behavioral questions we could be asking.

We conduct our analysis in two different parts. Firstly, we compare the training fitness in terms of the whole population. To do that, we calculate the average fitness for the whole population of GDTs; this process is done for each generation. Let us call this average Avg_{Fit}. Thus, we can observe how the GDTs' Avg_{Fit} changes over the 50 generations of a single run. We then repeat this procedure for each one of the 50 runs. Finally, we calculate the average, over these 50 runs, of Avg_{Fit}. Figure 4 presents these results. Each line in the graph denotes the average Avg_{Fit} for a different dataset. As we can see, the population of EDDIE 7 starts at generation 1 with an average fitness between 0.1-0.2, for all stocks. This quickly rises to 0.4-0.5 and stabilizes around 0.6, with half of the stocks slightly exceeding this level. On the other hand, EDDIE 8's population average fitness for all stocks starts from a much higher point, around 0.3. Fitness here also rises quickly to 0.5-0.6 and stabilizes between 0.6 and 0.7. As we can see, the average training fitness population of EDDIE 8 is somewhat higher than EDDIE 7's. It is obvious that EDDIE 8's grammar has allowed it to come up with better individuals in the first generation, and thus start with a population that has higher fitness.

For the second part of our analysis, we compare the fitness of the best individual (i.e. the GDT with the highest fitness) per generation; this fitness is called $Best_{Fit}$. So now instead of calculating the average fitness of the whole population for each generation, we just obtain the highest fitness. We can thus present how the highest fitness changes over the 50 generations of a single run. We then repeat this procedure for each one of the 50 runs. Finally we find the average, over these 50 runs, of $Best_{Fit}$. Figure 5 presents these results. In order to get a clearer idea of these results, we have divided them into two graphs per algorithm. The first column presents the graphs for EDDIE 7, and the second one for EDDIE 8. The graphs at the top are for the first 5 stocks (in alphabetical order) and the bottom graphs are for the remaining 5 stocks. We can see that results vary per stock for both algorithms, although they seem to follow the same pattern. The $Best_{Fit}$ values for EDDIE 7 start from a range of [0.58,0.66], at generation 1, and reach up to a range of [0.64,0.74], at the last generation. The datasets for EDDIE 8 seem to follow a very similar behavior: the $Best_{Fit}$ values start in the range of [0.58,0.67] and end up in the range of [0.65,0.75].

Table 4 also presents the average $BestFit$ values for the first and the last generation. Each stock has 4 values, 2 for EDDIE 7 and 2 for EDDIE 8. The top value represents the average $Best_{Fit}$ at generation 1, and the bottom value represents the average $Best_{Fit}$ at generation 50. EDDIE 8's $Best_{Fit}$ starts with higher fitness for 7 stocks. This means that there are 3 stocks which EDDIE 7 has better initial values: Carnival (0.6298), Hammerson (0.6121), Schroders (0.5935). In addition, at the end of the evolutionary procedure (generation 50), there are 2 stocks that EDDIE 7's $Best_{Fit}$ is higher than EDDIE 8's: BAT (0.7320), and Hammerson (0.6894). However, these differences from EDDIE 7 are relatively small (below 1%).

As we can see, there can be times where EDDIE 7 outperforms EDDIE 8, although this is only to a small degree. Nonetheless, this is quite interesting, because it indicates that there can be cases where EDDIE 8 might not be able to outperform its predecessor. Of course, at this moment this is only an indication that comes from results during the training period and this is why more analysis needs to be conducted.

4.2 Summary Results for Testing Period

In this section we present summary results for the two algorithms, after the GDTs were applied to the testing period. The first part presents the averages of the metrics we used and the second part presents the improvements and diminutions caused by the best GDT evolved by EDDIE 8.

4.2.1 Average Results

We first start with the average results for Fitness. In this way, we can have a general view of how the two algorithms have performed. We then move to the performance measures (RC, RMC and RF).

Figure 6 presents the average fitness results over the 50 runs for EDDIE 7 and EDDIE 8. As mentioned at the beginning of this section, the results have been normalized and are in the scale of

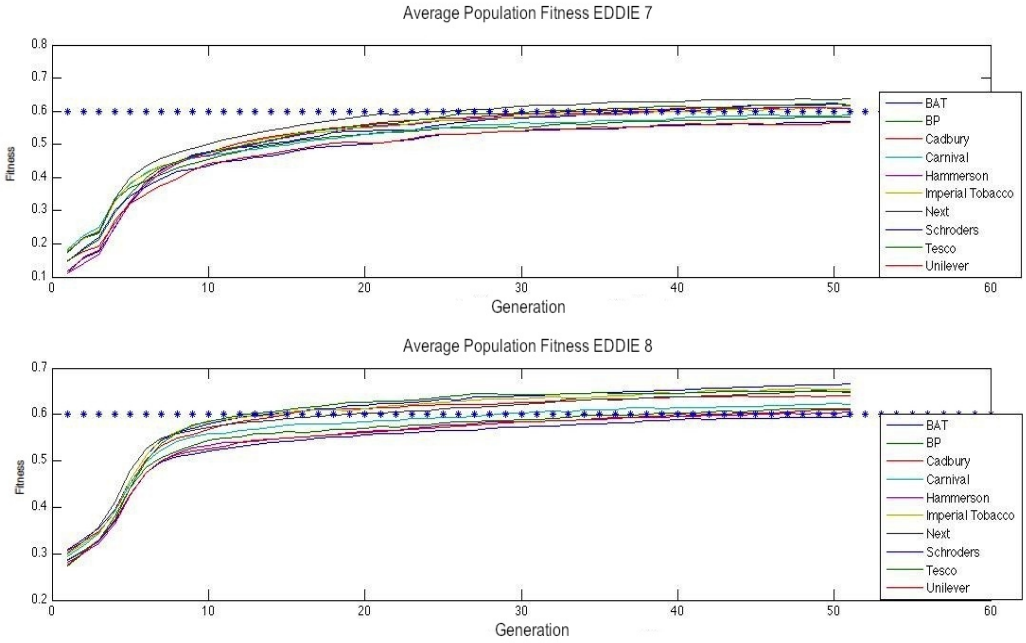


Figure 4 Average of the average fitness of the population of the GDTs for EDDIE 7 and EDDIE 8. This means that we first obtain the average fitness of the whole population, per generation. Then we find the average of this number over the 50 runs.

