Comput Econ (2018) 52:125–144
https://doi.org/10.1007/s10614-017-9665-9

An Automated Investing Method for Stock Market Based on Multiobjective Genetic Programming

Alexandre Pimenta[1,2] · Ciniro A. L. Nametala[1,2] ·
Frederico G. Guimarães[3] · Eduardo G. Carrano[3]

Accepted: 21 February 2017 / Published online: 2 March 2017

Abstract Stock market automated investing is an area of strong interest for the academia, casual, and professional investors. In addition to conventional market methods, various sophisticated techniques have been employed to deal with such a problem, such as ARCH/GARCH predictors, artificial neural networks, fuzzy logic, etc. A computational system that combines a conventional market method (technical analysis), genetic programming, and multiobjective optimization is proposed in this work. This system was tested in six historical time series of representative assets from Brazil stock exchange market (BOVESPA). The proposed method led to profits considerably higher than the variation of the assets in the period. The financial return was positive even in situations in which the share lost market value.

Keywords Genetic programming · Multiobjective optimization · Technical analysis · Stock exchange market · Feature selection · BOVESPA

✉ Frederico G. Guimarães
fredericoguimaraes@ufmg.br

Alexandre Pimenta
alexandre.pimenta@ifmg.edu.br

Ciniro A. L. Nametala
ciniro@ifmg.edu.br

Eduardo G. Carrano
egcarrano@ufmg.br

[1] Department of Computing, Instituto Federal Minas Gerais, Formiga, MG, Brazil

[2] Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil

[3] Department of Electrical Engineering, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil

1 Introduction

The efficient market hypothesis (Fama 1970), assumed by some economists, states that a stock price behaves as a random walk. It suggests that any price forecasting attempt consists on a futile effort because if returns were forecastable, market participants would use them to generate unlimited profits. The profit seeking behavior of market agents would therefore destroy any predictability pattern that may arise in the series for some time. This characteristic makes market time series non stationary and harder to forecast, nevertheless, temporary predictable patterns may arise in financial series and can be exploited. Instead of being the end of the story for forecasting methods in stock markets, the efficient market hypothesis is on the contrary a motivation for the development of innovative adaptive financial forecasting methods, since any conventional method would quickly become unsuccessful.

Despite the difficulties in forecasting financial time series, many efforts have been spent to better understand the stock market. Atsalakis (2009) an extensive study of conventional prediction methods is performed. In this comparison ARMA (autoregressive moving average models), ARIMA (autoregressive integrated moving average), ARCH (autoregressive conditional heteroscedasticity), and GARCH (generalized autoregressive conditional heteroscedasticity) models leaded to the best results (Atsalakis 2009). However, these standard methods assume stationarity of the time series and are deemed to be unsuccessful in practice under the efficient market hypothesis.

Recently, several computational intelligence (CI) based systems have been proposed for supporting stock market investments (Atsalakis and Valavanis 2009), such as neural networks and/or fuzzy logic. Atsalakis and Valavanis (2009) present a review of 150 publications that use those techniques. In this review, the authors collect evidence that CI based systems usually obtain better results than conventional methods. Genetic programming (GP) (Poli et al. 2008) has also been used for this purpose. Computational intelligence and hybrid fuzzy methods have been also applied to forecasting financial time series (Sadaei et al. 2016; Talarposhti et al. 2016) with promising results.

This work proposes a computational system for automated investment in stock markets. The system is a combination of multiobjective optimization (Kalyanmoy et al. 2002), genetic programming (Poli et al. 2008), and technical trading rules (Murphy 1999), which are commonly employed in financial time series evaluation. Additional procedures are used to improve the quality of results, such as automatic outlier detection/removal and feature selection. Finally, the decision is taken based on an ensemble. To best of the authors knowledge, such a combination is an innovation in the literature. The proposed method is an evolutionary method combining technical trading rules in a more sophisticated way with an adaptive process. Different classifiers in the ensemble could potentially learn different temporary patterns in the data and combine this into a single decision with a majority rule, giving more reliability to the action.

The proposed algorithm was applied to identify ideal moments for sending buying and selling orders for six shares of the Brazilian stock market BOVESPA. The experiments were performed in an evaluation window of two years, between February 2013 and February 2015. The results were compared with Buy and Hold (Murphy 1999), which estimates the variation of the asset on the period, and two other automated techniques, based on the technical analysis. In addition, a second evaluation window

(July 2015 to July 2016) was also considered in order to estimate how the proposed method would work in a very critical moment of Brazilian economics and politics.

This text is structured as follows: a brief conceptual background of some techniques employed in this manuscript is presented in Sect. 2, jointly with some related works; the proposed automated investment system is described in Sect. 3; results obtained by the proposed algorithm and three benchmark methods on six assets of BOVESPA are described in Sect. 4; finally, some concluding remarks are drawn in Sect. 5.

2 Conceptual Background

A brief conceptual background of genetic programming for time series forecasting and technical analysis is given along this section. Some references that employ these techniques in related problems are also presented.

2.1 Genetic Programming and Time Series Forecasting

Genetic Programming (GP) is an evolutionary computation algorithm proposed by Koza (1992). Its structure is very similar to a Genetic Algorithm (GA): it starts from an initial population, which is evolved continuously, into a generational process, through crossover, mutation, evaluation, and selection operations. However, differently from GA in which the individuals are represented by numerical chains, each candidate solution in GP represents a computer code, encoded as a tree. This representation makes it easier to optimize decision rules using GP rather than GA. As a drawback, this structure requires additional control on crossover and mutation operators, to deal with invalid solutions.

Genetic Programming (GP) has been applied in various fields of knowledge, such as pattern recognition, data mining, function regression, decision rule generation, time series forecast, etc (Espejo et al. 2010; Barros et al. 2012; Cortez 2002; Alfaro-Cid et al. 2014; Kattan et al. 2015). Currently, it is accepted as an important area of study in machine learning, mainly because of its simplicity, robustness, and its potential of being adapted for considerably different contexts and applications (Poli et al. 2008).

In several problems, it is necessary to predict phenomenon that occurs at certain regular time intervals. These problems are known as time series forecast problems, and the biggest challenge on them is to build the most generalizable model that represents the time series function accurately. Genetic Programming has been widely employed on time series prediction problems, such as illustrated in Cortez (2002), Alfaro-Cid et al. (2014), Kattan et al. (2015).

Automated investment in financial markets can be interpreted as a non-linear and non-stationary time series prediction problem. GP has been also applied to financial time series prediction (Allen and Karjalainen 1999; Potvin et al. 2004; Myszkowski and Rachwalski 2009; Pimenta et al. 2014; Vasilakis et al. 2013; Dabhi and Chaudhary 2015).

