Applied Soft Computing Journal 90 (2020) 106169

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Strongly-typed genetic programming and fuzzy inference system: An embedded approach to model and generate trading rules

Kevin Michell, Werner Kristjanpoller *

Departamento de Industrias, Universidad Técnica Federico Santa Maria, Av. España 1680, Valparaíso, Chile

ARTICLE INFO

Article history:
Received 29 May 2018
Received in revised form 30 December 2019
Accepted 12 February 2020
Available online 18 February 2020

Keywords:
Strongly typed genetic programming
Stock market predictions
Recommendation system
Trading rules
Fuzzy inference system

ABSTRACT

Generating trading signals is an interesting topic and a hard problem to solve. This work uses fuzzy inference system (FIS) and strongly typed genetic programming (STGP) to generate trading rules for the US stock market, a framework that we call FISTGP. The two embedded models have not been widely evaluated in financial applications, and according to the literature, their combination could improve forecasting performance. The fitness function used to train the STGP model is based on accuracy, optimizing the buy and sell signals, taking a different approach to the classic optimization of return–risk ratio. The rules are generated in a FIS framework, and the final signal depends on the amount of information that the investor relies on. The model is suited to each investor as a recommendation of when to change portfolio composition according to his or her particular criteria. Ternary rules are generated based on an economic interpretation, considering the risk-free rate as a part of more demanding rules. The model is applied to 90 of the most traded and active stocks in the US stock market. This approach generates important recommendations and delivers useful information to investors. The results show that the proposed model outperforms the Buy and Hold (B&H) strategy by 28.62% in the test period, considering excesses of return, with almost the same risk (1.28% higher). The other base models underperform in comparison to the B&H, with the proposed model also outperforming them.

1. Introduction

Establishing a framework for trading decisions is a well-known problem, which is addressed by many different techniques, from Markowitz [1] mean–variance approach to machine learning. Especially in the latter, there are a variety of algorithms that serve as decision support systems for investors and companies, which can produce useful information.

Formulating the trading problem is a difficult task. Technical indexes, such as moving averages (MA), relative strength index (RSI), rate of change (ROC), arms index (also known as TRIN, an acronym for TRading INdex), among others, help the investor to make decisions regarding portfolio composition. The combination of these technical indexes can actually be used as inputs to intelligent systems, which are able to detect important relationships over time [2,3].

Several studies show that machine learning can outperform the approaches of Markowitz [4–8]. The most used approach to determine the optimal configuration of a portfolio is to calculate stock weights [7,9]. Another approach is to develop a recommendation system based on the identification of trading rules, providing useful information to the investor of when to buy or to sell the stocks. In this regard, evolutionary algorithms (EA), such as genetic algorithms (GA; [10]) and genetic programming (GP; [11]), have been widely used because they are weak algorithms with respect to the information they need to solve general problems [8,12].

This paper proposes the combination of fuzzy inference systems and strongly typed genetic programming as a recommendation system framework to generate ternary trading rules, which we call FISTGP. The contribution of this framework to the literature is based on the following aspects: (1) The fitness function used to train the STGP model is based on optimizing signal accuracy instead of the optimization of the weights of stocks in the portfolio, where the signals are actually calculated in a FIS framework, (2) An exploratory analysis is performed, considering the investor's risk profile. This analysis is done using the STGP with FIS root node framework configuration, (3) the inclusion of risk-free rate (r_f) to generate the ternary rules (explained in detail in Section 3.2.5). Under the mentioned contributions, the proposed FISTGP framework is able to outperform the Buy and Hold (B&H) approach in excess return, two different markets indexes, as well as the FISTGP with Conditional Sharpe Ratio and PG using accuracy and Conditional Sharpe Ratio.

* Corresponding author.
E-mail address: werner.kristjanpoller@usm.cl (W. Kristjanpoller).

https://doi.org/10.1016/j.asoc.2020.106169

The major contribution of this study is the use of a fitness function focused on the optimization of signal accuracy as opposed to the optimization of return-on-risk ratios. The main motivation for this approach is based on the fact that it is possible to generate portfolio optimizations that imply greater benefit for the investor using rules that determine when it is convenient to invest, rather than the specific weight or price forecast. This apparent simplification of the objective function allows us to reduce the computation times, and as this study shows, it even generates better returns than the optimization of return/risk ratios. Thus, the main contribution is the practical implementation of a fitness function that has another approach to solving the problem, by treating the optimization of signals as the main objective in the identification of rules that allow us to generate excess returns, without affecting the level of risk. This novelty is expressed in the fact that the problem is simplified through simple ternary rules, where the system is forced to generate rules that can at least yield the same as the market. Empirically, in this study, it is shown that this approach does not harm the portfolio performance of the analyzed stocks, outperforming its counterparts that optimize return on risk (e.g. Conditional Sharpe ratio)

The remainder of this paper is organized as follows. In Section 3 we explain the proposed methodology. In Section 4, we explain the data used and the results obtained. Finally, in Section 5, we discuss the results and analyze their implications.

2. Related work

One of the first studies that employed GP to identify trading rules was Allen & Karjalainen [13]. They used standard GP to generate trading rules for the S&P500 index. Although GP could generate greater returns, it was not able to outperform the B&H strategy in the same period. Chen et al. [14] use GP along with a dynamic proportion portfolio insurance (DPPI) strategy as a risk multiplier. This changes the risk component of the market according to its actual change. The risk variables identified by the principal component analysis (PCA) are used to construct the trees using GP. Fitness is calculated as $f = \frac{r+1}{\sigma+1}$ where r is the return with transaction cost and σ is the standard deviation. They selected five stocks from the New York Stock Exchange (NYSE), and the data was divided into six different sets. The experiment shows that the use of DPPI generates better performance, and they identify which risk variables appear more often than others. Potvin, Soriano & Vallée [15] use GP to construct trading rules with an approach consisting of constructing the rules for each stock individually. The fitness is calculated as the difference between GP and Buy&Hold return. The data consisted of 14 Canadian companies and the periods divided into short and long term, with two sub-periods for each one. The empirical results indicate that, taking price and transaction volume data of the Toronto stock market to model, GP can outperform classical strategies. Ha & Moon [16] proposed GP to identify patterns in the cryptocurrency market. They focus the study on applying the signals in the market to increase investment return. Based on several experiments, they show that GP can create useful rules for trading, performing well in the test set (see Table 1).

There are many variations of the original GP algorithm of Koza, such as genetic network programming (GNP; [22]), multitree genetic programming (MGP; [23]) and strongly typed genetic programming (STGP; [24]). Each of these techniques allows the development of a different approach in the portfolio allocation problem [25]. Mabu et al. [26] used an extended GNP with rule accumulation (GNP-RA); this is, in each generation of the training period, the algorithm generates several rules, which are stored in the rule pool, making decisions based on them. Additionally, the authors enhanced the approach by adding up and down trends as well as the buying and selling frequency. Furthermore, they used the Sarsa [27] learning reinforcement (LR) in the proposed model (GNP-RA), in GNP Sarsa LR (GNP-SARSA), GNP-RA Sarsa (GNP-RA Sarsa) and Buy&Hold strategy. All these models were applied to test their profit performance. The simulations showed that the proposed approach effectively generates higher profits than conventional GNP and models using the Sarsa method. Dehghanpour & Esfahanipour [28] used a robust genetic programming model, in a constant proportion portfolio insurance approach, to determine where and how much to invest in risky and risk-free assets. They applied the model to 5 NYSE stocks, and managed to outperform the proposed benchmarks.

Of particular interest to this study is the application of STGP to the portfolio selection problem. Berutich et al. [17] used STGP with random sampling method (RSFGP) to discover trading rules to determine buy and sell signals in 21 stocks of the spanish market. STGP adds additional restrictions in which the sparse trees are constructed, generating programs in a specific way. RSFGP calculates fitness in randomly selected segments, instead of as the whole data set. The results indicate that this approach outperforms Buy&Hold, standard GP and volatility adjusted fitness GP. Furthermore, the approach is able to automatically manage a stock portfolio without a financial market expert. Manahov, Hudson & Linsey [18] used a special adaptive form of STGP learning algorithm to generate trading rules for different agents, evaluating the amount of wealth that each agent obtained over time. Data for their model evaluation was taken from Russel 1000, 2000 and 3000 indexes. The study generated profit compared with the random walk (RW; [29]), both in-sample and out-of-sample. Additionally, volume had no predictive power, implying that it is not a variable that can help develop trading rules. Moreover, the results were statistically significant and the predictive accuracy was better than the RW case (over 50%). Manahov [19] studied the effect of high frequency data with a STGP approach, to see if it is beneficial or not to the market, finding that it is actually harmful to institutional traders, on a millisecond basis.

Recommendation systems rely on algorithms that can process large quantities of information to make more accurate decisions. In selection, generally the algorithms present three options for the stocks, i.e., buy (1), sell (−1) and no trade (0), in a ternary rule system. However, this leaves no option for the investor when it is actually best for him or her to change the portfolio composition based on his or her particular criteria. Thus, the same recommendation could be taken or not depending on the information that the investor has, and how it is processed. Fuzzy logic (FL) [30], and specifically, one of its most used applications, fuzzy inference system (FIS), can be used to deal with this problem and to generate more realistic signals, providing possibilities of when a stock should be bought, sold or not traded instead of absolute signals, allowing the investor to choose when it is best for him or her to take into account the recommended signal.

