POLITECNICO DI TORINO

Master of Science in Computer Engineering

Master Degree Thesis

Comparing time series and associative classification approaches to quantitative stock trading

Supervisor:
prof. Elena Baralis

Co-Supervisor:
rsr. Luca Cagliero

Candidate:
Giuseppe Attanasio

Academic year 2017 – 2018

This thesis is licensed under a Creative Commons License, Attribution – Noncommercial – NoDerivative Works 4.0 International: see www.creativecommons.org. The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

I hereby declare that the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Abstract

Can stateful, time-independent representations make time series forecasting more accurate? Does *time* dimension become negligible in such situations?

This work addresses these research questions in the context of quantitative trading. In order to do so, two classes of forecasting methodologies have been adopted to build automated trading systems and run simulations on data from Italian stock exchange market.

The first class includes models from time series analysis, that typically require data to be in chronological order. The proposed method instead is built upon a logical pipeline including the translation of data to a different, time-independent representation and associative classification. Stock prices have been translated into a structured data format, where each time sample is characterized by a set of state variables, describing the key features of the series at the sampled time. The produced representation incorporates the key information about the series as a independent set of variable states, thus relaxing the temporal dependencies among sampling times. As state variables, several technical indicators and oscillators have been considered, typically used in technical analysis to forecast future price movements and trends directions.

Experiments have shown that the proposed strategy achieves more stable results in mid-term forecasting compared to time series techniques and that provides models simpler to configure and validate in real world situations.

Acknowledgments

Questi cinque anni sono stati lunghi, altalenanti a tratti, per felicità e soddisfazioni. Ho raccolto così tante esperienze che, a conti fatti, il numero di persone che sento di dover ringraziare è elevato.

Per quanto lo si provi a nascondere, il trasferimento in una nuova città è, sì, una nuova sfida, ma porta anche nuove pressioni e incertezze. Se ho potuto vivere quest'ultime con la giusta leggerezza d'animo, lo devo alla mia famiglia. Per cui grazie, per essere stati presenti dall'inizio e per esserlo tutt'ora.

Sono successe molte cose durante questo percorso, ciascuna mi ha permesso di crescere come persona. Il merito va, quindi, a chi queste cose le ha rese possibili. Per questo devo ringraziare il mio gruppo di amici storici, i C., che, pur crescendo, è rimasto lo stesso punto di ritrovo fondamentale, dimostrandomi come alcuni rapporti prescindono dalle distanze. Ci sono poi molte altre persone che mi hanno accompagnato, criticato, abbracciato, stuzzicato, spinto e aiutato in eventi della vita quotidiana. Tengo quindi a ringraziare chi, tra una battuta ed un'altra, ha saputo darmi, forse senza saperlo, un modello d'ispirazione, accademica e personale; chi mi ha tirato su la testa dai libri, per portarmi al campetto; chi è stato compagno di studio e di svago; chi mi ha dimostrato che sviluppare una propria idea, nata ad un tavolino, è divertente e stimolante. Se non vi avessi conosciuto sarebbe stato tutto profondamente diverso.

Infine, ancora un grazie va alla persona che ha avuto l'odioso ruolo di sopportarmi, quasi quotidianamente, sia all'interno che fuori l'Università. Sei stata un punto di conforto e confronto permanente. Grazie.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Stock exchange trading is the process of buying and selling stocks of any company that is publicly traded. From traders' point of view this means investing personal money on markets, predicting possible price movements or trends, in order to collect positive profit. In practice, any market operator opens a position when he or she thinks that is the right time to do it and closes it for the same reason. The profit associated to the trade is generally evaluated as relative difference between closing and opening prices. Stock trading is accessible to everybody. Nowadays several web platforms, acting as market brokers, allow online trading to common people who want to invest their own money. Nevertheless, trading on markets with positive results is far to be naïve: the process must be driven by a strategy. Trading strategies could come from trader experience, who knows how to analyze past price trends and to foresee profitable buying or selling actions. These kind of traders are also known as *traditionals* or *chartists*. Another approach is explored instead in quantitative trading. In quantitative market analysis, mathematical and statistical models and methods, build on past values of prices are adopted to take decision in place of the trader. Designing the model and applying it on traditional trading platforms generates an automated trading system. Usually, models output is used to generate profitable *buy* or *sell* signals. However, the choice of discriminant variables to be included in the model and the ways to trigger buy and sell operations based on these variables are the most critical factors for system success. This building phase has traditionally been attributed to *quantitative traders*. Lately, the possibility to generate such models for stock price forecasting has been explored, among others, in the fields of Machine Learning: quantitative variables are the results of data mining and pattern extraction algorithms. The novel approach presented in this work belong to this class of methodologies.

Ever since stock prices analysis started, mainly because of their intrinsic nature, time series analysis has been the principal instrument used to forecast their trends. Time series are one of the most common and popular data type produced still nowadays. A time series is an ordered set of values, where the order is given by the time which samples are collected at. Since every collection created with a temporal order respects this property, it is sufficiently correct to think time series as they were not strictly related to a single

domain. Data can represent temperatures for each hour of a room, monthly income of a company, daily people visiting a museum, and so on. All these examples give an hint on another important property of time series: they can be discrete or continuous. The ones mentioned above are examples of the former. Values that compose discrete time series are collected within intervals of fixed amount of time. Continuous time series present values also in between the intervals. Since stock index prices are usually collected with a fixed time frequency, this work focuses on discrete time series.
The high amount of data available and the wide spectrum of domains they can come from makes time series analysis a huge opportunity for scientific studies. One of the most common objective for time series analysis is to forecast future series values. Given a number of time steps already elapsed and collected, a mathematical model is trained to fit and represent past data. Then, the model is used to infer a data point in the future not yet seen. In the known literature, some techniques use to predict only one or a few values after it, performing the so called *short-term* forecasting, while other techniques forecast data much further in time, performing a *long-term* forecasting. Time series models are built considering time series as it is, i.e. a sequence in temporal order of values. Their accuracy relies on the assumption that adjacent data are related: under such conditions, values from a past time frame are a valuable benchmark to forecast future data. Since models are built considering data sorted in time, a different order would lead, most of the cases, to a different model. Hence, in this work these technique are said to be *time-dependent*. Model performances are influenced by several factors. One critical design choice regards the number of different stochastic variables used, this leads either to *univariate* or *multivariate* models. Univariate models are the easiest to understand and design, as their output only depends on a single stochastic variable. Multivariate ones, instead, are described by functions of multiple variables, usually in a certain form of relationship. Auto Regressive models are one of the most common univariate examples: each value of the series is given uniquely by a combination of its previous values.

Following the increasing adoption, during recent years, of Machine Learning and Data Mining techniques, several newer strategies have been proposed to tackle the forecasting task in quantitative analysis. It is the case of classification techniques. It happens quite often that input data come split in sub-categories or groups or *classes*. The division criteria can be known in advance, i.e. each input data has already a class label, hence a so-called *supervised* technique is required. If class labels are not known *apriori*, a domain expert can generate them, typically grouping data with certain criteria. Supervised classifiers are mathematical models that if are trained with enough labeled input data, fit their parameters to learn data properties. Then, they can be used to infer one of the learned labels if new unlabeled data is given as input. There exist several works that has explored the use of classifiers to forecast market trends, their direction and possible turning points. Commonly used supervised classifiers include *Support Vector Machines*, *Naive Bayes* or *Artificial Neural Networks*. This work proposes a novel technique that uses an associative classifier, *Live-and-Let-Live* (L^3). Associative classifiers are emerging techniques in the field of Data Mining, used mainly for predictive purposes. Such models, built upon a set of transactional records, extract specific association rules. The rule set highlights the most frequent occurrences of combinations of items in the data set. Each rule comes in the format $A \to B$, where A is a collection of items and B is the consequent inferred by the rule. Whenever such rules are adopted for classification purposes, the consequent B is a class label. The main advantage of associative classifiers models it that the set of rules can be interpreted by a human operator. In fact, an association rule could easily be described, in plain English, as as set of conditions that *cause* the rule consequent.

This thesis addresses several research questions. Does time series analysis on stock prices produce more accurate and profitable trading signals compared to time-independent techniques? If it do so, in which contexts? Are time-independent, interpretable models reliable enough to configure real world trading system? Which are the most relevant rules among a large rule set to be included in such trading systems?

A comparison between time-dependent approaches relying on time series analysis and a time-independent methodology based on associative classification has been designed to tackle the first question. To perform time-independent predictions, we transformed time series data into a structured data format, where each time sample is characterized by a set of state variables, which describe the key features of the series at the sampled time. The produced data representation incorporates the key information about the series as a independent set of variable states, thus relaxing the temporal dependencies among sampling times. As state variables, a variety of technical indicators and oscillators have been considered, typically used in technical analysis to describe the main trends of price series. We applied the L^3 associative classifier to the time-independent data representation in order to discover the most significant associations between the state variables and the future stock price.
There are quite a few advantages in using associative classification. The first and most important is that extracted rules may be expressed in plain English. This leads to a much easier interpretation - e.g. for domain experts. Then it is also known that associative classifiers have fairly high accuracy on structured data compared to other classification techniques. Also, if the transactional representation is built starting from technical indicators, the rules extracted by L^3 classifier are quite similar to the ones already used in quantitative analysis – e.g. $RSI < 30, MACD > 0 \to BUY$: if the RSI oscillator takes value less than 30 and MACD indicator is above zero then generate a buy trading signal for the analyzed stock. Furthermore, given the set of extracted rules, they may be analyzed individually. Since each rule carries a subset of the relevant information, its behavior as single trading rule is used to know how well that information captures price trends and, eventually, is able to generate positive profit. This additional analysis has involved the top-10 rules – ranking is done with quality metrics.

The comparison has been conducted on stocks belonging to the main Italian stock exchange benchmark index, the Financial Times Stock Exchange Milano Indice di Borsa – or *FTSE MIB*. Even though the gross profit is still a valuable index in the domain, careful attention has also been paid on the number of operations and the relative profit per operation: in real markets each trade is charged with several fees. Stocks data collected during the years 2011, 2013 and 2015 have been used to analyze market in different conditions.

