

Dynamic Index Trading Using a Gene Regulatory Network Model

Miguel Nicolau^(✉), Michael O'Neill, and Anthony Brabazon

Natural Computing Research and Applications Group, University College Dublin,
Dublin, Ireland

{Miguel.Nicolau,M.ONeill,Anthony.Brabazon}@ucd.ie

Abstract. This paper presents a realistic study of applying a gene regulatory model to financial prediction. The combined adaptation of evolutionary and developmental processes used in the model highlight its suitability to dynamic domains, and the results obtained show the potential of this approach for real-world trading.

1 Introduction

Recent work in the Evolutionary Computation field has seen a surge of interest in Genetic Regulatory Networks (GRNs) as models for computation [1, 4, 9, 10, 13]. In nature, GRNs are a key element of temporal gene expression regulation in biological organisms, providing the remarkable capacity of cells to respond to their ever-changing surrounding environment.

GRN-based algorithms combine the adaptive power of evolutionary processes with regulatory mechanisms that differential gene expression provides, leading to life-long conditional adaptation to the environment. This makes these algorithms especially useful for noisy and dynamic environments.

One such dynamic and hard to predict environment are financial markets. In this study, a GRN model is applied to the problem of index trading. Experiments were designed to make this problem as realistic as possible, hence using only raw historical prices and their transformations, and relatively short trading periods, focusing on the dynamic adaptation of the system. The results obtained again highlight the advantages and limitations of current GRN models, and their potential as computational devices, and further pave the future for their continued adaptation in the EC community.

This paper starts with a brief introduction to GRNs and the model used, in Section 2. This is followed by an overview of index trading and the methodology used (Section 3). Section 4 presents and analyses the results obtained, and finally conclusions and future work directions are drawn in Section 5.

2 Artificial Gene Regulatory Model

2.1 Background

Gene Regulatory Networks (GRNs) refer to the complex networks of gene regulation occurring in cell environments. Given a suitable environment, segments of

DNA encoding genes are transcribed into RNA strands, which, through a translation process, are used to form sequences of amino-acids, thus creating proteins. Some of these proteins are called *Transcription Factors*, and their role is to help create an environment that either enhances or inhibits the expression of genes. This leads to complex networks of regulation, with genes encoding proteins that themselves enhance or inhibit the expression of proteins from other genes.

2.2 The Model

Typically, artificial GRN models are a broad simplification of their biological counterpart. In this work, a model originally presented by Wolfgang Banzhaf [1] is used; it was shown to exhibit similar dynamics to real world GRNs [2], and has been applied to dynamic control problems (such as the pole-balancing benchmark [13] and index trading [12]).

The model consists of a binary linear genome, which is scanned for 32 bit binary promoter sequences, identifying gene locations. Once a promoter is found, the 2×32 bits preceding it represent two regulatory sites (an enhancer and a inhibitor), and the following 5×32 bits represent the gene contents (used to encode a protein); this is shown in Fig. 1.

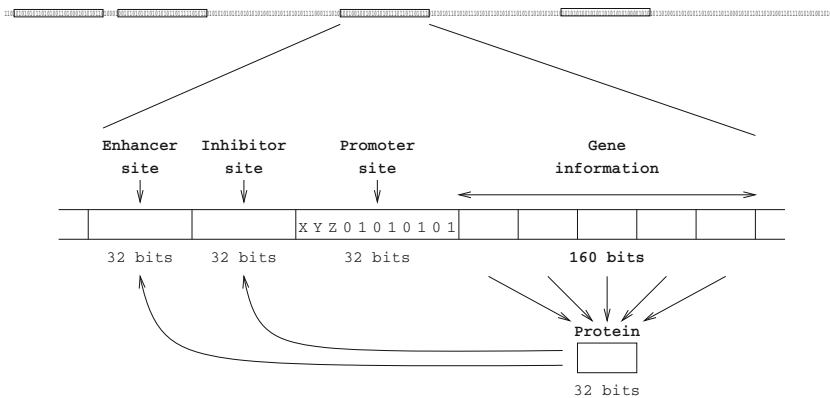


Fig. 1. Bit string encoding of a gene. If a promoter site is found, the gene information is used to create a protein, whose quantity is regulated by the attachment of proteins to the enhancer and inhibitor sites.

