

Designing Strategies for Autonomous Stock Trading Agents using a Random Forest Approach

Monira Aloud

Department of Management Information System
College of Business Administration, King Saud University, Saudi Arabia

Abstract—Machine learning-based autonomous agents are valuable for back-testing stock trading strategies, including algorithmic trading. Several studies in the financial literature have proposed artificial intelligence-based algorithms that support decision making for financial investment, but few studies have provided systematic processes for designing intelligent trading agents. This paper overviews the steps involved in designing agents that forecast stock prices in a trading strategy. These steps include data preprocessing, time-series segmentation, dimensionality reduction, clustering, and others. Our main contributions are: (i) a systematic process that guides the design and development of trading agents, and (ii) a random forest forecasting model.

Keywords—Decision trees; financial forecasting; machine learning; random forest; trading agents; trading strategy

I. INTRODUCTION

The success of a stock trading strategy depends on the quality of the information underlying its decision-making process. Most importantly, a successful trading strategy must quickly respond to data inputs for making informed decisions. Forecasting the level of a financial market index or individual stock is a classic problem. The efficient market hypothesis claims that forecasting stock prices is infeasible and that financial time series follow a random walk [1]. However, technical analysts consider that information regarding stocks and stock market indexes are reflected in historical prices and that prices can be forecasted. Financial market time-series forecasting has been widely studied in various disciplines, including finance, economics, engineering, computer science, and mathematics [2].

Financial market time series are commonly analyzed and forecasted by fundamental analysis [3, 4] and technical analysis [5–7]. The fundamental approach, typically used for long-term decision making on financial investments, focuses on the macroeconomic and company-specific factors that can influence the movements of stock index prices. Fundamental analysis is less common in the financial forecasting literature, because forecasting models that identify the reason(s) for stock price changes are difficult to construct. In contrast, technical analysis recognizes the patterns in a historical financial time series and implicitly assumes that some patterns are exploitable. Analytical and forecasting models based on technical analysis indicators are widely used in the financial literature (Murphy 1986, 1999).

Several machine learning techniques have been applied to financial time series [5, 8-13]. Machine learning techniques

identify the patterns in financial time series using a data driven approach. Common machine learning techniques for forecasting financial time series are decision trees [14], random forest (RF) [15-17], artificial neural networks (ANNs) [18-22] and evolutionary algorithms, such as genetic algorithms [23-27] and genetic programming (GP) [28-33]. Machine learning applications of financial time-series forecasting are reviewed in [34-35].

Automated trading in financial markets, including the foreign exchange (FX) and stock markets, is among the most logical applications of autonomous agents. The systematic design and evaluation of active trading strategies requires a detailed understanding of market mechanisms and operations and an awareness of the issues common to trading strategies. The present study provides a systematic process that guides the design and development of a stock market trading strategy for autonomous trading agents. The study also identifies the essential attributes representing the inputs of a machine learning algorithm for financial forecasting and discusses the potential and impact of these attributes. Designing and evaluating trading agents for financial markets is a non-trivial task giving the complexity and dynamic nature of financial data.

To accomplish our goals, we build a classification model that forecasts stock market prices using attribute selection and an RF algorithm. The characteristics and importance of the attributes are described in detail, and the most relevant attributes for forecasting and hence generating the trading rules are identified. The RF model is trained on the seasonality data's attribute space, directional changing price events, fundamental and technical indicators, and other financial data. Based on the attribute subset obtained from the attribute search space, the classifier computes the forecasting results, which are then used in designing the trading strategy. The impact of number of trees in the RF model and length of the training period are carefully evaluated. The agent strategy (and hence the performance of the generated trading rules) is evaluated in a stock market simulation of three datasets: the daily trade data of Standard and Poor (S&P100), Financial Times Stock Exchange (FTSE 100), and the Deutscher Aktienindex (DAX) indices of German stock. To identify the profitable trading opportunities over various time scales within the trading run, we connect the machine learning algorithm to the appropriate attribute space. Our RF learning algorithm can help traders to generate profitable trading strategies and bridge the gap between attribute selection and trading-strategy design.

The remainder of this paper is structured as follows. Section 2 briefly reviews the related literature. Section 3

*Corresponding Author

systematically describes the agent design process, including attribute selection and the RF algorithm. Section 4 demonstrates and examines the experimental results. Section 5 offers concluding remarks.

II. RELATED WORK

Agent-based modeling is an efficient and cost-effective method for analyzing complex dynamic phenomena, such as trading in financial markets and portfolio management [36-39]. By finding the optimal attribute sets and associations between the inputs and outputs of agent-based models, we can better understand the dynamics of financial markets and the modeled trading strategy. The financial literature includes several methods that explain and simplify the identification of attribute relationships through sampling and modeling [40-44].

Machine learning techniques provide a broad set of tools for the systematic design of trading and trading-agent algorithms [35, 45]. Among these techniques, supervised learning algorithms efficiently capture the nonlinear relationships between the inputs and outputs, inferring functional relationships from the data. Machine learning algorithms for stock forecasting include the support vector machine approach [46-48], ANNs [18-22], genetic algorithms [23-27], GP [28-33, 43, 44], support vector regression models [49], RF [15-17], extreme gradient boosting (XGB) [11, 17], and reinforcement learning [50]. Two reviews [34, 35] cover the recent significant studies on machine learning algorithms for stock market forecasting. The most relevant studies are [14, 16, 17]; and [51] who initially applied machine learning (particularly RF) to financial forecasting.

The work by [51] proposed a hybrid system that forecasts the future levels of a stock market index. They introduced a two-stage fusion approach using support vector regression (SVR) in the first stage and an ANN, RF, and SVR in the second stage, yielding three fusion prediction models: SVR-ANN, SVR-RF, and SVR-SVR. The second stage reduces the overall forecasting error. Patel et al. in [51] input ten technical analysis indicators to each forecasting model and emphasized the importance of providing adequate information to the model. To evaluate the effectiveness of their results, they compared the forecasting performance with single-stage models based on ANN, RF, and SVR.

The work by [17] applied deep ANNs, gradient-boosted decision trees (GBDTs), RF, and a combination of these methods (referred to as an ensemble) to the S&P 500 index. They forecasted the daily price returns by binary classification after training the four models on the momentum feature space. The average profits predicted by the ensemble (0.45% per day) were higher than those predicted by deep learning (0.33% per day), GBDT (0.37% per day), and RF (0.43% per day). Their results suggest that machine learning algorithms can provide sustainable profit opportunities in the short term.

The authors in [16] proposed an algorithmic trading strategy for the intraweek FX market trading based on an RF model and Probit regression. The RF model forecasted the medium-term evolution, whereas the Probit model was applied to the technical analysis indicators for predicting the price value of the next day. The profitability of the proposed model was evaluated on USD/EUR, USD/JPY, and USD/GBP from January 2014 to 2019. Their strategy improved the prediction accuracy and helped to identify the best times of entering and exiting the market.

The work in [14] built an experimental prediction framework based on a tree classifier, which analyzes the prices up to n days earlier, and forecasts whether prices will rise or fall. They trained two algorithms, RF and GBDT (using XGBoost), on technical analysis indicators, and varied the time interval of the analysis. The authors highlighted the importance of flexibility enabled by different features, each giving a specific interpretation.

The present study makes three contributions to the literature. First, it provides a simplified, systematic process for designing trading agents. We provide guidelines for determining the minimal set of attributes needed for an effective design, offering essential insights into agent design. Second, we evaluate the performance of the trading strategy using the existing standards in the financial forecasting literature. Third, we generate a diverse set of trading rules with a focus on attribute selection, allowing an extended training phase and trading rules that best exploit the RF algorithm.