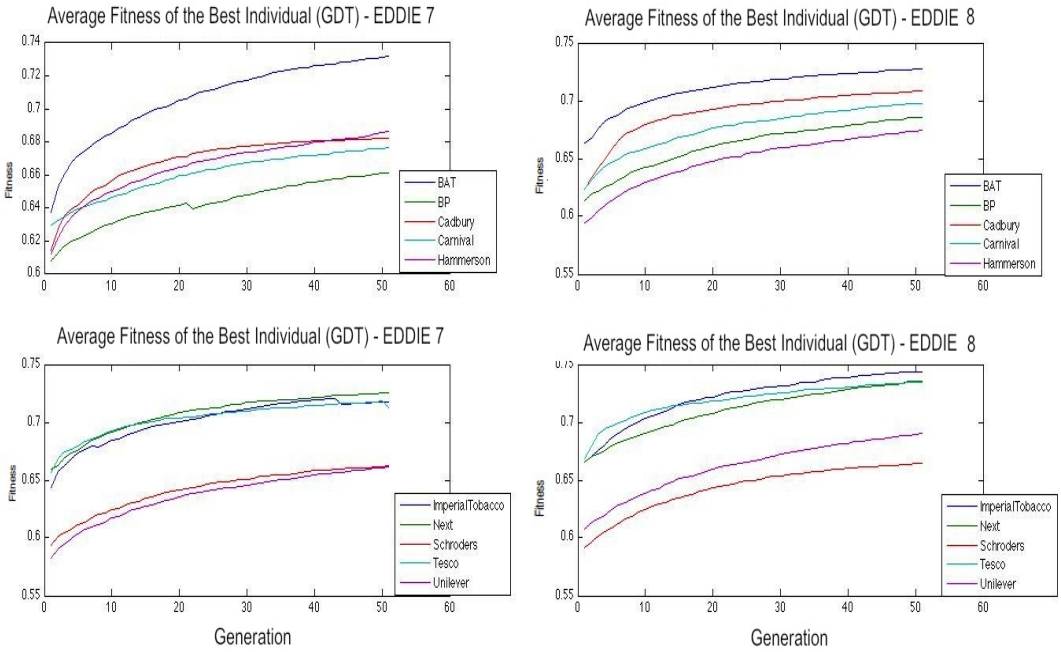


Figure 5 Average $Best_{Fit}$. We first obtain the best GDT's fitness per generation, for each one of the 50 runs. This happens for both algorithms. We then calculate the average of these fitness values (over the 50 runs) and present them in this figure. For the convenience of the reader, we have split the stocks into 5 per graph (by alphabetical order). The graphs in the first column are for EDDIE 7 and the others for EDDIE 8.

Table 4 Average $Best_{Fit}$ at generation 1 and 50, for EDDIE 7 and EDDIE 8, over the 10 stocks. Each stock has 4 values, 2 for EDDIE 7 and 2 for EDDIE 8. The top value represents the average $Best_{Fit}$ at generation 1, and the bottom value represents the average $Best_{Fit}$ at generation 50.

Stock		EDDIE 7	EDDIE 8
BAT	Generation 1	0.6373	0.6635
	Generation 50	0.7320	0.7273
BP	Generation 1	0.6079	0.6138
	Generation 50	0.6612	0.6860
Cadbury	Generation 1	0.6144	0.6236
	Generation 50	0.6822	0.7084
Carnival	Generation 1	0.6298	0.6235
	Generation 50	0.6763	0.6977
Hammerson	Generation 1	0.6121	0.5944
	Generation 50	0.6864	0.6743
Imp. Tobacco	Generation 1	0.6438	0.6651
	Generation 50	0.7178	0.7439
Next	Generation 1	0.6591	0.6655
	Generation 50	0.7257	0.7360
Schroders	Generation 1	0.5935	0.5915
	Generation 50	0.6626	0.6643
Tesco	Generation 1	0.6569	0.6684
	Generation 50	0.7123	0.7346
Unilever	Generation 1	0.5825	0.6073
	Generation 50	0.6613	0.6906

[0,1]. As we can see, EDDIE 8 performs better than EDDIE 7 in 5 stocks (BAT, BP, Carnival, Hammerson, Tesco) and worse in the other 5 (Cadbury, Imperial Tobacco, Next, Schroders, Unilever). In order to test for the statistical significance of these results, we use the Kolmogorov-Smirnov test (K-S). We find that EDDIE 8 is better in only 3 stocks (BP, Carnival, Hammerson) and worse in 4 (Cadbury, Next, Schroders, Unilever), at 5% significance level.

We get a similar picture for the rest of summary statistics results, namely RC, RMC and RF. Regarding the average RC (Fig. 7), EDDIE 8 is significantly better in 2 stocks only (Carnival, Hammerson), whereas it performs worse in 5 (Cadbury, Imperial Tobacco, Next, Schroders, Unilever). Figure 8 shows that EDDIE 8 is better in only 1 stock (Hammerson), in terms of average RMC, whereas EDDIE 7 performs better in 5 (BAT, BP, Next, Tesco, Unilever). Finally, Fig. 9 informs us that EDDIE 8 is better in 3 stocks (BAT, BP, Carnival, Tesco), in terms of RF, and worse in 5 (Cadbury, Imperial Tobacco, Next, Schroders, Unilever). The reader should bear in mind when reading the figures that we are interested in maximizing the values of Fitness and RC, and minimizing the values of RMC and RF. So when we say that EDDIE 8 performs better in terms of fitness and RC, it means that these values have increased; on the other hand, when we say that EDDIE 8 performs better in terms of RMC and RF, this means that these values have decreased. Finally, we should again mention that all of the results reported here have been tested by the K-S test and were found to be significant at 5% significance level.