Each protein encoded by a gene is a Transcription Factor (TF), that is, it is a regulatory protein, whose role is to affect the rate of expression of all genes (including the one that produced it). Proteins are 32 bit binary sequences, extracted using a majority rule from the 5 sequences of 32 bits that compose the gene information (i.e., if 3 or more equally located bits are set to 1, then the corresponding bit in the protein is also set to 1).

Proteins are bound to regulatory sites via an exclusive-or matching of their respective 32 bit signatures (i.e., the number of different bits in protein signatures and regulatory sites determines the regulatory strength). The enhancing and inhibiting signals regulating the production of each protein p_i are calculated as:

$$e_i, h_i = \frac{1}{N} \sum_{j=1}^N c_j \exp(\beta(u_j - u_{max})) \quad , \quad (1)$$

where N is the total number of proteins, c_j is the concentration of protein j , u_j is the number of complementary bits between the (enhancing or inhibitory) regulatory site and protein j , u_{max} is the maximum match observed in the current genome, and β is a positive scaling factor.

The concentration of protein p_i is calculated using a differential equation:

$$\frac{dc_i}{dt} = \delta(e_i - h_i)c_i \quad , \quad (2)$$

where δ is a positive scaling factor (representing a time unit). All the concentrations are normalised at each time step, ensuring that $\sum_i c_i = 1.0$ at all times; this results in competition for resources within the cell environment.

Input and Output. This model has been extended with the notion of inputs and outputs [13], to facilitate its application to computation. In order to encode inputs, extra regulatory proteins (EPs) are injected into the system. These are not produced by any gene, yet also contribute to the regulation of gene expression. They represent the variables required to describe the state of the environment, and their concentrations reflect the (normalised) value of those inputs.

To extract output signals, genes are divided into two classes: TF-genes (genes encoding transcription factors), and P-genes (encoding *product* proteins). The concentration of proteins produced by P-genes can then be used as output signals of the system. The approach taken here is the same as in previous studies [12].

3 Index Trading

In the Financial domain, an index is a composite measure of price changes in a portfolio of shares in a market. Investors who wish to proxy the return of the index can trade it using index funds (EFTs), which offer low expense ratios and high liquidity. These investments are very popular and are the focus of our study.

Evolutionary algorithms have been successfully applied to financial modelling; the reasons for their applicability include their ability to efficiently explore the search space, and uncover dependencies between input variables, leading to their proper inclusion in the final models [5]. Brabazon and O'Neill [3] provide an overview of the application of evolutionary computation to financial modelling.

3.1 Methodology

The trading methodology is based in previous studies [7, 12, 14], where a trader issues *buy*, *sell*, or *do nothing* signals for each day of the training or test periods. Starting with a capital of \$10000, if a buy signal is issued, 10% of the total funds (initial capital plus earnings) are invested in the index; this position is automatically closed after a ten day period. If a sell signal is issued, an investment of 10% is sold short, and also closed after ten days. This ensures that the system cannot overtrade at any point (i.e., issue a trade signal with no funds available)¹.

The profit or loss at the end of each trading period uses a conservative estimate of one-way trading costs and slippage of 0.2% and 0.3%, respectively. Uncommitted funds take into account a risk-free rate of return, which is approximated using the average interest rate over the entire dataset.

3.2 Datasets

The work presented here follows closely the methodology of previous applications of Grammatical Evolution [3, 14] and GRNs [12] to index trading, and uses three datasets, from the UK FTSE 100 index, the Japan Nikkei index, and the German Dax index. To keep the results comparable, all data is drawn from the period between 16/4/1991 and 21/10/1996; Fig. 2 plots each dataset. These were divided into four training periods and twelve test periods, of 90 days each, with the latter representing the period where the system has gone *live*.

These datasets highlight the potential risk of overfitting the training period. The FTSE training period exhibits a very unstable, slightly downwards trend, whereas the test period exhibits a clear growth trend. The Dax index shows a slight growth in the training period, with a sudden drop towards the end, which is somewhat consistent with the test period. Finally, the Nikkei training period exhibits a very strong decline in index value, followed by an unstable test period, consisting of medium term upwards and downwards trends.

