Economics Bulletin

Volume 32, Issue 1

An empirical case against the use of genetic-based learning classifier systems as forecasting devices

Jaqueson K. Galimberti
The University of Manchester and The Capes Foundation

Sergio da Silva
Department of Economics, Federal University of Santa Catarina

Abstract

We adapt a genetic-based learning classifier system to a forecast evaluation exercise by making its key parameters endogenous and taking into account the need of convergence of the learning algorithm, an issue usually neglected in the literature. Doing so, we find it hard for the algorithm to beat simpler ones based on recursive regressions and on the random walk in forecasting stock returns. We then argue that our results cast doubts on the plausibility of using learning classifier systems to represent agents process of expectations formation, an approach commonly found into the agent-based computational finance literature.

During first drafts of this paper Jaqueson K. Galimberti was at Federal University of Santa Catarina (Brazil), while several revisions and improvements were conducted at his current affiliation with additional financial support from The Capes Foundation through BEX 0597/10-4 process. We thank for comments on earlier versions of this paper received from Jaylson da Silveira, Leandro Coelho, and Maurício Nunes. We also thank for the report provided by one anonymous referee and to the associate editor assessment. The authors take full responsibility for any remaining errors, of course.
Citation: Jaqueson K. Galimberti and Sergio da Silva, (2012) "An empirical case against the use of genetic-based learning classifier systems as forecasting devices", *Economics Bulletin*, Vol. 32 No. 1 pp. 354-369.
Contact: Jaqueson K. Galimberti - jakaga2002@yahoo.com.br, Sergio da Silva - professorsergiodasilva@gmail.com.
Submitted: August 30, 2011. **Published:** January 25, 2012.

1 Introduction

The literature on agent-based computational finance provides a view on financial markets as composed of interacting groups of learning, bounded-rational agents, and models important features of these markets by means of computationally-oriented simulations of artificial markets. One of the main features of this approach is the explicit modeling of the process of inductive expectations formation on which agents are allowed to adapt their expectations regarding future returns according to the evolution of market states (see Wan et al., 2002; LeBaron, 2006, for reviews on this literature). The implementation of this task vary from model to model, but the most prominent use genetic algorithms (Lettau, 1997; Dawid, 1999), classifier forecasting systems (Arthur et al., 1997), fuzzy logic systems (Tay and Linn, 2001), genetic programming (Chen and Yeh, 2001), and artificial neural networks (Beltratti and Margarita, 1993).

This great variety of alternative models of inductive expectations illustrates the increasingly interest in economics for the modeling of learning in a bottom-up perspective. However, nearly all of these models are in some way set up *ad hoc* without clear justification (Brenner, 2006). Most studies on artificial stock markets have its empirics drawn from the final results obtained from the simulations of the entire model making it hard to identify whether the conclusions derived are directly related to the adopted model of expectations formation, or to any other component of the setup. This raises the questions: Are the results obtained from these studies sensitive to the choice of the expectational model? Are these computationally intensive algorithms needed? How realistic are they? Here we focus on these two latter questions and empirically evaluates the use of a genetic-based learning classifier system algorithm (GBLCSA) for stock returns prediction in a real stock market[1].

A classifier system is an adaptive rule-based system that models an environment considering a set of competing condition-action rules (Holland, 1986; Holland et al., 1986; Booker et al., 1989). When the forecasting environment is the stock market, classifier systems are used as inductive reasoning algorithms to mimic the process of expectation formation (Arthur et al., 1997; LeBaron et al., 1999; Palmer et al., 1999). To allow for learning capabilities to react in changing environments one needs to further consider rule-discovery algorithms, such as a genetic algorithm. Genetic algorithms (Holland, 1962, 1975) constitute a class of search, adaptation, and optimization techniques based on natural evolution. From a starting set of competing rules the algorithm applies selection, crossover, and mutation operators to create a new generation of rules. The process is then replicated a number of times to create successive generations until a termination criterion is met, and the final population of rules delivers a collection of solution candidates.

Here, we present a GBLCSA whose main properties are drawn from the above literature. In the context of a forecasting exercise, however, special care is required into the design of the attached learning algorithm so as to guarantee convergence has been achieved when a forecast is taken out of this algorithm. Given that genetic algorithms are in a great part stochastically determined, repeated executions of the algorithms over an identical set of input data can lead to distinct outputs. In this sense, by convergence of the learning algorithm we mean that (approximately[2]) the same series of forecasts are obtained from repeated executions of the algorithm. We deal with this issue by further developing the learning algorithm to make its key parameters endogenous.

We evaluate the forecasting performance of the GBLCSA using stock market (rather than

[1] As in most applications evaluating stock returns predictability, this study can also be related to the well known literature evaluating the hypothesis of efficient markets (Fama, 1991), but we do not explore this link in detail here.

[2] A precise definition of this approximation is given in the description of the algorithm.

artificially generated) data and taking two computationally simpler benchmarks to contrast with, namely a recursive regressions algorithm (RRA) and a random walk algorithm (RWA). We find that the GBLCSA fails to beat these simpler algorithms. Previous work employing the learning classifier systems to forecast exchange rates (Beltrametti et al., 1997; Stone and Bull, 2008) and stock market indices (Mahfoud and Mani, 1996; Allen and Karjalainen, 1999; Liao and Chen, 2001; Schulenburg and Ross, 2002; Armano et al., 2002; Chen et al., 2007) overall favors the use of classifier systems. The fact that we find divergent evidence here may be due to our concerns on the convergence issue and the correspondent methodological improvements proposed.

From these results, our argument then is that the support to any computational algorithm as representative of agents process of expectations formation depends crucially on its attractiveness as a forecasting device. A similar argument has been drawn by Branch and Evans (2006) to evaluate the empirical plausibility of recursive forecasting algorithms adopted in the literature modeling expectations through adaptive learning (see Evans and Honkapohja, 2001). We acknowledge, however, that it is not possible to achieve an incisive answer to this question solely by looking at agents incentives to use the forecasting device, specially for algorithms having a population interpretation as the one we analyse here (Vriend, 2000). Even if an algorithm has a poor forecasting performance, such that any kind of rationality would preclude its usage, it can still be the case that it stands as a representative of the market-wide population of heterogeneous agents. In this sense, our results stand just as case against the use of genetic-based learning classifier systems as forecasting devices.

The rest of this paper is organized as follows. Section 2 describes the forecasting algorithms. Section 3 presents data and discusses calibration issues. Section 4 shows the forecast evaluation results, while Section 5 concludes this study.