III. METHODS

As mentioned above, trading agents developed by machine learning techniques can forecast financial time-series data. This section sets a framework to describe the main components in designing and developing intelligent trading agents to guide developers and practitioners in building such agents. The works by [52, 53] present the operational steps of an intelligent trading system that focuses on forecasting price movements in financial time series.

Simplicity is an essential characteristic of trading agents, because intricate design and complex trading rules will confuse the trading decisions and the dynamics of the trading rules. Forecasting of a price time-series significantly depends on the set of attributes used in the model. Therefore, selecting the right set of attributes for the trading agent is essential. Understanding the role of the attributes and settings can benefit investment profitability.

Our design of an intelligent trading agent proceeds through the following steps: (1) data preprocessing, (2) time-series segmentation, (3) dimensionality reduction, (4) algorithm definition, (5) training, and (6) evaluation. Fig. 1 depicts the framework of designing an intelligent trading agent for predicting the stock market.

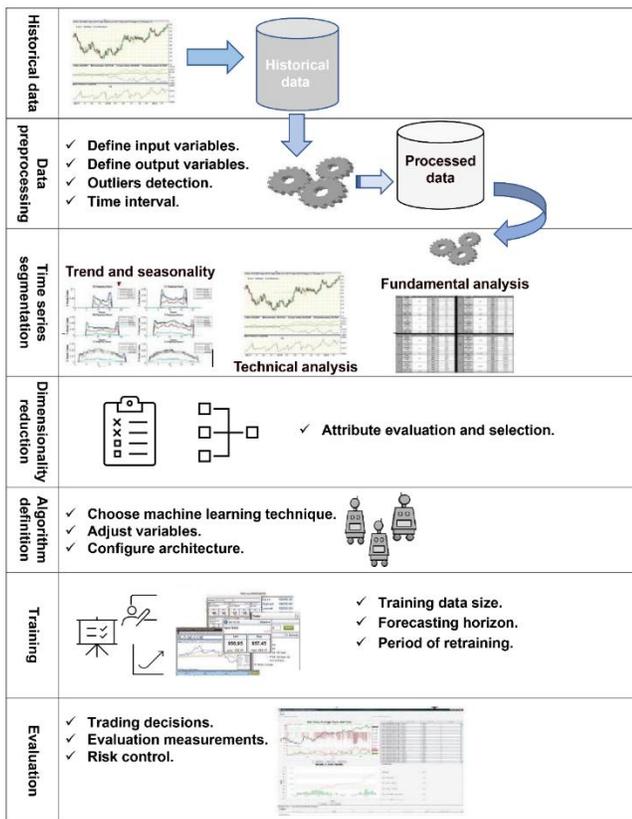


Fig. 1. Framework for Designing an Intelligent Agent for Stock Market Trading.

A. Data Preprocessing

The first step pre-processes the data utilized in the learning process. The main tasks in this step are defining the input and output variables, detecting the outliers, and structuring the data within a specified time interval. Data preprocessing ensures a reliable and accurate forecasting performance.

1) *Defining the input and output variables*: When designing a trading agent, we must first define the input variables used by the machine learning technique and the output signals to be forecasted. The sets of input and output variables define the trading rules and control risk, respectively. Technical analysis indicators are considered as the primary modeling inputs. Trading agents for short forecasting horizons are thought to require the technical analysis indicators, whereas those for long-term forecasting mainly use the fundamental input variables. Intraday seasonality statistics are the essential inputs of a forecasting model. The output variables are binary classifications of three class values signifying the buy, sell, and hold signals, which predict the direction and size of price movements throughout the trading period.

2) *Outlier detection*: After defining the input and output variables, the data are processed to remove noise and outliers, and the data elements are normalized and structured within a specified time interval. This process includes filtering the outliers and eliminating dimensionality from the data, data reduction, normalization, and detrending [54, 55]. Some studies

[54, 55, 56-58] suggest removing conditional transactions, late transactions, transactions reported out of sequence, and transactions with special settlement conditions. If these outlier transactions remain, they create noise and possibly erroneous data signals.

3) *Time interval*: As prices in a financial time series arrive at unequal time intervals, designing the trading rules is a challenging task. In particular, the time interval affects the technical analysis of the time series. Most of the reported intelligent forecasting models compute the technical parameters using the prices (open, high, low, and close) during a trading day, without measuring the magnitude of the price change. Therefore, the input data are considered as continuous data values (an interval-based dataset). This approach has proven effective in several studies [59-62] that assess whether today's closing asset price will extend yesterday's price trend.

Alternatively, trend-deterministic data can be analyzed within a directional change (DC) event framework [59, 61], which treats the input data as discrete entities to be sampled at irregular time intervals. The sampling interval is decided by the magnitude of a price change defined by the trader. The DC approach, which converts the discrete nature of a pricing series into a continuous DC event, has been critical in the FX market [43, 59-61, 63] and stock markets [28, 29, 64].

B. Time-Series Segmentation

The time-series data of financial prices are naturally dynamic and unevenly spaced in time. These attributes complicate the analysis phase and forecasting investment opportunities, as discussed in subsection 3.1.3. To reduce the number of dimensions for later analysis, the time series is commonly segmented in a preprocessing step. The resulting time series contains less data, allowing better detection of its significant patterns. Well-known financial time-series segmentation methods are intraday seasonality statistics [15, 60], scaling laws [61], and technical analysis indicators [6, 7]. By identifying trends and patterns in the segmented data, a trading agent can lower the computational costs and improve the forecasting accuracy.

1) *Trend and seasonality*: Seasonality analysis identifies periodic patterns in the historical data and formulates predictions based on the behavior patterns of earlier periods. Periodic past behaviors are assumed as the most likely future scenarios [14, 28, 65]. Therefore, seasonality statistics can reveal the most active trading hours of a stock index, the volume of transactions throughout a given day, and the probable development of trading flows. The intraday seasonality in a time series is defined as the hourly activity changes over a specified period of the time series. For example, the intraday seasonality can be represented by the aggregate number of pricing data points observed in each hour of a single day (hour 0-23), divided by the total number of pricing data points in the same period.

2) *Technical analysis indicator*: Technical analysis forecasts the prices and returns of actively traded financial instruments, such as individual stocks, market indices, and FX.

This approach assumes that patterns appear in historical prices. In statistical terms, technical indicators measure the various features in the time series of a given stock or market index, such as price trends, volatility, and momentum. Various technical indicators have been used for forecasting future price movements and developing rules of trading decisions [6, 66].

In this study, the trading agent was modeled using eight popular technical indicators calculated from historical price data. The calculation formulas of these indicators are given in Table I.

TABLE I. TECHNICAL ANALYSIS INDICATORS USED BY THE TRADING AGENTS IN THE PRESENT STUDY

Indicator	Formula
Moving Average	$MA(L, t) = \frac{p(t) - \left(\frac{1}{L} \sum_{i=1}^L p(t-i)\right)}{\frac{1}{L} \sum_{i=1}^L p(t-i)}$
Trade Break Out	$TRB(L, t) = \frac{p(t) - \max\{p(t-1), \dots, p(t-L)\}}{\max\{p(t-1), \dots, p(t-L)\}}$
Filter	$FLR(L, t) = \frac{p(t) - \min\{p(t-1), \dots, p(t-L)\}}{\min\{p(t-1), \dots, p(t-L)\}}$
Volatility	$VOL(L, t) = \frac{\sigma(p(t), \dots, p(t-L+1))}{\frac{1}{L} \sum_{i=1}^L p(t-i)}$
Momentum	$Mom(L, t) = p(t) - p(t-L)$
Momentum Moving Average	$MomMA(L, t) = \frac{1}{L} \sum_{i=1}^L Mom(L, t-i)$

The first technical indicator is the moving average (MA). The MA indicator continuously updates the average price within a specified window of time. In this study, the forecasting model inputs a 10-day MA and provides a short-term prediction. If the current price is above or below the MA, the price trend is upward or downward, respectively.