3.3 Data Preprocessing

In previous studies [3, 12], the data was pre-processed prior to evolution, with the raw prices initially transformed into a moving average with a 75 day gap, and then normalised into the range of 0 to 1. However, as the current study focuses on application to real market trading, both of these pre-processing steps are troublesome, as detailed below, and hence were not used.

Moving Average. Working with a moving average smoothes the price curve, but at a cost - the description of trends is also smoothed out. This is exemplified in Fig. 3, for the FTSE market. In the first year of data, for example, the market switches from upwards to downwards trends in a short period of time. This is clearly visible at the beginning of test period 1 (T1), where the index value is

¹ In the final 10 days of each period, all trade actions are ignored.

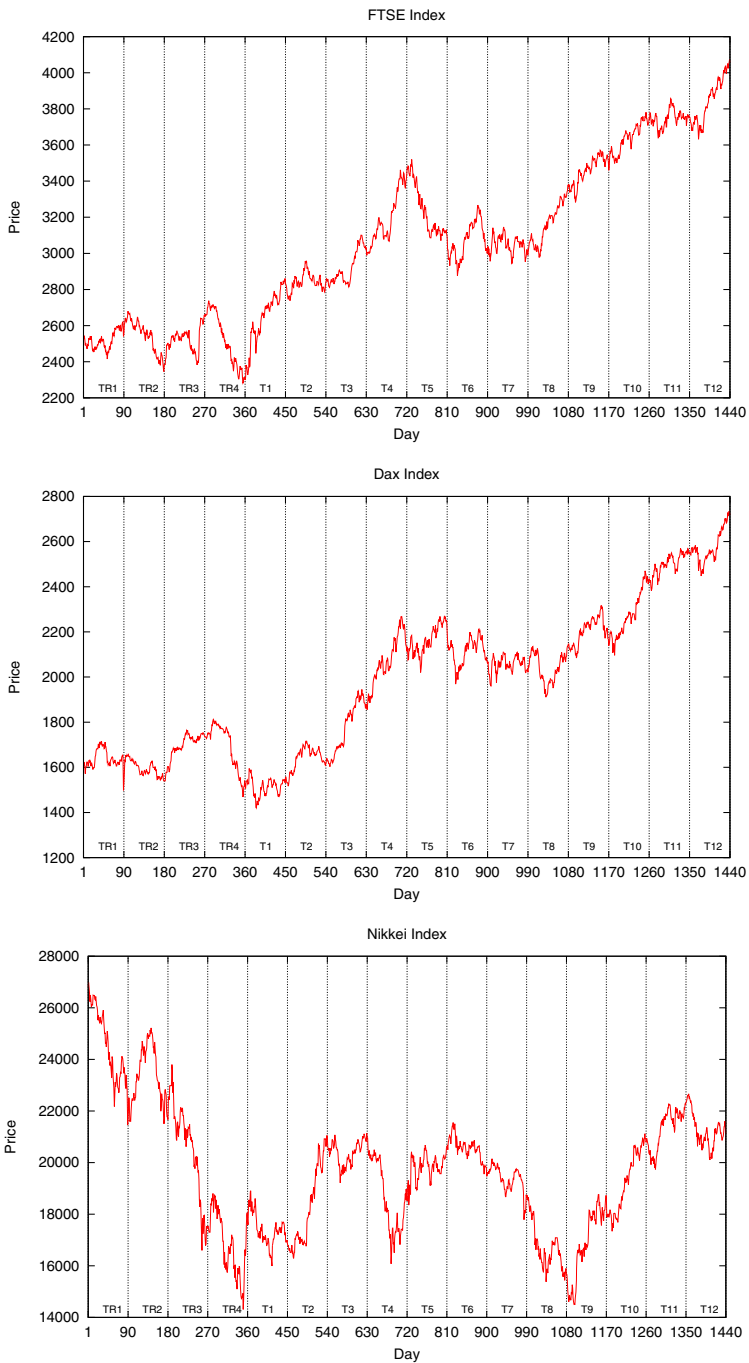


Fig. 2. Plots of the three markets used in this study, along with their training and live (test) periods. Each period consists of 90 trading days; data ranges from 16/4/1991 to 21/10/1996.

growing, but the $MA(75)$ is still reflecting the previous downwards trend. As the current experiment uses fairly short trading periods (90 days), a 75 day moving average is too slow to indicate the current trend of the index. Working with a smaller value ($MA(10)$) reduces this problem, but as all trading periods are of 10 days, the indication of trends is still sometimes deceptive.