Results have highlighted some key facts. Time-dependent models need to be refreshed quite often to achieve high-quality results, while time-independent ones obtain more stable results in mid-term analyses. The reason is that, since time series analysis models are sensitive to small price variations, they are influenced by the forecasting horizon and by the algorithm configuration, while in time-independent models the underlying, recurrent market trends are mainly considered. Therefore, configuring and validating the quality of associative models is simpler than applying time-dependent approaches.

Overview. This document is organized as follows. Chapter 2 describes trading fundamentals and gives more details on the most common trading strategies and how they are implemented. Follows a survey on time series modeling and forecasting in Chapter 3. Chapter 5 contains a detailed report on the design of the proposed trading system. In Chapter 6 experiments conducted are listed and the results are reported in Chapter 7. Finally, in Chapter 8 conclusions are set out and possible future works are suggested to further explore these topics.

Chapter 2

Trading fundamentals

This work focuses on trading strategies for Italian stock exchange market. The process of trading involves a seller and a buyer. Usually, several financial assets may be traded. In the specific case, stock, also called *shares*, are the selling unit. The stocks are provided by listed companies, in this case the ones belonging to the main Italian market index, the Financial Times Stock Exchange Milano Indice di Borsa, or *FTSE MIB*.
A stock index is a benchmark series created from a portion of the market. Typically, the most important companies are picked and a weighted average, based either on prices or capitalization strength, is used to compute the index value. FTSE MIB counts forty of the most important companies present in Italian market. It is not static: the stocks included can change across the years.

Traders' main goal is to increase profit on stock exchanges. When the agent opens a position, it bets on a possible future trend of the stock. In stock exchange markets, a position is known to be either *long* or *short*. Investing money on long positions is equivalent to buy a stock at a certain price, speculating that it will increase its value, in order to sell it later in future and make profit by the *buy-sell* difference. Somewhat differently, short trading is equivalent to bet that price will drop. The operator rents temporarily assets it does not actually own and sells them to the market. If price drops, assets are bought back from the market in order to return them, making profit by the *sell-buy* difference.

Different trading strategies have been developed during the years. A trading strategy requires the generation of buy and sell signals that are used to evaluate possible entry or leaving points. Since the beginning, two school of thoughts emerged: one pursuing the so called *technical analysis*, and one the *fundamental analysis*. Technical analysis is based on the main assumption that stock prices and volumes already contain and reflect all relevant information to evaluate their profitability and future trend. Technical analysts mainly study price charts to discover trends and use statistical indicators to decide when and where operate on the market. Fundamental analysts, instead, believe that price movements are consequence of fundamental economic forces. Since these factors influence prices, they are more likely to contain crucial information to forecast it. *Fundamentalists* try to understand advisability of an investment analyzing company financial stability, income statements or the *sentiment* of investors around the company and several other aspects not related to the price itself. Nevertheless, as stated by Murphy [1], despite different points of view, the two categories face the same problem of price forecasting.

There are many examples in recent human history of real world problems, initially studied and solved by human mind solely, that have been modeled and analyzed with mathematical and statistical approaches. Stock trend analysis is among them. The field that studies methodologies to figure out price variations with the help of mathematical models is known as *quantitative analysis*. In most general cases, quantitative financial analysts design models upon subjective rules, encoded in variables of the model, to test and optimize them - e.g. they may look for recurrent patterns and evaluate possible future trends with a given probability. The variables and outputs involved are interpreted by analysts that have prior experience on the field. One of the first examples is shown by Markowitz [2] on the possibility to use mathematical formulae to encode diversification in planning profitable stock portfolios.

Furthermore, there is a strict connection between quantitative analysts and technical ones. In fact, the former are likely to be considered *technicians* whenever technical indicators are included in the model. Quantitative trading systems implemented in this work belong to this category, since technical indicators are used in the pre-processing stage, as described in section 2.1.2.

2.1 Technical analysis

As stated by Murphy [1] in his book, technical analysis studies market action, in order to forecast price variations or trends. This is mainly done through reading and analyzing charts with historical data. Reversing the main rule of fundamental analysts that business are influenced by fundamental economic agents, the central assumption of technical analysis is that price already reflects every factor that could possibly influence the price on the market itself, hence analysts are only required to study price values to forecast market movements. It is clear that, under this assumption, if a price is detected to rise or fall it already reflects the fundamental ratio of supply and demand, so traders should react and be bullish or bearish. The two definitions belongs to fundamental analysis: a *bullish* investor believe that stock price will increase in the future, a *bearish* investor believes the contrary, so that the price will decrease over time.

Murphy suggests one more important rule underlying technical analysis: price move in trends. It is probably the main factor that influence traders: charts and also indicators are built upon past values mainly to identify trends. The sooner a trend is detected, the better is for a trader since he or she can trade in the direction the trend is heading.

Figure 2.1 shows a clear positive trend of Microsoft Corporation prices across the one year period 24/09/2017–24/09/2018.

The principal instrument used by technical traders is chart analysis. Murphy [1] describes the subtle difference between *technicians* and *chartists*. Even considering that

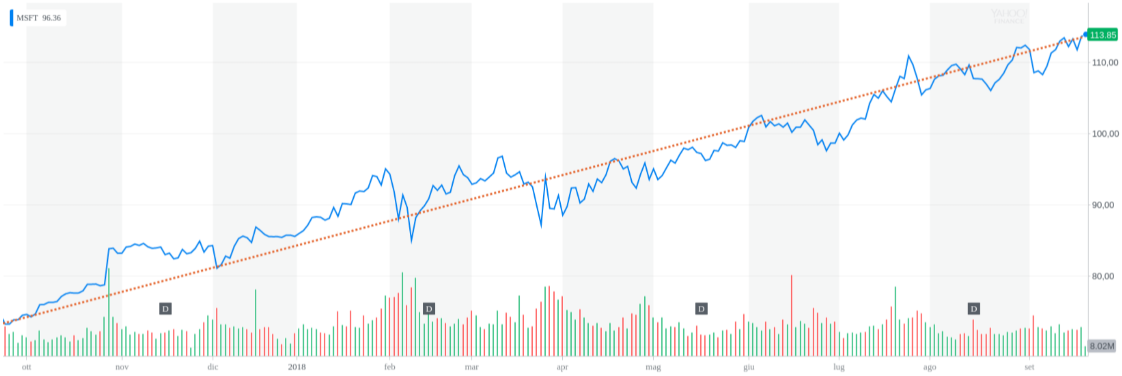


Figure 2.1: Line chart of Microsoft Corporation (MSFT) [3]

overlap exists between the two categories, chartists are considered to be traditional technical traders that strictly study charts and take decision based on their known experience. The interpretation still is subjective for the trader. Technician, instead, complement chart analysis with quantitative studies on statistical indicators. Also, intelligent trading models, or systems, can be programmed with rules to act mechanically and support trader decision. Trading systems proposed in this work try to convey time series analysis notions to build such an automated systems.

2.1.1 Charts

Charts are graphical instruments used by technical analysts to look for past trends and common behaviors of a given market share. They include history of the price and may contain several overlay for indicators. The most popular type of chart used is the *candlestick* chart. Figure 2.2 is an example that shows UniCredit prices across the period 24/06/2018–24/09/2018, with one day as unit of time.

Figure 2.2: Candlestick chart of UniCredit S.p.A. (UCG.MI) [3]

For each day a candle icon is used. Typically, two colors or empty / non-empty convention are used to differentiate two possible situation: either the closing price of the day is higher with respect to the opening one, either the contrary. In Figure 2.2 example, a green candle indicates a day in which price has increased. If the price went up, the bottom of the candle indicates the opening price and the top the closing one. On the other hand, if price decreased along the day, the top refers to the opening price and the bottom to the closing one. Each candle presents two stick at the top and bottom boundaries: the upper one extends up to the highest price obtained by the stock on that day, the lower one extends to the lowest price. Candlestick charts may or may not augmented with volumes bar. Typically volumes are present since they are displayed with non-intrusive graphical elements. In charts reported, at the bottom, volume bars show the absolute quantity of stock exchanged that day, either if it has been bought or sold. Just as graphical hint, their color reflects the candles one.

Another possibility is to use line charts. Nevertheless, their use is discouraged as they include only information on one price component, such as the closing values. Figure 2.1 is an example that depicts closing prices.

2.1.2 Technical indicators

Technical analysis make use of statistical indicators built upon past values of the stock components. They take into account *close*, *open*, *lowest* and *highest* prices as well as exchanged volumes. Usually, they are used for inter-day trading, i.e. with positions opened and closed in different days.

Indicators are used by traders as additional information to trend direction forecasting, turning point detection and in general to spot good entry or leaving point for stock trades. Since indicators can be evaluated day after day individually, in the proposed solution they may be regarded as *state variables* of a novel trading system. Even though such state is time-independent, many studies on technical analysis [1] [4], show that it is strictly correlated to price movements in the next future. Hence, this work proposes a sub set of indicators to build a time-independent representation that still contains valuable information for predictive purposes.

Each indicator has its own interpretation. When used by traders, they may be displayed as overlays onto charts so they get indexed by the same timeline. In Figure 2.3 there is an example of two moving average lines, twenty and fifty periods, plus the *Relative Strength Index* line in a candlestick chart of Atlantia S.p.A. stock, across the period 24/06/2018–24/09/2018. These indicators are detailed in sections 2.1.2 and 2.1.2.

Indicators may be bounded or unbounded. Bounded indicators are also know as *oscillators*. The most common are defined over the range $[0, 100]$ or over a symmetrical range with respect to the origin — e.g. $[-25, 25]$. Then, two threshold are defined to set boundaries of important areas: if the oscillator goes above or below the thresholds, valuable information can be inferred.

Figure 2.3: Moving averages and RSI indicators on Atlantia S.p.A. (ATL.MI) [3]

Over the years dozen of indicators have been designed by statistical economists and quantitative analysts. Those may be divided mainly in four classes: trend indicators, momentum oscillators, volatility indicators and volume indicators.
In the following sub sections there are theoretical notions useful to understand and interpret indicators. It is described how their trend over time is commonly used to infer trading signals and which stock components are part of their definition. Furthermore, a detailed description for the most common ones is reported.