To understand the trade break out (TBR) indicator, we must define two concepts: support and resistance. At the support and resistance points, the price trend is expected to stop declining and rising, respectively. The TBR assumes that price trends will reverse at the support and resistance points; however, when these points represent a break out, possibly because new information emerges on the market or (for an individual stock) the company, the price trend is expected to continue in the same direction. Traders watching these break-outs will buy when a price trend exceeds the point of resistance and sell when the price trend falls below its support level.

The filter (FLR) technical indicator signifies a buy or sell signal depending on whether the price trend has been reversed by a predefined threshold. For example, if the price rises by a defined percentage of its earlier low value (reversing a downward price trend), the FLR issues a “buy” order.

Price volatility (VOL) is the standard deviation calculated from the historical prices over a specified period. The VOL indicator suggests a buying opportunity when the volatility is declining, as the price may be trending upward. In contrast, a

cycle of increasing volatility may signify a downward price trend, presenting a “sell” opportunity.

Momentum (Mom) is a simple trend-following indicator defining the rate or strength of the price movement. The Mom indicator relates the very recent closing price to the previous closing price over a time interval specified by the analyst. The Mom indicator also determines the momentum MA (MomMA).

Return on investment (ROI) in the trading context (distinct from the ROI of the data in a company’s financial statements) relates the trading profits to the strategy’s capital. Here, the ROI is the ratio of the price return to invested capital over a specified period. In our decision support system framework, the average price return is determined throughout the previous n days. The ROI is real-valued and can be positive (indicating a profit) or negative (indicating a loss).

The relative strength index (RSI) is a momentum indicator that contrasts the sizes of the recent upward and downward price trends over a defined period. This measure evaluates the rate and change of an asset’s price movements. To define an upward price-trend change, we set $Uptrend_t = p_t - p_{t-1}$ if $p_t > p_{t-1}$ and 0 otherwise; conversely, we define a downward price trend as $Downtrend_t = p_{t-1} - p_t$ if $p_t < p_{t-1}$ and 0 otherwise. The relative strength $RSI(n)$ is then defined as the average of the last n upward price-trend changes divided by the average of the last n downward price-trend changes. The RSI fluctuates between zero (average upward price change = 0) and 100 (downward price change = 0). In typical scenarios, an asset or overall market is overbought when the RSI exceeds 70 and is oversold when the RSI falls below 30.

3) *Fundamental analysis*: A fundamental analysis of a time series of securities prices investigates the likely factors affecting the time-series dynamics. These factors are derived from various sources, including financial statements, liquidity measures, market capitalization, and earnings announcements. Several studies have confirmed the relationships between these financial data and future price returns [3, 67].

The core mechanism of fundamental analysis is estimating an asset’s fundamental value using the financial data from all available sources. A trader can compare a fundamental value with the asset’s current price and make a trade decision based on that comparison. If an asset’s price exceeds its fundamental value, it is regarded as overvalued, and a sell order is recommended. Conversely, if the price is below its fundamental value, the asset is considered undervalued and a buy is recommended.

The trading strategy can consider the fundamental price value [67] or more than one value [3]. In the present analysis, an agent considers one fundamental price value as in [67]. Throughout the model run, the asset’s fundamental value is described by an exogenous random procedure:

$$f_t = f_{t-1} + \eta_t, \tag{1}$$

where f_t denotes the fundamental value of the asset price at time t and η_t is a normal noise process with mean μ and standard deviation σ .

C. Dimensionality Reduction

Restricting the number of attributes in the trading-agent model is essential, as not all attributes are valuable for forecasting a trade signal, and some are irrelevant. Reducing the attribute number and feature dimensionality improves the learning time, shortening the computational time, and improving the scalability. This step is particularly important for adaptive learning algorithms, which depend on the learning speed to route the streaming data over a suitable timeframe. Reducing the dimensionality reduces the memory and data-storage requirements, and simplifies the behavior of the trading agents. It also captures the dependencies and correlations among the attribute features.

D. Algorithm Definition

During learning by a machine learning technique, the agents continuously interact with the environment and thus learn the optimal trading rules to achieve a specific goal. The setting involves current price movements and historical data of stock market trading provided by an extensive array of raw data and technical analysis indicators. When selecting a machine learning technique, these historical data and technical indicators are prerequisites for designing agents that observe the price time-series environment and learn the stock trading rules. Most of the total design time is expended in providing the agents with informative data and tools, which is performed in the first three phases (data preprocessing, time-series segmentation, and dimensionality reduction). Although each phase generates inputs for the subsequent phase, the progression is unsteady and moves back and forth between the phases to optimize the result. After processing and reducing the data, the trading rules are generated from the selected input data by a machine learning technique, and the output signals are forecasted.

1) *RF methodology*: Forecasting by a machine learning technique is subjected to many factors: processing speed, pattern learning capability, natural interpretation of the trading strategy, knowledge simplification after learning, flexibility and adaptability to changes, and strategic optimizers for the retraining and forecasting horizons. Decision trees and RF are popular algorithms for forecasting price changes and developing automated trading agents. The RF algorithm has been successfully used in feature pruning (attributes in our case) owing to its broad capability and strength. In this study, the RF algorithm generates the trading rules for the trading agent.

RF is a supervised machine learning algorithm involving multiple decision trees on different subspaces of the attribute space. One decision tree connects a decision to a set of descriptive variables or attributes. Each decision tree is formed as a set of variables and conditions, hierarchically structured, and continuously utilized in a dataset. The decision trees are diversified by generating them from several training datasets that are randomly resampled from the original dataset. In financial forecasting, a variety of decision trees offers autonomous numerical estimates of the arbitrage opportunity (i.e., increase in price change) rather than class labels for predictions. The RF algorithm learns the relationships among the following inputs: (a) row data (opening, closing, high and low prices, and the trading volume per day), (b) fundamental

analysis information, (c) eight technical indicators, (d) DC price events, and (e) intraday seasonality statistics. The algorithm then generates one of three trading signals (buy, sell, or hold). Fig. 2 depicts the design framework of the forecasting model based on the RF algorithm for stock market prediction.

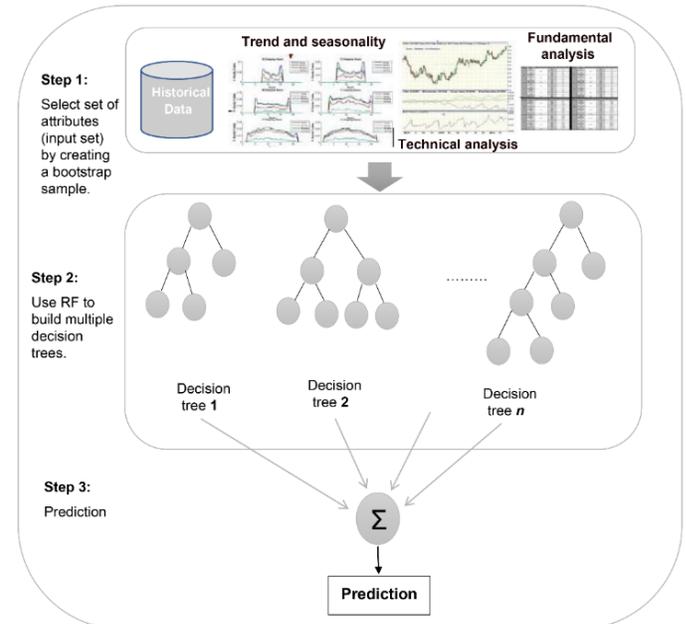


Fig. 2. Design Framework of a Forecasting Model based on the Random Forest (RF) Algorithm for Stock Market Prediction.