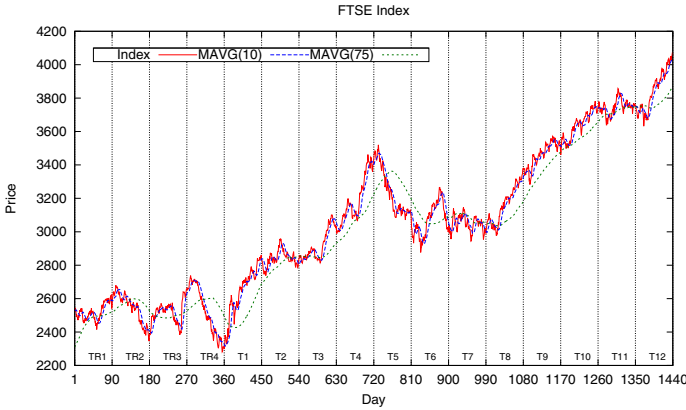


Fig. 3. Index closing values and 10 and 75-day moving average, for the FTSE market

Normalisation. Normalisation can also introduce problems. As the range of future index values cannot be known, a minimum and maximum value must be set for normalisation. On certain markets, this is problematic for model induction, because the range of values encountered in training might be quite different from the range of the test period. Fig. 4 highlights this problem on the FTSE market. As the training period has the range $[2281.0, 2737.8]$, normalising over this range would mean that all normalised values from test period $T2$ onwards would have a capped value of 1.0. Even if the full range of prices $[2281.0, 4073.1]$ were somehow guessed at the start, this would still be problematic, as the training period would only expose the models to a $[0, 0.255]$ range.

3.4 Technical Indicators

Rather than just working with raw and historical market price data, it is typical in the financial domain to derive information from the raw data series into *technical indicators*. These look to predict future price levels, or more generally market trends. Although a potentially infinite number of such indicators may exist, certain classes of indicators are regularly used by investors [7, 15]. The following were used in this study²:

² To minimise price range issues, all price data used is logarithmic; the n -day periods used are typical values from the literature.

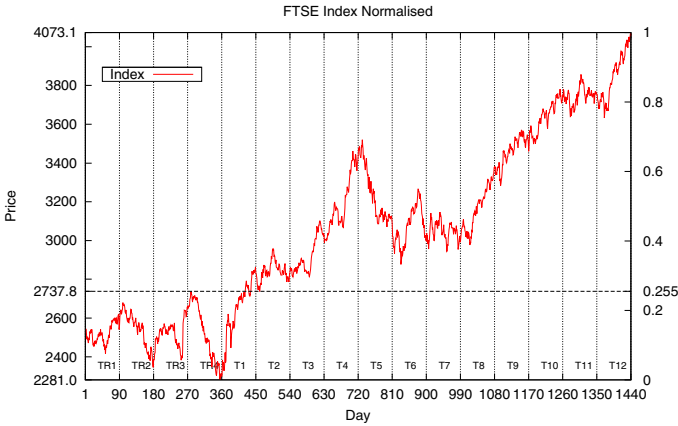


Fig. 4. FTSE index, normalised over the range [2281.0, 4073.1]

- **Moving Average Convergence Divergence (MACD)**. The MACD [11] is a popular indicator: it is typically calculated by subtracting the exponential moving average (EMA) of the last 12 days from the 26-day EMA.
- **Relative Strength Index (RSI)**. The RSI is a momentum indicator; it calculates an upward or downward charge per trading period, and returns the ratio of the EMA of these charges [17]. A 14-day RSI is used.
- **Stochastic Oscillator (sOsc)**. This indicator returns the relative location of the current price in relation to its full price range over a period of n days (a 14-day period was used); it tries to predict price turning points [6].
- **Premier Stochastic Oscillator (psOsc)**. The psOsc is based on an 8-day sOsc, which is smoothed using a 5-day double EMA [8]. This smooths and evens out the response to market changes.