Allen and Karjalainen (1999) used GP for generating negotiation rules. Their experiments were performed using daily candles of negotiation from 1928 to 1995 (SP 500 index). Arithmetic operators, max and min operators, rates, and conditional operators (if then else) were used to build the decision trees. Their results were found to be profitable, but they did not surpassed the strategy Buy and Hold (Murphy 1999).

Vasilakis et al. (2013) used GP to the daily prediction of the exchange rate (Euro/Dolar). The experiments were performed with daily candles from January of 1999 to October of 2009. In addition to the arithmetic operators used in Allen and Karjalainen (1999), It was included operators sqr, cube, cos, sin, tan, abs, exp, log, pow. The objective of this work eh generate an optimal mathematical expression, used for daily forecasts. Making operations daytrade, this work was compared with three traditional strategies (naive strategy, MACD and BUY and HOLD), in the three comparisons GP proposal presented better results.

Potvin et al. (2004), in addition to the operators used by Allen and Karjalainen (1999), the authors made use of some TA indicators. The experiments were performed with 14 shares listed on the Toronto Stock Exchange. The experiments were performed with daily candles from June 30th 1992 to June 30th 2000. The results were better when the shares were lateralized or falling. With the booming market, the Buy and Hold strategy was better.

Myszkowski and Rachwalski (2009), unlike Allen and Karjalainen (1999), Potvin et al. (2004), used two decision trees, one for buying and another for selling. Their experiments were performed at FOREX (Foreign Exchange) market trading Euro/US. Ten minute candles were used for purchase and sale operations. The decision trees combine logical operators (And, Or) and technical trading rules of TA. The results were unstable, but with positive return for individuals (trees) with height 5 and 6.

Pimenta et al. (2014) followed a similar proposal to the work of Myszkowski and Rachwalski (2009). The generated decision trees combine logical operators with TA indicators. The experiments were performed using shares from the Brazilian stock market BOVESPA, using daily candles from February 24th 2010 to February 28th 2014, with promising results. The financial goal (10% profit) was reached in 90% of the cases. The main criticism for this work is the no inclusion of financial loss in the limit. When compared to Myszkowski and Rachwalski (2009), Pimenta et al. (2014) differs essentially in three points: the negotiation system, the monitoring system, and the use of financial goals.

Dabhi and Chaudhary (2015) proposes a combination of wavelet and Postfix-GP, a postfix notation based genetic programming system, for financial time series prediction. The discrete wavelet transform approach is used to smoothen the time series by separating the fluctuations from the trend of the series. Their experiments were performed at four financial time series, two stock price series (Intel and Microsoft) and two stock market indexes series (NASDAQ Composite and S&P CNX Nifty) problems. It was used candles Daily, for periods ranging from September 12th 2007 to November 11th 2010 to (Intel and Microsoft), March 1th 2007 to March 22th 2011 to (NASDAQ), March 1th 2007 to April 9th 2001 to (S&P CNX). The result of this work was compared with genetic programming used in ECJ (Luke et al. 2004). The tests indicated that the proposed work was better in the two indices (NASDAQ e S&P CNX), and not come to a conclusive answer to the roles of companies (Intel e Microsoft).

In this work, the GP was used to generate purchase and sale rules for the Brazilian stock exchange market, BOVESPA. The algorithm presented in Pimenta et al. (2014) was a very preliminary version of the method proposed in this work. When compared to the former reference, the following improvements were introduced: (i) inclusion of an approach of multi-objective optimization, with the purpose to avoid overfitting; (ii) tests with positive results for the three types of financial series, downtrend, uptrend and lateralization; (iii) tests with better results than strategy Buy and Hold, even in series with high trend (iv) limits if financial loss. The particularities of the algorithm employed in this work are described in Sect. 3.3. For additional information about GP and its many variants, please refer to Koza (1992), Barros et al. (2012), Espejo et al. (2010), Cortez (2002), Alfaro-Cid et al. (2014), Kattan et al. (2015).

2.2 Technical Analysis

According to Murphy (1999), "the technical analysis can be defined as the study of prices, volumes, and open contracts of the market, in order to predict future price trends". This approach is focused on the movement of prices among themselves and not in the causes behind such variations. Technical analysis can be either classic, based on the identification of graphical patterns, or computational, which is focused on the evaluation of numerical indicators (Elder 1993).

Nowadays, it is possible to identify around fifty TA key indicators, which can be divided into four main categories (Elder 1993; Murphy 1999): trend followers, oscillators, band systems, and divergence identifiers. The main features of these categories are Murphy (1999):

- Trend followers: the indicators of this category identify the main movements of the asset prices at a certain period. Examples: Simple Moving Average, Exponential Moving Average, Donchian Channel, Hilo Activator.
- Oscillators: the indicators of this category monitor the price variations of the asset in a certain range in order to identify possible reverse points. Examples: Chaikin Oscillator, Oscillator Chaikin Volatiliy, Williams R.
- Band systems: band systems are constituted of three curves drawn around the prices. These curves are drawn from a particular distance of a moving average. The intermediate band is usually a simple moving average, and the intervals between the bands are determined by price volatility. When there is no defined trend, the rule is to sell when the price is above the upper band and buy when the price falls below the lower band. Examples: Keltner Channel, Bollinger Bands, Bollinger Oscillator.
- Divergence Identifier: these indicators are based on the principle that the whole trend goes through corrections. The divergences occur when comparing the behavior of the indicator in relation to the price movement of an asset. Examples: Accumulation and Distribution, On Balance Volume, Relative Strength Index.

All these TA indicators can be evaluated based on previous values of time series, such as asset value in market opening or closing, minimum and maximum values along the day, trading volume, etc.

In this work, the TA indicators are used as function nodes of GP for building purchase and sale decision rules.