To create a single decision tree, we start from the root node and recursively split the data into two child nodes using a set of rules based on the market domain. The tree-building terminates when a definite stop condition is reached. The RF algorithm selects the attributes by a random process rather than employing all attributes. The resampling is performed by a random module and the combinations of attributes for the data-splitting are optimized to acquire more representative nodes. IF-THEN rules are derived from the trained decision trees by tracing the path from the root node toward each leaf node in each decision tree. The obtained rules are merged into a single decision-rule set. Along each path, the attributes and variables of the nodes are determined under the rule conditions, and the class of the leaf node represents the result of the rules (the trading signals). Fig. 3 shows a possible decision tree generated by the RF algorithm, and the following code derives a decision rule from the decision tree.

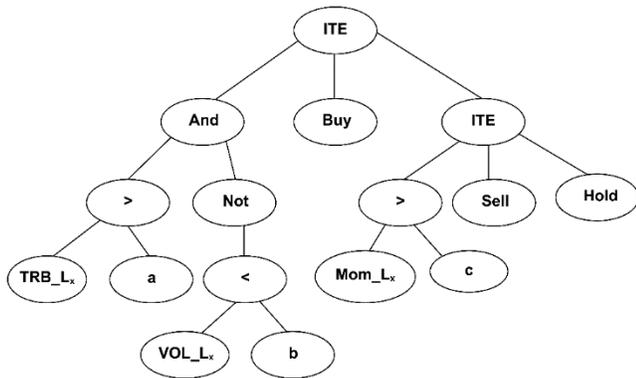
2) *RF algorithm*: The RF algorithm constructs an ensemble model of multiple unpruned decisions from random subsets of attributes that do not overfit the data. In this study, we build a RF of decision trees from bootstrapped training data. Therefore, instead of defining the best split among the entire set of input variables, we extract a subset of randomly selected attributes. In Algorithm 1 below, forests of decision trees are constructed using the training dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_T, y_T)\}$. The number of attributes is A and the number of trading days is T . Algorithm 1 is the core mechanism of the RF algorithm.

Algorithm 1: Random forest (RF) algorithm

```

1: procedure RF(D)           D is the training dataset
2:   f = new Array()        --> create a new forest
3:   for do i = 0 to N
4:     Di = bootstrapSamples(D) --> randomly select n data
from D
5:     attributei = RandomAttributeSelection(Di)
6:     DTi = new DecisionTree()
7:     DTi.train(Di, attributei)
8:     f.add(DTi)
9:   end for loop
10:  return forest f         --> return the new forest
11: end procedure RF(D)

```



```

If ((TRB_Lx > a) And (Not (VOL_Lx < b))) Then
  Buy
Else
  If (Mom_Lx > c) Then
    Sell
  Else
    Hold
  End if
End if

```

Fig. 3. A Possible Decision Tree Model in an RF (top), and Code Showing how a Decision Rule is Derived from the Decision Tree (Bottom).

The aim is to find a function $f : X \rightarrow Y$, where X is the attribute space and Y is the output trading signal space. The procedure is outlined below.

1. To form a bootstrap sample, randomly select n data from D drawn with or without replacement from the training dataset.
2. Among the A attributes, select $a \in A$ attributes to create a decision tree. Using a subset of data ensures variety and avoids the overfitting problem.
3. At each node, randomly select the attributes and optimize the data split (based on the Gini impurity). Note that the smaller the value of $Gini_D$, the higher is the purity of D . The Gini impurity measures a node's splitting quality. At each node N , it is defined as

$$Gini(N) = \sum_{i \neq j} p(w_i)p(w_j) \quad (2)$$

4. Grow n decision trees to the most considerable extent by reiterating steps 1 to 3 until the maximum number of

nodes is reached. The depth of each decision tree is controlled by the allowed maximum depth and the minimum number of required variables and constants at a leaf node. This so-called pruning criterion reduces the underlying complexity of the decision tree.

5. Define the mean prediction of all individual decision trees in the forest. The aggregate prediction represents the output at time t .

The examinations and visualizations were implemented in the R programming language [21, 22]. The following libraries and supplementary materials were also used: forecast [68], RF [69], gdata [70], ggplot2 [71], and reshape2 [72].

3) *Variable adjustment*: After choosing a machine learning technique, the input variables and output signals are adjusted to fit the desired algorithm. In this phase, the input dataset is decomposed for dividing the forecasting model into smaller sub-models. The machine learning technique (RF in this case) models the relationships among the input variables based on their initial definitions, and produces output signals. In the financial literature, machine learning techniques build the forecasting model using different numbers and types of input variables. Fundamental analysis offers numerous candidate attribute variables, such as firm size, cash flow, liquidity, and profitability. In contrast, technical indicators describe technical attributes such as moving averages, volatility, and momentum. In financial forecasting, the set of input variables for building a forecasting model (and thus designing a trading agent) is unlimited.

4) *Architecture configuration*: The architecture is configured by adjusting the set of input and output variables of the chosen machine learning technique. A practical design for a trading agent requires three main components. The first specifies the trading rules governing the entry and exit times of trades. Exploiting the forecasted output signals, these trading rules discover the optimal time to buy or sell an asset's quantity to maximize the investment profit. The second component evaluates the forecast and operates the risk control system. This component requires a set of rules that evaluate and protect the invested assets. Evaluation and risk control strategies include defining the profit objectives and risk appetites intended for each trade (e.g., maximum loss). The third component is the trading management mechanism, which manages the trading position size, resources, time window, and risk. This component emphasizes the impact of real-world constraints, such as transaction costs, news, transaction volume limits, and volatility [52].

E. Training

Once an agent has been developed from a training dataset and its variables have been adjusted on the validation set, the agent-trading strategy must be trained and evaluated on a new subset of the time-series data (namely, the test dataset). Here, a new subgroup is employed because the agent-trading strategy may be influenced by the training dataset. Therefore, it may overfit the data, producing an unrealistically high profitability. The training phase proceeds in three steps: testing the

performance of the methods, determining the forecasting horizon, and estimating the retraining period.

1) *Training dataset size*: Most of the financial forecasting models divide a financial time series into training and testing datasets. This approach is static because the same data are used throughout the complete testing period, with no updates. Splitting the data into training and testing datasets reduces the quantity of data available for learning and the resulting design may exclude various significant price events. Moreover, the forecasting performance sometimes depends on the time-series distribution used in the training and test datasets. This dependence is problematic, as the performance can be driven artificially high. Alternatively, the forecasting model can be trained by dynamic sliding-window cross-validation. In this approach, several evaluations are performed on small amounts of sequentially ordered data. Because the training-time window size may strongly affect the forecasting accuracy, we trialed sliding windows of different sizes.

2) *Retraining period*: Periodic retraining assesses the trading rules generated by the agent, thereby improving the average forecasting accuracy and accumulated profits. However, increasing the size of the training dataset over time may affect the forecasting performance. The price trends must be forecasted using an up-to-date training dataset representing the current market conditions. When designing the agent, the trading rules should be updated whenever the agent's wealth drops below a defined threshold value. This approach has been practiced in several automated agent-trading strategies [73].