4 Setup and Results

4.1 Encoding

The four technical indicators used ($MACD(26, 12)$, $RSI(14)$, $sOsc(14)$ and $psOsc(8, 5)$) were encoded using EPs, as explained in section 2.2. The choice of binary signature for the EPs can influence the system. In previous studies [12], signatures as different as possible from each other were chosen, but this created dependencies between them (i.e., for a regulatory site to fully match one, it had to fully ignore another). To try to minimise these dependencies, the following encodings were used:

```

MACD(26, 12): 00000000000000000000000000000000
RSI(14):      00000000000000001111111111111111
sOsc(14):    00000000111111110000000011111111
psOsc(8, 5): 00000000111111111111111110000000
    
```

The initial rate of expression of all genes in a model was initially the same, and the system was first allowed to settle for a maximum of 100000 regulatory iterations, or until all protein concentrations were stabilised; after this period, the trading session begins. To synchronise the GRN with the trading simulator, a trading signal was extracted every 2000 protein iterations.

To extract a trading signal from the network, the concentration of a given P-gene is used (all P-genes are tested, and the best result is chosen):

$$c_i \geq 66\% \rightarrow \text{BUY} \quad 66\% > c_i > 33\% \rightarrow \text{D/N} \quad c_i \leq 33\% \rightarrow \text{SELL}$$

This methodology thus encodes technical indicators as regulatory proteins, which influence the internal regulatory process of the genome, and therefore influence the resulting concentration of P-genes, which can then be interpreted as a trading signal. It is a very similar process as seen in previous applications of GRNs to time-series datasets [12, 13].

4.2 Evolutionary Setup

A (250+250)–*ES* evolutionary strategy was used to evolve the binary genomes: a population of 250 individuals is used to create 250 offspring, and the best 250 of all parents and offspring are used as the new parent population (a maximum of 100 iterations were allowed). The variation operator used was a bit-flip mutation, set to 1% and adapted by the 1/5 rule of Evolution Strategies [16].

4.3 Evaluation

Two approaches were used when deriving models: the first denominated *Fixed*, and the second *Dynamic*. Each was run independently on the three markets; the training periods (TR1→TR4, see Fig. 2) were used to derive a trading model.

At the end of the evolutionary process, the best *Fixed* approach model is applied to all test periods (T1→T12). The *Dynamic* approach, however, is only tested on period T1; it is then reprocessed in a smaller evolutionary process (50 ES iterations), using a moving window of 4 training periods each time: train in TR2→T1, test in T2; train in TR3→T2, test in T3; and so on.

As noted in Section 3, long term investments tend to produce good returns in historically upwards return indexes. A common passive investment strategy is *Buy & Hold* (B&H), where an investment is made and held for a long time. In order to evaluate the performance of the evolved traders, their performance was compared to a B&H strategy in both the training and test datasets.

As seen in previous studies [3, 12, 14], it would be inadequate to simply calculate fitness as the profit return, as this does not consider the risk of deploying an

evolved trader. A measurement of this risk is provided by the maximum draw-down, that is, the maximum cumulative loss of the system during each of the datasets. This can be incorporated into the fitness calculation by subtracting the maximum cumulative loss from the profit of each period.

4.4 Results and Analysis

For both approaches, 50 independent runs were done for each market. Table 1 shows the best models in each market, chosen by their TR1→TR4 training performance. As expected, both evolved traders do quite well on the training periods, both due to the obvious fact that they were optimised for those periods, but also because of the downwards trend of period TR1→TR4 on all markets (as highlighted in Section 3.2), which hampers the gains of the B&H benchmark.

Once the traders go *live*, the figures change considerably. In upwards trend markets like FTSE and DAX, the B&H benchmark performs very well, and is very hard to improve on that performance; only the dynamic approach was able to achieve better test performance, in the FTSE market. In the Nikkei market, however, with its fluctuating and slightly downwards trend, both traders achieved better performance. The Dynamic trader in particular is on par or superior to similar EC approaches found in the literature [3, 12, 14].