The easy interpretation of FIS architectures makes them an interesting tool to identify rules for trading. Dourra & Siy [31] propose a FIS that can map the relation of different technical indicators to evaluate stock prices, following the practice of market experts. Fasanghari & Montazer [32] develop a FIS for selecting superior stocks to encounter the uncertainty of stock in the portfolio and to create a solid rule generator model. Yunusoglu & Selim [33] propose a complete framework that takes investor preference and thresholds to select stocks, then ranking them to finally construct the portfolio, in a realistic and flexible way.

3. Methodology

The proposed framework consist of three phases: (1) preprocessing the data; (2) the embedded model; and, (3) the postprocessing of signals.

Table 1
Past studies on GP, STGP, and STGP-FIS.

Study	Data[a]	Training metric	Period	Trading system
Allen & Karjalainen, [13]	(1) S&P500 index	Excess return over B&H	1928–1995 daily	Crips
Chen et al. [14]	(5) NYSE stocks	Return over risk	2001–2004 daily	Crips
Potvin, Soriano & Vallée [15]	(14) Canadian companies	Excess return over B&H	1992–2000 daily	Crips
Ha & Moon [16]	(63) Cryptocurrency	Attractiveness of patterns	2016 every 30 min	Crips
Berutich et al. [17]	(21) Spanish market	Sum sterling ratio	2000–2013 daily	Crips
Manahov, Hudson & Linsey [18]	(3) Russell	Sterling ratio	1979–2012 daily	Crips
Manahov [19]	(1) High frequency S&P GSCI	Real - Expected	2014 milliseconds	Crips
Mousavi, Esfahanipour & Zarandi [20]	(15) TSE, Toronto and Frankfurt stock exchanges	Conditional sharpe ratio	2010–2014 daily	Fuzzy
Bahar, Zarandi & Esfahanipour [21]	(10) Tehran Stock Exchanges	Total profit	2015–2016 daily	Fuzzy

[a]Number indicates the number of stock or indexes used in the study.

1. First phase (pre-processing the data): The inputs of the algorithm are normalized. This is an important process because it helps the algorithm to perform better, considering that price and transaction volume have very different orders of magnitude. This will be explained in detail in Section 3.1
2. Second phase (proposed embedded model): The normalized data are the inputs in the embedded model, which consists of an STGP structure with a FIS node as the root. The model consists of an STGP algorithm, where its root node is an FIS algorithm, which generates signals as possibilities instead of as probability. This will be explained in detail in Section 3.2
3. Third phase (post-processing of signals): The signals, previously transformed into recommendations depending on the risk behavior of the investor, are related to the corresponding return, and the final performance is obtained. This will be explained in detail in Section 3.3

These three phases work in a rolling windows approach that can be summarized as

- Take a fixed time window of f days to adjust the model, and then forecast h days ahead. i.e., take day 1 to day f for training and then forecast the days $f+1$ to $f+h$ for the first window.
- Move h days forward and repeat step 1 (i.e., for the second window, the data for training would be from day $h+1$ to $f+h$ in order to forecast days $f+h+1$ to $f+2h$ and so on).
- Repeat this until the entire period has been covered.

For this study, h will take the values of 10 (two weeks), 22 (a month), 33 (month and a half), 66 (3 months) and 126 (half a year); and, f will take values of 33, 66, 126 and 252 (a year), as they resemble relevant periods in financial markets.

The particular configuration of the STGP, FIS and ternary rules are covered in detail on the following subsections. For reference of these three phases see Fig. 1 in which the whole process is presented.

3.1. First phase (pre-processing of the data)

We selected the most traded and active stocks in the US stock market, leaving 90 possible assets to conform the portfolio. The test period starts on January 08, 2003 and ends November 30, 2015. This leaves a test period of 3246 days in total. Characteristics of the stocks selected to test the models are in Appendix, in Tables 10–12. The proposed framework was applied over a rolling windows approach, as mentioned early. Transaction cost was set to 0.1% for buying and selling signals as seen in the literature [17,18,34]. Once the data is obtained, these are normalized, and thus have no problems with the internal work of the algorithm. This is important, because there is a huge difference between the orders of magnitude in the input data (price and transaction volume).

3.2. Second phase (proposed embedded model)

With the data obtained and normalized, we proceed to build and optimize the model.

3.2.1. Genetic programming (GP)

As described before, GP is able to solve highly non-linear problems, allowing the exploration of the entire non-differential space, making it interesting to be used in the stock market environment, especially in the generation of trading rules [35]. In general, GP is an evolutionary optimizer based on GA, but they differ mainly in solution representation; in the latter, chromosomes solve a pre-defined problem, and in the former, they first make decision trees that solve an undefined problem given input data and functions.

GP generates programs that are encoded as parse trees [36], and they are created by the following general pseudo code:

1. Initiate the population randomly (generation 0).
2. Evaluate the fitness of each program.
3. Until a program can successfully model the problem or another stopping criteria is met, do
   (a) Advance to next generation.
   (b) Apply genetic operators over the population of previous generation (selection, crossover and mutation).
   (c) Create the new population.
   (d) Run every program and obtain its fitness value.
4. Return the best individual so far.

3.2.2. Strongly-typed genetic programming (STGP)

STGP is a variation of standard GP. The main change is related to how the parse trees are conformed, because in STGP, there is a requirement that functions and terminals must have their data type specified in advance; thus, the population is only constructed by syntactically correct parse trees, reducing the search space considerably [37]. Compared to the complete search space of standard GP, which can have orders of 10^{30} to 10^{40} [24],

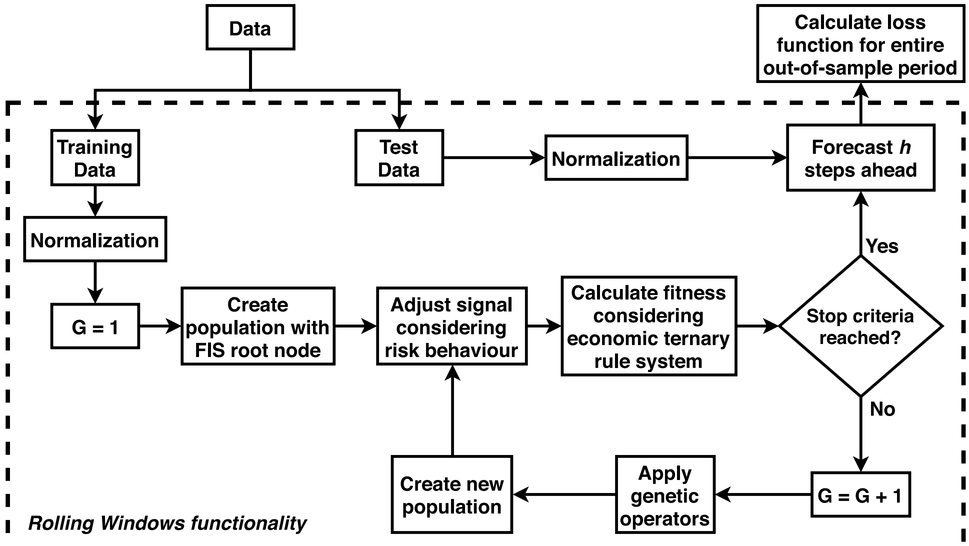

Fig. 1. Scheme of proposed framework. The first phase consists of the data, training data, test data, and normalization boxes. The second phase consists of all the boxes inside the dashed box (rolling window functionality), not considering the ones in the first phase. The third phase is related to the right-upper box (calculate loss function for entire out-of-sample period).

Table 2
Terminals and function of FISTGP.

Terminals[a]	Price and Transaction Volume of each stock to trade
Function[b]	Arithmetic operators ($+$, $-$, $\times$, $/$), Boolean operators (AND, OR, NOT), Relational operators ($<$, $>$), Maximum, Minimum and Square function
Root Node	FIS decision node

[a]Terminals refer to the input data for the STGP algorithm.
[b]Functions refer to the operations that are presumed to be sufficient to model the problem.

the STGP is much more parsimonious. Specifically, in an STGP environment, root nodes must return a value type that is required by the problem, and each non-root node must give a value type required by the parent. These two specifications make it possible to reduce the search space considerably.

3.2.3. Terminal and function set

The tree composition is based on terminals and functions. Table 2 shows terminals, functions and the specific root node.

Due to the STGP restriction of each node, the general form of the coded trees is sequential, with maximum, minimum, square and arithmetic operators in the lower part, relational operators in the middle and then boolean operators on top, following the requirement of the root node (boolean) and the chosen terminal set.

3.2.4. Genetic operators

Selection, crossover and mutation genetic operators are used to evolve the population. They work similarly as in the GP case, but considering the two additional restrictions that STGP imposes, explained in 3.2.2. The characteristic of this and the other configurations of the STGP algorithm are summarized in Table 3. Here, the population size and maximum generation have low values because, after several iterations, with higher values, the algorithm does not improve and takes considerably more time. This is based on the theoretical analysis of Chen et al. [38], where it is clearly stated that large population sizes are not helpful to evolutionary techniques. This is also supported empirically by Ashlock [39], where he demonstrated that the use of small population sizes achieves better performances in a variety of

Table 3
Parameters and settings of proposed STGP.

Parameters	Value
Total population size[a]	60
Maximum generation[a]	40
Maximum genome size	2048
Maximum genome depth	10
Initial selection type	Ramped half & half
Crossover probability	0.7
Mutation probability	0.2
Replacement[b]	Yes

[a]Small numbers to improve processing time, without clearly affecting the fitness of the proposed model [38,39].
[b]Offspring will replace worst performing agents.

problems. The crossover and mutation probabilities were also tested, given 0.7 and 0.2 the better performance respectively based on experimental results.