2 Algorithms

The GBLCSA is made up of sets of predictors, where each of them is a condition-forecast rule. While the condition part determines when a particular predictor is activated according to the current state of the market, for example, the forecast part includes a formula for next period returns. A strength value is also assigned to each predictor to measure its past accuracy and thus allow the activated predictors to be selected. Learning capabilities to react in a changing environment are then added. Here, a rule-discovery algorithm (usually a genetic algorithm) is coupled to the classifier system. Figure 1 shows the GBLCSA processing routine.

At each period the classifier system may be represented by a table consisting of N rules that map the states of the market into forecasting parameters. Strength values are associated with these parameters. The states are coded into binary strings made up of l positions, each of which taking on the values 1 or 0 for the current condition of the market. When data are either unavailable to define a market state value or when a rule is evolved as to make a state irrelevant, we denote the resulting state by the wildcard symbol #[3]. Though predefining a set of binary states limits the algorithm to forecast conditioned solely on this information set, the use of the wildcard character aims to remedy this deficiency by allowing the algorithm to dynamically ignore these variables and to focus on the piece of information really relevant (LeBaron et al., 1999).

The forecasting part of each rule is represented by linear equation parameters. Here, we

[3] Initialisation of the condition part of the N rules is done randomly with probabilities: $\# - 50\%$, $1 - 25\%$, and $0 - 25\%$. This set up usually has minor effects on the final results.

take two functional forms having the benchmark algorithms in mind. The first includes a drift,

$$\hat{r}^{drift}_{j,t} = \hat{a}_j + \hat{b}_j r_{t-1}, \tag{1}$$

and the second has no drift term,

$$\hat{r}^{nodrift}_{j,t} = \hat{b}_j r_{t-1}. \tag{2}$$

The parameters are initially set to uniformly distributed random values confined within the lower and upper bounds a_{inf}, a_{sup}, b_{inf}, b_{sup}. These limits are symmetrically set around zero and are picked to include the minimum and maximum values of the corresponding parameters when estimated by recursive regressions. Then, the parameters are left to evolve following the genetic algorithm described below.

The strength s_j of each rule is defined as the reciprocal of the average squared forecast error measure v^2_j, which is adjusted each period for the activated rules according to the exponentially weighted average of the squared forecast error,

$$v^2_{j,t} = \tau v^2_{j,t-1} + (1-\tau)\left(r_t - \hat{r}_{j,t}\right)^2, \tag{3}$$

where τ determines the horizon length considered while evaluating the forecasting performance of each rule. As the strength measure is also used as the selection criterion for the genetic algorithm, τ can be interpreted as a measure of the speed of adjustment of the forecasting rules.

As for the final forecast output provided by the GBLCSA each period, figure 1 shows that after a subset of forecasts is obtained one still has to define how to get a point forecast from them. Here, we suggest the construction of three predictors to obtain the final forecasts from the currently active rules: (1) select the forecast from the rule with greater strength; (2) compute a simple arithmetic mean over rules forecasts; and (3) compute a weighted average over rules forecasts, where the weights are given by the strength of each rule. We thus consider not only the narrower individual-based predictor commonly found in the literature (1) but also the wider population-based predictors (2) and (3).

As usual, for rule discovery we take a genetic algorithm (Holland, 1962, 1975) which is applied on the onset of each period. For the algorithm to operate a number of parameters must be first determined, such as population size, selection pressure, and probabilities of crossover and mutation. To mitigate the problem of dependence of the results on the choice of the initial parameters, we put forward a genetic algorithm with endogenous probabilities of crossover, mutation, and selection pressure. This makes our algorithm a self-adapting solution technique at the individual level (Gibbs et al., 2008). This procedure alleviates the tension in choosing between exploitation of currently known forecasting rules (through crossover) and exploration of seldom-tried forecasting rules (through mutation) that may respond more to the current state of the market. The genetic algorithm routine is described in algorithm 1 at the appendix. In what follows we provide details about each of its main operations.

We suggest the selection pressure S to be determined according to the Shannon entropy of population fitness (San Jose-Revuelta, 2007),

$$S = 2 + (N-2)\left(1 - \sum_{j=1}^{N} \tilde{v}^2_{j,t} \log_N \tilde{v}^2_{j,t}\right), \tag{4}$$

where S is rounded to the nearest integer, and $\tilde{v}^2_{j,t}$ is the normalized average squared error given by $\tilde{v}^2_{j,t} = {}^{v^2_{j,t}}/_{\sum^N_j v^2_{j,t}}$. Thus, selection pressure is increased whenever the diversity of rule strengths increases, as this diversity enhances the exploration of superior solutions.

We suggest the bit-to-bit probability of crossover $p_{c,i}$ to be determined by

$$p_{c,i} = \left({}^{c_i}/_{k^*}\right)\left({}^{k}/_{k^*}\right), \tag{5}$$

where c_i measures how many of the N rules have bit i matching with the current state of the market[4], k is the number of active rules, and k^* is a parameter (see algorithm 1 at the appendix). Whenever $p_{c,i}$ ends up greater than 1, the bit is directly cloned from the parents into the offspring. We consider the probability of mutation as simply the complement of the probability of crossover. Once the mutation operator is activated, the bit is changed to one of the two alternative values, but always favoring the value matching the current state of the market, i.e., a 75% probability of changing to a bit value matching the market state.

The forecasting parameters are also subject to genetic operations. Here, the probabilities of crossover are given by the absolute difference between the parents parameters and the range of possible values. Then, the crossover probability for the forecasting parameter $\hat{b}_j$, e.g, is given by

$$p_{c,b} = {}^{|\hat{b}_1 - \hat{b}_2|}/_{(b_{sup} - b_{inf})}. \tag{6}$$

As before, the probabilities of mutation are the complements of the probabilities of crossover. Equation (6) captures the pursuit of intensifying exploitation (exploration) search when the parents have diverse (similar) forecasting parameters.

However, given that the forecasting parameters are not bit-coded, but real-coded, its genetic operations are formulated distinctly (see Herrera et al., 2003). Here, the parameters of the offspring rule resulting from a crossover are generated by randomly drawing from a normal distribution with mean equal to the average between the parents parameters, and variance given by

$$\sigma_b^2 = {}^{(\hat{b}_1^2 - 2\hat{b}_1\hat{b}_2 + \hat{b}_2^2)}/_{4}, \tag{7}$$

This crossover operation carries some additional degree of exploration search given that the offspring parameters are randomly generated from an unbounded distribution. The mutation operation is executed by simply adding a random shock proportional to the probability of mutation.