4.2.2 Best GDTs

In this section, we investigate the improvements and diminutions caused by the best GDT that was evolved by EDDIE 8. From now on, we will be referring to this GDT as *Best-8*. *Best-8* is essentially the GDT with the highest fitness at the end of the training period, among all 50 runs. It is thus the best solution that EDDIE 8 could come up with, after these 50 individual runs. After obtaining *Best-8*, we apply it to the testing period. Likewise, we obtain the best GDT evolved by EDDIE 7, named *Best-7*, and also apply

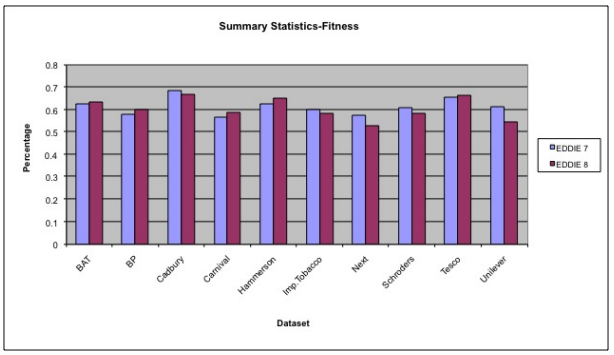


Figure 6 Summary results over 50 runs for fitness for EDDIE 7 and EDDIE 8. Results are normalized to [0,1] scale.

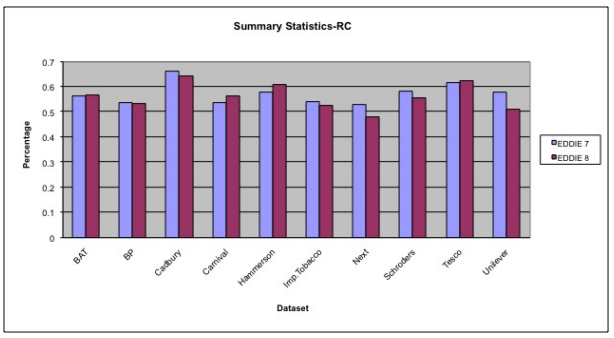


Figure 7 Summary results over 50 runs for RC for EDDIE 7 and EDDIE 8. Results are on the scale of [0,1].

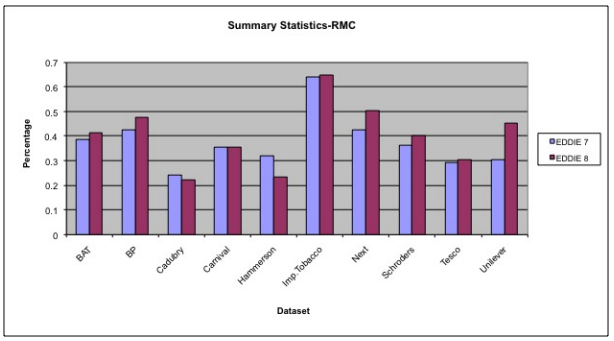


Figure 8 Summary results over 50 runs for RMC for EDDIE 7 and EDDIE 8. Results are on the scale of [0,1].

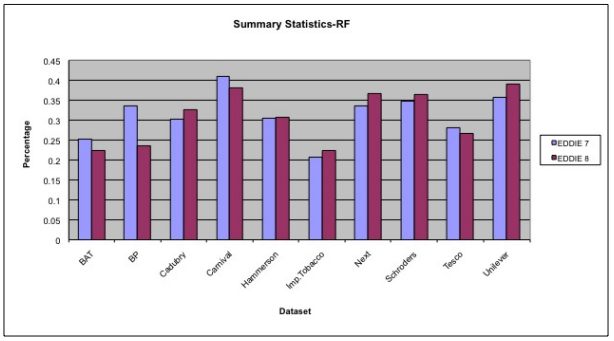


Figure 9 Summary results over 50 runs for RF for EDDIE 7 and EDDIE 8. Results are on the scale of [0,1].

it to the testing period.

The reason for choosing to compare the best GDTs is quite obvious. If an investor was using EDDIE to assist him with his investments, he would run the algorithm many times, and then pick the best GDT that was produced during training. Thus, by comparing *Best-7* and *Best-8*, we can get insight into which EDDIE version would be more effective to an investor's predictions.

Table 5 presents the improvements and diminutions caused by *Best-8*, after having calculated the differences between *Best-7* and *Best-8*, for each metric. Thus, an entry with positive sign indicates that *Best-8* has improved the results in that metric by the respective percentage. Likewise, an entry with negative sign indicates that *Best-8*'s results for that metric have declined by the respective percentage.

In addition, the last two rows of Table 5 present the mean of the above improvements and diminutions. Therefore, when we want to calculate the mean of improvements for Fitness, we sum up the values where Fitness is positive; we hence sum up the Fitness values for BAT (7.31), BP (1.05), Carnival (10.15), Tesco (3.27), Unilever (9.72) and then divide them by 5 (since that is the number of stocks with positive sign). Hence, when we want to calculate the mean of improvements for a metric, we calculate the mean for those values that have *positive sign*. On the other hand, when we want to calculate the mean of diminutions, we calculate the mean for those values that have *negative sign*. The same process stands for all metrics of the table.

Finally, apart from Fitness and the three metrics presented earlier in Sect. 2, Table 5 uses two additional metrics: Average Annualized Rate of Return (AARR), and Rate of Positive Return (RPR). Since the EDDIE application lies in finance, we consider that it would be beneficial to an investor to use as a reference performance criteria that are related to investment return. Obviously, the higher these metrics are, the higher the return for the investor. The formulas for these two additional metrics are presented in the Appendix. We should emphasize that these two metrics are given here only as reference, and are not part of the fitness function that EDDIE 7 and EDDIE 8 use.