Crossover and mutation operators allow the proposed model to explore the search space in greater depth. In particular, for this problem, this space is quite large, and although the use of STGP reduces the search only to valid programs, the combinations are high. Therefore, the genetic operators are fundamental to, first, expand the obtaining of better rules and second, to allow us to find the global optimum in each window, which must be optimized each time.

3.2.5. Risk free in the portfolio

Before considering the Risk Free and its importance for the ternary rules, we first define the return of an asset i in time t, $r_{i,t}$ as

$$r_{i,t} = \ln\left(\frac{P_{i,t}}{P_{i,t-1}}\right) \quad \forall t \in T,\ \forall i \in N$$

where $P_{i,t}$ is the price of the asset i in time t. Based on this, we propose the use of ternary rules, which are learned based on an ideal target of investment in a stock when its return is greater than the risk free return, r_f, selling when its negative, and in any other case make no trade. Mathematically, this can be expressed as

$$Target_{i,t} = \begin{cases} 1 & \text{if } r_{i,t} > r_{f_t} \quad \forall t \in T,\ \forall i \in N, \\ -1 & \text{if } r_{i,t} < 0 \quad \forall t \in T,\ \forall i \in N, \\ 0 & 0 \le r_{i,t} \le r_{f_t} \quad \forall t \in T,\ \forall i \in N, \end{cases} \tag{1}$$

where T is the period length, N is the total stocks and r_{f_t} is the risk free in time t. This approach entails an important implication: the requirement to generate the buying signal is higher. This means that the model must consider returns that are greater than r_f, instead of just considering the returns that are positive. This ternary relation has a clear economic interpretation in which the ideal objective is to generate rules that have greater returns than a r_f base. Notice that a 0 signals means that if the stock was in a buying state, it stays in that state, and if the stock was not in a buying state, it is recommended to invest in r_f.

That last analysis say that, when a particular stock is not in a buying state, r_f is considered in its place. This has two important implications. First, the algorithms must learn to identify excess return against a more demanding natural benchmark instead of just obtaining a positive return. Second, the problem becomes a classification problem, because the portfolio always has the N slot active, having in some cases the stock return and in other cases the r_f. Several studies in the literature have used GP for classification [40,41] with excellent results, and thus, applying the proposed model could generate better performance. The federal fund's rate of the Federal Reserve of St. Louis is considered as the r_f asset,[1] represented in percent and not seasonally adjusted.

3.2.6. Fuzzy inference system (FIS)

FIS is based on fuzzy logic (FL) to analyze and find relevant relationships in the analyzed data, following a methodology of classification by different methods, among which those of Sugeno and Mamdani stand out. This type of systems works based on IF-THEN rules, where the input variables are related with logical operators *AND*, *OR* and *NO*. In addition, each variable introduced into the system, as well as its output, has a particular curve or function that describes the possibility the variable represents in the system, which are called membership functions.

The FIS model basically has 5 stages, which can be summarized as: (1) Transform the input data into fuzzy elements; (2) Apply fuzzy operators; (3) Apply the implication method; (4) Group all outputs and, (5) Transform the output data to its original characterization. The most relevant aspects of this type of methodology are the choice of membership functions and the model learning method.

3.2.7. FIS node

We propose a ternary trading system, where the algorithm makes a recommendation to buy, sell or no trade. A FIS node is selected to obtain these rules, where Eq. (1) represents the target for learning. A sub cluster approach is taken [42], because it is not clear how the data is actually related. A subtractive clustering (SC) algorithm performs a cluster analysis to generate the fuzzy sets through the projection of those clusters in a multidimensional space to its corresponding axis [43]. This helps to identify the optimal amount of groups that can represent the data's behavior. A gaussian function was used as a membership function, due to the fact that this function has good convergence, capable of generating stable results independent of the input data. Finally, a post-processing of the fuzzy signals is conducted to generate the final $Signal_{i,t}$ for the i stock in time t as

$$Signal_{i,t} = \begin{cases} 1 & \text{if } FISrule_{i,t} > \xi \quad \forall t \in T,\ \forall i \in N, \\ -1 & \text{if } FISrule_{i,t} < -\xi \quad \forall t \in T,\ \forall i \in N, \\ 0 & \sim \quad \forall t \in T,\ \forall i \in N, \end{cases} \tag{2}$$

where $FISrule_{i,t}$ is the original signal generated by the FISTGP approach and ξ is a constant related to how prone the investor is to open or close with a specific amount of information (expressed as the possibility recommendation of the FIS node). This parameter will be analyzed for different investor profile.

3.2.8. Fitness evaluation

The most widely used fitness function to generate trading rules in the literature is the conditional Sharpe ratio (CSR). In this study, since one of the most important aspects of a trading expert system is to provide useful information to the investor, it is desirable to have an accurate signal instead of a specific portfolio composition. Thus, we present a new fitness function focus on the accuracy of the generated signals instead of the risk adjusted return. To do this, first, a target is defined as the ideal case of when to buy, sell or no-trade, as explained in Eq. (1). Then, the accuracy of each program can be calculated as

$$Accuracy_{i,t} = \begin{cases} 1 & \text{If } Signal_{i,t} = Target_{i,t} \quad \forall t \in T,\ \forall i \in N, \\ 0 & \text{If } Signal_{i,t} \neq Target_{i,t} \quad \forall t \in T,\ \forall i \in N, \end{cases} \tag{3}$$

where $Signal_{i,t}$ was explained in Eq. (2). Thus, the fitness of each program would be

$$fitness = -\sum_{t=1}^{T}\sum_{i=1}^{N} Accuracy_{i,t} \tag{4}$$

The study pseudo code is represented in algorithm 1.

Algorithm 1 FISTGP pseudo code

1: **procedure** RULE_GENERATION_PROCESS(data_stocks, data_target, trn, h, stocks, max_gen, popsize, functions, terminals)
2: **for** i $\leftarrow \{trn, trn+h, trn+2h, \dots, len(data\_stocks)\}$ **do**
3: $train\_stocks \leftarrow data\_stocks(i-trn : i-h, 1 : stocks)$
4: $train\_target \leftarrow data\_target(i-trn : i-h, 1)$
5: $G \leftarrow 1$
6: *population* $\leftarrow$ Create initial population (ramped half-and-half,
7: functions, terminals, popsize)
8: **while** $G < max\_gen$ **do**
9: **for** $inds \leftarrow population$ **do**
10: $rules \leftarrow ind(train\_stocks)$
11: $signals \leftarrow rules(\xi)$
12: $performance(ind) \leftarrow fitness(signals, train\_target)$
13: **end for**
14: **if** $\max(performance) \geq tolerance$ **then**
15: $G \leftarrow G + 1$
16: *population* $\leftarrow$ Apply genetics operators to population
17: **else**
18: **break while**
19: **end if**
20: **end while**
21: $best\_program \leftarrow population(\max(performance)$
22: $test\_stocks \leftarrow Normalization(data\_stocks(i-h : i, 1 : stocks))$
23: $forecast(i : i+h) \leftarrow test\_stocks$
24: **end for**
25: **end procedure**

Line 6 is related to the initialization of the STGP algorithm, thus population is an object that contains all the initial configuration of the STGP. This object suffers direct modification (as in line 16) that are not fully specified in the pseudo code.

This pseudo code presents the process performed by the model in a simplified way in order to generate the rules and to forecast the purchase, sale, or non-movement h steps ahead. In order for the algorithm to operate, it needs the price and volume data of each stock (*data_stocks*), the ideal signals obtained

[1] https://fred.stlouisfed.org/series/DFF.

with Eq. (1) (*data_target*), the training days (*trn*), the steps ahead (h), the functions (*functions*), and terminals (*terminals*) established in Table 3, the maximum generations of the STGP (*max_gen*) and the number of individuals per generation (*popsize*). With respect to the input *tolerance*, this refers to a minimum percentage of acceptance of assertiveness, which was set at 50%.

3.3. Third phase (post-processing of signals)

With the constructed model and having selected the best program for the particular period, we proceed to performing the post-processing of the signals to return.

3.3.1. FISTGP post process return

The signals explained in Eq. (2) are evaluated in terms of the corresponding return as

$$rp_{i,t} = \begin{cases} r_{i,t} & \text{if } Signal_{i,t-2} = 1 \quad \forall i \in N\ t \in \{3, \dots, T\}, \\ r_{f_t} & \sim \quad \forall i \in N\ t \in \{1, \dots, T\} \end{cases} \tag{5}$$

where $rp_{i,t}$ is the portfolio return associated with the stock i in time t. Thus, profit of the portfolio on the analyzed period can be calculated as

$$RP = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N} rp_{i,t} \tag{6}$$

We refer to this configuration (FISTGP with accuracy fitness function) as the FISTGP-ACC model.

3.4. Benchmarks

To measure how well the proposed model performs, different benchmarks are proposed. To compare the different configurations, the returns were passed to a daily average of the out-of-sample period.

3.4.1. FISTGP with traditional fitness

The first natural benchmark is to test the proposed model against the same model but with the classical fitness approach, which is risk adjusted. The conditional Sharpe ratio (CSR) is used to select the program and obtain the appropriate rules. The CSR is defined in Eq. (7).

$$CSR = \frac{R - \sum_{t=1}^{T} r_{f_t}}{CVaR_{1-\alpha}} \tag{7}$$

where R is the return generated by the model and $CVaR_{1-\alpha}$ is the conditional value at risk with $100(1-\alpha)\%$ confidence level. $CVaR_{1-\alpha}$, is obtained as follows:

$$CVaR_{1-\alpha} = -E[r_{i,t} | r_{i,t} \le -VaR_{1-\alpha}] \tag{8}$$

where $VaR_{1-\alpha}$ is the worst expected loss at $1-\alpha$ confidence level. We refer to this model (FISTGP with CSR fitness) as the FISTGP-CSR model.