The genetic algorithm (see algorithm 1 at the appendix) runs until the set of termination criteria are met. Clearly, these define the main targets of the rule discovery algorithm. Criterion 4a characterizes the adaptation process as state-guided, criterion 4b guarantees convergence of the forecasts, and criterion 4c prevents the algorithm to halt.

Although the termination criterion 4b reinforces the convergence of the genetic learning rule, the forecasts resulting from the GBLCSA still carry some degree of uncertainty due to genetic drift. We thus take averages of 10 runs of the algorithm as the final forecasts to attenuate this problem and also to check for convergence. Ten runs proved to be enough to clear the resulting forecasts from this uncertainty driven by genetic drift.

We now turn to discuss the benchmark algorithms. A recursive regressions algorithm (RRA) is devised to access the same set of information as that of the GBLCSA, that is, the l bits of market states. At each period, a return forecast is obtained by first estimating

$$\hat{r}_t^{drift} = \hat{\alpha}_0 + \sum_{i=1}^{l} \hat{\alpha}_i b_{i,t-1} + \hat{\beta}_0 r_{t-1} + \sum_{i=1}^{l} \hat{\beta}_i b_{i,t-1} r_{t-1}, \tag{8}$$

[4]A unit (1) value is summed to c_i for each rule where the bit i is found to match exactly with the corresponding bit of the market state, and half a unit ($^1/_2$) if the wildcard character # is found.

and

$$\hat{r}_t^{nodrift} = \hat{\beta}_0 r_{t-1} + \sum_{i=1}^{l} \hat{\beta}_i b_{i,t-1} r_{t-1}, \tag{9}$$

using the least squares as well as all the past information. Then, the dependent variable is projected to the next period by considering the estimated parameters and the current state of the market. Equation (8) is related to the one that includes a drift in the GBLCSA (equation (1)), while equation (9) is related to the driftless equation (2).

As a simpler benchmark, we also take a random walk algorithm (RWA) related to the martingale hypothesis (Campbell et al., 1997), which posits that the best forecast of tomorrow's price is simply today's price. The driftless random walk forecast of the next period return is thus zero, while the forecast from the random walk with drift is given by the past returns average.

3 Data and Calibration

To make our case we take the stock of the company Petrobras listed on the Sao Paulo Stock Exchange (Bovespa). We take just one stock because the computational cost to run the GBLCSA is high. The data comprise 276 data points, T, ranging from January 1987 to December 2009. A portion of the data, H, is left out for out-of-sample analysis. We set H to a generous value of 168 (January 1996 to December 2009). This leaves the starting in-sample estimation of the RRA with a number of observations two times the number of parameters estimated. Choosing a large H we reduce the risk of size distortions in the model evaluation stage (Ashley, 2003).

We consider 23 state conditions (table I) which involve information based on both technical and fundamental indicators. The technical indicators are historical moving averages compared with the current level of a variable. The fundamental indicators are price-earnings, price-dividends, price-book, and price-market index ratios along with real exchange rate, oil price, and a dummy for macro stabilization period.

The data consist of accumulated values of one year and are taken from Economatica, Ipeadata, and the Brazilian Central Bank. The real interest rate series is constructed from the interbank certificate of deposit (CDI) rate accumulated over the month minus the inflation rate, and then compounded to a yearly basis. The IPCA consumer price index is used as a deflator. The oil price refers to the WTI quote in US dollars. The market index (of Bovespa) takes part of the same base of prices to form bit 22.

The parameters determining the size of the search space for the forecasting equation, a_{inf}, a_{sup}, b_{inf}, b_{sup}, are set in accordance to the sum of their related parameters in (8) and (9) as obtained from: $\left(a_{inf}, a_{sup}\right) = \mp \max\left|\hat{\alpha}_0 + \sum_{i=1}^{23} \hat{\alpha}_i b_{i,t-1}\right| = \mp 0.5231 \simeq (-0.55, 0.55)$ and $\left(b_{inf}, b_{sup}\right) = \mp \max\left|\hat{\beta}_0 + \sum_{i=1}^{23} \hat{\beta}_i b_{i,t-1}\right| = \mp 3.1392 \simeq (-3.15, 3.15)$ for the specification with drift, and $\left(b_{inf}, b_{sup}\right) = \mp \max\left|\hat{\beta}_0 + \sum_{i=1}^{23} \hat{\beta}_i b_{i,t-1}\right| = \mp 1.7711 \simeq (-1.8, 1.8)$ for the driftless one.

The remaining parameters N, k^*, e, and τ are calibrated with an eye on the convergence and predictive performance of the forecasts provided by the algorithm. Predictive performance is tracked by the mean squared prediction error, while convergence is captured by the correlation coefficient of the individual forecast series obtained from five repetitions of the algorithm. Parameters N and k^* showed no significant relationship with either predictive performance or convergence, a result possibly generated by the proper design of the termination criteria. We then set the lower values $N = 100$ and $k^* = 10\%$ to cut down the computational cost of running the algorithm. Setting parameter e to 50% helped find convergence. Parameter τ presented a negative relationship with both convergence and predictive performance, thus suggesting a trade-off between accuracy and confidence in the forecasts provided by the GBLCSA. Setting a unique value for τ would have masked this trade-off; we then considered three possible values: $\tau = \{0.3, 0.6, 0.9\}$.

This parameter setting along with the two forecasting specifications and the three proposed predictors for the GBLCSA rendered us with a total 18 distinct series of forecasts. The benchmark algorithms each provided two distinct series related to the drift and driftless specifications.

4 Results

Figures 2 and 3 show the evolution of forecast errors for each algorithm following the approach common in literature (see Enders, 2004, pp. 79-86). Here, we consider a mean squared prediction error (MSPE) ratio test as well as the Diebold and Mariano (DM) (1995) test. These are applied to the alternative specifications of the GBLCSA: with and without drift along with the three different predictors. Further, the algorithms (GBLCSA, RRA, and RWA) and the alternative specifications of the GBLCSA are individually evaluated by their predicted signal hit rate.

In the MSPE ratio test, an F-statistic is constructed to compare two competing forecasting models using the ratio of their MSPEs. Under the null hypothesis of equivalent forecasting performance,

$$F = \Sigma_{t=1}^{H}\left(r_t - \hat{r}_{1,t}\right)^2 / \Sigma_{t=1}^{H}\left(r_t - \hat{r}_{2,t}\right)^2, \tag{10}$$

has a standard F-distribution with (H, H) degrees of freedom under three assumptions: (1) the forecast errors are normally distributed with zero mean; (2) the forecast errors are serially uncorrelated; and (3) the forecast errors are contemporaneously uncorrelated with one another.