It is interesting to observe the behaviour of both evolved traders; Fig. 5 plots the best Fixed and Dynamic FTSE traders. As the training period TR1→TR4 has no clear trend, cautious traders that mostly take no risk are evolved, profiting from rate of interest returns in funds not invested. Only the TR4 period exposes the system to a downwards trend.

Once the evolved Fixed trader goes live, it can be seen that it keeps the same cautious behaviour. However, in the periods T1→T12, the market exhibits an upwards trend, which the trader seldom identifies. This is clearly visible in the period T8→T12. The Dynamic trader, however, is constantly exposed to the changing market trend, and adapts to a more aggressive (and profitable) buying behaviour. This is again clearly visible in the period T8→T12, where at each new live period, more and more buy actions are generated.

Although the better approach, the Dynamic trader is not always the best. In the Nikkei market, for example, it is not fast enough to adapt to the instability of the index, leading to periods (T3, T4, T6, T9 and T11) where the Fixed approach generates more profit; these are periods of sudden trend change, where the Dynamic trader has been trained on the previous period. At the end of all test periods, however, the Dynamic trader performance is still clearly superior.

Table 1. Best evolved traders compared to Buy & Hold benchmark, on the FTSE, Dax and Nikkei markets (net profit in dollars)

Period	FTSE			Dax			Nikkei		
	Buy & Hold	Static	Dynamic	Buy & Hold	Static	Dynamic	Buy & Hold	Static	Dynamic
TR1 (1 to 90)	35.69	209.59	209.59	-1413.58	284.90	284.90	-4094.96	1156.8	1156.8
TR2 (91 to 180)	-1734.99	137.16	137.16	34.7648	259.92	259.92	-16.34	955.46	955.464
TR3 (181 to 270)	1100.9	1267.02	1267.02	1267.77	161.96	161.96	-3989.76	2126.39	2126.39
TR4 (271 to 360)	-2650.98	928.87	928.87	-2341.59	1447.94	1447.94	255.12	2718.93	2718.93
Total	-3249.38	2542.64	2542.64	-2452.64	2154.72	2154.72	-7845.94	6957.58	6957.58
Test									
T1 (361 to 450)	2402.85	227.71	227.71	34.82	-581.02	-581.02	-940.44	175.93	175.93
T2 (451 to 540)	-124.51	39.19	598.78	289.72	200.16	196.48	2451.63	51.91	450.61
T3 (541 to 630)	617.82	719.40	799.46	1538.52	363.96	594.47	49.81	57.78	36.78
T4 (631 to 720)	1206.9	1068.92	1300.30	1317.47	-26.33	1213.62	-2221.33	1393.61	976.81
T5 (721 to 810)	-2010.04	209.59	285.05	594.23	295.88	319.75	1060.15	-935.99	-605.42
T6 (811 to 900)	-675.91	247.55	108.72	-1281.42	204.61	249.87	-882.82	214.32	63.37
T7 (901 to 990)	-181.46	355.13	271.38	-328.61	165.47	220.07	-834.97	-233.17	-174.09
T8 (991 to 1080)	1239.55	209.59	196.85	520.64	120.79	624.01	-3051.26	2114.46	2159.62
T9 (1080 to 1170)	337.91	209.59	198.44	55.51	236.78	365.18	1901.6	284.28	-479.75
T10 (1171 to 1260)	722.71	209.59	564.64	1304.42	217.09	314.08	1136.03	-1475.39	1249.46
T11 (1261 to 1350)	-23.49	209.59	216.08	412.75	192.73	328.63	653.651	614.37	-282.48
T12 (1351 to 1440)	795.61	209.59	327.07	669.43	-102.78	516.86	-1021.36	141.93	488.72
Total	4307.09	3915.443	5094.48	5127.48	1287.34	4362.00	-1699.94	2404.04	4059.56