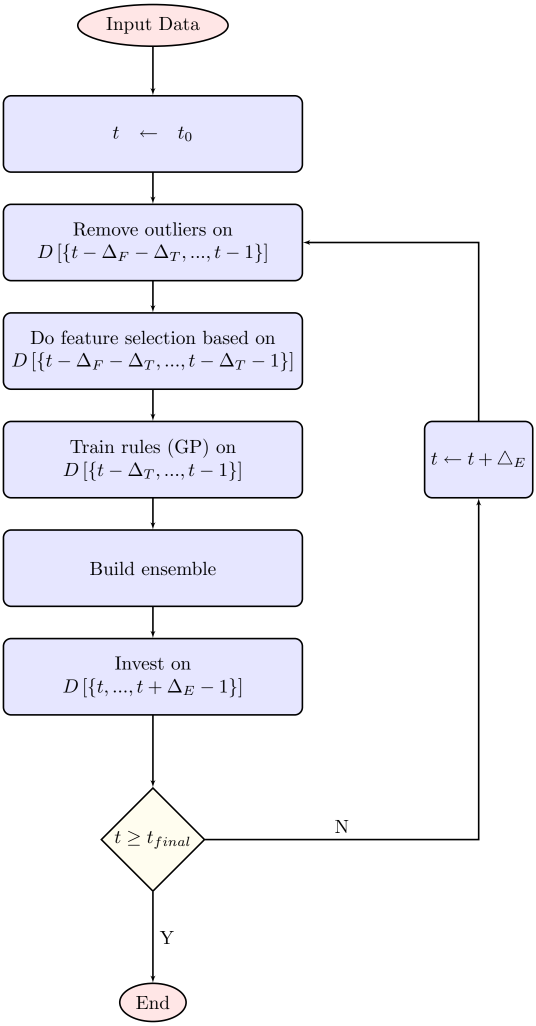
Input Data
$t \leftarrow t_0$
Remove outliers on $D\left[\{t-\Delta_F-\Delta_T,...,t-1\}\right]$
Do feature selection based on $D\left[\{t-\Delta_F-\Delta_T,...,t-\Delta_T-1\}\right]$
Train rules (GP) on $D\left[\{t-\Delta_T,...,t-1\}\right]$
Build ensemble
Invest on $D\left[\{t,...,t+\Delta_E-1\}\right]$
$t \geq t_{final}$
N
Y
End
$t \leftarrow t+\triangle_E$

Fig. 1 General scheme of the proposed automated investment system. In this scheme: $D[\{t_I, \ldots, t_F\}]$ is the financial data series between start time t_I and final time t_F; $\Delta_F > 0$, $\Delta_T > 0$, and $\Delta_E > 0$ are the sizes of feature selection, training, and evaluation windows, respectively

3 Proposed Automated Investment System

The general scheme of the proposed automated investment system is shown in Fig. 1. It is possible to note that the system works in a sliding window loop, which stops when the target date (t_{final}) is reached.

Three time windows are considered in each iteration, in sequence: the feature selection window $\{t - \Delta_F - \Delta_T, \ldots, t - \Delta_T - 1\}$ (past data), the training window $\{t - \Delta_T, \ldots, t - 1\}$ (past data), and the evaluation window $\{t, \ldots, t + \Delta_E - 1\}$ (forecast window). The proposed system can be summarized into the following five main operations that are performed in each iteration:

1. Remove outliers on $D\left[\{t - \Delta_F - \Delta_T, \ldots, t - 1\}\right]$: outliers are identified and removed from feature selection and training windows.

2. Do feature selection based on $D\,[\{t - \Delta_F - \Delta_T, \ldots, t - \Delta_T - 1\}]$: the candidate attributes are evaluated on the feature selection window. The best attributes are considered in training phase, while the remaining ones are ignored in the current iteration.
3. Train rules (GP) on $D\,[\{t - \Delta_T, \ldots, t - 1\}]$: in this step, the GP is applied to the training interval in order to build purchase rules and sale rules.
4. Build ensemble: the rules obtained by GP are used to build an ensemble, which will take the decisions on investment step.
5. Invest on $D\,[\{t, \ldots, t + \Delta_E - 1\}]$: finally, the ensemble is employed to identify if one of three operations is performed at the end of each day in the evaluation window: (i) to buy; (ii) to sell, or; (iii) to stay.

Each one of these steps is described with more details in the next subsections.

3.1 Outlier Detection and Removal

According to the literature, outliers are defined with respect to a supposed underlying distribution or a theoretical model. If these change, the observation may be no more outlying (Yadolah 2008). In other words, “outliers are observations that do not follow the pattern of the majority of the data” (Cook and Hawkins 1990).

In financial time series, outliers are abnormal variations of the share price, which are often an indication of an external factor that could not be modelled without privileged information. These values can be a serious problem in training phase, since they can lead to distortions in the evaluation of the candidate buying and selling rules.

A mechanism for automatic detection and removal of outliers is proposed in this work. This mechanism, which is based on Locally Weighted Scatterplot Smoothing (LOWESS) (Cleveland 1981), works as follows:

1. Build a smooth approximation of the share price curve using LOWESS.
2. Evaluate the differences between the observed data and the respective point on the smooth approximation.
3. Set the maximum acceptable negative difference as $MaxNeg \leftarrow Q_1 - F_{low}(Q_3 - Q_1)$, in which Q_1 and Q_3 are the first and third quartiles of the differences, and F_{low} is a scale parameter set by the user.
4. Set the maximum acceptable positive difference as $MaxPos \leftarrow Q_3 + F_{low}(Q_3 - Q_1)$.
5. Remove any training point whose difference (dif) lies in one of the following conditions: $dif < MaxNeg$ or $dif > MaxPos$.

An example of the application of such procedure is shown in Fig. 2, for BBAS3 stock price. It is possible to note that it filters sudden variations of the stock price.

3.2 Feature Selection

Twelve technical analysis indicators were considered as candidate attributes for genetic programming, such as shown in Table 1. These indicators were chosen because they

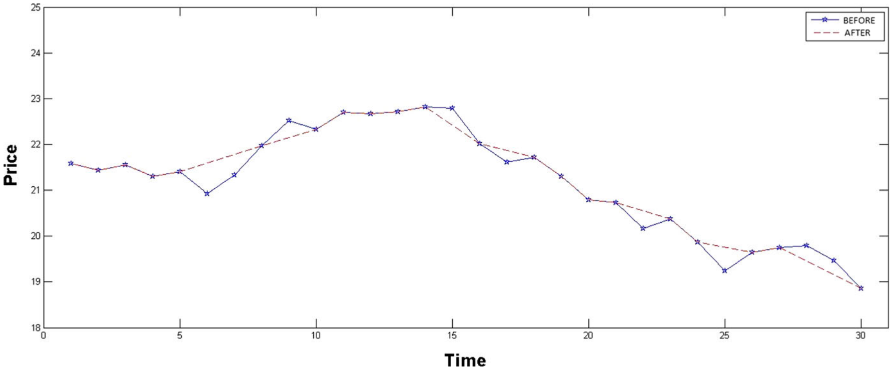

Fig. 2 Example of data (BBAS3 between May 7th 2013 and June 19th 2013) before (*black continuous line*) and after (*red dashed line*) outlier detection and removal procedure. (Color figure online)

Table 1 Technical analysis indicators considered

Indicator	Name	Category
SMA	Simple Moving Average	Trend follower
EMA	Exponential Moving Average	Trend follower
CO	Chaikin Oscillator	Oscillator
OCV	Oscillator Chaikin Volatility	Oscillator
WR	Williams R	Oscillator
VO	Volume Oscillator	Oscillator
PO	Price Oscillator	Oscillator
BB	Bollinger Bands	Band system
AD	Accumulation and Distribution	Divergence identifier
OBV	On Balance Volume	Divergence identifier
RSI	Relative Strength Index	Divergence identifier
S	Stochastic	Divergence identifier

are representative for the main groups of TA indicators: trend followers, oscillators, systems of bands and divergence identifiers.