Because we needed to compare 22 series, the test was applied 231 times. Results are shown in table II. Panel A shows the comparisons of the three GBLCSA predictors through 18 pairwise tests. We find that predictor 2 and 3 performed better than predictor 1. Panel B shows the comparisons of the drift and driftless specifications through 11 pairwise tests. The driftless specifications generally performed better, especially for the RRA and the GBLCSA. Panel C shows the comparisons between the GBLCSA, the RRA, and the RWA through 28 pairwise tests. Evidence is mixed regarding the RRA as opposed to the RWA; the driftless specification of the RWA beats that of the RRA, whereas results are reversed for the specification with drift. The RWA outperforms the GBLCSA for the specifications with drift. However, the GBLCSA seems to outperform the RWA for the driftless specifications though this is not statistically robust. Overall, the RRA outperforms the GBLCSA. In the driftless specification, this happens in all the nine tests at the one percent significance level.

The Diebold-Mariano test relaxes some assumptions of the MSPE-ratio test by considering the mean differential loss from using the forecasts coming from two competing models, that is,

$$DM = \frac{1}{H}\sum_{t=1}^{H}\left[\left(r_t - \hat{r}_{1,t}\right)^2 - \left(r_t - \hat{r}_{2,t}\right)^2\right] / \sqrt{\hat{\sigma}^2_{NW}/(H-1)}, \tag{11}$$

where $\hat{\sigma}^2_{NW}$ is the Newey-West's variance estimator. Under the null of similar forecast accuracy, DM has a t-distribution with $H-1$ degrees of freedom. Considering the DM test allows one to assess whether the previous results are robust to possible violations in the assumptions of the MSPE ratio test. Table III shows that the results are overall the same. The RRA and the RWA outperform the GBLCSA, though with reduced statistical significance.

Considering the number of times one algorithm forecast changes direction (signal) in comparison with the changing signal of actual data allows us to further compare the predictive performance of the alternative algorithms. Dividing the number of time of changing signal by the total of forecasts leads to the hit rate measure, that is, the percentage of correctly predicted signals. Under the null of randomly generated signals, the number of signal matches has a binomial distribution $B(168, 0.5)$. Table IV shows the hit rates for each algorithm. Again, the RRA outperforms the others, especially in the specification with drift.

5 Conclusion

We consider a relatively standard genetic-based learning classifier system algorithm and make its key parameters endogenous. By making the probabilities of crossover, mutation, and selection pressure endogenous, we mitigate the problem of dependence of the results on the choice of initial parameters. This makes our algorithm a self-adapting solution technique at the individual level, also facilitating the convergence of the search process.

The forecasting performance of the improved algorithm is then contrasted with simpler ones: a random walk and recursive regressions. We consider stock market data, rather than data artificially generated. We find that the genetic-based classifier system fails to beat the simpler algorithms. This result is at odds with the majority of the literature, which favors the classifier systems but usually also considers most of its parameters as exogenous and neglect the need of convergence of the learning algorithm in order to prevent diffuse forecasts.

Exploring some alternative specifications our analysis also evidenced that wider predictors based on the population of rules into the classifier system outperform the narrower individual-based predictor which is commonly employed in the literature. The driftless specifications of the algorithms generally performed better, specially for those forecasting algorithms making use of market states information. However, evidence is mixed regarding the recursive regressions as opposed to the random walk model: the driftless specification of the recursive regressions beats that of the random walk, whereas results are reversed for the specification with a drift term. The random walk also outperforms the classifier system for the specifications with drift.

Overall, the recursive regressions outperforms the learning classifier system irrespective of the specification. These results are corroborated by two test statistics: the mean squared prediction error ratio test and the Diebold-Mariano test. Also, considering the percentage of correctly predicted signals we find that the recursive regressions outperform the other algorithms, especially in the specification with a drift. In short, our results cast doubts on the plausibility of using learning classifier systems to represent agents process of expectations formation. We emphasise, though, that the evidence presented is not incisive on this issue as it stands just as case against the use of genetic-based learning classifier systems as forecasting devices. Further research is needed for a comprehensive evaluation on this matter.

References

Allen, F. and R. Karjalainen (1999). Using genetic algorithms to find technical trading rules1. *Journal of Financial Economics 51*(2), 245–271.

Armano, G., M. Marchesi, and A. Murru (2002). Nxcs: Hybrid approach to stock indexes forecasting. In S. H. Chen (Ed.), *Genetic Algorithms and Genetic Programming in Computational Finance*, pp. 125–158. Boston, MA, USA: Springer.

Arthur, W. B., J. Holland, B. LeBaron, R. Palmer, and P. Tayler (1997). Asset pricing under endogenous expectations in an artificial stock market. In W. B. Arthur, S. Durlauf, and D. Lane (Eds.), *The Economy as an Evolving Complex System II*, pp. 15–44. Reading, MA, USA: Addison-Wesley.

Ashley, R. (2003). Statistically significant forecasting improvements: how much out-of-sample data is likely necessary? *International Journal of Forecasting 19*(2), 229–239.

Beltrametti, L., R. Fiorentini, L. Marengo, and R. Tamborini (1997). A learning-to-forecast experiment on the foreign exchange market with a classifier system. *Journal of Economic Dynamics and Control 21*(8-9), 1543–1575.

Beltratti, A. and S. Margarita (1993). Evolution of trading strategies among heterogeneous artificial economic agents. In *Proceedings of the second international conference on From animals to animats 2: simulation of adaptive behavior: simulation of adaptive behavior*, Cambridge, MA, USA, pp. 494–501. MIT Press.

Booker, L. B., D. E. Goldberg, and J. H. Holland (1989). Classifier systems and genetic algorithms. *Artificial Intelligence 40*, 235–282.

Branch, W. A. and G. W. Evans (2006). A simple recursive forecasting model. *Economics Letters 91*(2), 158–166.

Brenner, T. (2006). Agent learning representation: Advice on modelling economic learning. In L. Tesfatsion and K. L. Judd (Eds.), *Handbook of Computational Economics*, Volume 2 of *Handbook of Computational Economics*, Chapter 18, pp. 895–947. Elsevier.

Campbell, J. Y., A. W. Lo, and A. C. MacKinlay (1997). *The Econometrics of Financial Markets*. Princeton, NJ, USA: Princeton University Press.