Trading signals

Trading signals are usually generated by spotting two different situations: crossovers and divergence. *Crossovers* may happen either between one indicator and its own moving average line — known as *signal line*, or between two different indicators. The first case is mainly due to the nature of moving averages. As described in sub section 2.1.2, they are lagged indicators, hence they react to changes with a certain delay in time. Consequently, if the indicator line crosses its own moving average it could be symptom that the measure the indicator monitors is going to change. Figure 2.4 shows an example of Moving Average Convergence Divergence indicator that crosses its signal line in two different days.
The alternative crossover case involves two different indicators. One example could be the 20-periods moving average that crosses above the 50-periods one. This event is usually interpreted as symptom of an uprising trend.

Divergence instead is evaluated between indicators and stock components related to the indicator itself. If prices are rallying on a rising trend but an indicator, whose growth is semantically related to the growth of prices, is decreasing, signals indicate that positive trend is going to end. In the context of quantitative analysis, relationships between indicators and stock components may be quantified and signals may be programmed and inserted as part of automated trading systems. The latter is used to impersonate traders in finding future trend and decide to operate either long or short operations.

Figure 2.4: MACD lines on UnipolSai Assicurazioni S.p.A. (US.MI). 12-period and 26-period Moving Averages build the MACD indicator (purple) and 9-period Moving Average is the signal line (yellow). [3]

Stock components

Technical indicators are defined using mathematical formulae that have stock price components as variables. Common indicators uses the following values, referred to a single day t:

- Opening price: O_t;
- Closing price: C_t;
- Highest price: H_t;
- Lowest price: L_t;
- Volume exchanged: V_t.

In the following section, unless it is specified a different choice, indicators are referred to the closing price. As described below, this choice is the most appropriate for the majority of technical indicators. Nonetheless, there exists case where other components are required.

Trend indicators

Trend indicators should be used to assess trend strength, duration and direction. Also, used with crossover techniques, can signal trend reversal. Here are listed the most common one.

Simple Moving Average (SMA). Simple Moving Average is one of the most used and easy to interpret technical indicator. Each day, its value is given by the arithmetic mean of preceding n components. SMA is not related to a specific price component, nevertheless closing price is the most commonly used (2.1). Some traders use *mid-point* value, calculated as the day's range divided by two or the arithmetic mean between close, highest and lowest prices.

$$SMA_t(n) = \frac{1}{n}\sum_{i=1}^{n} C_{t-i} \tag{2.1}$$

Some considerations should be done. Moving averages are *lagged* – or *follower* – indicators, since the react to changes with a certain delay. They are used to support the idea that a new trend is started or an old one has changed direction, but they cannot predict these events. Also, evaluating the arithmetic mean, the series of values given by moving averages reflects the original one but with smoothed shapes.
The parameter n describes the sensitivity to price movements: fast moving averages have a small n, slower ones have a greater n. Technicians use to look for crossovers between SMAs of different speed. When a faster SMA crosses a slower one price trend is likely reversing. In Figure 2.3 this behavior is clear: Atlantia S.p.A. prices fell right after the tragedy of *Morandi* bridge in Genoa and the faster 20-period SMA crossed 50-period SMA.

Simple Moving Average equally weights values in the considered time frame. Some times this would not be the best choice. One possible alternative is to give a heavier weight to more recent days. Other types of moving averages, known as weighted averages, address the problem.

Exponential Moving Average (EMA). Exponential Moving Average is a weighted average: past values do not have all the same weight. It's evaluation requires three steps:

1. evaluate the initial value as:
$$EMA_0(n) = SMA_{n+1}(n)$$
2. evaluate the current weighting multiplier as:
$$\omega_n = \frac{2}{n+1}$$
3. evaluate current EMA value as:
$$EMA_t(n) = EMA_{t-1}(n) + \omega_n \cdot (C_t - EMA_{t-1}(n))$$

This calculus requires the evaluation of EMA values back to the initial $EMA_0(n)$.

Moving Average Convergence Divergence (MACD). This indicator is calculated as the difference between two Exponential Moving Averages, one faster with a short period and one slower with a long period. The classical implementation is:

$$MACD_t(12,26) = EMA_t(12) - EMA_t(26)$$

MACD values can be positive or negative. Whenever MACD crosses the zero, a new trend is expected. In particular, if the crossing is negative-to-positive an uptrend has just started, otherwise a positive-to-negative transition suggests an ongoing downtrend.
Moreover, a 9-period signal line is commonly used. The interpretation of crossovers between the indicator and its signal line still remains the same.

Average Directional Index (ADX). The index is a directional indicator developed by the engineer and technical analyst Welles Wilder [5]. The indicator makes use of *Average True Range* indicator, *Plus Directional Index* (+DI) and *Minus Directional Index* (-DI) directional indicators. While +DI and -DI could indicate trend direction, ADX suggests its strength. It is calculated through the following procedure:

1. evaluate for each day *True Range* (TR), *Plus Directional Movement* (+DM) and *Minus Directional Movement* (-DM) as:

$$TR_t = max((H_t - L_t), |H_t - C_{t-1}|, |L_t - C_{t-1}|)$$

$$+DM_t = \begin{cases} H_t - H_{t-1}, & \text{if } H_t - H_{t-1} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$-DM_t = \begin{cases} L_{t-1} - L_t, & \text{if } L_{t-1} - L_t > 0 \\ 0, & \text{otherwise} \end{cases}$$

2. smooth each TR, +DM, -DM with Wilder's smoothing technique over a 14-days period — e.g. smoothing on TR:

$$TR_0(14) = \sum_{i=1}^{14} TR_i$$

$$\cdots$$

$$TR_t(14) = TR_{t-1}(14) - \frac{TR_{t-1}}{14} + TR_t$$

3. evaluate +DI and -DI as:

$$+DI_t(14) = 100 \cdot \frac{+DM_t(14)}{TR_t(14)}, \quad -DI_t(14) = 100 \cdot \frac{-DM_t(14)}{TR_t(14)}$$

4. evaluate *Directional Movement Index* (DX) as:

$$DX_t = 100 \cdot \frac{|+DI_t(14) - -DI_t(14)|}{+DI_t(14) + -DI_t(14)}$$

5. finally evaluate ADX smoothing DX with another Wilder's technique over a 14-days period:

$$ADX_0(14) = \frac{1}{14} \sum_{i=1}^{14} DX_i$$

$$\cdots$$

$$ADX_t(14) = \frac{1}{14}(13 \cdot ADX_{t-1} + DX_t)$$

Average Directional Index measures the strength of a running trend. Wilder has suggested a well known interpretation. If ADX is above 25, a strong trend is present. If ADX is comprised between 20 and 25 trend as a contained speed. If ADX is below 20, no trend is present.

Momentum oscillators

Oscillators are a specific type of indicators that oscillates in a bounded range. They are used as support to trading decision with chart inspection and complementary to other technical indicators.
Momentum oscillators may be used both to assess trend strength and to highlight overbought or oversold conditions. The *momentum* is the speed at which a price trend is running: its knowledge is crucial for traders that want to spot eventual reversal. *Overbought* condition verifies whenever a stock has gained a huge hype on the market and it has been bought way more with respect to normal conditions. The opposite verifies with *oversold* situation, when the stock is massively sold on the market. Both cases could signal a trend reversal, caused by unsustainable market conditions.

Oscillators are characterized by an upper and a lower band of values. If oscillator enters in either one of the two bands, an atypical market condition could be present. Typically, values within the upper band suggests an overbought condition, while lower band indicates an oversold condition. In the following paragraphs the most common oscillators are listed, with their relative bands of attention.

Percentage Price Oscillator (PPO). This oscillator monitors the percentage difference between two moving average lines. The most common version uses 12-period and 26-period moving averages, making it the relative counter part of MACD:

$$PPO_t = 100 \cdot \frac{EMA_t(12) - EMA_t(26)}{EMA_t(12)} = \frac{MACD_t(12{,}26)}{EMA_t(12)} \tag{2.2}$$

Just like MACD, two types of crossover should be monitor: the ones with respect to the zero — i.e. positive-to-negative or negative-to-positive, and the ones relative to a signal line, typically set to PPO 9-period moving average.

Relative Strength Index (RSI). RSI has been designed by Wilder [5]. It provides the magnitude of recent price changes, mainly to highlight overbought or oversold condition. The index uses the concept of *Relative Strength*, with a time frame of 14 past days:

$$RS_t(14) = \frac{\sum_{i \in U} C_t - C_{t-1}}{\sum_{i \in D} C_t - C_{t-1}}$$

where U is the set of days whose closing price has been higher that the previous day close, also known as *up days*. On the contrary, D contains *down days*, whose price closed lower with respect to the previous day close. The numerator represents the average gain in past 14 days, the denominator instead the average loss.
Then, RSI is evaluated as:

$$RSI_t = 100 - \frac{100}{1 + RS_t} \tag{2.3}$$

RSI oscillates between 0 and 100. Traders may use 30 and 70 as thresholds respectively for oversold and overbought band borders. More conservative operators may also use 20 and 80 as thresholds. In any case, when RSI enters in upper band, the bullish market is likely going to end. The same applies for lower band but it reversal is due to a bearish market that oversold the stock.

Money Flow Index (MFI). It is an extension of RSI, where also the trading volume has been included. Like RSI, MFI is defined as:

$$RSI_t = 100 - \frac{100}{1 + MFR_t} \tag{2.4}$$

where *Money Flow Ratio* substitutes Relative Strength. MFR uses the notion of *typical price* (TP), *Raw Money Flow* (RMF), *Positive Money Flow* (+MF) and *Negative Money Flow* (-MF). The time frame is 14 days long:

$$\begin{aligned} TP_t &= \frac{C_t + H_t + L_t}{3} \\ RMF_t &= TP_t \cdot V_t \end{aligned}$$

Then:

$$MFR_t(14) = \frac{\sum_{i \in U} RMF_i}{\sum_{i \in D} RMF_i}$$

where U and D are up and down days considering the typical price for the gains and losses. Also MFI oscillates between 0 and 100, hence the pairs 30–70 or 20–80 can be used as thresholds for oversold and overbought conditions.