F. Forecasting Evaluation

Among the many performance measures of forecasting models for trading agents, the ROI is probably the most popular choice.

1) *Evaluation measure*: The ROI represents the realized profits or losses of the trading strategy relative to the amount of money invested in the design. In our analysis, each trading decision was evaluated using the mean absolute percentage error (MAPE), mean absolute error (MAE), relative root mean squared error (rRMSE), and mean squared error, which are respectively calculated as follows:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|R_t - F_t|}{|R_t|} \times 100 \quad (3)$$

$$MAE = \frac{1}{n} \sum_{t=1}^n \frac{|R_t - F_t|}{|R_t|} \quad (4)$$

$$rRMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n \left(\frac{R_t - F_t}{R_t} \right)^2} \quad (5)$$

$$MSE = \frac{1}{n} \sum_{t=1}^n (R_t - F_t)^2 \quad (6)$$

Here, R_t and F_t are the actual and forecasted values, respectively.

The investment performance was evaluated with three performance indicators: Rate of correctness (RC), rate of missing chances (RMC), and rate of failure (RoF). These performance measures are calculated by Eqs. 7, 8, and 9,

respectively. The four criteria in these equations are defined in the contingency table (Table II).

$$RC = \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

$$RMC = \frac{FN}{FN+TP} \quad (8)$$

$$RF = \frac{FP}{FP+TP} \quad (9)$$

TABLE II. CONTINGENCY TABLE OF VARIABLES IN THE INVESTMENT PERFORMANCE INDICATORS

	Forecasted Negative	Forecasted Positive
Real negative	# of True Negative (TN)	# of False Positive (FP)
Real positive	# of False Negative (FN)	# of True Positive (TP)

2) *Risk control*: A successful trading agent requires a risk control mechanism that protects the invested assets by a set of trading rules such as a stop-loss limit order. Risk control strategies must define the profit objectives and risk appetites of each trade. Here we employed the Sharpe ratio (SR), which assesses the investment performance relative to risk. The SR is calculated by Eq. 10. Note that the higher the SR, the better is the risk-adjusted return, and (by inference) the better are the trading rules. According to Eq. 10,

$$SR = \frac{\mu - RF}{\sigma} \sqrt{n} \quad (10)$$

where μ is the mean of the portfolio returns, RF is the risk-free rate, σ is the portfolio's standard deviation, and n is the number of observations in the time series.

IV. RESULT

The previous section described the essential features and parameters of an autonomous trading agent. To handle the various possible mixtures of such features and parameters, a systematic design approach is required for an autonomous trading agent. This section details the verification experiments of the proposed approach. The results are expected to assist the design of autonomous trading agents and decision support systems for investment arbitrage in stock markets.

A. Data

The effectiveness of the proposed autonomous trading-agent design was tested on three stock market indices: Yahoo Finance S&P500 (US) [74], Yahoo Finance FTSE100 (UK) [75], and the Yahoo Finance DAX (GER) [76]. For the analysis, we extracted four years of historical data from September 1, 2015 to August 30, 2019. These three market indices were chosen for their popularity and strong liquidity in the global financial market. The study period covers various market conditions and time-series scenarios. The technical indicators were computed from the close, high, low, and opening levels of the three stock indices. All data were obtained from the Yahoo Finance websites.

B. Trading Strategies for Comparison

To verify the efficiency and effectiveness of the autonomous trading agent, we compared its trading performance with those of five well-known trading strategies: Buy and Hold (B&H),

technical analysis (TA), fundamental analysis (FA), zero-intelligence with constraint agents (ZI), and GP. The B&H strategy is a reactive investment strategy that emphasizes buying several stock assets and holding them for an extended period, regardless of fluctuations in the market-price time series. The TA strategy forecasts an asset's price movement using TA indicators (see subsection 3.2.2), and the FA strategy analyzes the potential influencing factors of the asset-price time series (subsection 3.2.3). The ZI randomly places orders under given budget constraints. However, such benchmarking is based only on the architecture of the proposed RF agent. Whether our RF proposal outperforms the existing machine learning and artificial intelligence techniques is an open question. Does the proposed RF-based agent architecture effectively outperform the other architectures based on other forecasting techniques? To answer this question, we added a fifth trading strategy based on GP, an evolutionary approach that extends genetic algorithms. GP evolves the programs of artificial intelligence to generate applicable solutions in a decision tree structure. Decision trees are generated by combining various defined functions and terminals. Both the function and terminal sets are derived from the problem search domain. Here we employed the GP trading agent designed by [29], which searches over the space of TA indicators.

C. Trading Performance Results

In the experiments, we varied the time horizon and combination of input attributes (parameters). The reported values were averaged over 30 runs in each trading model. After determining the agent design's most significant attributes based on the forecasting performance, the dataset was divided into two subsets using sliding-window cross-validation. The subsets were used for training and validating the RF algorithm.

After adjusting the model settings, we compared the performances of our RF-based agent design and the five competing trading agents: B&H, TA, FA, ZI, and GP. In the first evaluation, we measured the investment effectiveness of the RF-based agent using the ROI and SR as the investment performance metrics (a higher value indicates better performance). Table III compares the performance data of the six agents for each market index.

TABLE III. COMPARISON OF ANNUALIZED RETURN ON INVESTMENT (ROI) AND SHARPE RATIO (SR) OBTAINED BY VARIOUS TRADING AGENTS FROM SEPTEMBER 1, 2015 TO AUGUST 30, 2019

Stock Index	RF	B&H	TA	FA	ZI	GP
S&P500						
ROI	18.5%	2.4%	11.5%	2.62%	1.6%	11.9%
SR	1.7	0.97	1.2	0.9	0.1	1.1
FTSE100						
ROI	9%	-2.7%	6.6%	-3.6%	-27.6%	7.5%
SR	1.3	-0.2	1.1	-0.5	-0.01	1.03
DAX						
ROI	12.2%	-0.32%	10.2%	0.21%	-11.7%	10.8%
SR	1.4	-0.1	1.3	0.01	-0.6	1.4

^aRF: random forest; B&H: Buy and Hold; TA: technical analysis; FA: fundamental analysis; ZI: zero-intelligence with constraint agents; GP: genetic programming

The trading agent based on the RF model significantly outperformed the other trading agents over the four-year study period. The RF model achieved an average annualized profit of 13.22%. On the S&P500 data, the SR of the RF-based model was 1.72, followed by 0.97 for the B&H strategy, 1.21 for the TA agent, and 1.12 for the GP agent. The RF strategy also delivered the highest ROI and SR on the FTSE100 and DAX datasets. The results of the B&H and FA agents were comparable, as demonstrated in Table III.

The RF, TA, and GP agents employing the TA indicators significantly outperformed the remaining three trading agents. Therefore, generating the trading rules using TA indicators promises high investment results. However, determining the possibility of future trading achievement based on the ideal combination of TA indicators and its time horizon is a challenging task.

The RF algorithm limits the number of attributes used without compromising the profitability of the trading strategy. In particular, the RF-based trading agent can outperform the B&H approach when the selected attributes (inputs to the machine learning algorithm) are appropriately combined.

In summary, after studying the literature on automated trading agents and testing our RF-based approach, we conclude that a developed level of intelligence and diversity of the chosen attributes is essential for boosting the investment returns of the trading agent. Learning and adaptation intelligence can improve the ability of automated agents to process the market information and dynamics of price time series and thereby adapt to financial market events and different circumstances.