Chen, M.-Y., K.-K. Chen, H.-K. Chiang, H.-S. Huang, and M.-J. Huang (2007). Comparing extended classifier system and genetic programming for financial forecasting: an empirical study. *Soft Computing 11*, 1173–1183.

Chen, S.-H. and C.-H. Yeh (2001). Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market. *Journal of Economic Dynamics and Control 25*(3-4), 363–393.

Dawid, H. (1999). *Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models* (2nd ed.). Srpinger-Verlag.

Diebold, F. X. and R. S. Mariano (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics 13*(3), 253–63.

Enders, W. (2004). *Applied Econometric Times Series* (2nd ed.). Wiley Series in Probability and Statistics. John Wiley & Sons.

Evans, G. W. and S. Honkapohja (2001). *Learning and expectations in macroeconomics*. Frontiers of Economic Research. Princeton, NJ: Princeton University Press.

Fama, E. F. (1991). Efficient capital markets: Ii. *Journal of Finance 46*(5), 1575–617.

Gibbs, M. S., G. C. Dandy, and H. R. Maier (2008). A genetic algorithm calibration method based on convergence due to genetic drift. *Information Sciences 178*, 2857–2869.

Herrera, F., M. Lozano, and A. M. Sanchez (2003). A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems 18*, 309–338.

Holland, J. H. (1962). Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery 9*, 297–314.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: University of Michigan Press.

Holland, J. H. (1986). Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel rule-based systems. In R. Michalski, J. Carbonel, and T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Volume 2. San Mateo, CA, USA: Morgan Kaufmann.

Holland, J. H., K. J. Holyoak, R. E. Nisbett, and P. R. Thagard (1986). *Induction: Processes of Inference, Learning, and Discovery*. Cambridge, MA, USA: The MIT Press.

LeBaron, B. (2006). Agent-based computational finance. In L. Tesfatsion and K. L. Judd (Eds.), *Handbook of Computational Economics*, Volume 2 of *Handbook of Computational Economics*, Chapter 24, pp. 1187–1233. Elsevier.

LeBaron, B., W. B. Arthur, and R. Palmer (1999). Time series properties of an artificial stock market. *Journal of Economic Dynamics and Control 23*(9-10), 1487–1516.

Lettau, M. (1997). Explaining the facts with adaptive agents: The case of mutual fund flows. *Journal of Economic Dynamics and Control 21*(7), 1117–1147.

Liao, P. and J. Chen (2001). Dynamic trading strategy learning model using learning classifier systems. In *Proceedings of the 2001 Congress on Evolutionary Computation*, Volume 2, pp. 783–789.

Mahfoud, S. and G. Mani (1996). Financial forecasting using genetic algorithms. *Applied Artificial Intelligence 10*(6), 543–565.

Palmer, R. G., W. B. Arthur, J. H. Holland, and B. LeBaron (1999). An artificial stock market. *Artificial Life and Robotics 3*(1), 27–31.

San Jose-Revuelta, L. M. (2007). A new adaptive genetic algorithm for fixed channel assignment. *Information Sciences 177*, 2655–2678.

Schulenburg, S. and P. Ross (2002). Explorations in lcs models of stock trading. In P. L. Lanzi, W. Stolzmann, and S. W. Wilson (Eds.), *Advances in Learning Classifier Systems. Proceedings of the Fourth International Workshop*, Volume 2321 of *Lecture Notes in Artificial Intelligence*, pp. 151–180. New York, NY, USA: Springer.

Stone, C. and L. Bull (2008). Foreign exchange trading using a learning classifier system. In L. Bull, E. Bernada-Mansilla, and J. Holmes (Eds.), *Learning Classifier Systems in Data Mining*, Volume 125 of *Studies in Computational Intelligence*, pp. 169–189. Springer.

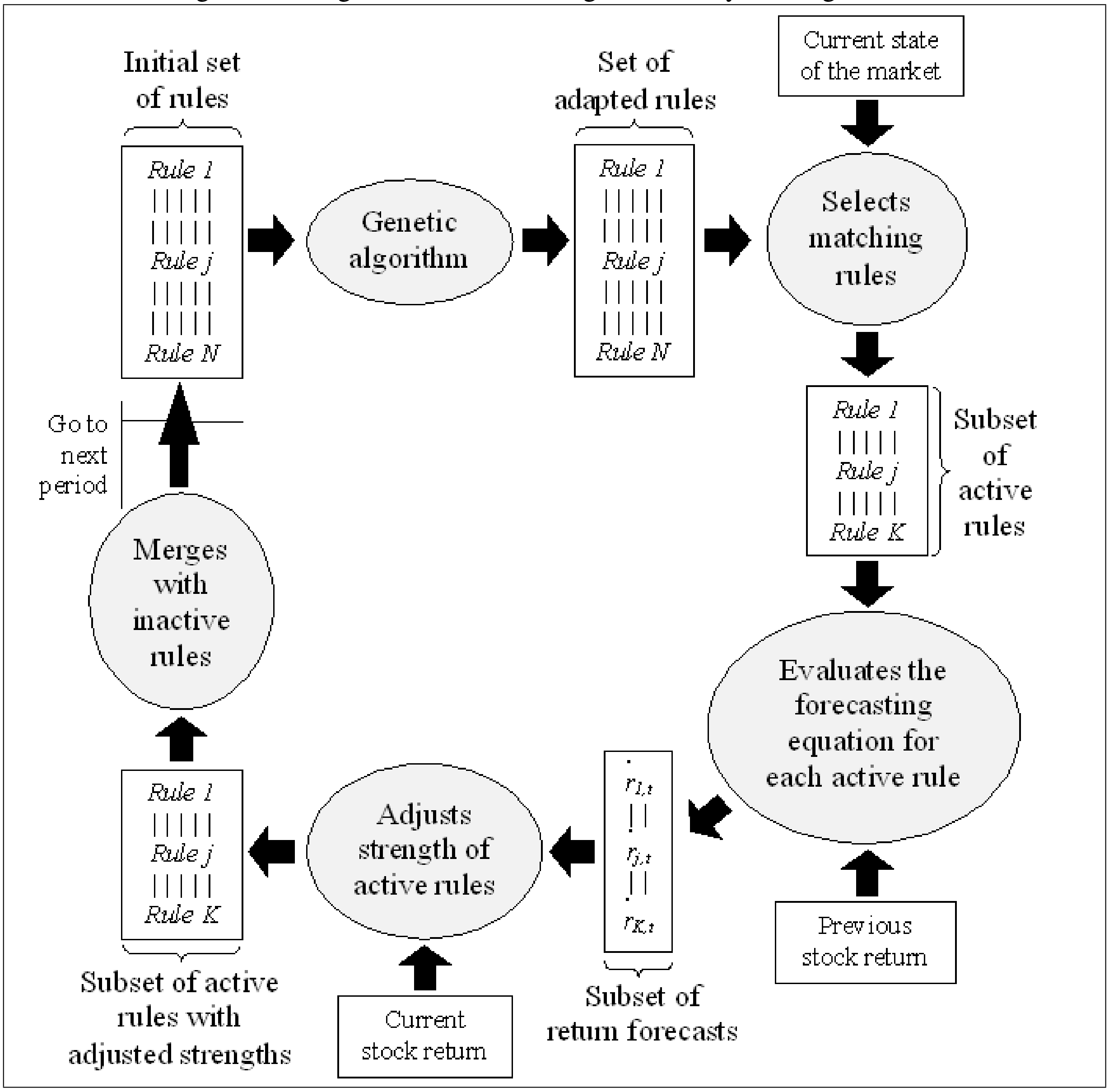
Tay, N. S. P. and S. C. Linn (2001). Fuzzy inductive reasoning, expectation formation and the behavior of security prices. *Journal of Economic Dynamics and Control 25*(3-4), 321–361.