True Strength Index (TSI). This oscillator was introduced by Blau [6]. Its calculation involves several smoothing steps to make the indicator less sensible to noisy variations of stock price. It make use of *Double Smoothed Price Change* (DSPC) and *Absolute Double Smoothed Price Change* (ADSPC):

$$\begin{aligned} PC_t &= C_t - C_{t-1} \\ DSPC_t &= EMA_{PC,t}(EMA_{PC,t}(25), 13) \\ ADSPC_t &= EMA_{|PC|,t}(EMA_{|PC|,t}(25), 13) \end{aligned}$$

Then:

$$TSI_t = 100 \cdot \frac{DSPC_t}{ADSPC_t} \tag{2.5}$$

It's interpretation is straightforward. The oscillator can be either positive or negative: in the first case prices are going to rise, while in the other they are likely going to fall. As a consequence, the center line crossover is the most common signaling situation. Additionally, two symmetrical threshold could be used to identify oversold and overbought conditions, like -25–25 or -50–50.

Stochastic Oscillator (SO). Market theory says that in stocks trending upward prices will close near to the highest recent price and in down trending conditions the same applies for the lowest recent price. Stochastic oscillator has been designed to catch this behavior:

$$\begin{aligned} \%K_t &= 100 \cdot \frac{C_t - L_t(14)}{H_t(14) - L_t(14)}, \quad \text{where:} \\ H_t(14) &= \max(H_i), \quad i \in \{t-1, t-2, \cdots, t-14\} \\ L_t(14) &= \min(L_i), \quad i \in \{t-1, t-2, \cdots, t-14\} \end{aligned} \tag{2.6}$$

Typically a 3-period signal line is used to detect crossovers. Moreover, since the indicator oscillates in [0,100], two positive thresholds can define oversold and overbought regions: common values are 30–70 or 20–80.

SO compares current closing price to the lowest in a recent time frame.
Williams %R instead relates the current closing price with the highest price in the recent window.

$$\%R_t = -100 \cdot \frac{H_t(14) - C_t}{H_t(14) - L_t(14)} \tag{2.7}$$

It is a similar momentum indicator: %R generates the same curve of %K but scaled to different values.

Volatility indicators

These indicators help to detect periods in which market is more volatile, when stocks use to change prices with sharp movements. In such conditions trading become more difficult and different signals should be taken into account. Volatility indicators do not show trends, their strength or directions, but give indications on how smoothly the market is likely going to move around current prices.

Average True Range (ATR). Among this class of indicators, Average True Range (ATR) by Wilder [5] is the most popular one. The volatility is encoded in the absolute measure of *True Range*:

$$TR_t = max((H_t - L_t), |H_t - C_{t-1}|, |L_t - C_{t-1}|)$$

Then:

$$\begin{aligned} ATR_0(14) &= \frac{1}{14} \cdot \sum_{i=0}^{13} TR_i \\ ATR_t(14) &= \frac{1}{14} \cdot (ATR_{t-1}(14) \cdot 13) + TR_t \end{aligned} \tag{2.8}$$

It is clear that True Range is an absolute value that is strictly related to the range of stock considered. Hence ATR values coming from different stocks are not comparable.

Strong and sharp movements in stock price lead to an high True Range and, in turn, to an high Average True Range value. The index is monitored by traders because these conditions are commonly accompanied by trend reversal. However, ATR itself should not be used alone but, like other oscillators, should be a support for other trading strategies.

Volume indicators

Volume indicators combine prices values and volumes to give traders indications on sell or buy pressure. One of the rule used by technical traders is that rising in prices should be linked to rises in volumes. A divergence could suggest that trend is not going to last. Here are listed the most common indicators used.

Percentage Volume Oscillator (PVO). It is a momentum oscillator that, like Price Percentage Oscillator, monitors the momentum, or speed of change, smoothing a stock component, in this case the Volume. It is defined as:

$$PVO_t = 100 \cdot \frac{EMA_{V,t}(12) - EMA_{V,t}(26)}{EMA_{V,t}(12)} \tag{2.9}$$

Accumulation Distribution Line (ADL). This volume based indicator measures the flow of investments on a stock, given is historical prices and volumes values:

$$ADL_t = ADL_{t-1} * MFV_t \tag{2.10}$$

where the *Money Flow Volume* is:

$$MFV_t = V_t \cdot \frac{[(C_t - L_t) - (H_t - C_t)]}{H_t - L_t}$$

On Balance Volume (OBV) This volume indicator is a cumulative measure introduced by Granville [7]. It measures buying and selling pressures. The volume of each day is added to the total if price closed above opening, while it is subtracted if it was a down day. Its formula is then:

$$OBV_t = OBV_{t-1} \pm V_t \tag{2.11}$$

Chapter 3

Time series forecasting

Building a valuable trading system means, on top of all, taking the right decision when it is time to open, close or keep a stock quantity in the personal wallet. Decisions are primarily guided by the forecast on future value and its statistical confidence.

In classical forecasting techniques, future values are obtained through a mathematical model. Such models are usually build fitting existing data: the *shape* of the model approximates real values with some deviation error. Models are defined by functions of a specific number of parameters and input variables. Fitting these models to past data is equivalent to find model independent parameters that drive the function itself to be as much as *similar* to the real, yet unknown, function that distributes original values. Finally, forecasting is allowed by the model representation itself. Once it has been trained, its function already reflects the data distribution, hence is possible to obtain one or more output feeding it with *out-of-sample* time steps. The function outcomes represent predicted values.

The increasing study of algorithms and methodologies in the fields of *Artificial Intelligence*, *Machine Learning* and *Data Mining* has made possible the development of novel time series forecasting techniques. Machine Learning and Data Mining areas in particular are experimenting nowadays a huge hype in scientific community. Machine Learning takes the task of *Computer Science* one step further: algorithms and methods are now designed in a way that machines can incrementally learn from new data, eventually reprogramming themselves changing the inner shape of the model. Data Mining techniques are programmed to look for underlying associations between data: whatever recurrent pattern or rule is discovered, it can be used to infer properties on a new data that respects that pattern or rule.
It is worth mention that Machine Learning and Data Mining techniques aim at building novel models – typically classifiers – that are not necessarily implemented by means of mathematical formulae. They are made to summarize different information, such as recurrent patterns or underlying rules, rather than the distribution of past values. The building process itself is different and usually is strictly dependent of the technique considered.

In the following sections, theoretical notions on both the classes of methods are reported. Furthermore, several representative models for both the categories are described.

3.1 Statistical models

Time series analysis is a well know topic explored in both computer sciences and statistics fields. Classical models are the result of studies that have begun during the first half of past century. Ever since, those work have been focused on defining models that could appropriately fit data with certain statistical properties over time, such as the distribution, variation range, mean and standard deviation.

At the beginning, time series have been associated and studied both as discrete or continuous signal over time. Mathematicians Paley and Wiener [8] firstly introduced *Auto Regressive Integrated Moving Average* (ARIMA) models in thirties, linear models then used as discrete time filtering technique. Filters, known as digital filters in discrete time domain, are used to manipulate series transforming the information they intrinsically carry [9].

It became clear very soon that time series analysis should have embraced those models. *Auto Regressive* (AR), *Auto Regressive Moving Average* (ARMA) and *Auto Regressive Integrated Moving Average* (ARIMA) models are general and flexible enough to fit many of the real world time series. Furthermore, like other linear models, they provide forecasting capabilities out of the box. Whenever functions have been shaped to fit the known data, the approximated series is obtained by evaluating model function one step at a time, in chronological order. Hence, forecasting consists in extending one step ahead the time horizon. If it is required to generate multiple values, the model simply takes into account any value previously predicted.

Nonetheless, designing Auto Regressive models requires several choices that could make the model to fit better or worse to different types of time series. More details are given in section 3.1.2. In addition, there are cases where it is not possible to understand the right configuration in advance, hence an empirical study has to be conducted. Consequently, probably the main contribution in time series modeling was given a few decades later: Box and Jenkins published in their book, now at its fourth edition [11], a methodology to find out the most accurate ARIMA model that best fit the current series, known nowadays as *Box-Jenkins Method*.

The book gives a complete overview of time series analysis and forecasting. It describes practical application fields for time series analysis. Not only time series forecasting is considered, but also the possibility to estimate a transfer function for dynamic systems that are subject to inertia, or the study on how stochastic processes could be used to monitor the behavior of physical phenomena. The book also explores a fundamental notion for time series analysis and forecasting: the parallelism between time series and stochastic processes. Any series can be interpreted as an instance of a stochastic process. This allows to use probabilistic laws and search for statistical properties like *stationarity*.

Box and Jenkins analyze different type of models, linear and non linear, univariate and multivariate. In the following section, main focus is given to ARIMA *univariate linear* models. The latter are subject to deeper examination in the book itself where dedicated chapters provide an iterative methodology – the Box-Jenkins method, to find out their best parameters to fit input data. Later in time, other works have been published following notions of the aforementioned book – e.g. Chafield [12] continued the study of ARIMA models and their design, providing more application examples.

Auto Regressive models – or processes – are not the only possibility available among univariate forecasting techniques. Statistical *Linear Regression* is one of the most common and easy to implement model. Other univariate procedures, such as *Moving Averages* or *Exponential Smoothing* can address random variations by smoothing the original curve. More complex Exponential Smoothing schemes can deal with time series that have an underlying trend. *Holt-Winter*'s model can tackle both a trending evolution and seasonality in data distribution.