D. Forecasting Accuracy

In the second part of the study, we analyzed the forecasting accuracy of the RF and GP agents. As shown in Table IV, the RF- and GP-based forecasting models achieved strong RC, RMC, and RoF. For the RF agents, the RC ranged from 72.08% to 81.92% (average 77.22%), the RMC varied from 17.11% to 24.46% (average 20.90%), and the RoF ranged from 5.84% to 18.45% (average 13.18%). Clearly, the RF agents outperformed the GP agents in forecasting accuracy. We must clarify that the results in Table III influenced the accuracy of the forecasting performance. Moreover, selecting the right set of attributes improved the accuracy of classifying the training dataset, but does not guarantee better trading-strategy performance. Our findings suggest a substantial relationship between the profits generated and the fundamental/technical indicators, which may be important for creating practical trading rules.

The results of the RF algorithm emphasize the importance of attribute selection (data preprocessing, time-series segmentation, dimensionality reduction, and variable adjustments) as a preprocessing stage. The proposed agent design can assist (to some extent) the forecasting of stock prices. Accordingly, we conclude that when designing a trading agent to predict stock prices, we must examine the attribute types used to build the trading strategy and their values.

TABLE IV. SUMMARY OF FORECASTING PERFORMANCES IN 30 RUNS OVER THE TESTING PERIOD, MEASURED BY THE RATE OF CORRECTNESS (RC), RATE OF MISSING CHANCES (RMC), AND RATE OF FAILURE (RoF)

Stock Index	R C	RM C	RoF	MA E	MAP E	RMS E
S&P500						
RF	81.9 2	21.15	5.84	0.006	0.002	0.006
GP	74.3 1	28.21	9.52	0.009	0.003	0.009
FTSE100						
RF	77. 65	17.1 1	15.2 5	0.00 9	0.007	0.009
GP	71. 46	23.1 2	19.2 6	0.01 0	0.006	0.009
DAX						
RF	72. 08	24.4 6	18.4 5	0.00 7	0.004	0.008
GP	68. 13	28.0 4	19.6 4	0.00 9	0.004	0.009

^a MAE: mean absolute error; MAPE: mean absolute percentage error; RMSE: root mean squared error

E. Statistical Significance Test

The reliability and precision of the findings were tested in a paired t-test, which contrasts the means of two agents' datasets with variable ROIs. For this purpose, we divided the dataset into two equal subsets. An adequate paired t-test requires a roughly normally distributed variation of the data. Therefore, the normality was checked through a normal probability plot and a Jarque–Bera test.

The paired t-test was applied by specifying two values: the t-statistic value tsv and the t-critical value (two-tail variable) tc under the 95% confidence level. The first and second datasets were represented by A and B , respectively, where $a \in A$ and $b \in B$. The means of A and B were denoted as m_A and m_B , respectively. Both datasets were of equal size n . The numeral i indexed the record in both datasets, where $1 \leq i \leq n$. The null hypothesis stated that B is pairwise statistically equivalent to A , where $m_A - m_B = 0$. We define a new dataset D with n records, in which each form is specified by $d_i = a_i - b_i$, $d_i \in D$. The mean and standard deviation of this set were denoted by m_D and σ , respectively. The tsv was calculated as

$$tsv = \frac{\mu_D}{SE(\mu_D)} \quad (11)$$

where $SE(\mu_D)$ is the standard error of the mean difference, defined as

$$SE(\mu_D) = \frac{\sqrt{\sigma}}{n} \quad (12)$$

The t-critical value was determined from the t-distribution table. If $|tsv| < |tc|$, the difference between datasets A and B is trivial, and we can accept the null hypothesis. Table V shows the t-statistic (tsv) and Pearson correlation ($corr$) values of the paired t-test at the 95% confidence level. The results of the six trading agents on the first part of the dataset approximated those on the second part of the dataset with 95% confidence. Furthermore, the correlation coefficients demonstrate a high level of dependency between the results of the two datasets.

TABLE V. VALUES OF THE T-STATISTIC VALUE (TSV) AND PEARSON CORRELATION (CORR) OF THE PAIRED T-TEST AT THE 95% CONFIDENCE LEVEL (THE T-CRITICAL TWO-TAILED VALUE = 2.07)

Agent Type	tsv	$corr$
RF	3.54E-09	+0.98
B&H	3.01E-10	+0.94
TA	3.61E-09	+0.98
FA	2.82E-09	+0.95
ZI	2.32E-09	+0.97
GP	3.36E-09	+0.97

V. DISCUSSION

This study designed a simple automated intelligence agent for trading in the financial markets. An algorithmic (automated) trading agent is vital for developing trading strategies and forecasting models. This study provides a systematic approach for designing a trading agent using the RF approach. The approach was experimentally tested on four years of historical data of three stock indices: S&P500, FTSE100, and DAX.

The contribution of our work was twofold. First, we presented the main steps of building an intelligent automated trading agent for stock market trading. We categorized the main stages into five essential components: (i) the primary goal of the trading strategy (e.g., short term, long term, and high frequency); (ii) the context of the financial market domain; (iii) the input variables used; (iv) the proposed machine learning technique, and (v) diversity of the data sources.

Our second contribution was the design and implementation of adaptive RF trading agents for financial markets using various parameters (TA indicators and fundamental factors). It was hypothesized that trading agents learn, adapt, and identify periodic patterns in price time series. The RF trading agent evolved the trading rules of financial market trading investment. After the learning process, the RF algorithm achieved higher returns and better forecasting accuracy than the Buy & Hold trading strategy and agents based on technical and FA. We believe that this outperformance was achieved by selecting the right set of attributes (i.e., the inputs to the machine learning algorithm).

In future research, we could adapt machine learning techniques to dynamically select the appropriate set of attributes based on the market conditions. For this purpose, data reduction and segmentation could be combined with the sequential steps of developing an intelligent selection approach, as proposed in this study.

As another research direction, we could systematically test diverse intelligence trading strategies in the automated intelligence-based approach, and hence assess the required level of intelligence for designing autonomous trading agents. Examining and understanding the impact of complexity on a trading strategy might improve the effectiveness of that strategy. Nonetheless, a straightforward trading-strategy design, such as that developed in this study, might be more suitable than a complex one. To conclude, we should explore different artificial intelligence techniques and their effects on the effectiveness and trading rules of the strategy.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their useful comments and suggestions. The authors thank the Deanship of Scientific Research and RSSU at King Saud University for their technical support.