Logical purchase and sale rules, with different parametrizations, were built with combinations of the TA indicators. These rules, which are shown in Table 2, should be read as the following examples:

- BUY :: $SMA_{21}(i) > SMA_{21}(i-3)$: if the SMA evaluated for 21 days ending today is higher than the SMA evaluated for 21 days ending three days before, then buy the share (up trend).
- SELL :: $RSI(i) > 80$: if the RSI is higher than 80, then sell the share since it seems to be overbought.

From Table 2, it is possible to note that both purchase and sale rules can be built with 36 different logical candidate functions each. It would imply an extremely large

Table 2 Logical rules based on TA indicators

Type	Rule	Parameters
BUY	$SMA_P(i) > SMA_P(i-3)$	$P \in \{8, 9, 21\}$
BUY	$EMA_P(i) > EMA_P(i-3)$	$P \in \{8, 9, 21\}$
BUY	$And(CO_P(i-1) < 0, CO_P(i) > 0)$	$P \in \{10, 15, 20\}$
BUY	$OCV_P(i) < 0$	$P \in \{10, 15, 20\}$
BUY	$WR_P(i) < -90$	$P \in \{8, 9, 21\}$
BUY	$VOL_P(i) > 300$	$P \in \{8, 14, 21\}$
BUY	$PO_P(i) < -10$	$P \in \{8, 12, 21\}$
BUY	$And(AssetPrice(i-1) < BB_P(i-1), AssetPrice(i) > BB_P(i))$	$P \in \{14, 20, 24\}$
BUY	$And(AD(i) > AD(i-P), AssetPrice(i) > AssetPrice(i-P))$	$P \in \{1, 2, 3\}$
BUY	$And(OBV(i) > OBV(i-P), AssetPrice(i) > AssetPrice(i-P))$	$P \in \{1, 2, 3\}$
BUY	$RSI(i) < P$	$P \in \{10, 20, 30\}$
BUY	$S(i) < P$	$P \in \{10, 15, 20\}$
SELL	$SMA_P(i) < SMA_P(i-3)$	$P \in \{8, 9, 21\}$
SELL	$EMA_P(i) < EMA_P(i-3)$	$P \in \{8, 9, 21\}$
SELL	$And(CO_P(i-1) > 0, CO_P(i) > 0)$	$P \in \{10, 15, 20\}$
SELL	$OCV_P(i) > 60$	$P \in \{10, 15, 20\}$
SELL	$WR_P(i) > -10$	$P \in \{8, 9, 21\}$
SELL	$VOL_P(i) > 300$	$P \in \{8, 14, 21\}$
SELL	$PO_P(i) > 10$	$P \in \{8, 12, 21\}$
SELL	$And(AssetPrice(i-1) > BB_P(i-1), AssetPrice(i) < BB_P(i))$	$P \in \{14, 20, 24\}$
SELL	$And(AD(i) < AD(i-P), AssetPrice(i) < AssetPrice(i-P))$	$P \in \{1, 2, 3\}$
SELL	$And(OBV(i) < OBV(i-P), AssetPrice(i) < AssetPrice(i-P))$	$P \in \{1, 2, 3\}$
SELL	$RSI(i) > P$	$P \in \{70, 80, 90\}$
SELL	$S(i) > P$	$P \in \{80, 85, 90\}$

search space, which could slow the GP convergence significantly. A feature selection procedure was proposed to mitigate such a problem, as follows:

1. For each logical rule i in Table 2:
   (a) Make $Correct(i) \leftarrow 0$.
   (b) For each day j in feature selection window ($j \in \{t - \Delta_F - \Delta_T, \ldots, t - \Delta_T - 1\}$):
      i. If i is a buying rule:
         A. If rule i indicates "*true*" and the share price increased at day $j+1$, then make $Correct(i) \leftarrow Correct(i) + 1$.
         B. If rule i indicates "*false*" and the share price did not increase at day $j+1$, then make $Correct(i) \leftarrow Correct(i) + 1$.
      ii. If i is a selling rule:
         A. If rule i indicates "*true*" and the share price did not increase at day $j+1$, then make $Correct(i) \leftarrow Correct(i) + 1$.

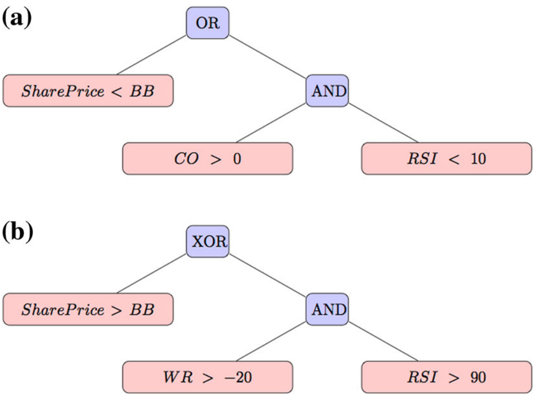

Fig. 3 Example of GP individual. **a** Buying rule. **b** Selling rule

Table 3 Individual decision making

Purchase rule	Sale rule	Decision
False	False	Stay
True	False	Buy
False	True	Sell
True	True	Stay

   B. If rule i indicates "*false*" and the share price increased at day $j + 1$, then make $Correct(i) \leftarrow Correct(i) + 1$.

2. Select the N_a rules with higher $Correct$ values for buying, and the N_a rules with higher $Correct$ values for selling (N_a is a parameter set by the user).

This process reduces the cardinality of the search space explored by GP. It is applied to the feature selection window, which is located immediately before the training window. Finally, it is important to remark that feature selection and training are not performed in the same time window because it would lead to data overfitting.

3.3 Genetic Programming Algorithm

The investment rules of the proposed automated system are generated using a genetic programming algorithm. In this algorithm, each individual encodes two logical rules, one for purchasing and one for selling shares. These rules are formed by the attributes (AT based rules) that survived after feature selection and the logical operators AND, OR and XOR. An example of a candidate individual is shown in Fig. 3.

It should be noticed that this encoding scheme is very interesting, because it makes the implementation of crossover and mutation operators straightforward: since all nodes are logical, any node operation replacement operation leads to a valid solution.

Each individual in the population must take one of three possible decisions at a time: (i) to buy the share; (ii) to sell the share, or; (iii) to stay in the current state. This decision is made based on the signals emitted by purchase and sales rules, such as shown in Table 3.