Vriend, N. J. (2000). An illustration of the essential difference between individual and social learning, and its consequences for computational analyses. *Journal of Economic Dynamics and Control 24*(1), 1–19.

Wan, H. A., A. Hunter, and P. Dunne (2002). Autonomous agent models of stock markets. *Artificial Intelligence Review 17*, 87–128.

A Supplementary Material (by order of appearance into the main text)

Figure 1: The genetic-based learning classifier system algorithm.



Algorithm 1 The genetic algorithm routine.

1. From the starting subset of active rules (k-rules), elitism sets in to directly replicate a maximum of $E = eNk^*$ active rules into the next generation of rules, favoring stronger rules if $k > E$, where $e \in [0,1]$ is the elitism parameter.

2. $N - E$ offspring rules are then generated by the following steps:

   (a) Two parents are selected by an S-sized tournament selection from the set of initial rules.

   (b) Bit-strings for each of the two offspring are generated by applying a bit-to-bit crossover or by cloning the two-parent bit strings.

   (c) A bit-to-bit mutation operation is then applied.

   (d) The forecasting parameters of the offspring are obtained by either crossing-over or cloning the forecasting parameters of the parents.

   (e) A mutation operation is then executed on the resulting forecasting parameters.

   (f) The offspring strengths are set by mixing their parent variances v_1^2 and v_2^2 with a weighted average; here, the weights are determined by the absolute difference between the predicted returns of the offspring rule and the predicted returns of the parents rule, that is,

   $$v_o^2 = \left[\left|\hat{r}_{o,t} - \hat{r}_{2,t}\right| / \left(\left|\hat{r}_{o,t} - \hat{r}_{1,t}\right| + \left|\hat{r}_{o,t} - \hat{r}_{2,t}\right| \right) \right] v_1^2 + \left[\left|\hat{r}_{o,t} - \hat{r}_{1,t}\right| / \left(\left|\hat{r}_{o,t} - \hat{r}_{1,t}\right| + \left|\hat{r}_{o,t} - \hat{r}_{2,t}\right| \right) \right] v_2^2,$$

   where v_o^2 is the variance measure of the offspring rule and $\hat{r}_{o,t}$, $\hat{r}_{1,t}$, and $\hat{r}_{2,t}$ are the predicted returns of the offspring, the first parent, and the second parent rules, respectively.

   (g) Finally, the strengths of the offspring are adjusted according to its performance (using equation (3)). A window of past data with size determined by 24τ is considered, which imposes a maximum of 24 previous periods for evaluation and turns τ into a measure of memory size. The resulting variance is then averaged with that obtained from the previous step.

3. The $N - E$ offspring rules are merged with the subset of E active rules found in step 1.

4. The resulting population of rules is finally considered as a new initial set of rules and input in step 1; the routine goes on until three termination criteria are jointly met:

   (a) The proportion of active rules k over the classifier system size N is greater than or equal to a k^* minimum proportion of active rules;

   (b) The last three generations of forecasting rules lead to return forecasts with the same signal for each predictor;

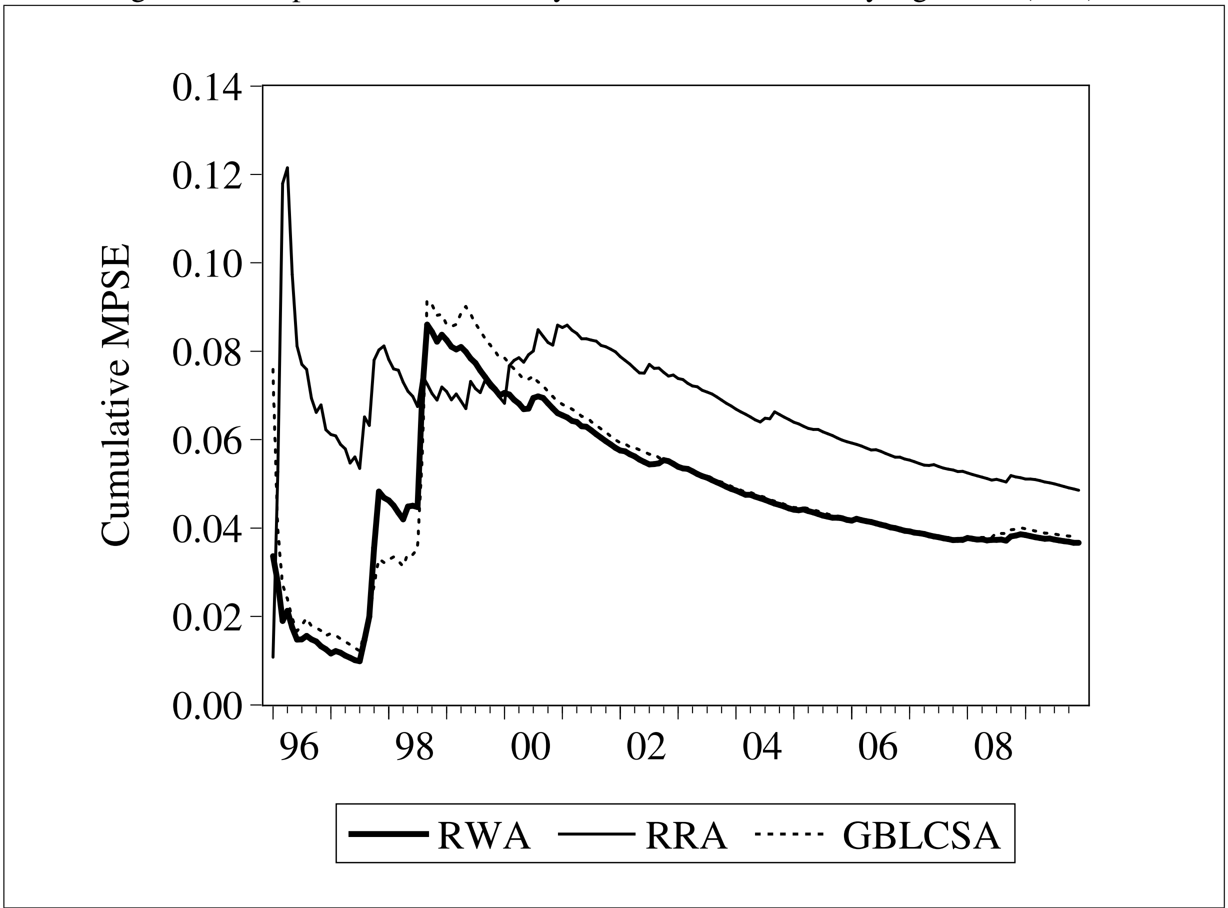
   (c) The replications are restricted to a maximum of 100 loops.