3.1.1 Model characterization

Any time series can be seen in the form:

$$Y = Y_t, \quad t \in \theta \tag{3.1}$$

where θ is the index set that represents ordered time steps and Y_i is the *ith* value of the series. Models fit time series data in the sense of representing, with a mathematical function, the relationship between each time step and the series value. In the most general definition, with *model*, it is intended any mathematical function:

$$f : \mathbf{R}^n \to \mathbf{R}^n \tag{3.2}$$

such that time series generated by the model may be defined as:

$$\hat{y} = f(t) \tag{3.3}$$

where t is the only independent variable, usually representing time, f is a mathematical function that has t as domain and $\hat{y}$ as co-domain and $\hat{y}$ is the time series value at time t. This definition suits well for continuous models or whenever the output at time t can be described as a mathematical function of t itself – see (3.6).
Since time series are normally indexed by positive time steps and stock prices are real numbers, such functions would become:

$$f : \mathbf{N}^1_+ \to \mathbf{R}^1 \tag{3.4}$$

As discussed before, this work, like many other in the known literature, analyzes *discrete* time intervals. Discrete time series – or signals – may be explicitly defined by a mathematical equation known as *recurrence relation*. Such representation presents each value as a combination of several other preceding terms. A general relation can be:

$$\hat{y}_t = r(t, x_1, x_2, \cdots, x_n) \tag{3.5}$$

where r stands for the recurrence relation and $(x_1, x_2, \cdots, x_n)$ are stochastic variables whose past values define the sequence value at time t. A well known relation composes the *Fibonacci* sequence:

$$y_n = y_{n-1} + y_{n-2}$$

Normally, recursive representations do not include initial states for input variables – this is not true for Fibonacci sequence, where we know that $y_0 = 1$ and $y_1 = 1$. If the latter are provided, a relationship with functions in the form of (3.3), known as *closed formulae*, can be found: specifically, a closed function that satisfies the sequence can be found. The process is known as *solving a recurrence relation*.

The dimensionality n of the relation is worth attention. Models can fit *univariate* or *multivariate* data. Univariate functions consist in mathematical combinations of one single variable that is, usually, the one of interest. Multivariate ones, instead, describe series value as a combination of more than one stochastic variable. Those models are more complex as they have to take into account relationship, correlations and differences between ranges of inputs. This comparative studies focuses on univariate models to build trading systems.

Finally, the mathematical nature of each model distinguishes them between *linear or not in input variables* and *linear or not in parameters*. The well known equation of straight line (3.6) is an example of both linear in variables and linear in parameters.

$$y(t) = m \cdot t + q \tag{3.6}$$

While the equation to define position in rectilinear uniformly accelerated motion (3.7) is an example of model linear in parameters but non linear in variables, because of a squared term.

$$x(t) = x_0 + v_0 t + \frac{1}{2} a t^2 \tag{3.7}$$

The model used by *Single Exponential Smoothing* (SES) is instead an example of linear model in variables but non linear in parameters (3.20). In the context of time series, non linearity in parameters is typically used to generate a smoothed series from the original one. This approach has two main advantages. Random variations are smoothed so the real trend of the series becomes more clear. Additionally, when SES is used in forecasting, predictions are mostly influenced by near-in-time values, since the contribution of past values decreases exponentially going back in time.

This work compares a novel approach built upon associative classification with several univariate, auto regressive, linear in variable and non linear in parameters models. These are described in the following sections.

3.1.2 Forecasting with univariate stochastic models

As stated by Box, Jenkins, Reinsel, *et al.* [11], univariate time series models consider the fundamental assumption that adjacent observation of are related. Under this conditions forecasts made on a certain time ahead – also known as *time horizon* – are valuable. Hence, the search of representative mathematical models is justified.

While modeling real world problems with mathematical functions, there are two possible class of models. In previous section *deterministic models* have been described – see (3.6) and (3.7). Once the function has been defined and the input domain is known, the outcome of these models is fixed. Time series modeling, instead, deals with *stochastic models*. For such models values are supposed to depend on many unknown variables, hence they cannot be predicted exactly. Any time series is hence a realization, with an associated statistical probability, of a stochastic model – or stochastic process.

In order to implement forecast, a stochastic model with a known form should be selected. Box, Jenkins, Reinsel, *et al.* [11] suggest that many of the real world time series are sufficiently well represented by auto regressive integrated moving average models. However, statistical properties of the series changes the way models are chosen. Specifically, *how statistical properties change* over the time is crucial – i.e. if the process, whose the series is a concrete representation, is stationary or not.
A stochastic stationary process preserves its mean level and variance over the time. A lot of simplification could be done in series modeling if the process is known to be stationary. Even though within auto regressive class there are models that can tackle non stationary processes, a forecasting technique strictly requires a preliminary study of values distribution to pick the right model.

Box-Jenkins Method

The model identification is the first part of what is known as *Box-Jenkins* methodology to find the model that best fit the series. As said before the detection of *stationarity* is crucial. Plotting the series can help: if an evident trend is present, the series is likely non stationary. Other techniques can be used, such as the *auto correlation plot* (ACP). In order use auto regressive models, stationarity is required. If the input series presents a non stationary trend it has to be differenced d times until the dth difference becomes a stationary series. Then the order of the model should be chosen. If an ARIMA model is used, there are auto regressive, moving average and differentiation order to pick – see sub section 3.1.4. The choice strictly depends on the data distribution and it can be simplified looking at statistical metrics over the time – e.g. ACP should be used to identify the auto regressive order: if auto correlation is still over a given threshold after n time step, including n previous values in the model could lead to more accurate predictions.
The second step is *model estimation*. Models parameters should fit the known data and, to do so, several mathematical formulae are used. An optimization strategy is followed, typically trying to maximize or minimize a given measure. The two most used approaches are *Non linear least squares* and *Maximum likelihood estimation*. The latter is presented by Brockwell, Davis, and Calder [13].
The third step regards *model validation*. Once the model has been produced residuals with respect to the real series are evaluated. A well designed models produces a stationary distribution of residuals.

Mathematical definitions

Box defines several mathematical indicators to better define auto regressive models. Any time series is made of a sequence of values indexed by the time dimension t. If the current value of the series is z_t, then a forecast distant n time steps is defined as:

$$\hat{z}_t(n) \tag{3.8}$$

The distance between forecast and real value the series has assumed after n steps is known as *forecast error*. If the simple difference is considered, the error becomes:

$$e_t(n) = z_{t+l} - \hat{z}_t(n) \tag{3.9}$$

The error function – or *loss function* – is typically the target of optimization while fitting any model to data. An often employed strategy is the minimization of the *Mean Squared Error*:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2 \tag{3.10}$$

but there exist other possibilities.
Useful mathematical operators are:

- backward shift operator: $B_{z_t} = z_{t-1}$, $B^m_{z_t} = z_{t-m}$;
- forward shift operator: $F_{z_t} = z_{t+1}$, $F^m_{z_t} = z_{t+m}$;
- backward difference operator: $\nabla z_t = z_t - z_{t-1} = (1-B)z_t$

3.1.3 Linear filter model

It can be considered a general stochastic model. The time series is generated by a set of independent *shocks* a_t sampled from a *white noise process* – i.e. a fixed distribution with zero mean and variance σ_a^2. These shocks are translated to the time series with a linear filter:

$$z_t = \mu + \Psi(B)a_t \tag{3.11}$$

where:

$$\Psi(B) = 1 + \Psi_1 B + \Psi_2 B^2 + \cdots$$

If z_t is stationary, μ is its constant mean.

3.1.4 Autoregressive models

Autoregressive models define the series as a finite, linear aggregate of previous values of the process itself. Depending on the terms included, different models could be generated.

Auto Regressive (AR(p))

The model estimated for AR processes is given by:

$$z_t = \mu + \phi_1 z_{t-1} + \phi_2 z_{t-2} + \cdots + \phi_p z_{t-p} + a_t$$
$$\bar{z}_t = z_t - \mu = \phi_1 \bar{z}_{t-1} + \phi_2 \bar{z}_{t-2} + \cdots + \phi_p \bar{z}_{t-p} + a_t$$

$$\Phi(B)\bar{z}_t = a_t \tag{3.12}$$

where $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ is known as *autoregressive operator* and p is the order of the process.

Moving Average (MA(q))

The model estimated for MA processes is given by:

$$\bar{z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}$$

$$\bar{z}_t = \Theta(B) a_t \tag{3.13}$$

where $\Theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$ is known as *moving average operator* and q is the order of the process.

Auto Regressive Moving Average (ARMA(p,q))

These models provide more flexibility adding both auto regressive terms and random sample from a white noise. The model is given by:

$$\bar{z}_t = \phi_1 \bar{z}_{t-1} + \cdots + \phi_p \bar{z}_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}$$

$$\Phi(B)\bar{z}_t = \Theta(B) a_t \tag{3.14}$$

Auto Regressive Integrated Moving Average (ARIMA(p,d,q))

ARIMA models address non stationary processes. They introduce one more factor to difference the input series. Specifically the differentiation order d must lead to a *dth* differenced series that is stationary. Hence the original series can be associated to a generalized auto regressive operator that includes differentiation:

$$\varphi(B) = \phi(B)(1 - B)^d \tag{3.15}$$

The recurrence equation becomes:

$$\begin{aligned} \varphi(B)\bar{z}_t &= \Theta(B) a_t, \\ \phi(B)(1 - B)^d \bar{z}_t &= \Theta(B) a_t, \\ \phi(B)\delta_t &= \Theta(B) a_t \end{aligned} \tag{3.16}$$

where $\delta_t = (1 - B)^d \bar{z}_t = \nabla^d z_t$, using the backward difference operator.