What we can observe from Table 5 is that *Best-8* does better than Best-7 for 5 stocks in terms of Fitness (BAT, BP, Carnival, Tesco, Unilever), for 4 stocks in terms of RC (BAT, Carnival, Tesco, Unilever), for 4 stocks in terms of RMC (BAT, Carvival, Imp. Tobacco, Schroders), for 6 stocks in terms of RF (BAT, BP, Carnival, Hammerson, Tesco, Unilever), for 5 stocks in terms of AARR (BAT, BP, Imp. Tobacco, Tesco, Unilever), and for 5 stocks in terms of RPR (BAT, BP, Carnival, Tesco, Unilever). The differences in the values of the metrics are often quite big; for instance, EDDIE 8 has improved the Fitness of BAT and Carnival by 7.31 and 10.15%, respectively. What is even more remarkable is the differences in AARR: 31.03% for Tesco, and 48.81% for Unilever. Similar extremes can be observed for the diminutions. However, it seems that the improvements of *Best-8* have a greater impact than its diminutions.

To make this clearer, let us move our focus to the last two rows of the table, where the mean of *Best-8*'s improvements and diminutions in all metrics is presented. As we can see, improvements have on average had a greater effect than diminutions (6.30% vs -5.22% [Fitness], 8.00% vs -3.83% [RC], 11.64% vs -3.70% [RMC], 7.10% vs -6.96% [RF], 25.55% vs -16.61% [AARR], 8.62% vs -6.82% [RPR]). This is a very important result, because it indicates that an investor using EDDIE 8's best GDT would on average gain more than if he was using EDDIE 7's best tree.

4.2.3 Discussion on the summary statistics results

So far we have presented summary statistics for EDDIE 7 and EDDIE 8. From what we saw in the previous sections, EDDIE 7 outperforms EDDIE 8 in more stocks, in terms of all average statistics (Fitness, RC, RMC and RF). On the other hand, EDDIE 8 outperforms EDDIE 7 in terms of the average results of the best GDT.

An interesting observation from the above is that although EDDIE 8's best GDT can on average perform better than the one of EDDIE 7, this superiority is not reflected in the mean values of Fitness, RC, RMC, and RF. EDDIE 8 is able to come up with very good GDTs, sometimes even better than EDDIE 7's. However, the problem is that it does not come up with such trees often enough. Figure 10 illustrates this problem. It presents the relationship between performance (i.e. fitness) (x-axis) and precision (y-axis). It is divided into two parts. The top graph (Fig. 10a) presents the performance-precision values for stocks where EDDIE 8's average fitness is lower than EDDIE 7's. Let us denote these two fitness values by $ED8_{Fit}$ and $ED7_{Fit}$, respectively. The bottom graph (Fig. 10b) presents the performance-precision relationship for stocks where $ED8_{Fit} > ED7_{Fit}$.

What we can observe from Fig. 10 is that EDDIE 8 always has lower precision than EDDIE 7 for stocks where $ED8_{Fit} < ED7_{Fit}$. This indicates that EDDIE 8's GDTs are spread in a bigger fitness range than the ones of EDDIE 7. It seems that *there is something preventing EDDIE 8 from having results with high fitness more often*. The picture is exactly opposite in Fig. 10b, where $ED8_{Fit} > ED7_{Fit}$. We can see that here EDDIE 8 is not having difficulties finding good solutions, with precision at least as good as EDDIE 7's.

To summarize, the conclusions we can draw are the following:

- EDDIE 8 can perform better than EDDIE 7
- However, there are stocks where $ED8_{Fit} < ED7_{Fit}$
- EDDIE 8's best GDT does on average better than EDDIE 7's best GDT
- EDDIE 8's precision is lower than EDDIE 7's, for stocks where $ED8_{Fit} < ED7_{Fit}$. This does not happen for stocks where $ED8_{Fit} > ED7_{Fit}$
- Therefore, there is something which prevents EDDIE 8 from returning high fitness GDTs more often. This *unknown factor* reduces EDDIE 8's precision and only happens when $ED8_{Fit} < ED7_{Fit}$.

Hence, our next goal is to identify the reason that EDDIE 8 cannot return high fitness GDTs more often, for the stocks where $ED8_{Fit} < ED7_{Fit}$. One explanation could be that there is something special in the nature of the patterns of these stocks. We therefore need to deepen our analysis and try to explain *when* and *why* EDDIE 8 outperforms EDDIE 7.

5 Artificial Datasets

So far, the experiments were tested under 10 empirical datasets. As we saw, results cannot be considered conclusive, since it is not yet clear why EDDIE 8 cannot always outperform EDDIE 7. This section attempts to provide an answer to this question, by presenting some previously derived results,[27] where we used artificial datasets.

Table 5 Improvements and diminutions of *Best-8* in Fitness and the other metrics, for all 10 stocks. The data presented in the first ten rows is basically the difference between metric values of EDDIE 8 and EDDIE 7. Finally, the last two rows of the table present the mean of the improvements and diminutions of the *Best-8* to the metrics.