The training of rules was modeled as a bi-objective optimization problem, whose objectives are: (i) to maximize the financial return (R), and; (ii) to minimize the classifier complexity (C). The first function is simply the accumulated profit in the training time window, as shown in Eq. (1). In this equation, $P(d)$ is the profit at day d. Obviously, if $P(d) > 0$, it represents a profit, and if $P(d) < 0$, it represents a loss.

$$R = \sum_{d=t-\Delta_T}^{t-1} P(d) \tag{1}$$

The second objective is intended to reduce the probability of overfitting. According to the *Bias-Variance Dilemma*, more complex classifiers have more structural flexibility to accurately model the training data, often presenting small error on the training set, but poor generalization performance (Abbass 2001). Nowadays, the control of complexity of regressors and classifiers is an usual practice. In the proposed algorithm, the complexity was measured as the maximum depth[1] in the trees representing purchase and sale rules. This objective is shown in Eq. (2). In this equation D_B and D_S are the depths of buying and selling rules, respectively.

$$C = \max(D_B, D_S) \tag{2}$$

The GP was implemented using the Grow method for generating initial solutions, point crossover, and shrink mutation (Koza 1992). The selection is performed using the Crowding Binary Tournament and the Fast Non-dominated Sorting, such as they were proposed in the Non-dominated Sorting Genetic Algorithm 2 (NSGA2) (Kalyanmoy et al. 2002).

3.4 Ensemble

An ensemble is a decision committee in which the votes of several "judges" are combined in order to reach a final decision. The idea behind this method is to take advantage of the good local behavior of each of the judges, in order to increase the accuracy and the reliability for the global scenario.

This kind of approach is widely employed in computational intelligence, specially for neural networks (Hansen and Salamon 1990; Perrone and Cooper 1992; Opitz and Shavlik 1996). In these cases, several classifiers, usually neural networks with different topologies and/or parameters, are employed to classify the same input pattern, and their votes are combined using some specific rule, such as majority, arithmetic mean, weighted average, etc.

In this work, the decision committee is formed by the individuals that compose the final Pareto-set approximation delivered by the genetic programming algorithm. The final decision is taken based on majority, i.e. the operation (to buy, to sell, or to stay) with more votes is executed.

[1] The depth of a tree is the length of the longest path between the root and the leaves (Cormen et al. 2009).

Based on the training problem formulation, it is reasonable to accept that the Pareto-set approximation is composed of rules varying from very simple (underfitted) to very complex ones (overfitted). It is expected that their combination indicates unbiased decisions. Finally, it is important to remark that such a strategy was tested against several single rule decision approaches, and it always leaded to better results.

3.5 Trading Module

The last component of the proposed method is the trading module, which is responsible to execute the purchase and sale orders on the evaluation window. This module was built based on six main features:

- The decisions are based on ensemble outputs.
- It is possible to make share rentals, if necessary.
- The available budget can be leveraged by a constant factor F_{lev}.
- Purchase and sale operations are performed at market close.
- It stops trading when a maximum loss $MaxLoss$ or a maximum gain $MaxGain$ is reached (stop loss and stop gain conditions).
- The position is closed at the end of the evaluation window.

A share rental operation occurs when the system indicates a sale, but the share is not owned. In this case, the share is rented and sold, and the return of the operation is used to cover the rental in a near future. This type of operation is often referred as short selling (Murphy 1999; Elder 1993).

The system can also leverage[2] the available financial resources in F_{lev} percent through debt. This is also a common practice in the stock markets that is used to increase profits (Murphy 1999; Elder 1993). However, it also increases the operation risks.

The trading module is applied at each day of the evaluation window. After each run, the system should be in one of four states: (i) out of market (initial); (ii) bought; (iii) sold, or; (iv) out of market (close). The actions performed vary accordingly to the current state, as follows:

Out of market (initial): this is the state in the first day of the evaluation window. The state of the system on the next day will be defined by the action indicated by the ensemble, as follows:

- To stay: the system will remain on out of market (initial) state.
- To buy: it will spend all available money on share acquisition, moving to bought state.
- To sell: it will rent the share and sell it, moving to sold state.

Bought: in this state, the ensemble will indicate one of two possible transitions:

- To stay/To buy: the system will remain on bought state.
- To sell: it will sell all available shares before and will move to sold state.

[2] In finance, leverage is the general term for any technique that is used to multiply the profitability through debt.

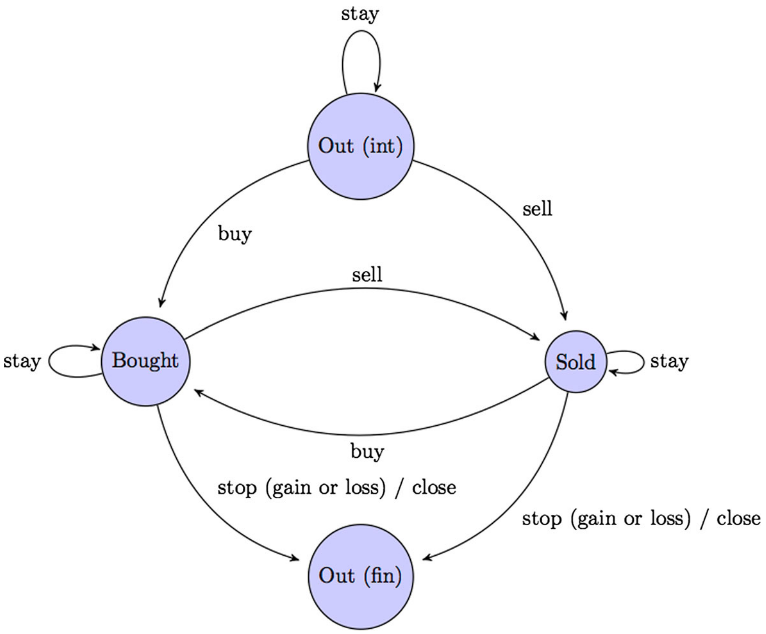

Fig. 4 Transition diagram of the trading module. In this figure, Out (int) and Out (fin) represent the Out of market (*initial*) and Out of market (*final*) states

In addition, the system can move to out of market (final) state if one of the following conditions becomes valid: (i) stop loss is reached; (ii) stop gain is reached, or; (iii) it is at the last day of the evaluation window.

Sold: in this state, the ensemble will indicate one of the following transitions:

- To stay/To sell: the system will remain on sold state.
- To buy: it will spend all available money on share acquisition.

The system can move to out of market (final) state under the same conditions related for bought state.

Out of market (final): this is the only state that is not reached through the ensemble output. The system moves to this state under three situations:

- When the evaluation window finishes.
- If the stop loss condition is reached.
- If the stop gain condition is triggered.