3.2 Regression models

Since time series data may be regarded as bi-dimensional sequences of values indexed by the time dimension, a regression analysis can be performed. Linear regression models fit data with a model linear in variables and parameters. The model function is:

$$y(t) = \beta x + q \tag{3.17}$$

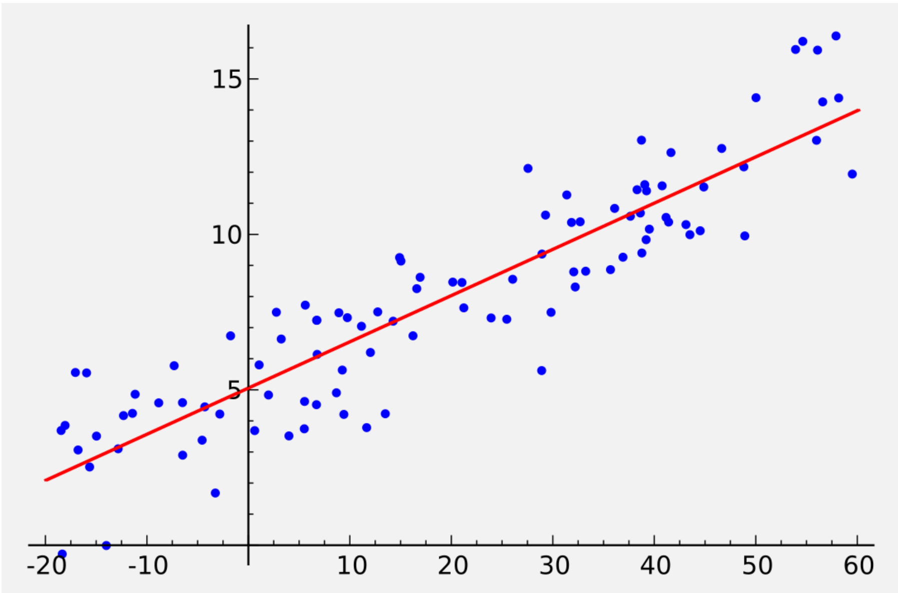


Figure 3.1: Linear model depicted in a scatter plot. [14]

When the model is built, several optimization techniques can be used to find β and q values. *Ordinary Least Squares* is one of the most commonly used: since the fitted curve generate residuals for each data point, parameters are chosen to minimize the sum of squared residuals.

3.3 Smoothing techniques

Two of the most common approaches to smooth time series are Moving Averages and Exponential Smoothing. Both are easy to represent with a recurrence equation, hence their use can easily be extended to forecasting. In a strict mathematical term, they are also univariate auto regressive models since each time step is a linear combination of previous values of the series itself.

3.3.1 Simple Moving Average (SMA)

Simple moving average performs smoothing through arithmetic mean on a past time frame of fixed dimension n.

$$\bar{z}_t = \frac{1}{n}\sum_{i=1}^{n} z_{t-i} \tag{3.18}$$

Its use as a technical indicator has been described in section 2.1.2.

3.3.2 Simple Exponential Smoothing (SES)

One critical restriction of SMA is that it attributes the same weight to every time step in the window. For certain types of time series, better performances are obtained if nearest points have a greater influence in forecast, with respect to values farther in the past. Any smoothed value is evaluated as:

$$\bar{z}_t = \alpha z_{t-1} + (1-\alpha)\bar{z}_{t-1}, \quad 0 < \alpha < 1, \quad t \geq 3 \tag{3.19}$$

The recurrent formula is obtained developing the recursive equation:

$$\bar{y_t} = \alpha \cdot \sum_{i=1}^{t-2} (1-\alpha)^{i-1} y_{t-i} + (1-\alpha)^{t-2}\bar{y_2}, \quad t \geq 2 \tag{3.20}$$

α is the *dampening factor*. With an high value, weights have a quick exponential dampening going back in the past and the smoothing is less accented. The contrary happens if with a small α: values away in the past still have influence in the current smoothing.

3.3.3 Holt-Winter's model

Holt-Winter's models is a smoothing model derived from Simple Exponential Smoothing. It addresses both the presence of a trend and the presence of seasonality in data distribution. Whenever a trend is present, smoothing techniques suffer the deviation from the standard mean level. The same applies for seasonality: if recurrent cycles are present the information can potentially increase forecasting accuracy if encoded in the model. Holt-Winter's formula presents two additional factor with respect to SES:

$$\bar{z}_t = \alpha \frac{z_{t-1}}{I_{t-L}} + (1-\alpha)(\bar{z}_{t-1} + b_{t-1}), \quad 0 < \alpha < 1 \tag{3.21}$$

where α is again the dampening factor, b_t stands for the trend factor and I_t for the seasonal one.

$$\begin{aligned} b_t &= \gamma(\bar{z}_t - \bar{z}_{t-1}) + (1-\gamma)b_{t-1}, && 0 < \gamma < 1 \\ I_t &= \beta \frac{z_t}{\bar{z}_t} + (1-\beta)I_{t-L}, && 0 < \beta < 1 \end{aligned} \tag{3.22}$$

L instead is the number of points that composes a season – e.g. is $L = 5$ for stock data that are sampled on week business days.

3.4 Data Mining and Machine Learning models

During recent years Machine Learning and Data Mining fields have been intensely explored by scientific community. Machine Learning is generally embodied by all the techniques that learn from input data – usually referred as *training* data – building an *experienced* model. Notions learned are then used to take decisions whenever a new incoming data is fed to the model. Once training algorithms are fixed, human hand is not even present: such techniques are considered automated decision-taking systems in all respects.

In the last few years we experienced an enormous increase in data produced from electronic devices and the growth is still continuing today: smart, connected devices started populating cities, manufacturing buildings and several other environments. The sudden big amount of data to process has created mainly two research opportunities for computer scientists, statisticians and data analysts: the development of efficient techniques to process data and the research of hidden information, at least to human eyes and experience. Data mining field embraces all techniques that address such problems. Data mining algorithms aims to discover pattern, rules and correlations between items that, instead, would hardly to be interpreted manually, either for the complexity or the quantity.

One of the most common task achieved with these techniques is classification. It is the process of assigning a class label to an unlabeled data. As said before, the model learns how to take this decision by processing training data. The two most common training strategies are supervised and unsupervised training.
In supervised techniques, training data is labeled hence models learn relationship between data and their label. In unsupervised techniques, it is the model itself that learns class distribution from unlabeled training data.

Different approaches both belonging to Machine Learning and Data Mining, have been designed and tested across many domains. In the following paragraphs a brief description of some widely used techniques is reported.

Naive Bayes. It is a family of supervised classifiers that make use of Bayes' Theorem on conditional probability [15]. The word *Naive* is consequence of the assumption that data attributes are mutually independent.

These models require a few data to obtain classification performances comparable to other advanced techniques. Additionally, the models produced can be inspected and interpreted.

Decision trees. Decision trees are supervised classifiers whose internal representation can be depicted as a tree. Each node of the tree testes one attribute of a structured data set and branching is created following the possible values that the attribute can assume in its domain – e.g. for the attribute *Gender* one branch for *Male* and one for *Female* are created. The leafs then are class labels.

Decision trees work well with structured data and the model obtained is easy to interpret.

Support Vector Machines. Support Vector Machines [16] are another supervised classification technique. Their training is headed to a geometrical objective, in fact the process aim is to search an hyper-plane that separates classes.

Initial implementations were designed to deal linearly distributed data. Recently, SVM implementations are some of the best performing classification techniques also with non linear distributions.

Artificial Neural Networks. There exist unsupervised artificial neural networks implementation and supervised ones. The latter are the most common ones. ANNs are complex structures that embody several logical processing units, the neurons, that mainly performs simple operations, like multiplications and additions. The model structure recalls human brain: many neurons are interconnected and organized in layers. Connections are weighted and the learning phases consists in updating those weights to better known how to distinguish classes. Many supervised ANNs have been recently designed to address classification problems from different domains. As of today, they find application in computer vision tasks, like image and object classification [17], speech recognition, natural language processing and others, where they outperform other classification techniques.

Associative classifiers Associative classifiers are a well known Data Mining supervised technique, used mainly for predictive purposes [18]. These classifiers are built from a structured data set where a set of attributes is the target of mining. The training process involves a rule extraction algorithm that searches for correlations between attributes values and class labels. Such information is encoded in the model using association rules. Each rule correlates a set of conditions on attribute values with a class label:

$$C_1 \wedge C_2 \wedge C_3 \wedge \cdots \wedge C_n \Rightarrow label$$

where:

$$C : \{attribute = value\}$$

A more compact representation is:

$$X \Rightarrow label$$

where X is the item set of conditions. The quality of each rule is evaluated mainly by two metrics: its support and confidence. The support is namely the portion of records that matches the rule conditions:

$$sup(X) = \frac{|X \subseteq d, d \in D|}{|D|} \tag{3.23}$$

where D is the whole data set. The confidence is measures how many times the inference suggested by the rule is correct:

$$conf(X) = \frac{sup(X \Rightarrow label)}{sup(X)} \tag{3.24}$$

The inference of class label is straightforward. Whenever an unlabeled record is fed to the model, a check against known rules is performed: if the input matches all conditions of a rule, that rule consequent is used as label. Association rules provide a huge advantage with respect to other classification technique. Since they can represented in plain English, they are easy to interpret. This allows, for example, domain experts to inspect and analyze them.

L^3 associative classifier

Live-and-Let-Live[1] is an associative classifier that implements a *lazy* pruning strategy on mined rules [19].

As described in previous section, an associative classifier model is built running a rule extraction algorithm on training data. Typically, these algorithms mine a large number of rules to let the model cover also rare conditions that represent minorities.

However this approach introduces a problem: a huge rule set is generated and it's exploration becomes soon computationally unfeasible as the initial set grows. In this cases pruning techniques are used. The choice is justified also supposing that classification performances increase if only *high quality* rules are retained. Support-based or confidence-base pruning – i.e. discarding rules under certain support or confidence thresholds – are usually employed. But pruning, in turn, has a side effect to consider: if many rules are removed from the knowledge base, also useful information could be lost.

L^3 classifier addresses the problem proposing a trade off strategy. Its model is built using a *lazy* pruning technique: in order to preserve as much information as possible, only rules that solely misclassify training data are discarded, regarded as *harmful* rules. Then remaining set is split in *used* rules, that have classified at least one record, and *spare* rules, that have not classified any training record but still remain useful for classification, since they are not harmful. Used rules better characterize data properties, hence they are used as rules of *first level*. Spare rule set, that is bigger in size, is kept to cover some of the cases where first level rules could not catch slight variations of inputs. Hence, the classification is done in two steps: at first a check against first level rules is performed. If no rule matches the record, a check on second level, spare, rules is done.