Stock	Fitness	RC	RMC	RF	AARR	RPR
BAT	7.31%	8.00%	8.07%	5.66%	10.95%	4.08%
BP	1.05%	-1.67%	-1.86%	7.45%	17.07%	5.26%
Cadbury	-10.48%	-11.33%	-17.32%	-6.48%	-7.03%	-7.11%
Carnival	10.15%	10.67%	13.87%	7.86%	-6.19%	10.90%
Hammerson	-0.22%	-0.33%	-1.94%	0.57%	-3.29%	-0.03%
Imp.Tobacco	-1.85%	-1.33%	12.97%	-7.83%	19.90%	-5.31%
Next	-7.59%	-7.33%	-7.07%	-8.28%	-17.92%	-5.31%
Schroders	-0.96%	-1.00%	12.22%	-5.27%	-48.61%	-9.55%
Tesco	3.27%	3.00%	-3.43%	6.05%	31.03%	5.46%
Unilever	9.72%	10.33%	-6.52%	13.91%	48.81%	17.38%
Mean Improvement	6.30%	8.00%	11.64%	7.10%	25.55%	8.62%
Mean Diminution	-4.22%	-3.83%	-3.70%	-6.96%	-16.61%	-6.82%

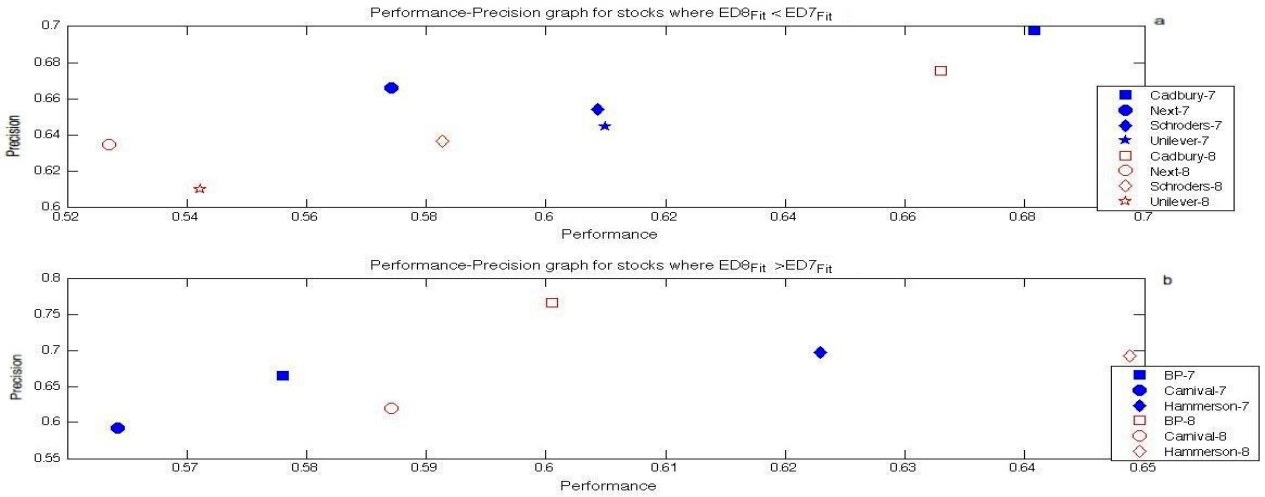


Figure 10 Performance-precision relationship. The x-axis presents the performance (average fitness) and the y-axis the average precision over 50 runs. The top graph (a) presents the stocks where $ED8_{Fit} < ED7_{Fit}$. The bottom graph (b) presents the stocks where $ED8_{Fit} > ED7_{Fit}$.

The reason for using artificial datasets was twofold. First of all, a potential drawback of experimental work with real data is that we cannot be sure that there are always patterns in the data. As the result, a failure of an algorithm to find patterns could also be attributed to this fact. Of course, someone could argue that in our current work both EDDIE 7 and EDDIE 8 have managed to find patterns and that EDDIE 7 just happens to be better in more cases. Nonetheless, using our own artificial dataset can reassure us of the existence of such patterns. At the same time, artificial datasets can guarantee the absence of any noise. The second reason we used artificial datasets was that we could have control over the nature of the patterns. This was very important, because it enabled us to study the weaknesses and strengths of the algorithms, i.e. with what kind of data would EDDIE 7 or EDDIE 8 perform better.

In order to study how different patterns can affect the results, we created two different datasets, one with patterns from EDDIE 7's vocabulary only, and one with patterns from the extended vocabulary of EDDIE 8. It was then found that when patterns came from EDDIE 7's vocabulary, EDDIE 7 would perform significantly better than EDDIE 8. This was an interesting observation, because although EDDIE 7's patterns were included in EDDIE 8's search space, the latter seemed to have difficulties in finding those patterns. On the other hand, when we tested the two versions under the dataset that had patterns from EDDIE 8, EDDIE 7 was outperformed by EDDIE 8, which was of course something we anticipated.

Thus, what we concluded from the experiments under artificial datasets, was that results were indeed affected by the patterns in the dataset. More importantly, it seems that there is a trade-off between 'searching in a bigger space' and 'search effectiveness'. Hence, when patterns come from EDDIE 7's limited vocabulary, EDDIE 8 is having difficulties in searching effectively in such a small search space. The solutions are indeed in its search space, but because they come from a very small area of it, EDDIE 8 cannot only focus its search in this area. The search space has increased exponentially and there is an obvious trade-off between the more expressive language that EDDIE 8 provides and the search efficiency of EDDIE 7.

6 Extending the Artificial Datasets' Results

So far, we have made two valuable observations in Sect. 4 and 5:

- EDDIE 8 has lower precision than EDDIE 7, for stocks where $ED8_{Fit} < ED7_{Fit}$
- EDDIE 8 performs better than EDDIE 7 (on artificial datasets), when patterns come from EDDIE 8's vocabulary. If, on the other hand, patterns come from EDDIE 7's vocabulary, then EDDIE 8 is having difficulties discovering them, and thus ends up with lower performance

We also said at the end of Sect. 4 that a plausible explanation for EDDIE 8's lower precision is that the nature of the patterns in the data prevents EDDIE 8 from performing well more often. Now, after having the insight from the artificial datasets' results, we want to see if we can apply our conclusions to the empirical datasets. We shall therefore move our focus to the indicators that EDDIE 8's GDTs use and examine their relation with EDDIE 7's vocabulary. We saw earlier that if patterns in the hidden function come from EDDIE 7's vocabulary, then EDDIE 8 is having difficulties discovering them. This is what we are going to investigate now with the empirical datasets. Our aim is to show that when $ED8_{Fit} < ED7_{Fit}$, it is because the GDTs of EDDIE 8 contain a high percentage of indicators that come from the vocabulary of EDDIE 7, or indicators very close to it. If this happens, it means that EDDIE 8 needs to look for patterns in a very small search space, and thus faces difficulties in doing so.