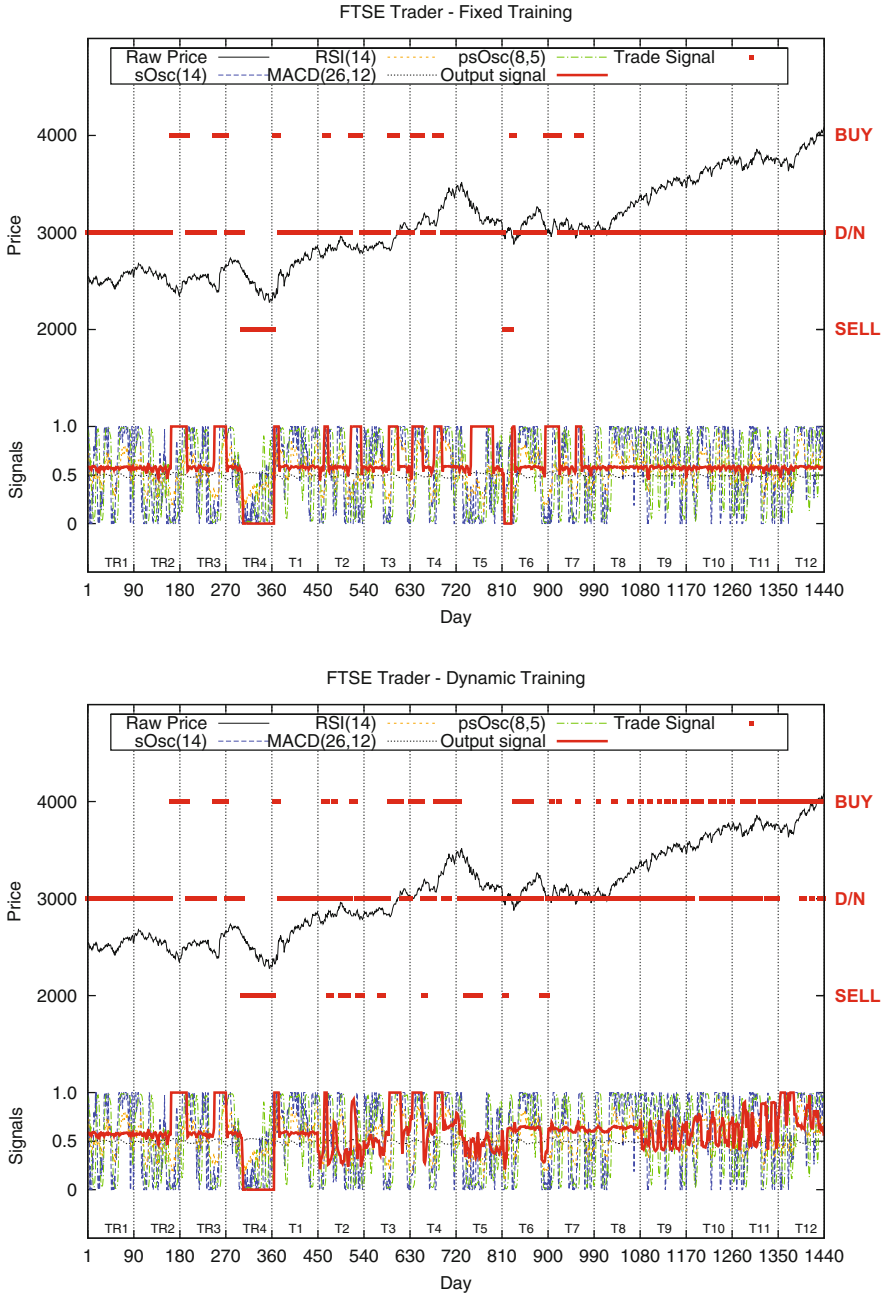


Fig. 5. Best Fixed (top) and Dynamic (bottom) traders for the FTSE market. The top of each plot shows the index value and the generated trade action (buy, sell or do nothing), and the bottom shows the inputs to the GRN (technical indicators) and the generated output signal.

5 Conclusion

In this study, a realistic simulation of applying a GRN model to index trading was presented. Different aspects of feature selection were analysed, and two approaches were applied to three market indexes.

The results obtained show the potential of applying developmental systems to real-world dynamic problems, but also their limitations. The applied developmental system seems unable to adapt to all market fluctuations in unseen data (Static approach), and still requires an extra evolutionary process to adapt to new market tendencies (Dynamic approach). But even the latter approach exhibits signs of overfitting its training data.