REFERENCES

- [1] Malkiel, B., 1973. *A Random Walk Down Wall Street*: W. W. Norton, & Company, New York.
- [2] Dacorogna, M., R. Gençay, U. Müller, R. Olsen, and O. Pictet, 2001. *An introduction to high-frequency finance*. San Diego: Academic Press.
- [3] LiCalziLicalzi, M., and P. Pellizzari. 2003. Fundamentalists clashing over the book: a study of order-driven stock markets. *Quantitative Finance* 3:470–80. doi: 10.1088/1469-7688/3/6/306.
- [4] Menkhoff, L., R. R. Rebitzky, and M. Schröder. 2009. Heterogeneity in exchange rate expectations: evidence on the chartist-fundamentalist approach. *Journal of Economic Behavior & Organization* 70:241–52. doi: 10.1016/j.jebo.2009.01.007.
- [5] Dempster, M. H., T. W. Payne, Y. Romahi, and G. P. Thompson. 2001. Computational learning techniques for intraday fx trading using popular technical indicators. *IEEE Transactions on Neural Networks* 12:744–54. doi: 10.1109/72.935088.
- [6] Leigh, W., R. Purvis, and J. M. Ragusa. 2002. Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. *Decision Support Systems* 32:361–77. doi: 10.1016/S0167-9236(01)00121-X.
- [7] Murphy, J., 1986. *Technical analysis of futures markets*. New York Institute of Finance.
- [8] Hsu, M. W., S. Lessmann, M. C. Sung, T. Ma, and J. E. V. Johnson. 2016. Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications* 61:215–34.
- [9] Lahmiri, S. 2017. Modeling and predicting historical volatility in exchange rate markets. *Physica. Part A* 471:387–95. doi: 10.1016/j.physa.2016.12.061.
- [10] Lahmiri, S. 2018. A technical analysis information fusion approach for stock price analysis and modeling. *Fluctuation & Noise Letters* 17. doi: 10.1142/S0219477518500074
- [11] Pinto, J. M., R. F. Neves, and N. Horta. November 2015. Boosting trading strategies performance using vix indicator together with a dual-objective evolutionary computation optimizer. *Expert Systems with Applications* 42:6699–716. doi: 10.1016/j.eswa.2015.04.056.
- [12] Teixeira, L. A., and A. L. I. de Oliveira. 2010. A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications* 37:6885–90. doi: 10.1016/j.eswa.2010.03.033.
- [13] Yao, J., and C. L. Tan. 2000. A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing* 34:79–98. doi: 10.1016/S0925-2312(00)00300-3.
- [14] Basak, S., S. Kar, S. Saha, L. Khaidem, and S. R. Dey. 2019. Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics & Finance* 47:552–67. doi: 10.1016/j.najef.2018.06.013.
- [15] Booth, A., E. Gerding, and F. McGroarty. 2014. Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications* 41:3651–61. doi: 10.1016/j.eswa.2013.12.009.
- [16] Chihab, Y., Z. Bousbaa, M. Chihab, O. Bencharef, and S. Ziti. 2019. Algo-trading strategy for intraweek foreign exchange speculation based on random forest and probit regression. *Applied Computational Intelligence & Soft Computing* 2019:1–13. doi: 10.1155/2019/8342461.
- [17] Krauss, C., X. A. Do, and N. Huck. 2017. Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S&P 500. *European Journal of Operational Research* 259:689–702. doi: 10.1016/j.ejor.2016.10.031.
- [18] Chang, P.-C., C.-H. Liu, J.-L. Lin, C.-Y. Fan, and C. S. P. Ng. 2009. A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications* 36:6889–98. doi: 10.1016/j.eswa.2008.08.077.
- [19] Gutjahr, S., M. Riedmiller, and J. Klingemann, 1997. Daily prediction of the foreign exchange rate between the US dollar and the German mark using neural networks. In *Joint PACES/SPICES Conference* 492–8.
- [20] Kamruzzaman, J., and R. Sarker, 2003. Comparing ANN based models with ARIMA for prediction of forex rates. *Asor Bulletin* 22:2-11.
- [21] The R Core Team, 2017. *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.
- [22] Zimmermann, H. G., R. Neuneier, and R. Grothmann. 2001. Multi-agent modeling of multiple FX-markets by neural networks. *IEEE Transactions on Neural Networks* 12:735–43. doi: 10.1109/72.935087
- [23] Aguilar-Rivera, R., M. Valenzuela-Rendón, and J. J. Rodríguez-Ortiz. 2015. Genetic algorithms and Darwinian approaches in financial applications: A survey. *Expert Systems with Applications* 42:7684–97. doi: 10.1016/j.eswa.2015.06.001.
- [24] Allen, F., and R. Karjalainen. 1999. Using genetic algorithms to find technical trading rules. *Journal of Financial Economics* 51:245–71. doi: 10.1016/S0304-405X(98)00052-X.
- [25] Bauer, R., 1994. *Genetic algorithms and investment strategies*. New York: John Wiley & Sons.
- [26] Mahfoud, S., and G. Mani. 1996. Financial forecasting using genetic algorithms. *Applied Artificial Intelligence* 10:543–66. doi: 10.1080/088395196118425.
- [27] Versace, M., R. Bhatt, O. Hinds, and M. Shiffer. 2004. Predicting the exchange traded fund dia with a combination of genetic algorithms and neural networks. *Expert Systems with Applications* 27:417–25. doi: 10.1016/j.eswa.2004.05.018.
- [28] Aloud, M. 2017. Adaptive GP agent-based trading system under intraday seasonality model. *Intelligent Decision Technologies*. IOS Press 11:235–51. doi: 10.3233/IDT-170291.
- [29] Aloud, M. 2018. Investment opportunities forecasting: a genetic programming-based dynamic portfolio trading system under a directional-change framework. *Journal of Computational Finance* 22:1–35. doi: 10.21314/JCF.2018.346.
- [30] Chen, S.-H., and C.-H. Yeh. 2001. Evolving traders and the business school with genetic programming: a new architecture of the agent-based artificial stock market. *Journal of Economic Dynamics & Control*. Elsevier 25:363–93. doi: 10.1016/S0165-1889(00)00030-0.
- [31] Dunis, C. L., J. Laws, and A. Karathanasopoulos. 2013. Gp algorithm versus hybrid and mixed neural networks. *The European Journal of Finance* 19:180–205. doi: 10.1080/1351847X.2012.679740.
- [32] Kampouridis, M., and E. Tsang, 2010. EDDIE for investment opportunities forecasting: extending the search space of the GP. In *Proceedings of the IEEE congress on evolutionary computation*. Barcelona, Spain: 2019, 26.
- [33] Vasilakis, G. A., K. A. Theofilatos, E. F. Georgopoulos, A. Karathanasopoulos, and S. D. Likothanassis. December 2013. A genetic programming approach for eur/usd exchange rate forecasting and trading. *Computational Economics* 42:415–31. doi: 10.1007/s10614-012-9345-8.
- [34] Gandhmal, D. P., and K. Kumar. 2019. Systematic analysis and review of stock market prediction techniques. *Computer Science Review* 34:100190. doi: 10.1016/j.cosrev.2019.08.001
- [35] Henrique, B. M., V. A. Sobreiro, and H. Kimura. 2019. Literature review: machine learning techniques applied to financial market prediction. *Expert Systems with Applications* 124:226–51. doi: 10.1016/j.eswa.2019.01.012.
- [36] Arthur, W. 1991. Designing economic agents that act like human agents: A behavioral approach to bounded rationality. *American Economic Review* 81:353–9.
- [37] Chan, N. T., B. LeBaron, A. W. Lo, and T. Poggio, 1999. Agent-based models of financial markets: A comparison with experimental markets.
- [38] Cristelli, M., L. Pietronero, and A. Zaccaria, 2011. Critical overview of agent-based models for economics. *Quantitative Finance Papers* 1101.1847.
- [39] Farmer, J. D., and D. Foley. 2009. The economy needs agent-based modelling. *Nature* 460:685–6. doi: 10.1038/460685a.