One more thing to say is that here there are no hidden functions that EDDIE 8 is trying to discover. When dealing with empirical datasets, we have "solutions". A solution should be considered as the GDT that had the highest fitness at the end of the training period, and was then applied to the testing period. This GDT was the best solution EDDIE 8 could come up with for that specific run, for that specific dataset.

Let us now have a look into the components of the best solution of EDDIE 8, which as we said in Sect. 4.2.2 is called *Best-8*. A reminder that *Best-8* is obtained by first getting the "best GDT" (solution) per each individual run. We thus have 50 best GDTs, which presumably have high fitness. We finally pick the best one among them. *Best-8* is therefore the best tree that EDDIE 8 could find among a total of 50 runs[3]. We want to examine the components of *Best-8*, and calculate the percentage of indicators that come from the vocabulary of EDDIE 7. Figures 11 and 12 present us these results, for stocks where $ED8_{Fit} < ED7_{Fit}$ (11) and stocks where $ED8_{Fit} > ED7_{Fit}$ (12). The x-axis shows the number of days that an indicator of EDDIE 8 is away from the pre-specified indicators of EDDIE 7. For instance, "+/-1" means that EDDIE 8's indicator has a distance from EDDIE 7's indicators by a +1 or -1 day. Thus, since EDDIE 7's indicators have lengths 12 and 50 days, EDDIE 8's indicators in this example could be 11, 12, 13, 49, 50 and 51. The y-axis presents the percentage of EDDIE 8's indicators that come from EDDIE 7's vocabulary.

As we can see from Fig. 11, even though none of the 4 stocks' *Best-8* trees are using any indicators from the vocabulary of EDDIE 7 (all stocks have 0% at +/- 0 days), they are using indicators in a very close range. To be more specific, 50-60% of the *Best-8* indicators for these 4 stocks are close to indicators from EDDIE 7's grammar, in a range of [-4,+4] days; this percentage increases to 50-80% for range [-6,+6] days.

[3]The ideal case would of course be to consider as a solution a GDT that would be able to fit the data with 100% accuracy. However, this is not possible in forecasting with real data. We should thus consider as a solution other GDTs, which would be able to fit the data sufficiently well. In our case, we choose the best performing GDT out of 50 runs.

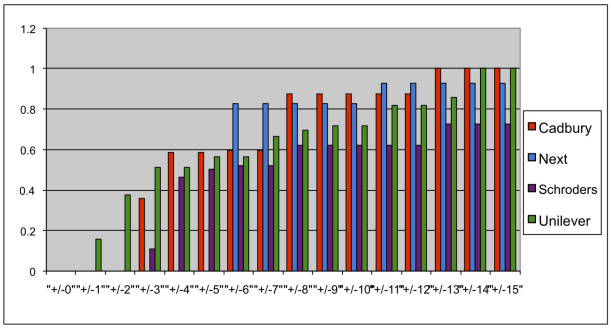


Figure 11 Percentage of EDDIE 8's indicators that are close to EDDIE 7's vocabulary, for the stocks that $ED8_{Fit} < ED7_{Fit}$. This percentage can be viewed in the y-axis. The x-axis presents the number of days that an EDDIE 8's indicator is away from the pre-specified indicators of EDDIE 7

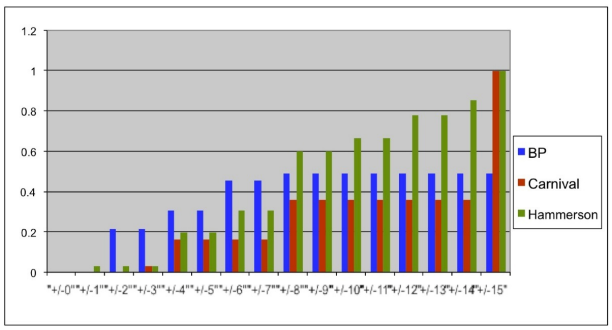


Figure 12 Percentage of EDDIE 8's indicators that are close to EDDIE 7's vocabulary, for the stocks that $ED8_{Fit} > ED7_{Fit}$. This percentage can be viewed in the y-axis. The x-axis presents the number of days that an EDDIE 8's indicator is away from the pre-specified indicators of EDDIE 7.

On the other hand, for stocks where EDDIE 7 is outperformed by EDDIE 8 (Fig. 12), the previous percentage is much lower. For the range of [-4,+4] days, *Best-8* for all 3 stocks has a percentage of 18-30%. For the range of [-6,+6] days, this percentage increases only a little, and is in the range of 18-44%, which is clearly much lower than the percentages we observed in Fig. 11.

Our theory hence seems to be verified. EDDIE 8's performance is indeed affected by the nature of the patterns in the GDTs. When these patterns come from EDDIE 8's broader vocabulary, then EDDIE 8 has no problem finding these GDTs. On the other hand, when solutions come from a very small space (in our case a search space around the one of EDDIE 7), then EDDIE 8 is having difficulties focusing there. This, as a consequence, affects EDDIE 8's performance results, which become poorer than the ones of EDDIE 7.