3.4.2. Markets

Another natural benchmark is the market. Proposing a framework that can earn excess of returns must be tested against the return obtained by the market, in order to be interesting enough to investors. Because of this, two well-known market indicators were selected as benchmarks: the Down Jones Composite Average (DJA); and, Standard & Poor's 500 (S&P500). These two measures reflect the reality of the US stock market quite well.

3.4.3. B&H strategy

The B&H strategy is the classical benchmark when trading rules are generated. As its name indicates, it consists of buying one or more stocks and holding them until the end of the period, earning the accumulated return of every passing day. This is the simplest investment strategy, and therefore, it is desirable that the generated rules earn more profit. This strategy is calculated as

$$bh_t = \frac{1}{N}\sum_{i=1}^{N} r_{i,t} \quad \forall t \in T \tag{9}$$

$$B\&H = \frac{1}{T}\sum_{t=1}^{T} bh_t \tag{10}$$

where bh_i is the B&H of stock i for all the period and $B\&H$ is the B&H for the portfolio for all the period.

3.4.4. Risk-free rate

As explained in 3.2.5, the portfolio can select the r_f as a possible asset. Thus, if the recommendation of the algorithm is to not have the stock in the portfolio, the final return will be the same as r_f. This makes the rule generation process more robust, because the incorrect selection of a stock would affect the portfolio performance more. The proposed model must be able to at least outperform the r_f.

3.5. Ratios

The following ratios are considered to measure the performance of the proposed method and benchmarks

- *Sharpe Ratio*: Shows the relationship between return and risk associated with a specific strategy. It is calculated as

$$SR = \frac{R - \sum_{i=1}^{T} r_{f_t}}{sd} \tag{11}$$

  where R is the return of the particular strategy being evaluated and sd is the standard deviation.
- *Conditional Sharpe Ratio*: Similar to the latter, it shows the relationship between the return and the worst risk scenario associated with a specific strategy. It is calculated as in Eqs. (7) and (8).
- *Buying return vs. Risk Free*: Records the return generated in each buying period, and then compares it to the return obtained by the r_f in the same period. The ratio is formulated as

$$Br\,vRf = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{BS_i}\sum_{bt=1}^{BT_i} Ibrrf_{i,bs} \tag{12}$$

  where $Ibrrf_{i,bs}$ is the individual analysis of the return associated with the stock i in the buying state bs against the return from the r_f over the same period. Considering $Isbs_{i,t}$ as the individual analysis when stock i is in a buying state in time t and $Ifd_{i,t}$ as the final day t of stock i buying state we have

$$Isbs_{i,t} = \begin{cases} 1 & \text{if } Signal_{i,t} = 1 \quad \forall i \in N, \forall t \in T, \\ 0 & \text{if } \sim \end{cases} \tag{13}$$

$$Ifd_{i,t} = \begin{cases} 1 & \text{if } Isbs_{i,t} = 0 \quad \forall i \in N, \forall t \in T, \\ & \ AND\ Isbs_{i,t-1} = -1 \\ 0 & \text{if } \sim \quad \forall i \in N, \forall t \in T \end{cases} \tag{14}$$

  and BS_i is

$$BS_i = \sum_{t=1}^{T} Ifd_{i,t} \quad \forall i \in N \tag{15}$$

Finally, the return of every buying state bs must first be calculated as

$$Ifrbs_{i,bs} = \frac{1}{S^{bs}} \sum_{s=1}^{S^{bs}} rp_{i,s} - \frac{1}{S^{bs-1}} \sum_{s=1}^{S^{bs-1}} rp_{i,s} \quad \forall i \in N, \quad \forall bs \in BS_i \tag{16}$$

where $rp_{i,s}$ is the return of stock i when it is in a buying state bs and S^{bs} indicates the final day of the buying state period bs for stock i. The r_f return associated to the same period is calculated as

$$rfbs_{i,bs} = \frac{1}{S^{bs}} \sum_{s=1}^{S^{bs}} r_{f_s} - \frac{1}{S^{bs-1}} \sum_{s=1}^{S^{bs-1}} r_{f_s} \quad \forall i \in N, \quad \forall bs \in BS_i \tag{17}$$

where r_{f_s} is the return of the r_f associated with the buying state bs. Then, $Ibrrf_i$ is expressed as

$$Ibrrf_{i,bs} = \begin{cases} 1 & \text{if } Ifrbs_{i,bs} > rfbs_{i,bs} \quad \forall i \in N, \forall bs \in BS_i, \\ 0 & \text{if } Ifrbs_{i,bs} > rfbs_{i,bs} \quad \forall i \in N, \forall bs \in BS_i \end{cases} \tag{18}$$

- *Not buying state against B&H*: Records the return generate in each not buying state, and compares it to the B&H in the same period. The ratio is formulated as

$$NBSvsBH = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{NBS_i} \sum_{nbs=1}^{NBS_i} Inbsbh_{i,nbs} \tag{19}$$

where $Inbsbh_{i,nbs}$ is the individual analysis of the return associated with the stock i in the not buying state nbS against the return from the $bh_{i,t}$ over the same period. Considering $Insbs_{i,t}$ as the individual analysis when stock i is not in a buying state in time t and $Infd_{i,t}$ as the final day t of stock i not in buying state we have

$$Insbs_{i,t} = \begin{cases} 1 & \text{if } Signal_{i,t} = 0 \quad \forall i \in N, \forall t \in T, \\ & \;\; OR\ Signal_{i,t} = -1 \\ 0 & \text{if } \sim \end{cases} \tag{20}$$

$$Infd_{i,t} = \begin{cases} 1 & \text{if } Isbs_{i,t} = 0 \quad \forall i \in N, \forall t \in T, \\ & \;\; AND\ Isbs_{i,t-1} = 1 \\ 0 & \text{if } \sim \quad \forall i \in N, \forall t \in T \end{cases} \tag{21}$$

and NBS_i is

$$NBS_i = \sum_{t=1}^{T} Infd_{i,t} \quad \forall i \in N \tag{22}$$

Finally, the return of every element that is in a not buying state bs must first be calculated as

$$Ifrnbs_{i,nbs} = \frac{1}{S^{nbs}} \sum_{s=1}^{S^{nbs}} rp_{i,s} - \frac{1}{S^{nbs-1}} \sum_{s=1}^{S^{nbs-1}} rp_{i,s} \quad \forall i \in N, \quad \forall nbs \in NBS_i \tag{23}$$

where $rp_{i,s}$ is the return of stock i when it is not in a buying state nbs and S^{nbs} indicates the final day of the not buying state period nbs for stock i. The $bh_{i,t}$ return associated to the same period is calculated as

$$bhnbs_{i,nbs} = \frac{1}{S^{nbs}} \sum_{s=1}^{S^{nbs}} r_{i,t} - \frac{1}{S^{nbs-1}} \sum_{s=1}^{S^{nbs-1}} r_{i,t} \quad \forall i \in N, \quad \forall nbs \in NBS_i \tag{24}$$

where r_{f_s} is the return of the r_f associated with the buying state bs. Then, $Ibrrf_i$ is expressed as

$$Inbsbh_{i,nbs} = \begin{cases} 1 & \text{if } Ifrnbs_{i,nbs} > bhnbs_{i,nbs} \quad \forall i \in N, \forall nbs \in NBS_i, \\ 0 & \text{if } Ifrnbs_{i,nbs} > bhnbs_{i,nbs} \quad \forall i \in N, \forall nbs \in NBS_i \end{cases} \tag{25}$$

- *Average Assertiveness*: It indicates the average of both latter ratios, which is how the model, in general, performs according to the rules explained in (1). It is calculated as

$$AA = \frac{1}{N} \sum_{i=1}^{N} \left[\left(\frac{1}{BS_i} \sum_{s}^{BS_i} Ibrrf_{i,nbs} \right) + \left(\frac{1}{NBS_i} \sum_{ns}^{NBS_i} Inbsbh_{i,nbs} \right) \right] \tag{26}$$

were $Inbsbh_{i,nbs}$ is calculated as in Eq. (25) and $Ibrrf_{i,nbs}$ is calculated as in Eq. (18).

4. FISTGP simulations

4.1. Results

Several experiments were conducted to test the proposed framework. Specifically, thirteen combinations of training days and forecasting horizons were selected, as explained in Section 3. Specifically, we consider the combinations as valid where the training period was at least twice the forecasting ahead period. Thus, of the 20 possible cases to be considered, only 13 were analyzed: (ordered as the first number representing training days and the second number as forecasting days ahead) 33-10, 66-10, 66-22, 66-33, 126-10, 126-22, 126-33, 126-66, 252-10, 252-22, 252-33, 252-66, and 252-126. The main objective of a well-defined portfolio is the balance between return and risk, and thus, the most important ratio would be the conditional Sharpe ratio (CSR), which indicates a good measure of both characteristics; therefore, model discrimination is conducted by this particular ratio. In Fig. 2, the STGPFIS-ACC and STGPFIS-CSR models are presented, together with the B&H, DJA and S&P500 as benchmarks, with a ξ equal to 0.5.