Future work will address these issues. The field of Epigenetics shows us that states of cellular organisms can be transmitted to offspring: a similar artificial process could transmit the regulatory state of parents to offspring (in the Dynamic approach), transferring the state of the market to new models, which will trade in later periods. This will allow newly created models to retain a better *historical* state, derived from trading in all previous periods.

References

1. Banzhaf, W.: Artificial regulatory networks and genetic programming. In: Riolo, R., Worzel, B. (eds.) *Genetic Programming Theory and Practice*, ch. 4, pp. 43–62. Kluwer Publishers, Boston (November 2003)
2. Banzhaf, W., Kuo, P.D.: Network motifs in natural and artificial transcriptional regulatory networks. *Biological Physics and Chemistry* **4**(2), 85–92 (2004)
3. Brabazon, A., O'Neill, M.: *Biologically Inspired Algorithms for Financial Modelling*. Springer (2006)
4. Cussat-Blanc, S., Bredeche, N., Luga, H., Duthen, Y., Schoenauer, M.: Artificial gene regulatory networks and spatial computation: A case study. In: Lenaerts, T., et al. (ed.) *Proceedings of ECAL 2011*, pp. –. MIT Press (2011)
5. Iba, H., Nikolaev, N.: Genetic programming polynomial models of financial data series. In: *Proceedings of CEC 2000*, vol. 2, pp. 1459–1466 (2000)
6. Lane, G.C.: Lanes stochastic. *Technical Analysis of Stocks and Commodities* **2**(3), 80 (1984)
7. LeBaron, B., Lakonishok, J., Brock, W.: Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance* **47**(5), 1731–1764 (1992)
8. Leibfarth, L.: Premier stochastic oscillator. *Stocks and Commodities V* **26**(8), 30–36 (2008)
9. Leier, A., Kuo, P.D., Banzhaf, W., Burrage, K.: Evolving Noisy Oscillatory Dynamics in Genetic Regulatory Networks. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) *EuroGP 2006*. LNCS, vol. 3905, pp. 290–299. Springer, Heidelberg (2006)
10. Lopes, R.L., Costa, E.: ReNCoDe: A Regulatory Network Computational Device. In: Silva, S., Foster, J.A., Nicolau, M., Machado, P., Giacobini, M. (eds.) *EuroGP 2011*. LNCS, vol. 6621, pp. 142–153. Springer, Heidelberg (2011)
11. Murphy, J.J.: *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Prentice Hall Pr. (1999)

12. Nicolau, M., O'Neill, M., Brabazon, A.: Applying Genetic Regulatory Networks to Index Trading. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part II. LNCS, vol. 7492, pp. 428–437. Springer, Heidelberg (2012)
13. Nicolau, M., Schoenauer, M., Banzhaf, W.: Evolving Genes to Balance a Pole. In: Esparcia-Alcázar, A.I., Ekárt, A., Silva, S., Dignum, S., Uyar, A.Ş. (eds.) EuroGP 2010. LNCS, vol. 6021, pp. 196–207. Springer, Heidelberg (2010)
14. O'Neill, M., Brabazon, A., Ryan, C., Collins, J.J.: Evolving Market Index Trading Rules Using Grammatical Evolution. In: Boers, E.J.W., Gottlieb, J., Lanzi, P.L., Smith, R.E., Cagnoni, S., Hart, E., Raidl, G.R., Tijink, H. (eds.) EvoIASP 2001, EvoWorkshops 2001, EvoFlight 2001, EvoSTIM 2001, EvoCOP 2001, and EvoLearn 2001. LNCS, vol. 2037, p. 343. Springer, Heidelberg (2001)
15. Pring, M.J.: Technical Analysis Explained: The Successful Investor's Guide to Spotting Investment Trends and Turning Points. McGraw-Hill (1991)
16. Rechenberg, I.: Evolutionsstrategie 1994. Frommann-Holzboog, Stuttgart (1994)
17. Wilder, J.W.: New Concepts in Trading Technical Systems. Trend Research (1978)