7 Conclusion

In this paper we presented two investment opportunities forecasting algorithms, EDDIE 7 and EDDIE 8. EDDIE 8 is an extension of EDDIE 7, because it extends its grammar. Traditionally, EDDIE 7 and other similar GP algorithms use predefined indicators to forecast the future movements of the price with a pre-specified period length. In this approach, we suggested that it should be left to the GP to decide the optimal period length. EDDIE 8 is thus an improvement to the previous algorithm because it has richer grammar and also because it can come up with solutions that EDDIE 7 could never discover. In addition, the improvements introduced by the best GDT evolved by EDDIE 8, called *Best-8*, have on average a greater impact than its diminutions. This is quite an important finding, because it indicates that an investor using EDDIE 8's best GDT would on average gain more than if he was using EDDIE 7's best tree. More specifically, we found that EDDIE 8's best tree was, on average, able to outperform EDDIE 7's best tree in terms of all performance measures. In addition, *Best-8* had significantly higher average annual return (AARR) than *Best-7*, which means that on an annual basis, an investor would make more profit if he was using EDDIE 8. The above thus allows us to characterize EDDIE 8 as a successful extension to its predecessor, and a valuable forecasting tool.

However, there seems to be a trade-off between 'discovering new solutions' and 'effective search'. Results from 10 empirical datasets from FTSE100 showed that EDDIE 8 cannot always outperform EDDIE 7. In order to further understand this behaviour, we used previously derived results under artificial datasets[27]. Those results were suggesting that EDDIE 8 can outperform EDDIE 7, as long as the solutions come from its own vocabulary. If they come from EDDIE 7's, then EDDIE 8 is having difficulties finding these solutions, due to the fact that it has to look in EDDIE 7's narrow search space. These results were also verified by our empirical datasets.

We can thus conclude that the current version of EDDIE 8 has its limitations. Nevertheless, EDDIE 8 is still a very valuable tool, due to the fact that it can guarantee significantly higher profits than its predecessor, as already explained.

Future research will focus on improving EDDIE 8's search effectiveness. Of course there are different ways to do this. A promising way that we have already investigated is hyper-heuristics[32–35], where we created a framework for financial forecasting[36]. The results were promising, because hyper-heuristics led to a significant decrease in the RMC; another promising result was that the search became more effective, since more areas of the search space were visited. We believe that more sophisticated hyper-heuristic frameworks, which will include more heuristics than the ones used in[36], can lead to even better results. We therefore intend to focus in that direction. Moreover, another direction of our research could be to produce some new search operators or to create a new constrained fitness function.

Acknowledgements

This work has been supported by the EPSRC grant No. EP/P563361/01

Appendix

Technical Indicators

The following section presents the technical indicators that the GP is using, along with their formulas. We performed a sort of standardization in order to avoid having a very big range of numbers generated by GP, because this would increase the size of the search space even more. Given a price time series [$P(t)$, $t \geq 0$], and a period of length L, Equations (5) - (10) present these formulas.

Moving Average (MA)

$$\text{MA(L,t)} = \frac{P(t) - \frac{1}{L}\sum_{i=1}^{L} P(t-i)}{\frac{1}{L}\sum_{i=1}^{L} P(t-i)} \tag{5}$$

Trade Break Out (TBR)

$$\text{TBR(L,t)} = \frac{P(t) - max\{P(t-1),\ldots,P(t-L)\}}{max\{P(t-1),\ldots,P(t-L)\}} \tag{6}$$

Filter (FLR)

$$\text{FLR(L,t)} = \frac{P(t) - min\{P(t-1),\ldots,P(t-L)\}}{min\{P(t-1),\ldots,P(t-L)\}} \tag{7}$$

Volatility (Vol)

$$\text{Vol(L,t)} = \frac{\sigma(P(t),\ldots,P(t-L+1))}{\frac{1}{L}\sum_{i=1}^{L} P(t-i)} \tag{8}$$

Momentum (Mom)

$$\text{Mom(L,t)} = P(t) - P(t-L) \tag{9}$$

Momentum Moving Average (MomMA)

$$\text{MomMA(L,t)} = \frac{1}{L}\sum_{i=1}^{L} Mom(L,t-i) \tag{10}$$

Additional Performance Measures

Here we present the formulas for the two additional metrics AARR and RPR, as presented in Ref. 28. We would once again like to remind the reader that these metrics should be used for reference only, since they are not part of the fitness function.

Hypothetical Trading Behaviour*: We assume that when a positive position is predicted by a GDT, one unit of money is invested in a stock reflecting the current closing price. If the closing price does rise by r% or more at day t within the next n trading days, we then sell the portfolio at the closing price of day t. If not, we sell the portfolio on the n_{th} day, regardless of the price.*

Given a positive position predicted, for example, the i_{th} positive position, for simplicity, we ignore transaction cost, and annualize its return by the following formula, presented in (B1):

$$ARR_i = \frac{255}{t} * \frac{P_t - P_0}{P_0} \tag{B1}$$

Where P_0 is the buy price, P_t is the sell price, t is the number of days in markets, 255 is the number of total trading days in one calendar year. Given a GDT that generates N_+ number of positive positions over the period examined, its average *ARR* is shown in (B2):

$$AARR = \frac{1}{N}\sum_{i=1}^{N_+} ARR_i \tag{B2}$$

RPR ((B3)) refers to the ratio of the number of signals, which turn out to achieve positive returns, to the total number of positive positions predicted, where a specific GDT is invoked for a finite period

$$RPR = \frac{1}{N_+}\sum_{i=1}^{N_+} I_i \tag{B3}$$

where

$$I_i = \begin{cases} 1 & if ARR_i \geq 0 \\ 0 & otherwise \end{cases}$$

and

$$0 < i \leq N_+$$

where N_+ is the number of positive positions generated by the GDT, and ARR_i is an annualized rate of return for the i_{th} signal.

References

[1] S.-H. Chen, *Genetic Algorithms and Genetic Programming in Computational Finance*. Springer-Verlag New York, LLC, 2002.

[2] J. Binner, G. Kendall, and S.-H. Chen, eds., *Applications of Artificial Intelligence in Finance and Economics*, vol. 19 of *Advances in Econometrics*. Elsevier, 2004.

[3] A. Bouchachia, "Radial basis function nets for time series prediction," *International Journal of Computational Intelligence Systems*, vol. 2, no. 2, pp. 149–157, 2009.