The system will remain on this state until the end of the evaluation window. A transition diagram of the proposed trading module can be seen in Fig. 4.

This system was designed assuming three premises:

1. The purchase/sale operations are always executed.
2. The operation itself does not affect the current share price.
3. The operation costs are ignored.

Table 4 Share descriptions

Share	Company	Sector	Trend	Website
BBAS3	Banco do Brasil S.A	Finance	Lateralization	www.bb.com.br
BOVA11	ETF	Finance	Low	www.bmfbovespa.com.br
CMIG4	CEMIG S.A	Energy	Lateralization	www.cemig.com.br
EMBR3	EMBRAER S.A	Industry	High	www.embraer.com
GBR4	GERDAU S.A	Industry	Low	www.gerdau.com
VALE5	VALE S.A	Commodite	Low	www.vale.com

Although these premises may sound too restrictive, it is possible to adopt some strategies in order to make them valid. At first, there are times of the day in which the share market has more liquidity, i.e. it is often possible to complete a buy or a sell operation at this time. In addition, shares from bigger companies have more liquidity and are less affected by small or medium orders. Finally, the operation costs might not be significant if the number of operations is small.

These aspects justify some choices adopted in this work. We perform the operations at market close because this is the time of day in which BOVESPA has more liquidity. We chose shares whose impact of small and medium orders is almost insignificant. Finally, the system works in swing trade, which reduces the number of operations considerably.

It is important to emphasize that the trading module is the only part of the proposed system in which such premises are assumed. Therefore, if one of them is not valid in a given situation, it is possible to maintain the remaining parts of the method unchanged and to modify only the trading module.

4 Results

The proposed automated investment system was applied to six shares (BBAS3, BOVA11, CMIG4, EMBR3, GGBR4, and VALE5) from BOVESPA, in a test window of 514 working days, between 02.05.2013 and 02.02.2015. These assets were selected based on the following criteria:

- They have impact in the BOVESPA index and they present liquidity for purchase and sale operations in daily candles.
- They are also traded in the New York Stock Exchange, through the American Depositary Receipts (ADRs), which ensures more liquidity to the assets.
- They cover four different sectors of the Brazilian economy: commodities, energy, finance, and industry.
- The series presented diverse behaviors in the test period: BOVA11, GGBR4, and VALE5 were in a downtrend; BBAS3, and CMIG4 presented lateralization, and; EMBR3 was in an uptrend.

A brief description of such shares is given in Table 4.

Table 5 Parameters adopted on the proposed automated system

Parameter	Value
Δ_F (in days)	30
Δ_T (in days)	30
Δ_E (in days)	10
F_{low}	0.3
N_a	10
Population size	60
Crossover ratio (per pair)	0.90
Mutation ratio (per individual)	0.40
Number of generations	60
F_{lev}	100%
$MaxLoss$	6%
$MaxGain$	12%

The proposed method was set with the parameters shown in Table 5. These parameters were chosen based on preliminary tests performed on test windows earlier than February 2013.

The proposed method was compared with other three approaches:

Buy and Hold (B&H): this is a passive investment strategy in which an investor buys stocks and holds them for a long period of time, regardless market fluctuations. An investor who employs a buy-and-hold strategy actively selects stocks, but once in a position, is not concerned with short-term price movements.

Best logical TA based rule committee (TA-10): this strategy also works in a sliding window loop. It uses the same feature selection, ensemble, and trading modules proposed in this work. In each iteration, the feature selection module is employed for indicating the N_a buying and the N_a selling TA based logical rules that are the most suitable for the current interval. These rules are then used to build the ensemble, which takes the decisions on the evaluation window. The training window is not considered in this approach, since GP is not employed. In this work we assumed $N_a = 10$.

All logical TA based rule committee (TA-36): this strategy is very similar to the previous one. The only difference is that feature selection is not employed. Therefore, the ensemble is built with all 72 logical TA based rules.

The financial returns obtained by the proposed method and the three reference ones are shown in Table 6. The values reported for the proposed algorithm are based on five independent runs and they are expressed using an $average\% \pm standard\ error\%$ notation. The performances of the other methods are expressed as single values because they are deterministic (i.e. they always obtain the same results when applied to the same time window).

From Table 6, it is possible to note that the proposed method clearly outperformed the other methods. The financial returns obtained by the automated system were always positive and considerably above the variation of the share in the same period (B&H

Table 6 Financial return obtained by the proposed method and the three reference ones

Share	Proposed	B&H (%)	TA-10 (%)	TA-36 (%)
BBAS3	77.61% ± 3.16%	0.60	6.17	37.77
BOVA11	33.37% ± 2.80%	−20.44	10.03	8.35
CMIG4	54.19% ± 5.78%	10.83	29.42	21.09
EMBR3	71.62% ± 6.14%	46.57	13.49	−38.71
GGBR4	48.50% ± 2.38%	−44.37	0.78	−5.79
VALE5	39.88% ± 4.62%	−48.75	−22.10	3.63
Portfolio	54.20%	−9.26	6.30	4.39

results). It is important to note that the method was able to obtain significant profit even in situations in which the asset depreciated considerably, such as BOVA11, GGBR4, and VALE5. A direct comparison with TA-10 strategy shows that the GP had an important impact on results: the average difference between the methods vary from 23 to 70 percentage points depending on the share. The comparison of TA-10 and TA-36 suggests that the feature selection module also has a positive impact on final results: the TA-10 strategy is better than the TA-36 strategy in 5 out of 6 shares.

The evolution of a theoretical investment portfolio composed of the six shares (with the initial budget being divided equally between the shares) is indicated in the last line of the table (Portfolio label). It is possible to note that the proposed method led to a significant profit, while the other methods obtained returns lower than non-risk applications (≈10%).

The statistical comparison approach proposed in Carrano et al. (2011) was applied to compare the financial returns obtained by the proposed method with the ones observed for the three reference methods. The hypotheses were tested using one sample Student T-tests and the significance was corrected using Bonferroni Correction (18 comparisons are performed). Under the confidence level of 99%, it is possible to assume that the differences observed between the proposed approach and the reference methods are statistically relevant, since the biggest p-value observed on comparisons was lower than 10^{-4} (the null hypothesis could be rejected for p-values lower than 5.56×10^{-4}).

The price evolution of the shares versus the performance of the proposed system can be seen in Figs. 5 (BBAS3), 6 (BOVA11), 7 (CMIG4), 8 (GGBR4), 9 (EMBR3), and 10 (VALE5). From these figures it is possible to note that the proposed system works well when the behavior of the share is well defined (uptrend or downtrend). When there is an inversion on trend, the system usually takes some time to respond to such a change but, after accommodation, it leads to positive return again.