It can be seen that, for this particular ξ, the model performs relatively better than the market in all the experiments, both for STGPFIS-ACC and STGPFIS-CSR. Nonetheless, nearly half of them manage to outperform the B&H strategy. However, if we consider the best training-forecast combination for the STGPFIS-ACC model (which is obtained with 66 days for training and 33 days for forecast horizon), CSR outperforms the B&H ratio by 85.71%, the S&P500 ratio by 164.06%, the DJA ratio by 186.44% and the STGPFIS-CSR ratio by 9.74%. These results indicate that the combination of STGP and FIS is actually able to identify good relationships in the data and to provide efficient recommendations that can outperform the market (considering that the best model in the STGPFIS-CSR approach also outperforms the B&H (69.23%), S&P500 (140.63%) and the DJA (161.02%) ratios).

However, the return obtained by the model was not able to outperform the B&H return, reaching 3.21E−04 against the 3.62E−04 of the basic strategy (−11.40%). This is precisely expected because of the result obtained by the CSR ratio, meaning that the risk was diminished considerably more (CVaR of the STGPFIS-ACC was 0.0156 and of the B&H 0.0335, implying a 53.43% less risk) than the return, in the well-known relationship of these two indicators (e.g. more risk, more return) (see Table 4).

Regarding rule generation, the three ratios to test this show interesting results. In buying periods, the best STGPFIS-ACC was able to correctly identify these states 61.80% of the time when it was actually convenient, which is to say that the model return

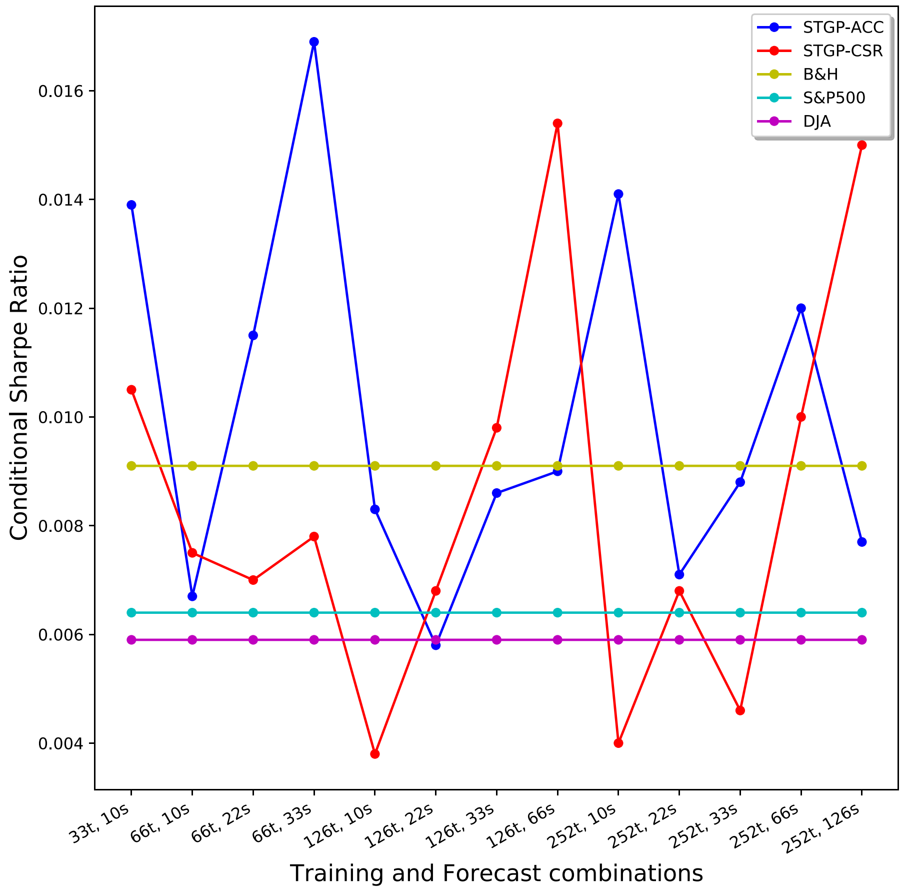

Fig. 2. Conditional sharpe ratio for models and benchmarks. The figure shows all the computed models' performances in terms of return adjusted by risk (through the CSR). Each point represents the mean daily performance of that model for a specific training-forecast combination. Notice that B&H, S&P500 and DJA are constant because they do not require training, so their performance is the same regardless of the mentioned combination. The x-axis ticks are read as training-forecast, i.e., for the first tick, 33t, 10s, means that the algorithm takes 33 days for training, and then forecasts 10 days-ahead in a rolling windows approach.

Table 4
Analysis of proposed framework for 0.5 recommendation information.

Ratio	STGP-ACC[a]	STGP-CSR[b]	B&H	DJA	S&P500
Return	3.21E−04	3.02E−04	3.62E−04	2.17E−04	2.50E−04
Standard deviation	0.0065	0.0065	0.0136	0.0136	0.0112
Sharpe ratio	0.0404	0.0376	0.0225	0.0143	0.0159
Conditional VaR	0.0156	0.0159	0.0335	0.0273	0.0302
Conditional sharpe ratio	0.0169	0.0154	0.0091	0.0059	0.0064
Buying return vs. risk free[c]	61.80%	52.85%	–	–	–
Not buying state against B&H[d]	47.33%	45.66%	–	–	–
Average assertiveness[e]	54.45%	49.24%	–	–	–

[a]Best configuration was achieved with 66 days for training and 33 forecast horizon.
[b]Best configuration was achieved with 126 days for training and 66 forecast horizon.
[c]Shows how many times the algorithm correctly identifies when it is convenient to buy (or stay in a buying state).
[d]Shows how many times the algorithm correctly identifies when it is convenient to sell (or stay in a not buying state).
[e]Shows the general performance of the algorithm in terms of its ability to generate the exact theoretical rule (based on the ternary rule system).

was higher than r_f in those periods. On the other hand, the model reached 47.33% of the assertiveness of when it is not convenient to stay in these states, which is to say that the model return was higher than *B&H* return in these periods. This is reflected in the assertiveness average, where the model reached 54.45%. These results indicate that the proposed rules were relatively learned by the model, identifying more states where it was convenient to buy but failing to identify the state where it was not convenient to invest in the stock.

The ξ parameter was selected arbitrarily, so two additional values for it were also tested. The 0.5 value can be interpreted as a neutral relationship with the possibilities that the model gives to the investor. So, 0.25 and 0.75 were tested, the former being an investor that requires less information to consider a recommendation to be true and the latter, an investor that requires more. This sensitization can be seen in Tables 6 and 8, respectively, in the Appendix. From this, the best models (considering CSR) for both STGPFIS-ACC and STGP-CSR are presented in Table 5.

Here, we notice that the model performs better when the threshold for selecting an option as a valid recommendation is diminished for the STGPFIS-ACC model. All the ratios are better, since the importance of the return. In this configuration, the

Table 5
Sensitization of recommendation information parameter.

Ratio	STGP-ACC[a]	STGP-CSR[b]	B&H	DJA	S&P500
Return	4.47E−04	3.02E−04	3.62E−04	2.17E−04	2.50E−04
Standard deviation	0.0067	0.0065	0.0136	0.0136	0.0112
Sharpe ratio	0.0585	0.0376	0.0225	0.0143	0.0159
Conditional VaR	0.0158	0.0159	0.0335	0.0273	0.0302
Conditional sharpe ratio	0.0247	0.0154	0.0091	0.0059	0.0064
Buying return vs. risk free[c]	63.46%	52.85%	–	–	–
Not buying state against B&H[d]	53.17%	45.66%	–	–	–
Average assertiveness[e]	58.28%	49.24%	–	–	–

[a]The best model was achieved with 0.25 for ξ parameter, 66 days for training and 33 forecast horizon.
[b]The best model was achieved with 0.50 for ξ parameter, 126 days for training and 66 forecast horizon.
[c]Shows how many times the algorithm correctly identifies when it is convenient to buy (or stay in a buying state).
[d]Shows how many times the algorithm correctly identifies when it is convenient to sell (or stay in a not buying state).
[e]Shows the general performance of the algorithm in terms of its ability to generate the exact theoretical rule (based on the ternary rule system).

Table 6
0.25 recommendation information parameter.

Model	Return	SR	CSR	Model	Return	SR	CSR
33-10	3.47E−04	0.0429	0.0180	126-66	2.32E−04	0.0244	0.0100
66-10	1.46E−04	0.0074	0.0048	252-10	2.19E−04	0.0242	0.0098
66-22	4.66E−04	0.0519	0.0218	252-22	2.94E−04	0.0314	0.0127
66-33	4.47E−04	0.0585	0.0247	252-33	2.47E−04	0.0242	0.0096
126-10	1.65E−04	0.0155	0.0063	252-66	2.64E−04	0.0318	0.0127
126-22	2.48E−04	0.0277	0.0112	252-126	2.01E−04	0.0266	0.0108
126-33	2.25E−04	0.0241	0.0098				

Table 7
0.50 recommendation information parameter.

Model	Return	SR	CSR	Model	Return	SR	CSR
33-10	2.82E−04	0.0334	0.0139	126-66	2.07E−04	0.0221	0.0090
66-10	1.79E−04	0.0074	0.0067	252-10	2.32E−04	0.0342	0.0141
66-22	2.59E−04	0.0285	0.0115	252-22	1.79E−04	0.0177	0.0071
66-33	3.21E−04	0.0404	0.0169	252-33	2.02E−04	0.0223	0.0088
126-10	1.96E−04	0.0202	0.0083	252-66	2.42E−04	0.0289	0.0120
126-22	1.62E−04	0.0144	0.0058	252-126	1.76E−04	0.0197	0.0077
126-33	2.15E−04	0.0213	0.0086				

Table 8
0.75 recommendation information parameter.