[4] D. Marcek, M. Marcek, and J. Babel, "Granular RBF NN approach and statistical methods applied to modelling and forecasting high frequency data," *International Journal of Computational Intelligence Systems*, vol. 2, no. 4, pp. 353–364, 2009.

[5] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *Computational Intelligence Magazine, IEEE*, vol. 4, pp. 24 –38, may. 2009.

[6] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Systems*, vol. 47, no. 2, pp. 115 – 125, 2009.

[7] T. Roh, "Forecasting the volatility of stock price index," *Expert Systems with Applications*, vol. 33(4), pp. 916–922, 2007.

[8] L. Cao and F. Tay, "Financial forecasting using support vector machines," *Neural Computing & Applications*, vol. 10, pp. 184–192, 2001.

[9] L. Cao and F. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1506–1518, 2006.

[10] T. Gestel, J. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. Moor, and J. Vandewalle, "Financial time series prediction using least squares support vector machines within the evidence framework," *IEEE Transactions on Neural Networks*, vol. 12, pp. 809–821, 2001.

[11] M. Lam, "Neural network techniques for financial performance prediction: integrating fundamental and technical analysis," *Decision Support Systems*, vol. 37(4), pp. 567–581, 2004.

[12] D. West, S. Dellana, and J. Qian, "Neural network ensemble strategies for financial decision applications," *Computers & Operations Research*, vol. 32(10), pp. 2543–2559, 2005.

[13] B. Wong and Y. Selvi, "Neural network applications in finace: A review and analysis of literature (1990-1996)," *Information & Management*, vol. 34, pp. 129–139, 1998.

[14] G. Zhang, B. Patuwo, and M. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, pp. 35–62, 1998.

[15] S. Bhattacharyya, O. Pictet, and G. Zumbach, "Knowledge-intensive genetic discovery in foreign exchange markets," *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 169 –181, apr. 2002.

[16] M. Dempster, T. Payne, Y. Romahi, and G. Thompson, "Computational learning techniques for intraday FX trading using popular technical indicators," *Neural Networks, IEEE Transactions on*, vol. 12, pp. 744 –754, jul. 2001.

[17] Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu, "Credit rating analysis with support vector machines and neural networks: a market comparative study," *Decision Support Systems*, vol. 37, pp. 543–558, 2004.

[18] K. Kim, "Artificial neural networks with evolutionary instance selection for financial forecasting," *Expert Systems with Applications*, vol. 30, pp. 519–526, 2006.

[19] E. Tsang, J. Li, S. Markose, H. Er, A. Salhi, and G. Iori, "EDDIE in financial decision making," *Journal of Management and Economics*, vol. 4(4), 2000.

[20] J. Li and E. Tsang, "Improving technical analysis predictions: An application of genetic programming," in *The 12th International FLAIRS Conference*, (USA), pp. 108–112, 1999.

[21] E. Tsang, S. Markose, and H. Er, "Chance discovery in stock index option and future arbitrage," *New Mathematics and Natural Computation, World Scientific*, vol. 1(3), pp. 435–447, 2005.

[22] J. Koza, *Genetic Programming: On the programming of computers by means of natural selection*. Cambridge, MA: MIT Press, 1992.

[23] R. Poli, W. Langdon, and N. McPhee, *A Field Guide to Genetic Programming*. Lulu.com, 2008.

[24] R. Edwards and J. Magee, "Technical analysis of stock trends," *New York Institute of Finance*, 1992.

[25] M. O'Neill, T. Brabazon, C. Ryan, and J. Collins, "Evolving market index trading rules using grammatical evolution.," in *Proceedings of EvoIASP*, 2001.

[26] I. Dempsey, M. O'Neill, and A. Brabazon, "Live trading with grammatical evolution," in *Proceedings of the Grammatical Evolution Workshop*, 2004.

[27] M. Kampouridis and E. Tsang, "EDDIE for investment opportunities forecasting: Extending the search space of the GP," in *Proceedings of the IEEE Conference on Evolutionary Computation*, (Barcelona, Spain), pp. 2019–2026, 2010.

[28] J. Li, *FGP: A Genetic Programming-ased Financial Forecasting Tool*. PhD thesis, Department of Computer Science, University of Essex, 2001.

[29] S. Martinez-Jaramillo, *Artificial Financial Markets: An agent-based Approach to Reproduce Stylized Facts and to study the Red Queen Effect*. PhD thesis, CFFEA, University of Essex, 2007.

[30] M. Austin, G. Bates, M. Dempster, V. Leemans, and S. Williams, "Adaptive systems for foreign exchange trading.," *Quantitative Finance*, vol. 4(4), pp. 37–45, 2004.

[31] J. Backus, "The syntax and semantics of the proposed international algebraic language of Zurich," in *International Conference on Information Processing*, pp. 125–132, UNESCO, 1959.

[32] E. Özcan, B. Bilgin, and E. E. Korkmaz, "A comprehensive analysis of hyper-heuristics," *Intelligent Data Analysis*, vol. 12, no. 1, pp. 3–23, 2008.

[33] E. Hart, P. Ross, and J. Nelson, "Solving a real-world problem using an evolving heuristically driven schedule builder," *Evol. Comput.*, vol. 6, no. 1, pp. 61–80, 1998.

[34] E. Burke, B. MacCloumn, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper heuristic for timetabling problems," *European Journal of Operational Research*, vol. 176, pp. 177–192, 2006.

[35] E. Burke, E. Hart, G. Kendall, J. Newall, P. Ross, and S. Schulenburg, *Hyper-heuristics: An emerging direction in modern search technology*, pp. 457–474. Kluwer, 2003.

[36] M. Kampouridis and E. Tsang, "Using hyperheuristics under a GP framework for financial forecasting," in *Proc. Fifth International Conference on Learning and Intelligent Optimization (LION5)* (C. A. Coello Coello, ed.), Lecture Notes in Computer Science, Springer, 2011. forthcoming.