Since spare rule set is typically large, its dimension is then reduced using a compact rule representation. A smaller structure is much easier to manage and it allows to overcome possible information loss given by the minimum support threshold filtering.

[1] L^3 classifier has been used in this work as core algorithm while designing the trading system. Stock prices have been translated to a structured representation thanks to technical indicators. Even though the choice of an associative classifier is not compulsory, it is advisable when working with structured data.

Chapter 4

Related works

Quantitative techniques are well known mediums, both among technical traders and statisticians, to build efficient automated decision taking system. When, in the last century, digitalization of most of the stock market assets began, a huge opportunity was created to introduce machines and computer programming to stock data analysis and forecast.
Such programs became soon essential tools for technical analysts. From one side, more accurate real-time charts were possible, while on the other side, technical indicators started coming out once the support of machines computation became less expensive and available to anyone.

Then, with the foundations of technical analysis well established, the development of quantitative analysis was able to begin. Technical experts started to quantify, study and optimize variables and indicators through mathematical models and computer programming. First automated trading systems were developed: they encoded prior consolidate expertise of traders and statisticians.

Such a solid background both from statistical theory and economics, together with a strong knowledge of computer science algorithms, has created breeding ground for today's data analysts. In fact, during the past decade, many Machine Learning (ML) and Data Mining (DM) techniques have been designed, implemented and tested against financial data.

Since the beginning, stock trend forecasting has been the core of any automated trading system. While Machine Learning and Data Mining models are, most of the cases, built with no regard of the application domain, there have been published many examples of such techniques applied to price forecasting or trend reversals detection.

Among data mining algorithms, Baralis, Cagliero, Cerquitelli, *et al.* [20] use weighted sequence mining techniques to pick each day's most profitable stocks and operate intraday trading on them. Another example is Chen and Chen [21] where pattern recognition techniques have been used to forecast a bullish turning point in stock prices. Also textual data analysis has been integrated in an automated intraday stock recommendation system, by Geva and Zahavi [22].

Pure stock market indices prediction is addressed with multiple types of supervised

classifiers. The use of Volume Weighted Support Vector Machines is presented by Żbikowski [23].
Following the huge hype that Artificial Neural Networks recently experienced, many works propose their adoption in time series forecasting [24] and stock price prediction [25] [26] [27] [28] [29] [30]. ANNs have also been employed in hybrid approaches: Zhong and Enke [31] proposes a two step procedure to tackle stock price forecasting, involving dimensionality reduction on inputs and neural networks.

Ensemble of classifiers are quite a common strategy to improve classification performances. Patel, Shah, Thakkar, *et al.* [32] have implemented a two step hybrid system that include Support Vector Regression, Neural Networks and Random Forest classifiers. Tsai, Lin, Yen, *et al.* [33] uses majority voting and bagging in groups of Decision Tree, Neural Networks and Logistic Regression classifiers. A turning signal detection and forecast system is developed in [34] using an ensemble of Neural Networks. There are examples of stock trading decision support systems implemented with the combination of feature weighted SVM and k-nearest neighbor algorithm [35], using NN and particle swarm optimization [36] [37], with NN and SVM [38] or NNs optimized through a genetic algorithm [39].
The hybrid systems itself could be subject to optimization. Kim, Min, and Han [40] show a evolutionary technique over a set of different classifiers.

Not only supervised classifiers have been employed. In the known literature also unsupervised techniques such as clustering algorithms have been used. Teixeira and De Oliveira [4] propose a trading strategy supported by nearest neighbor classification.

Trading systems presented in this work have been implemented through L^3 associative classifier. Its main advantage over other classifiers, as pointed also in [41], is that extracted rules may be interpreted by domain experts. The interpretability of a model is crucial whenever it is intended to discover how automated system operated and why it took specific decisions. Nevertheless, it is worth noting that methods described in this work have a common point with many others: the use of technical indicators. Many related works have used a common set of them as main source of information [21] [23] [39] [41] [42] [43].

Machine Learning and Data Mining techniques have outperformed their predecessors in classification within many application fields. On the other hand, time series forecasting can count on many well known traditional techniques from statistics and mathematics. Hence, the performance comparison has been an interesting research topic since the beginning and, just like in this document, there are works that test the two categories. Ho, Xie, and Goh [44] show how recurrent neural networks and ARIMA models obtain the same performances on compressor failures prediction, both performing better on short-term forecast. Tang, De Almeida, and Fishwick [45] instead demonstrate how ARIMA give away against Neural Networks designed *ad-hoc* to tackle time series of severe complexity.
Despite the adoption of a different classifier, results presented in this work confirm somewhat this behavior.

Chapter 5

Methods

This chapter introduces a novel methodology for stock exchange index quantitative trading. The practical realization is an automated trading system that is supposed to trade stocks on markets. The core of this system is stock series forecasting. Prices variations are forecast using associative classification: each day, the direction of the price the day after is predicted.

The model has been tested on a complete trading environment, with real-world data from Italian stock exchange, where the system has automatically decided when and where enter or leave the market. In the context of this comparative study, several experiments have been performed to test profit performances of proposed solution against trading systems built upon traditional forecasting techniques.

Time-based models are built on historical data. This means that an underlying constraint is present: it is required that values are sorted in time. They can have any distribution: models try to summarize it at the best, but different distributions would lead to different models and so to different forecast.

The approach proposed is built on *Live-and-Let-Live* (L^3) associative classifier. Associative classification is a supervised technique employed in Machine Learning and Data Mining fields: models are built relying on the fact that recurrent pattern and intra series relationships, correlations and rules are present and can be mined. Associative classifiers need input data formatted as a set of records, each representing a set of key-value attributes: since time series do not satisfy these conditions, they must be processed ahead-of-time. Attributes extracted during this step are critical for classification quality.
Since technical analysis theory provides dozens of indicators, with known formulas and confirmed validity, this work proposes an ahead-of-time processing of stock values using those indicators. The output of this stage is a new data set, made of records of items – or *itemsets* – that are fed into associative classifier for training.
Hence, the solution proposed by this work is comprehensive of three main part:

- stock pre-processing, to generate a new data set suitable for L^3 classifier;
- model generation, to obtain a set of associative rules useful to label new trading days;

- trading simulation, where the model is used to take trading decision – i.e. open and close market positions to obtain profit.

5.1 Stock pre-processing

Stock pre-processing has a crucial role over the process. There are two main objectives to focus on:

- a time-ordered series should be transformed to a time-independent format. Result data set has to contain records – or *tuples* – that generate always the same model, even if fed to L^3 classifier with different order;
- valuable information should be kept and coded in the model. Hence, data transformation has to keep underlying semantics that classifier can discover mining rules over the records.

This work proposes to use technical indicators, described in section 2.1.2. Thanks to their purpose and tested validity, they represent optimal information to code and store. Hence, some of the most representative have been introduced in the new data format: each new record created contains a set of key-value pairs, each of them strictly related to one or more indicators.

Initial data set is made of stocks financial data from a portion of the Italian stock exchange market. These stock are the ones belonging to the main Italian stock benchmark index, FTSE MIB. In order to make results more general, robust and valuable, data from three years have been considered: 2011, 2013 and 2015. The point behind this choice is to make the algorithm operate over markets with different financial conditions. From an economic point of view, Italian market has been strongly decreasing, in average, during 2011, then is has decreased, even if with less severity in 2013 and finally showed an average increasing trend during 2015. Between those years some companies left the index and some others entered. For this reason the study keeps data separated, analyzing each year independently from the other two.

Financial markets assign to each stock, each day, an opening and a closing price. Those two values are store separately because price can fluctuate over the day. Actually, intra-day trading strategies have been developed to exploit this behavior, but this is not the case.

Table 5.1: Samples from 2011 ENEL.MI stock values that show the initial data format.

Day	Open	Close	High	Low	Volume
0	3.47	3.5	3.52	3.27	50049479
1	2.95	2.91	2.96	2.81	51661752
2	...	...	...	...	...

Since price oscillates, each day has two more valuable values: a maximum and a minimum. Thus, stocks may be represented as in Table 5.1. Therefore, each day is characterized by:

- a integer index representing time;
- four continuous attributes;
- one integer attribute.

Furthermore, it is normal that during the day stocks get exchanged between buyers and sellers. This is why market moves and price changes. Stock shifts are usually related to facts that have happened in the near period or that are happening, directly or indirectly, to the publicly traded company.

Indicators evaluation

Each day of the year has been transformed using technical indicators – they are described in section 2.1.2. Each indicator has been evaluated, the result is either an integer or a floating point value and it can be bounded or not. Those values have been adopted as attributes for the new record. In principle, each time step would be associated a correspondent record with indicators. But, since several indicators are lagged – i.e. require a fixed number of past data to be evaluated – the total number of generated records is lower. Specifically, if n is the maximum number of time steps required among all indicators, the first day where each of them has a value would be the *n+1-th* one. As a consequence:

$$size(R) = size(I) - n \tag{5.1}$$

where I is the initial data set, composed of stock values one per day, and R is the data set with indicators formatted. Table 5.2 is a possible format of this new representation.

Attributes definition

The next step of pre-processing is crucial, since it requires considering relationships between some indicators, within the same record. The purpose is to give the classifier information typically analyzed by technical traders for such indicators. In the known literature, technical and quantitative traders focuses on differences between moving averages, either equally

Table 5.2: Time series transformed with technical indicators

ID	SMA(5)	SMA(20)	RSI(14)	PPO(12, 26)	...
0	1.18	1.22	55	1.51	...
1	1.20	1.23	57	1.43	...
2	1.21	1.23	56	0.6	...
...	...	...	...	...	...

or exponentially weighted, and between directional indicators. Several known pattern are monitored and, if conditions are satisfies, traders may foresee a future trend.

Since, as stated my Murphy [1], prices tend to move in trends, also a trend reversal may be foreseen. Chart in 2.3 shows a clear example of one shorter period moving average crossing below the longer period one, right after the tragedy of Morandi bridge in Genoa. The trading event, in that case, is suggesting a future down-rising trend.