- [40] Alfi, V., M. Cristelli, L. Pietronero, and A. Zaccaria. 2009. Minimal agent based model for financial markets I: origin and self-organization of stylized facts. *The European Physical Journal. Part B* 67:385–97. doi: 10.1140/epjb/e2009-00028-4.
- [41] Alfi, V., M. Cristelli, L. Pietronero, and A. Zaccaria. 2009. Minimal agent based model for financial markets II: Statistical properties of the linear and multiplicative dynamics. *The European Physical Journal. Part B* 67:399–417. doi: 10.1140/epjb/e2009-00029-3.
- [42] Aloud, M., M. Fasli, E. Tsang, A. Dupuis, and R. Olsen. 2017. Modeling the high-frequency FX market: an agent-based approach. *Computational Intelligence* 33:771–825. doi: 10.1111/coin.12114.
- [43] Aloud, M., and M. Fasli. 2017. Exploring Trading Strategies and Their Effects in the Foreign Exchange Market. *Computational Intelligence* 33:280–307. doi: 10.1111/coin.12085.
- [44] Martinez-Jaramillo, S., 2007. *Artificial financial markets: an agent based approach to reproduce stylized facts and to study the Red Queen effect* [PhD thesis]. Martinez: Centre for Computational Finance and Economic Agents (CCFEA), University of Essex, 2007.
- [45] Gerlein, E. A., M. McGinnity, A. Belatreche, and S. Coleman. 2016. Evaluating machine learning classification for financial trading: an empirical approach. *Expert Systems with Applications* 54:193–207. doi: 10.1016/j.eswa.2016.01.018.
- [46] Kim, K. j. 2003. Financial time series forecasting using support vector machines. *Neurocomputing* 55:307–19. doi: 10.1016/S0925-2312(03)00372-2.
- [47] Lu, C.-C., and C.-H. Wu, 2009. Support vector machine combined with garch models for call option price prediction. In *Proceedings of the 2009 international conference on artificial intelligence and computational intelligence* Vol. 01, AICI '09:35–40. Washington, DC, USA, IEEE Computer Society.
- [48] Tay, F. E. H., and L. Cao. 2001. Application of support vector machines in financial time series forecasting. *Omega* 29:309–17. doi: 10.1016/S0305-0483(01)00026-3.
- [49] Pai, P.-F., K.-P. Lin, C.-S. Lin, and P.-T. Chang. 2010. Time series forecasting by a seasonal support vector regression model. *Expert Systems with Applications* 37:4261–5. doi: 10.1016/j.eswa.2009.11.076.
- [50] Pendharkar, P. C., and P. Cusatis. 2018. Trading financial indices with reinforcement learning agents. *Expert Systems with Applications* 103:1–13. doi: 10.1016/j.eswa.2018.02.032.
- [51] Patel, J., S. Shah, P. Thakkar, and K. Kotecha. 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications* 42:259–68. doi: 10.1016/j.eswa.2014.07.040.
- [52] Cavalcante, R. C., R. C. Brasileiro, V. L. F. Souza, J. P. Nobrega, and A. L. I. Oliveira. 2016. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications* 55(Suppl. c):194–211. doi: 10.1016/j.eswa.2016.02.006.
- [53] Palit, A. K., and D. Popovic, 2005. *Computational intelligence in Time series forecasting: theory and engineering applications (advances in industrial control)*. Secaucus, NJ: Springer-Verlag New York, Inc.
- [54] Masry, S., M. Aloud, A. Dupuis, R. Olsen, and E. Tsang, 2010. High frequency FOREX market transaction data handling. In 4th CSDA International Conference on Computational and Financial Econometrics, London, UK.
- [55] Pyle, D., 1999. *Data preparation for data mining*. 1st ed. San Francisco, CA: Morgan Kaufmann Publishers, Inc.
- [56] Brownlees, C. T., and G. M. Gallo. 2006. Financial econometric analysis at ultra-high frequency: data handling concerns. *Computational Statistics & Data Analysis* 51:2232–45. doi: 10.1016/j.csda.2006.09.030.
- [57] Falkenberry, T., 2002. *High frequency data filtering*. Tick Data, Inc.
- [58] Hautsch, N., 2012. *Econometrics of financial high-frequency data*. Berlin, Heidelberg: Springer-Verlag.
- [59] Aloud, M., E. Tsang, R. B. Olsen, and A. Dupuis. 2011. A directional-change events approach for studying financial time series. *Economics Discussion Paper* 2011:28.
- [60] Aloud, M., M. Fasli, E. Tsang, A. Dupuis, and R. Olsen. 2013. Stylized facts of trading activity in the high frequency FX market: an empirical study. *Journal of Finance & Investment Analysis* 2:145–83.
- [61] Glattfelder, J. B., A. Dupuis, and R. B. Olsen. 2011. Patterns in high-frequency FX data: discovery of 12 empirical scaling laws. *Quantitative Finance* 11:599–614. doi: 10.1080/14697688.2010.481632.
- [62] Müller, U. A., M. M. Dacorogna, R. B. Olsen, O. V. Pictet, M. Schwarz, and C. Morgeneegg. 1990. Statistical study of foreign exchange rates, empirical evidence of a price change scaling law, and intraday analysis. *Journal of Banking & Finance* 14:1189–208. doi: 10.1016/0378-4266(90)90009-Q.
- [63] Gypteau, J., F. E. B. Otero, and M. Kampouridis, 2015. Generating directional change based trading strategies with genetic programming. In *Lecture notes in Computer Science*, ed. A. M. Mora, and G. Squillero 9028:267–78. Springer International Publishing:267–78. doi: 10.1007/978-3-319-16549-3_22.
- [64] Aloud, M. 2016. Profitability of directional change based trading strategies: the case of Saudi stock market. *International Journal of Economics & Financial Issues (IJEFI)* 6:87–95.
- [65] Ito, T., and Y. Hashimoto. 2006. Intraday seasonality in activities of the foreign exchange markets: evidence from the electronic broking system. *Journal of the Japanese & International Economies*. Elsevier 20:637–64. doi: 10.1016/j.jjie.2006.06.005.
- [66] Murphy, J. J., 1999. *Technical analysis of financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance, New York.
- [67] Farmer, J. D. 2002. Market force, ecology and evolution. *Industrial & Corporate Change* 11:895–953. doi: 10.1093/icc/11.5.895.
- [68] Hyndman, R. J., M. O'Hara-Wild, C. Bergmeir, S. Razbash, and E. Wang, 2017. *Forecast: forecasting functions for time series and linear models*, R package version 8.1. Accessed January 19, 2021. <https://rdrr.io/cran/forecast/>.
- [69] Liaw, A., and M. Wiener, 2002. Classification and regression by randomForest. *R News* 2:18–22.
- [70] Warnes, G. R., B. Bolker, G. Gorjanc, G. Grothendieck, A. Korosec, T. Lumley, D. MacQueen, A. Magnusson, and J. Rogers, 2017. *Gdata: various R programming tools for data manipulation*, R package version 2.18.0. Accessed January 19, 2021. <https://cran.r-project.org/web/packages/gdata/index.html>.
- [71] Wickham, H., 2016. *Elegant graphics for data analysis*. 2nd ed. Cham, Switzerland: Springer International Publishing:Ggplot2.
- [72] Wickham, H. 2007. Reshaping data with the reshape package. *Journal of Statistical Software* 21. doi: 10.18637/jss.v021.i12.
- [73] Martinez-Jaramillo, S, and E. P. K. Tsang. 2009. An heterogeneous, endogenous and coevolutionary GP-based financial market. *IEEE Transactions on Evolutionary Computation* 13:33–55. doi: 10.1109/TEVC.2008.2011401.
- [74] Yahoo. *Finance* (S&P 500). Accessed February 7, 2021. <https://finance.yahoo.com/quote/%5EGSPC/history/>.
- [75] Yahoo *finance* (FTSE 100). Accessed February 7, 2021. <https://uk.finance.yahoo.com/quote/%5EFTSE/>.
- [76] Yahoo. *Finance* (DAX). Accessed February 7, 2021. <https://finance.yahoo.com/quote/%5EGDAXI/>.