Model	Return	SR	CSR	Model	Return	SR	CSR
33-10	2.30E−04	0.0262	0.0108	126-66	2.81E−04	0.0345	0.0141
66-10	2.22E−04	0.0086	0.0078	252-10	1.61E−04	0.0135	0.0054
66-22	1.68E−04	0.0144	0.0057	252-22	1.33E−04	0.0116	0.0047
66-33	1.78E−04	0.0148	0.0059	252-33	1.55E−04	0.0141	0.0056
126-10	1.57E−04	0.0119	0.0047	252-66	2.15E−04	0.0230	0.0091
126-22	1.68E−04	0.0161	0.0064	252-126	2.43E−04	0.0275	0.0115
126-33	1.82E−04	0.0220	0.0087				

model manages to outperform the B&H return (in 28.62%) with almost the same risk (1.28% higher). This indicates that the model with the best configuration is able to optimize the relationship between risk and return, and manages to dominate the other strategies. This is also reflected in the CSR ratio in the test period, were its increment is 46.15% with respect to the same model with ξ parameter 0.5.

Ratios regarding assertiveness also improve. It is important to identify when it is not convenient to invest in the stocks, and the model improvement in this case is around 11 percentage points, leaving the strategy with more general accuracy for identifying the correct state (58.28%).

A final sensitization was performed, considering the starting point. Because the year 2003 was relatively arbitrary, and due to the fact that this year is post asian crisis, we take 4 more different starting points, considering 2001, 2002, 2004 and 2005, described in Table 9 in the Appendix. We can notice that the best model configuration remains the same, considering a ξ parameter of 0.25, 66 days for training and 33 days forecast horizon. Of all the configurations tested, none achieve better performance than the 2003 case, but all manage to outperform the 4 benchmarks, and thus, we can say that the framework does not depend on a specific period, but that it can dynamically adapt and perform better than the market and B&H strategy.

5. Discussion and conclusion

In this paper, we proposed a fuzzy inference system, strongly-typed genetic programming (FISTGP) embedded model to generate trading rules for 90 of the most traded and active stocks in the US stock market. The main contribution of this study was the combination of both models in a financial application, the fitness function and the ternary rules based on r_f. Additionally, the sensitization of the information parameter as well as the length sensitization of the training set and the forecast step were performed.

The model was able to successfully identify the relationship within the data to generate trading rules that outperformed the classical B&H approach and the model in the different variants regarding fitness and r_f. conditional Sharpe ratio in the testing period was higher than the market and B&H strategy, and also in the best configuration for the framework, the return obtained was higher. The rule generation process was learned by the model, achieving an average of 58% of accuracy in detecting the correct state.

5.1. Stock characteristics

This tables show the different B&H depending on the starting point for the test period.

Solving the trading decision problem with a selection approach in the fitness function allows us to generate more accurate rules, which are clear and understandable for an investors that are interested in entering financial markets. Accurate rules generated by the framework led to better average performance in all the test periods.

The combination of FIS and GP architectures has been explored in the literature. Mousavi, Esfahanipour & Zarandi [20] used a MGP model in combination with a Takagi–Sugeno–Kang (TSK) inference system to create dynamic portfolios. The decisions are based on the most important technical index, determined by the step-wise regression. Additionally, a Wavelet decomposition was used to reduce the noise. Specifically, the MGP was used to learn the TSK system rules. It was applied to the TSE (emerging market), the Toronto and Frankfurt index (mature markets). Considering a

Table 9
Different starting point cases.

2001 model[a]	0.25 CSR	0.50 CSR	0.75 CSR	Model	0.25 CSR	0.50 CSR	0.75 CSR
33-10	0.0139	0.0097	0.0078	126-66	0.0076	0.0043	0.0051
66-10	0.0035	0.0041	0.0035	252-10	0.0065	0.0068	0.0045
66-22	0.0185	0.0077	0.0052	252-22	0.0098	0.0052	0.0051
66-33	0.0212	0.0126	0.0067	252-33	0.0083	0.0062	0.0080
126-10	0.0017	0.0036	0.0045	252-66	0.0112	0.0080	0.0065
126-22	0.0070	0.0020	0.0030	252-126	0.0103	0.0073	0.0066
126-33	0.0078	0.0053	0.0051				

2002 model[a]	0.25 CSR	0.50 CSR	0.75 CSR	Model	0.25 CSR	0.50 CSR	0.75 CSR
33-10	0.0156	0.0117	0.0096	126-66	0.0085	0.0063	0.0064
66-10	0.0041	0.0050	0.0045	252-10	0.0084	0.0086	0.0055
66-22	0.0198	0.0089	0.0069	252-22	0.0105	0.0058	0.0059
66-33	0.0219	0.0133	0.0078	252-33	0.0082	0.0063	0.0084
126-10	0.0033	0.0057	0.0065	252-66	0.0110	0.0086	0.0073
126-22	0.0094	0.0038	0.0046	252-126	0.0098	0.0074	0.0067
126-33	0.0079	0.0064	0.0060				

2004 model[a]	0.25 CSR	0.50 CSR	0.75 CSR	Model	0.25 CSR	0.50 CSR	0.75 CSR
33-10	0.0153	0.0115	0.0095	126-66	0.0073	0.0055	0.0053
66-10	0.0016	0.0039	0.0044	252-10	0.0080	0.0106	0.0068
66-22	0.0190	0.0090	0.0078	252-22	0.0113	0.0055	0.0056
66-33	0.0221	0.0139	0.0083	252-33	0.0083	0.0069	0.0080
126-10	0.0049	0.0071	0.0079	252-66	0.0101	0.0093	0.0084
126-22	0.0091	0.0046	0.0058	252-126	0.0086	0.0059	0.0056
126-33	0.0074	0.0067	0.0060				

2005 model[a]	0.25 CSR	0.50 CSR	0.75 CSR	Model	0.25 CSR	0.50 CSR	0.75 CSR
33-10	0.0140	0.0105	0.0086	126-66	0.0064	0.0047	0.0043
66-10	0.0007	0.0030	0.0034	252-10	0.0074	0.0101	0.0061
66-22	0.0181	0.0079	0.0068	252-22	0.0102	0.0048	0.0050
66-33	0.0213	0.0133	0.0078	252-33	0.0075	0.0061	0.0071
126-10	0.0048	0.0068	0.0076	252-66	0.0092	0.0087	0.0080
126-22	0.0075	0.0036	0.0049	252-126	0.0085	0.0060	0.0060
126-33	0.0071	0.0059	0.0048				

[a]Indicates starting point of the test period.

Table 10
Description of stocks 1–30.

Stocks	Symbol	B&H 2001	B&H 2002	B&H 2003	B&H 2004	B&H 2005
Apple Inc.	AAPL	1.27E−03	1,24E−03	1.47E−03	1.46E−03	1.21E−03
Applied Materials, Inc.	AMAT	4.98E−05	8.16E−06	1.26E−04	−1.17E−05	1.30E−04
Apache Corporation	APA	1.60E−04	2.40E−04	2.20E−04	9.47E−05	3.78E−05
Boeing Co.	BA	2.95E−04	4.48E−04	5.41E−04	5.00E−04	4.75E−04
Bank of America Corp.	BAC	3.27E−05	−6.73E−05	−1.21E−04	−1.78E−04	−2.58E−04
Best Buy Co. Inc.	BBY	2.94E−04	5.40E−05	2.70E−04	4.11E−05	1.98E−05
Baker Hughes Incorporated	BHI	1.15E−04	1.68E−04	2.19E−04	2.17E−04	1.34E−04
Bristol-Myers Squibb Co.	BMY	1.36E−04	2.50E−04	4.69E−04	4.39E−04	5.28E−04
Citigroup Inc.	C	−5.25E−04	−5.69E−04	−5.17E−04	−6.69E−04	−7.36E−04
Caterpillar Inc.	CAT	4.00E−04	3.87E−04	4.40E−04	2.87E−04	2.60E−04
CBS Corporation Common Stock	CBS	1.49E−04	1.59E−04	2.00E−04	2.08E−04	2.77E−04
Celgene Corporation	CELG	8.78E−04	9.63E−04	1.12E−03	1.00E−03	1.02E−03
Chesapeake Energy Corporation	CHK	−1.32E−04	2.16E−06	−7.64E−05	−2.70E−04	−3.51E−04
Comcast Corporation	CMCSA	2.48E−04	3.10E−04	4.30E−04	3.74E−04	4.29E−04
Comcast Corporation	COF	8.38E−05	1.39E−04	3.15E−04	1.12E−04	2.92E−05
ConocoPhillips	COP	3.74E−04	3.77E−04	4.65E−04	3.90E−04	3.24E−04
Costco Wholesale Corporation	COST	4.23E−04	4.25E−04	5.87E−04	5.55E−04	5.10E−04
Cisco Systems, Inc.	CSCO	−5.85E−05	1.15E−04	2.29E−04	6.13E−05	1.78E−04
CVS Health Corp.	CVS	3.43E−04	5.69E−04	6.62E−04	5.97E−04	5.62E−04
Chevron Corporation	CVX	3.41E−04	3.40E−04	4.38E−04	3.86E−04	3.42E−04
E I Du Pont De Nemours And Co.	DD	2.27E−04	2.56E−04	2.67E−04	2.74E−04	2.69E−04
Deere & Co.	DE	4.13E−04	4.44E−04	4.62E−04	3.85E−04	3.75E−04
Walt Disney Co.	DIS	4.09E−04	4.98E−04	6.11E−04	5.56E−04	5.64E−04
Dow Jones	DOW	2.35E−04	2.50E−04	3.00E−04	2.13E−04	1.64E−04
Devon Energy Corp.	DVN	1.45E−04	2.81E−04	2.56E−04	1.92E−04	1.19E−04
eBay Inc.	EBAY	5.71E−04	4.10E−04	4.17E−04	2.56E−04	1.02E−04
Emblem Corp.	EMC	−2.41E−04	1.32E−04	3.88E−04	2.06E−04	2.36E−04
EOG Resources Inc.	EOG	5.03E−04	6.41E−04	6.84E−04	6.84E−04	6.13E−04
Ford Motor Company	F	−4.41E−05	4.10E−05	1.84E−04	1.46E−05	6.52E−05
Freeport-McMoRan Inc.	FCX	2.90E−04	1.75E−04	1.22E−04	−1.80E−04	−1.34E−04

Table 11
Description of stocks 31–60.