A second experiment was carried out to evaluate the proposed system under crisis situations. This experiment considered the evaluation window between 07/07/2015 and 07/14/2016 (260 working days). In this period, Brazil went through a very turbulent economical and political crisis. In economical terms, the Brazilian currency (Real) lost around 50% of its value with regard to dollar since 2014. In the politics scenario, the recently elected president, Dilma Roussef, suffered a long impeachment process,

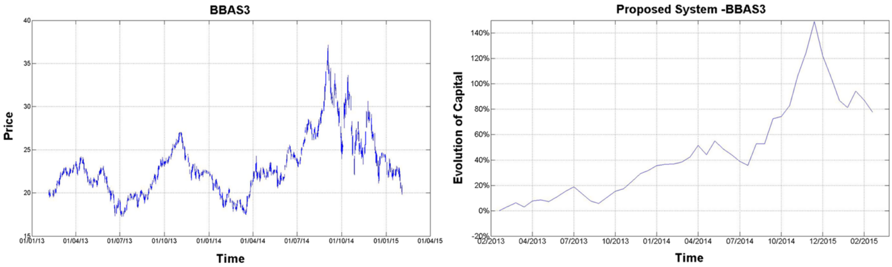

Fig. 5 BBAS3—price evolution versus performance of the proposed system (02.05.2013–02.02.2015)

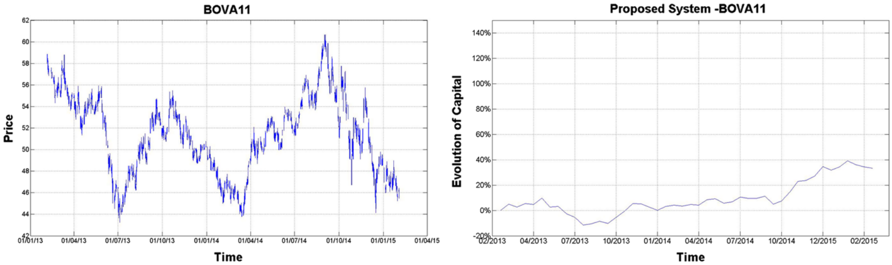

Fig. 6 BOVA11—price evolution versus performance of the proposed system (02.05.2013–02.02.2015)

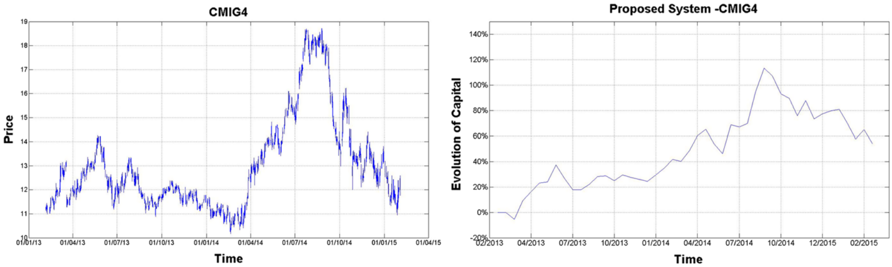

Fig. 7 CMIG4—price evolution versus performance of the proposed system (02.05.2013–02.02.2015)

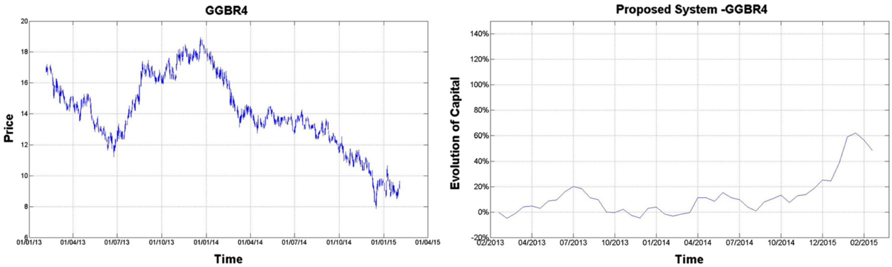

Fig. 8 GGBR4–price evolution versus performance of the proposed system (02.05.2013–02.02.2015)

being deposed from Brazilian government in August 2016 (Processo de impeachment de dilma rousseff 2016). Obviously, this scenario imposed strong oscillation to the financial market, with an overall downtrend on BOVESPA shares.

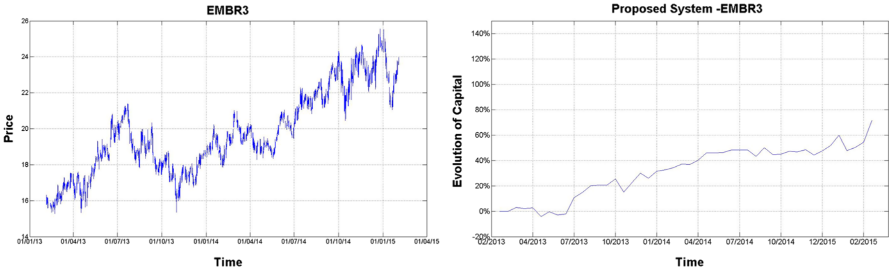

Fig. 9 EMBR3—price evolution versus performance of the proposed system (02.05.2013–02.02.2015)

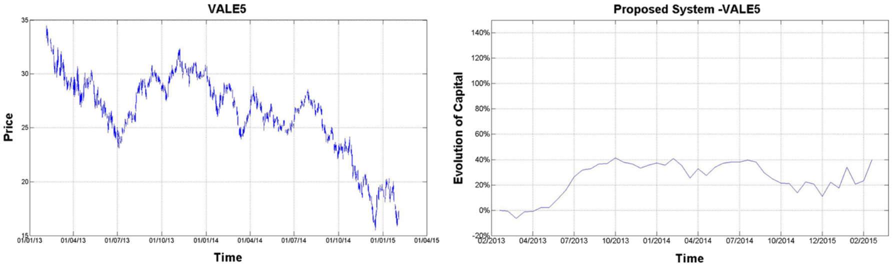

Fig. 10 VALE5—price evolution versus performance of the proposed system (02.05.2013–02.02.2015)

Table 7 Financial return obtained by the proposed method and the three reference ones (crisis period)

Share	Proposed	B&H (%)	TA-10 (%)	TA-36 (%)
BBAS3	19.83% ± 2.72%	−18.33	5.73	11.13
BOVA11	3.16% ± 0.15%	5.69	9.87	−1.03
CMIG4	23.33% ± 4.61%	−22.42	8.76	3.47
EMBR3	14.36% ± 1.44%	−23.62	12.18	6.43
GGBR4	−2.11% ± 1.99%	−3.28	4.51	3.92
VALE5	50.58% ± 6.18%	−8.78	2.03	−3.14
Portfolio	18.19%	−11.79	7.18	3.46

The results obtained by the proposed algorithm and the other benchmark methods in this second evaluation window are shown in Table 7. The proposed algorithm outperformed B&H in five, TA-10 in four, and TA-36 in five out of six shares. In addition, the average return of the theoretical investment portfolio was considerably higher than the other methods and above low-risk applications.