Table I: Condition bits.

Bit	Type	Condition	Bit	Type	Condition
1	Technical	$P_t > P_{t-1}$	12	Fundamental	$(P_t/D_t)\, r_t > 1$
2	Technical	$P_t > MA\,(P,6)$	13	Tech./Fund.	$(P_t/D_t) > (P_{t-1}/D_{t-1})$
3	Technical	$P_t > MA\,(P,12)$	14	Tech./Fund.	$(P_t/D_t) > MA\,(P/D,6)$
4	Fundamental	$(P_t/E_t)\, r_t > 1$	15	Tech./Fund.	$(P_t/D_t) > MA\,(P/D,12)$
5	Tech./Fund.	$(P_t/E_t) > (P_{t-1}/E_{t-1})$	16	Tech./Fund.	$x_t > x_{t-1}$
6	Tech./Fund.	$(P_t/E_t) > MA\,(P/E,6)$	17	Tech./Fund.	$x_t > MA\,(x,6)$
7	Tech./Fund.	$(P_t/E_t) > MA\,(P/E,12)$	18	Tech./Fund.	$x_t > MA\,(x,12)$
8	Fundamental	$(P_t/B_t) > 1$	19	Tech./Fund.	$o_t > o_{t-1}$
9	Tech./Fund.	$(P_t/B_t) > (P_{t-1}/B_{t-1})$	20	Tech./Fund.	$o_t > MA\,(o,6)$
10	Tech./Fund.	$(P_t/B_t) > MA\,(P/B,6)$	21	Tech./Fund.	$o_t > MA\,(o,12)$
11	Tech./Fund.	$(P_t/B_t) > MA\,(P/B,12)$	22	Fundamental	$P_t/I_t > 1$
			23	Fundamental	$t >$July 1994

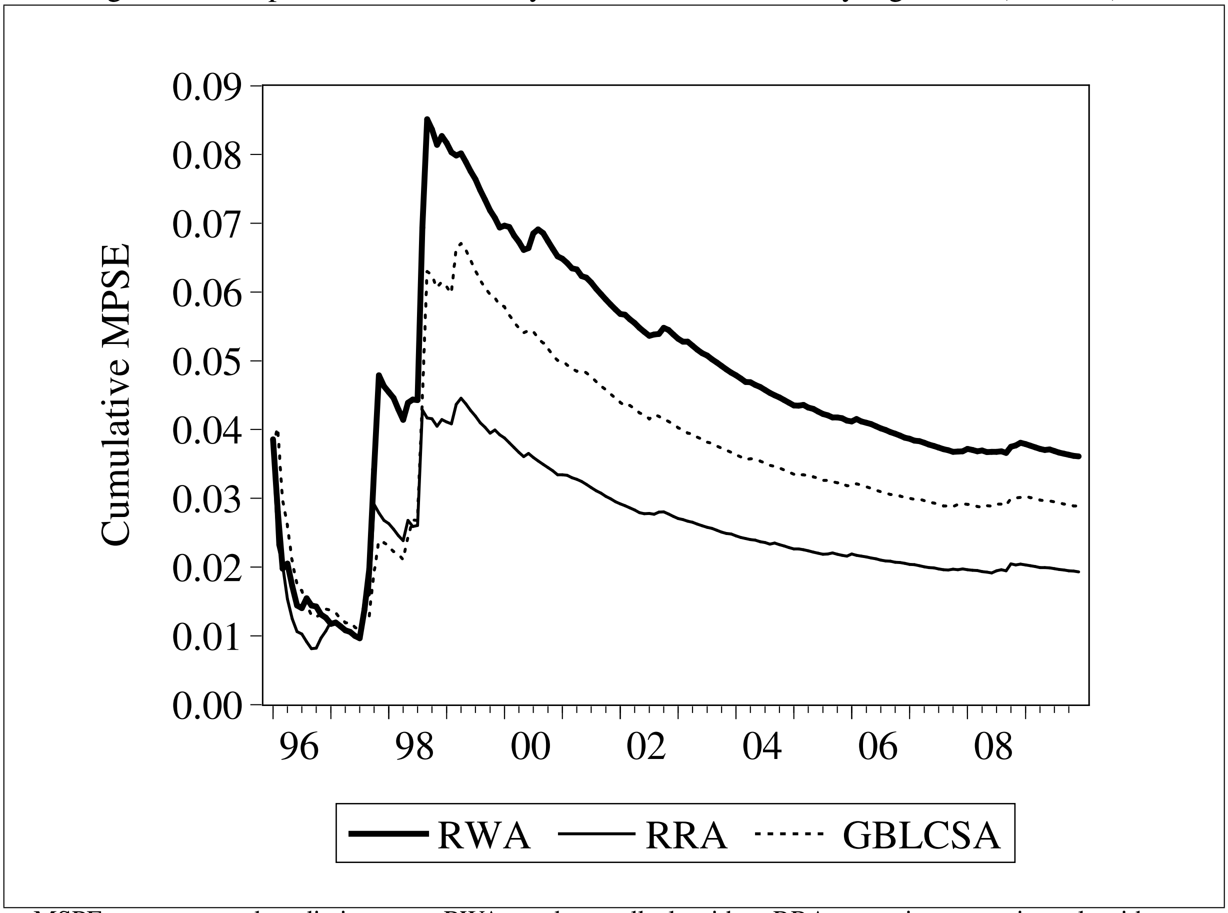
P_t : stock price; r_t : interest rate; E_t : corporate earnings per share; B_t : corporate accounting patrimonial value per share; D_t : corporate dividends per share; x_t : exchange rate; o_t : oil price; I_t : stock market index; $MA\,(\bullet, m)$: moving average over the past m months.