Stocks	Symbol	B&H 2001	B&H 2002	B&H 2003	B&H 2004	B&H 2005
FedEx Corporation	FDX	3.84E−04	3.19E−04	3.29E−04	3.07E−04	2.06E−04
General Electric Company	GE	−8.95E−07	5.09E−05	1.76E−04	1.14E−04	6.73E−05
Gilead Sciences, Inc.	GILD	9.82E−04	9.49E−04	9.72E−04	8.73E−04	9.26E−04
Corning Incorporated	GLW	−2.35E−04	2.08E−04	5.21E−04	2.16E−04	2.2Ee-04
Goldman Sachs Group Inc.	GS	1.96E−04	2.40E−04	3.41E−04	2.65E−04	2.61E−04
Halliburton Company	HAL	2.64E−04	6.19E−04	5.01E−04	4.13E−04	3.11E−04
Home Depot Inc.	HD	3.65E−04	3.61E−04	6.51E−04	5.31E−04	5.23E−04
Hess Corp.	HES	2.80E−04	3.42E−04	3.97E−04	4.19E−04	3.22E−04
Honeywell International Inc.	HON	3.05E−04	4.10E−04	5.33E−04	4.71E−04	5.00E−04
HP Inc.	HPQ	2.21E−05	1.10E−04	1.57E−04	1.12E−04	1.56E−04
IBM Common Stock	IBM	1.92E−04	9.69E−05	2.15E−04	2.05E−04	2.08E−04
Intel Corporation	INTC	1.26E−04	8.89E−05	3.13E−04	1.14E−04	2.73E−04
Johnson & Johnson	JNJ	2.78E−04	2.66E−04	2.92E−04	3.38E−04	2.91E−04
Juniper Networks, Inc.	JNPR	−3.76E−04	1.01E−04	3.86E−04	1.41E−04	7.34E−05
JPMorgan Chase & Co.	JPM	2.22E−04	2.71E−04	3.79E−04	2.92E−04	3.00E−04
KLA-Tencor Corp	KLAC	3.06E−04	1.85E−04	2.97E−04	1.98E−04	3.35E−04
The Coca-Cola Co.	KO	1.94E−04	2.89E−04	3.12E−04	2.92E−04	3.82E−04
Kohl's Corporation	KSS	−3.62E−05	−7.88E−05	−1.67E−06	5.26E−05	4.54E−05
Eli Lilly And Co.	LLY	1.02E−04	1.60E−04	2.10E−05	2.05E−04	2.93E−04
Lowe's Companies, Inc.	LOW	5.59E−04	4.01E−04	4.88E−04	3.94E−04	4.21E−04
McDonald's Corporation	MCD	4.25E−04	5.16E−04	7.03E−04	6.16E−04	5.81E−04
Medtronic PLC	MDT	1.15E−04	1.84E−04	2.09E−04	2.13E−04	2.19E−04
Merck & Co., Inc.	MRK	9.09E−06	1.41E−04	1.38E−04	1.92E−04	3.48E−04
Morgan Stanley	MS	−1.10E−04	−3.92E−05	4.84E−05	−5.26E−05	−5.30E−05
Microsoft Corporation	MSFT	3.38E−04	2.32E−04	3.13E−04	3.33E−04	3.43E−04
Motorola Solutions Inc.	MSI	1.19E−05	9.23E−05	2.33E−04	1.00E−04	6.83E−05
Micron Technology, Inc.	MU	−2.14E−04	−2.35E−04	1.13E−04	1.54E−05	1.21E−04
Newmont Mining Corp.	NEM	6.54E−05	3.86E−05	−7.86E−05	−2.65E−04	−2.44E−04
Nike Inc.	NKE	6.52E−04	6.89E−04	8.08E−04	7.41E−04	7.09E−04
National-Oilwell Varco, Inc.	NOV	2.32E−04	4.48E−04	4.53E−04	4.83E−04	3.62E−04

Table 12
Description of stocks 61–90.

Stocks	Symbol	B&H 2001	B&H 2002	B&H 2003	B&H 2004	B&H 2005
NetApp Inc.	NTAP	−1.86E−04	8.88E−05	3.01E−04	1.28E−04	9.89E−06
NVIDIA Corporation	NVDA	4.85E−04	1.14E−04	6.25E−04	4.66E−04	5.48E−04
Oracle Corporation	ORCL	9.63E−05	2.83E−04	3.66E−04	3.65E−04	4.19E−04
Occidental Petroleum Corporation	OXY	5.84E−04	5.90E−04	6.03E−04	5.08E−04	4.44E−04
PepsiCo, Inc.	PEP	2.78E−04	3.03E−04	3.63E−04	3.53E−04	3.42E−04
Pfizer Inc.	PFE	4.61E−05	8.86E−05	1.61E−04	1.18E−04	2.44E−04
Procter & Gamble Co.	PG	2.74E−04	2.91E−04	2.74E−04	2.49E−04	2.21E−04
Potash Corporation of Saskatchewan Inc.	POT	4.70E−04	5.64E−04	5.91E−04	5.30E−04	3.75E−04
QUALCOMM, Inc.	QCOM	9.79E−05	2.63E−04	3.57E−04	2.50E−04	1.13E−04
Transocean Ltd.	RIG	−2.67E−04	−1.88E−04	−8.26E−05	−1.18E−04	−3.27E−04
Starbucks Corporation	SBUX	6.64E−04	7.15E−04	7.79E−04	6.91E−04	5.45E−04
Schlumberger Limited	SLB	2.30E−04	3.53E−04	4.65E−04	4.07E−04	3.68E−04
AT&T Inc.	T	9.47E−05	1.46E−04	2.32E−04	2.78E−04	3.13E−04
Target Corporation	TGT	2.75E−04	2.18E−04	3.37E−04	2.81E−04	2.18E−04
Time Warner Inc.	TWX	3.14E−05	5.16E−05	3.13E−04	2.56E−04	2.65E−04
Texas Instruments Incorporated	TXN	1.11E−04	2.54E−04	4.51E−04	2.88E−04	4.13E−04
Tyco International PLC	TYC	5.96E−06	1.19E−05	3.65E−04	2.36E−04	1.58E−04
UnitedHealth Group Inc.	UNH	5.55E−04	5.59E−04	5.45E−04	4.86E−04	3.74E−04
Union Pacific Corporation	UNP	5.74E−04	5.64E−04	5.93E−04	6.11E−04	6.68E−04
United Parcel Service, Inc.	UPS	2.43E−04	2.65E−04	2.46E−04	2.17E−04	1.83E−04
U.S. Bancorp	USB	2.93E−04	3.32E−04	3.25E−04	2.62E−04	2.42E−04
United Technologies Corporation	UTX	3.16E−04	3.85E−04	4.21E−04	3.18E−04	3.18E−04
Valero Energy Corporation	VLO	6.03E−04	6.14E−04	6.92E−04	6.68E−04	5.02E−04
Verizon Communications Inc.	VZ	1.58E−04	1.60E−04	2.04E−04	2.63E−04	2.44E−04
Walgreens Boots Rg	WBA	2.38E−04	3.06E−04	3.56E−04	3.59E−04	3.25E−04
Wells Fargo & Co.	WFC	2.89E−04	3.72E−04	3.62E−04	3.25E−04	3.16E−04
Wal-Mart Stores Inc.	WMT	9.37E−05	7.68E−05	1.21E−04	1.09E−04	1.12E−04
United States Steel Corporation	X	−1.71E−04	−1.99E−04	−1.30E−04	−4.60E−04	−6.18E−04
Exxon Mobil Corporation	XOM	2.64E−04	3.03E−04	3.56E−04	3.25E−04	2.73E−04
Yahoo! Inc.	YHOO	2.16E−04	3.51E−04	3.88E−04	1.16E−04	−1.78E−05

conditional Sharpe ratio as the fitness function, their model outperforms the classical models (B&H, Momentum) and also other machine learning approaches. Bahar, Zarandi & Esfahanipour [21] used GNP with reinforcement learning and FIS to generate ternary trading rules. First, they used a Wavelet transform to eliminate noise as a pre-processing step. Then, they introduced the FIS structure inside the transition and process nodes of the GNP. It was applied over 10 TSE companies, and simulations demonstrated that the model is able to obtain excess return and increase the algorithm's accuracy, outperforming several other strategies.

Although both studies use different algorithms in combination with FIS, they achieve great performances. However, a key difference of our study is the fitness function, focused on the accuracy of the generated rules instead of the return–risk relationship.

Also, we explore the diversification aspect of investment considering the 90 most traded stock in the U.S. stock market, which makes the study more practical not just for investors, but also for fund management. Finally, another difference is that both studies do not use rolling windows to make the forecast. This approach allows the model to see patterns that have a direct economic interpretation, i.e. a three-month period is the optimal training period to forecast five weeks in advance. This kind of information is very useful for investors, as they can prepare and anticipate how to move their portfolios.