5 Conclusion

An automated system for investing in stock markets is proposed in this work. This system combines multiobjective optimization, genetic programming, technical analysis and feature selection in order to identify suitable moments for executing buying and selling orders. Up to the authors knowledge, such a combination is unique in the literature.

The system was tested in six BOVESPA shares (BBAS3, BOVA11, CMIG4, EMBR3, GGBR4, and VALE5) for two periods: (i) February 2013 to February 2015; (ii) July 2015 to July 2016. The results achieved were very promising. The system obtained financial returns considerably above the stock variation price on the same period, and it outperformed other two automated investment strategies. In addition, it was able to obtain significant profit even in situations of strong depreciation of the asset, which is a remarkable result.

Another contribution of this work is the study of the BOVESPA stock market. In Atsalakis and Valavanis (2009) the authors cite more than 100 works related with stock market investment, but only one is applied to Brazilian market. However, it is important to emphasize that the proposed method could be easily adopted for other financial markets, without further changes. The methods employed (technical analysis, outlier filtering, feature selection, genetic programming, and ensembles) are not based on the specifics of any stock market, and can be easily applied to other kinds of financial time series. The authors believe that the only required changes would be on the method parameters (Table 5). Adjusting StopGain and StopLoss parameters should receive special attention, taking into account that the values used here are tailored for the Brazilian market, which presents high volatility.

Acknowledgements The authors would like to thank the Brazilian agencies CAPES, CNPq, and FAPEMIG for the financial support. Funding was provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico and Fundação de Amparo à Pesquisa do Estado de Minas Gerais.

References

Abbass, H. A. (2001). A memetic pareto evolutionary approach to artificial neural networks. *Lecture Notes in Artificial Intelligence*, *2256*, 1–12.

Alfaro-Cid, E., Sharman, K., & Esparcia-Alcázar, A. I. (2014). Genetic programming and serial processing for time series classification. *Evolutionary Computation*, *22*(2), 265–285.

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

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying Stock Market Forecasting Techniques - Part I: Conventional Methods. *Journal of Computational Optimization in Economics and Finance*, *2*(1), 45–92.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques—Part ii: Soft computing methods. *Expert Systems with Applications*, *36*(3), 5932–5941. Times Cited: 53 54.

Barros, R. C., Basgalupp, M. P., De Carvalho, A. C. P. L. F., & Freitas, A. A. (2012). A survey of evolutionary algorithms for decision-tree induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *42*(3), 291–312.

Carrano, E. G., Wanner, E. F., & Takahashi, R. H. C. (2011). A multi-criteria statistical based comparison methodology for evaluating evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *15*, 848–870.

Cleveland, W. S. (1981). *Lowess: A program for smoothing scatterplots by robust locally weighted regression*. London: American Statistician.

Cook, R. D., & Hawkins, D. M. (1990). Unmasking multivariate outliers and leverage points: Comment. *Journal of the American Statistical Association*, *85*(411), 640–644.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). Cambridge: MIT Press.

Cortez, P. A. R. (2002). *Modelos Inspirados na Natureza para a Previsão de Séries Temporais. 2002. 188 f.* PhD thesis, Tese (Doutorado em Informática)–Departamento de Informática, Universidade do Minho, Braga.

Dabhi, V. K., & Chaudhary, S. (2015). Financial time series modeling and prediction using postfix-gp. *Computational Economics*, 1–35.

Elder, A. (1993). *Trading for a living: Psychology, trading tactics, money management* (Vol. 31). New York: Wiley.

Espejo, P. G., Ventura, S., & Herrera, F. (2010). A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *40*(2), 121–144.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *12*(10), 993–1001.

Kalyanmoy, D., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Kattan, A., Fatima, S., & Arif, M. (2015). Time-series event-based prediction: An unsupervised learning framework based on genetic programming. *Information Sciences*, *301*, 99–123.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection* (Vol. 1). Cambridge: MIT Press.

Luke, S., Panait, L., Balan, G., Paus, S., Skolicki, Z., Popovici, E., et al. (2004). *A java-based evolutionary computation research system* (**online**).http://cs.gmu.edu/~eclab/projects/ecj.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Harmondsworth: Penguin.

Myszkowski, P. B., & Rachwalski, Ł. (2009). Trading rule discovery on warsaw stock exchange using coevolutionary algorithms. In *Proceedings of the international multiconference on computer science and information technology* (vol. 3, pp. 81–88).

Opitz, W. D., & Shavlik, J. W. (1996). Actively searching for an effective neural network ensemble. *Connection Science*, *8*(3–4), 337–354.

Perrone, M. P., & Cooper, L. N. (1992). *When networks disagree: Ensemble methods for hybrid neural networks*. Technical report, DTIC Document.

Pimenta, A., Carrano, E. G., Guimaraes, F. G., Nametala, L., Aparecido, C., & Takahashi, R. H. C. (2014). Goldminer: A genetic programming based algorithm applied to brazilian stock market. In *2014 IEEE symposium on computational intelligence and data mining (CIDM)* (pp. 397–402). IEEE.

Poli, R., Langdon, W. B., McPhee, N. F., & Koza, J. R. (2008). *A field guide to genetic programming*. Raleigh: Lulu.com.

Potvin, J.-Y., Soriano, P., & Vallée, M. (2004). Generating trading rules on the stock markets with genetic programming. *Computers & Operations Research*, *31*(7), 1033–1047.

Processo de impeachment de dilma rousseff. (2016). http://pt.wikipedia.org/wiki/Processo_de_impeachment_de_Dilma_Rousseff/.

Sadaei, H. J., Enayatifar, R., Guimaraes, F. G., Mahmud, M., & Alzamil, Z. A. (2016). Combining ARFIMA models and fuzzy time series for the forecast of long memory time series. *Neurocomputing*, *175*, 782–796.

Talarposhti, F. M., Sadaei, H. J., Enayatifar, R., Guimaraes, F. G., Mahmud, M., & Eslami, T. (2016). Stock market forecasting by using a hybrid model of exponential fuzzy time series. *International Journal of Approximate Reasoning*, *70*, 79–98.

Vasilakis, G. A., Theofilatos, K. A., Georgopoulos, E. F., Karathanasopoulos, A., & Likothanassis, S. D. (2013). A genetic programming approach for EUR/USD exchange rate forecasting and trading. *Computational Economics*, *42*(4), 415–431.

Yadolah, D. (2008). *The concise encyclopedia of statistics*. Berlin: Springer.