Figure 2: Comparison of recursively accumulated MSPEs by algorithm (drift).



MSPE: mean squared prediction error; RWA: random walk algorithm; RRA: recursive regressions algorithm; GBLCSA: Genetic-based learning classifier system algorithm; the GBLCSA series of accumulated MSPE refers to that obtained from the second type of predictor (arithmetic mean) and with a τ value of 0.9.

Figure 3: Comparison of recursively accumulated MSPEs by algorithm (driftless).



MSPE: mean squared prediction error; RWA: random walk algorithm; RRA: recursive regressions algorithm; GBLCSA: Genetic-based learning classifier system algorithm; the GBLCSA series of accumulated MSPE refers to that obtained from the second type of predictor (arithmetic mean) and with a τ value of 0.9.

Table II: Summary of results obtained with the MSPE ratio test.

Testing pair	No. tests	Mean MSPE ratios	Number of times that the 1st beats the 2nd: Total	Significant at 10%	5%	1%	Number of times that the 2nd beats the 1st: Total	Significant at 10%	5%	1%
Panel A: GBLCSA Predictors Compared										
Predictor 1 vs Predictor 2	6	1.3781	0	0	0	0	6	5	4	3
Predictor 1 vs Predictor 3	6	1.2009	0	0	0	0	6	3	1	0
Predictor 2 vs Predictor 3	6	0.8734	6	0	0	0	0	0	0	0
Panel B: Driftless vs Drift Specifications										
Overall	11	0.7354	11	8	6	4	0	0	0	0
RWA driftless vs drift	1	0.9853	1	0	0	0	0	0	0	0
RRA driftless vs drift	1	0.3974	1	1	1	1	0	0	0	0
GBLCSA driftless vs drift	9	0.7452	9	7	5	3	0	0	0	0
Panel C: Algorithms Compared										
RWA vs RRA	2	1.3124	1	1	1	0	1	1	1	1
Driftless comparison	1	1.8704	0	0	0	0	1	1	1	1
Drift comparison	1	0.7544	1	1	1	0	0	0	0	0
RWA vs GBLCSA	18	0.8874	12	8	7	3	6	1	0	0
Driftless comparison	9	1.0153	3	2	1	0	6	1	0	0
Drift comparison	9	0.7596	9	6	6	3	0	0	0	0
RRA vs GBLCSA	18	0.7748	15	10	9	9	3	1	0	0
Driftless comparison	9	0.5428	9	9	9	9	0	0	0	0
Drift comparison	9	1.0068	6	1	0	0	3	1	0	0

MSPE: mean squared prediction error; GBLCSA: Genetic-based learning classifier system algorithm; Predictor 1: forecast obtained from the active rule with greater strength; Predictor 2: forecast obtained as a simple arithmetic mean of the active rules; Predictor 3: forecast obtained as a strength-weighted average of the active rules; RWA: random walk algorithm; RRA: recursive regressions algorithm.

Table III: Summary of results obtained with the Diebold-Mariano test.

Testing pair	No. tests	Mean *DM*	Number of times that the 1st beats the 2nd				Number of times that the 2nd beats the 1st			
			Total	Significant at			Total	Significant at		
				10%	5%	1%		10%	5%	1%
Panel A: GBLCSA Predictors Compared										
Predictor 1 vs Predictor 2	6	0.0130	0	0	0	0	6	6	6	5
Predictor 1 vs Predictor 3	6	0.0078	0	0	0	0	6	6	6	4
Predictor 2 vs Predictor 3	6	-0.0052	6	6	6	6	0	0	0	0
Panel B: Driftless vs Drift Specifications										
Overall	11	-0.0130	11	11	9	1	0	0	0	0
RWA driftless vs drift	1	-0.0005	1	1	1	0	0	0	0	0
RRA driftless vs drift	1	-0.0293	1	1	1	1	0	0	0	0
GBLCSA driftless vs drift	9	-0.0125	9	9	7	0	0	0	0	0
Panel C: Algorithms Compared										
RWA vs RRA	2	0.0024	1	0	0	0	1	1	1	1
Driftless comparison	1	0.0168	0	0	0	0	1	1	1	1
Drift comparison	1	-0.0119	1	0	0	0	0	0	0	0
RWA vs GBLCSA	18	-0.0065	12	9	6	3	6	2	1	0
Driftless comparison	9	-0.0005	3	2	1	0	6	2	1	0
Drift comparison	9	-0.0125	9	7	5	3	0	0	0	0
RRA vs GBLCSA	18	-0.0090	15	9	9	8	3	0	0	0
Driftless comparison	9	-0.0173	9	9	9	8	0	0	0	0
Drift comparison	9	-0.0006	6	0	0	0	3	0	0	0

DM: Diebold-Mariano test statistic; GBLCSA: Genetic-based learning classifier system algorithm; Predictor 1: forecast obtained from the active rule with greater strength; Predictor 2: forecast obtained as a simple arithmetic mean of the active rules; Predictor 3: forecast obtained as a strength-weighted average of the active rules; RWA: random walk algorithm; RRA: recursive regressions algorithm.

Table IV: Hit rates by algorithm.

Algorithms	Drift specifications		Driftless specifications	
	Hit-rate (%)	p-value	Hit-rate (%)	p-value
RWA	53.57	0.1579	47.62	0.7054
RRA	58.93	0.0083	56.55	0.0378
GBLCSA	50.13	0.4693	50.06	0.4693

RWA: random walk algorithm; RRA: recursive regressions algorithm; GBLCSA: Genetic-based learning classifier system algorithm; p-value: probability of obtaining a greater hit rate according to a binomial distribution $B(168, 0.5)$; the hit-rates for the GBLCSA refer to averages within the different specifications for this algorithms.