The STGP algorithm is restrictive in the sense that the functions provided are assumed to be sufficient to solve the problem. This can generate a regular performance because, as expected, it is difficult to know in advance all the functions that are required to correctly solve the problem. However, in this study, considering just some basic functions, the algorithm nevertheless is able to achieve a great performance. This brings the opportunity to consider other kinds of functions that can help the algorithm to perform even better, taken the approach already seen in the literature such as [20] taking technical index. This can be included inside the algorithm as functions as an extension of the presented work. Another approach with respect to the STGP algorithms is to explore other kinds of tree-based methodologies in which a root node can be modified to include the FIS algorithm. This can be an interesting search, based on the ternary rule system proposed in the present study, and if possible, the fitness function.

6. Future work

A deeper development with respect to the configuration of the fuzzy node could generate better results, with respect to a more specific membership function for the problem, as well as the algorithm for learning rules. Also, an extended analysis of the ξ parameter could be studied, testing its implications in the overall performance of the framework.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2020.106169.

CRediT authorship contribution statement

Kevin Michell: Conceptualization, Methodology, Software, Writing - original draft, Validation. **Werner Kristjanpoller:** Conceptualization, Methodology, Supervision, Validation, Writing - review & editing.

Acknowledgments

We would like to thank the financial support from Dirección General de Investigación , Innovación y Postgrado (DGIIP), UTFSM, Valparaíso, Chile, through the ‘‘Programa de Incentivos a la Iniciación Científica (PIIC)’’ initiative.

Appendix

A.1. Simulations results

These tables are presented as (*training days* - *forecast horizon*). Tables 6 to 8 show the different cases for the recommendation parameter. Table 9 shows the different starting points analysis.

References

[1] H. Markowitz, Portfolio selection, J. Finance 7 (1) (1952) 77–91.
[2] A. Esfahanipour, S. Mousavi, A genetic programming model to generate risk-adjusted technical trading rules in stock markets, Expert Syst. Appl. 38 (7) (2011) 8438–8445.
[3] M. Gunasekaran, K. Ramaswami, S. Karthik, Integration of sra, anfis and capm for stock portfolio management, CSI Trans. ICT 1 (4) (2013) 291–300.
[4] R.J. Bauer, Genetic Algorithms and Investment Strategies, John Wiley & Sons, 1994.
[5] M.A. Boyacioglu, D. Avci, An adaptive network-based fuzzy inference system (anfis) for the prediction of stock market return: the case of the istanbul stock exchange, Expert Syst. Appl. 37 (12) (2010) 7908–7912.
[6] A. Fernández, S. Gómez, Portfolio selection using neural networks, Comput. Oper. Res. 34 (4) (2007) 1177–1191.
[7] A. Fan, M. Palaniswami, Stock selection using support vector machines, in: International Joint Conference on Neural Networks, 2001. Proceedings. IJCNN'01, Vol. 3, IEEE, 2001, pp. 1793–1798.
[8] M.A. Dempster, C.M. Jones, A real-time adaptive trading system using genetic programming, Quant. Finance 1 (4) (2001) 397–413.
[9] T.-J. Chang, S.-C. Yang, K.-J. Chang, Portfolio optimization problems in different risk measures using genetic algorithm, Expert Syst. Appl. 36 (7) (2009) 10529–10537.
[10] H. Holland John, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan, USA, 1975.
[11] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT press, 1992.
[12] M.C. Roberts, Technical analysis and genetic programming: constructing and testing a commodity portfolio, J. Futures Mark. 25 (7) (2005) 643–660.
[13] F. Allen, R. Karjalainen, Using genetic algorithms to find technical trading rules, J. Financ. Econ. 51 (2) (1999) 245–271.
[14] J.-S. Chen, C.-L. Chang, J.-L. Hou, Y.-T. Lin, Dynamic proportion portfolio insurance using genetic programming with principal component analysis, Expert Syst. Appl. 35 (1) (2008) 273–278.
[15] J.-Y. Potvin, P. Soriano, M. Vallée, Generating trading rules on the stock markets with genetic programming, Comput. Oper. Res. 31 (7) (2004) 1033–1047.
[16] S. Ha, B.-R. Moon, Finding attractive technical patterns in cryptocurrency markets, Memet. Comput. 10 (3) (2018) 301–306.
[17] J.M. Berutich, F. López, F. Luna, D. Quintana, Robust technical trading strategies using gp for algorithmic portfolio selection, Expert Syst. Appl. 46 (2016) 307–315.
[18] V. Manahov, R. Hudson, P. Linsley, New evidence about the profitability of small and large stocks and the role of volume obtained using strongly typed genetic programming, J. Int. Financ. Mark. Inst. Money 33 (2014) 299–316.
[19] V. Manahov, The rise of the machines in commodities markets: new evidence obtained using strongly typed genetic programming, Ann. Oper. Res. 260 (1–2) (2018) 321–352.
[20] S. Mousavi, A. Esfahanipour, M.H.F. Zarandi, Mgp-intactsky: Multitree genetic programming-based learning of interpretable and accurate tsk systems for dynamic portfolio trading, Appl. Soft Comput. 34 (2015) 449–462.
[21] H.H. Bahar, M.H.F. Zarandi, A. Esfahanipour, Generating ternary stock trading signals using fuzzy genetic network programming, in: Fuzzy Information Processing Society (NAFIPS), 2016 Annual Conference of the North American, IEEE, 2016, pp. 1–6.
[22] H. Katagiri, K. Hirasama, J. Hu, Genetic network programming-application to intelligent agents, in: 2000 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 5, IEEE, 2000, pp. 3829–3834.
[23] D.P. Muni, N.R. Pal, J. Das, A novel approach to design classifiers using genetic programming, IEEE Trans. Evol. Comput. 8 (2) (2004) 183–196.
[24] D.J. Montana, Strongly typed genetic programming, Evol. Comput. 3 (2) (1995) 199–230.
[25] S. Mousavi, A. Esfahanipour, M.H.F. Zarandi, A novel approach to dynamic portfolio trading system using multitree genetic programming, Knowl.-Based Syst. 66 (2014) 68–81.
[26] S. Mabu, K. Hirasawa, M. Obayashi, T. Kuremoto, Enhanced decision making mechanism of rule-based genetic network programming for creating stock trading signals, Expert Syst. Appl. 40 (16) (2013) 6311–6320.
[27] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction. No. 1, MIT press, Cambridge, 1998.
[28] S. Dehghanpour, A. Esfahanipour, A robust genetic programming model for a dynamic portfolio insurance strategy, in: 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), IEEE, 2017, pp. 201–206.

[29] M. Kac, Random walk and the theory of brownian motion, Amer. Math. Monthly 54 (7) (1947) 369–391.
[30] R. Bellman, R. Kalaba, L. Zadeh, Abstraction and pattern classification, J. Math. Anal. Appl. 13 (1) (1966) 1–7.
[31] H. Dourra, P. Siy, Investment using technical analysis and fuzzy logic, Fuzzy Sets and Systems 127 (2) (2002) 221–240.
[32] M. Fasanghari, G.A. Montazer, Design and implementation of fuzzy expert system for tehran stock exchange portfolio recommendation, Expert Syst. Appl. 37 (9) (2010) 6138–6147.
[33] M.G. Yunusoglu, H. Selim, A fuzzy rule based expert system for stock evaluation and portfolio construction: An application to istanbul stock exchange, Expert Syst. Appl. 40 (3) (2013) 908–920.
[34] A. Karatahansopoulos, G. Sermpinis, J. Laws, C. Dunis, Modelling and trading the greek stock market with gene expression and genetic programing algorithms, J. Forecast. 33 (8) (2014) 596–610.
[35] A. Engelbrecht, S. Rouwhorst, L. Schoeman, A building block approach to genetic programming for rule discovery, in: Data Mining: A Heuristic Approach, 2001, pp. 174–189.
[36] A. Abraham, N. Nedjah, L. de Macedo Mourelle, Evolutionary computation: from genetic algorithms to genetic programming, in: Genetic Systems Programming, Springer, 2006, pp. 1–20.
[37] T.D. Haynes, D.A. Schoenefeld, R.L. Wainwright, Type inheritance in strongly typed genetic programming, Adv. Genet. Program. 2 (2) (1996) 359–376.
[38] T. Chen, K. Tang, G. Chen, X. Yao, A large population size can be unhelpful in evolutionary algorithms, Theoret. Comput. Sci. 436 (2012) 54–70.
[39] W. Ashlock, Using very small population sizes in genetic programming, in: IEEE Congress on Evolutionary Computation, 2006. CEC 2006, IEEE, 2006, pp. 319–326.
[40] S. Sakprasat, M.C. Sinclair, Classification rule mining for automatic credit approval using genetic programming, in: IEEE Congress on Evolutionary Computation, 2007. CEC 2007, IEEE, 2007, pp. 548–555.
[41] D. Zhang, M. Hifi, Q. Chen, W. Ye, A hybrid credit scoring model based on genetic programming and support vector machines, in: Fourth International Conference on Natural Computation, 2008. ICNC'08. Vol. 7, IEEE, 2008, pp. 8–12.
[42] S.L. Chiu, Fuzzy model identification based on cluster estimation, J. Intell. Fuzzy Systems 2 (3) (1994) 267–278.
[43] E. Lughofer, E. Klement, Online adaptation of takagi-sugeno fuzzy inference systems, in: Proceedings of CESA, 2003.