One more detail has been considered. Stocks in the market move between different absolute price ranges. The majority of indicators, e.g. every oscillator, gives a relative value to be interpreted independently of the range. Nevertheless, some of them, like moving averages or their differences, are absolute measures. Hence, as showed in (5.2), to make comparable records from different stocks, absolute attributes have been transformed as relative variation between one variable indicator and a fixed one, whenever it has been possible.

$$RSMA_t(5,20) = \frac{SMA_t(20) - SMA_t(5)}{SMA_t(5)} \tag{5.2}$$

However, indicators that base their statistical significance on absolute measures still remains. Whichever experiment that requires to consider in the same input set records belonging to different stocks uses an alternative representation, identical to the one described above except for the absolute indicators that have been discarded.
One final step is required: in order to make L^3 associative classifier work properly, a class label has been assigned to each new record. This is a critical point since, as described in section 3.4, class labels are used as rule *consequent* by the classifier. In other words labels should describe an event happening to the stock that is implied by values assumed by attributes. This work proposes to use as class label the percentage variation between the current day closing prices and the following one. Hence, each record will additionally have a class label defined as:

$$Label_t = 100 \cdot \frac{C_{t+1} - C_t}{C_t} \tag{5.3}$$

Discretization

L^3 requires records whose attributes have a categorical domain. Since categorical attributes are needed, pre-processing requires also a step to go from continuous new attributes to discrete ones, i.e. a *discretization* stage. Discretization requires to:

- split a continuous domain in a finite number of intervals;
- assign to each interval a categorical value;
- assign to each data points that falls into an interval the same categorical attribute.

It has not been possible to use a uniform strategy for intervals definition. Technical indicators are different, each of them has its own range and meaningful intervals. For some of them a two interval discretization has been used, for other a three interval one. The categorical representation of each attribute has been chosen as positive integer numbers:

$$a_d = \begin{cases} 1, & \text{for } a_c \text{ in first interval} \\ 2, & \text{for } a_c \text{ in second interval} \\ 3, & \text{for } a_c \text{ in third interval} \\ \ldots \end{cases} \tag{5.4}$$

where a_c is the old continuous attribute transformed it a discrete one, a_d.

Discretization of class labels follows the same strategy. The choice of these intervals is critical because it codes the *sensibility* that trading system will have: predicted class labels will generate trading signals, as described in section 5.3.
The trading system has been designed with a threshold of 1% for price variation to define intervals. Empirical results shows that 1% is a good compromise between lower thresholds that generates more noise signals and higher ones that rarely generate signals, making impossible to test the system. Categorical attributes meaningful to technical analysis have been chosen:

$$L_{t,d} = \begin{cases} BUY, & \text{for } Label_t \geq 1 \\ HOLD, & \text{for } -1 < Label_t < 1 \\ SELL, & \text{for } Label_t \leq -1 \end{cases} \tag{5.5}$$

Using a three-classes classifier is interesting because two different trading signals could be generated, either if BUY or SELL are predicted. Nonetheless, also the adoption of binary classifier has been considered for trading system. Such classifiers can easily be created considering price variations towards only one direction. As a consequence, two more classifiers are possible:

$$L_{t,d} = \begin{cases} BUY, & \text{for } Label_t \geq 1 \\ NOT\_BUY, & \text{for } Label_t < 1 \end{cases} \tag{5.6}$$

$$L_{t,d} = \begin{cases} SELL, & \text{for } Label_t \leq -1 \\ NOT\_SELL, & \text{for } Label_t > -1 \end{cases} \tag{5.7}$$

Table 5.3 shows a possible representation after the discretization phase with three classes. Each record represents a set of discrete values that indicators assumed that day. This structured representation, that is now time-independent, is convenient not only for associative classifiers. Many Machine Learning supervised classifiers such as Support Vector Machines, Artificial Neural Networks or Naive Bayes can be fed with these data.

A complete description of new attributes is given. Table 5.4 lists properties that have been used for input set containing data from multiple stock. Table 5.5 instead describes the ones used when data did come from a single stock, hence in cases where absolute attributes were usable.

Table 5.3: Final representation of new data set after pre-processing stage.

ID	RSMA(5, 20)	RSMA(20, 50)	RSI(14)	PPO(12, 26)	...	Label
0	2	1	2	3	...	SELL
1	2	3	1	2	...	BUY
2	1	1	3	3	...	HOLD
...	...	...	...	...	...	...

Table 5.4: Attributes defined for L^3 classifier used with data from multiple stocks.

Attribute	Description	Intervals
RSMA(5, 20)	Relative difference between SMA(5) and SMA(20)	a <0 <b
RSMA(8, 15)	Relative difference between SMA(8) and SMA(15)	a <0 <b
RSMA(20, 50)	Relative difference between SMA(20) and SMA(50)	a <0 <b
REMA(5, 20)	Relative difference between EMA(5) and EMA(20)	a <0 <b
REMA(8, 15)	Relative difference between EMA(8) and EMA(15)	a <0 <b
REMA(20, 50)	Relative difference between EMA(20) and EMA(50)	a <0 <b
MACD	Moving Average Convergence/Divergence	a <0 <b
AO(14)	Aroon Oscillator	a <0 <b
ADX(14)	Average Directional Index	a <20 <b
WD(14)	Difference between Positive Directional Index (DI+) and Negative Directional Index (DI-)	a <0 <b
PPO(12, 26)	Percentage Price Oscillator	a <0 <b
RSI(14)	Relative Strength Index	a <30 <b <70 <c
MFI(14)	Money Flow Index	a <30 <b <70 <c
TSI	True Strength Index	a <-25 <b <25 <c
SO(14)	Stochastic Oscillator	a <20 <b <80 <c
CMO(14)	Chande Momentum Oscillator	a <-50 <b <50 <c
ATRP(14)	Average True Range Percent: ratio, in percentage, between Average True Range and Close	a <30 <b
PVO	Percentage Volume Oscillator	a <0 <b
OBVP	On Balance Volume Percentage: On Balance Volume index evaluated with percentage variations	a <0 <b

5.2 Model generation and classification

Model generation is done through the available L^3 implementation. The produced model is defined by a set of association rules divided in first level rules and second level rules as described in section 3.4. Each rule specifies, under certain conditions, which is the trading signal predicted by the classifier.
One of the main advantages of this representation is that, using same conventions, rules may be transformed back into phrases open to experts interpretation. An example is given

Table 5.5: Attributes defined for L^3 classifier used when data came from a single stock

Attribute	Description	Intervals
RSMA(5, 20)	Relative difference between SMA(5) and SMA(20)	a <0 <b
RSMA(8, 15)	Relative difference between SMA(8) and SMA(15)	a <0 <b
RSMA(20, 50)	Relative difference between SMA(20) and SMA(50)	a <0 <b
REMA(5, 20)	Relative difference between EMA(5) and EMA(20)	a <0 <b
REMA(8, 15)	Relative difference between EMA(8) and EMA(15)	a <0 <b
REMA(20, 50)	Relative difference between EMA(20) and EMA(50)	a <0 <b
MACD	Moving Average Convergence/Divergence	a <0 <b
AO(14)	Aroon Oscillator	a <0 <b
ADX(14)	Average Directional Index	a <20 <b
WD(14)	Difference between Positive Directional Index (DI+) and Negative Directional Index (DI-)	a <0 <b
PPO(12, 26)	Percentage Price Oscillator	a <0 <b
RSI(14)	Relative Strength Index	a <30 <b <70 <c
MFI(14)	Money Flow Index	a <30 <b <70 <c
TSI	True Strength Index	a <-25 <b <25 <c
SO(14)	Stochastic Oscillator	a <20 <b <80 <c
CMO(14)	Chande Momentum Oscillator	a <-50 <b <50 <c
ATRP(14)	Average True Range Percent: ratio, in percentage, between Average True Range and Close	a <30 <b
PVO	Percentage Volume Oscillator	a <0 <b
FI(13)	Force Index	a <0 <b
FI(50)	Force Index	a <0 <b
ADL	Accumulation Distribution Line	a <0 <b
OBVP	On Balance Volume Percentage: On Balance Volume index evaluated with percentage variations	a <0 <b

below.

$$RSMA(5,20) < 0 \land RSMA(20,50) < 0 \land RSI(14) \geq 80 \land \cdots \rightarrow SELL \tag{5.8}$$

The model – i.e. the association rules – have been applied to build a trading system. Additionally, because of their readability and interpretability, they have been collected for further analysis. Specifically several other experiments have been conducted on single rules to evaluate performance and understand how relevant each rule is in the automated trading system.

Once L^3 model has been obtained, days belonging to the test set have been classified. Test set varies depending on the validation strategy used, as described in section 6. Classification results can be depicted as an array, where each day has its associated label – Figure 5.1.

The proposed implementation saves classification results this way. This kind of format is convenient: whenever different trading strategies or simulations should run on the same classification array, model building and classification steps are required once.

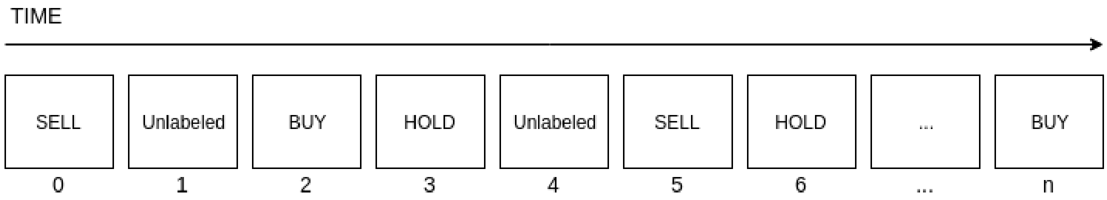


Figure 5.1: Example of classification array. Its first item refers to the first day in test set.

5.3 Trading simulation

Once labels have been obtained two different trading strategies have been tested. The process starts from the first day belonging to the test set and consists in taking, day after day, a trading decision. Such decisions may involve buying or selling a stock. In this work, quantities have not been considered. Instead, only the percentage profit has been collected. The absolute gross profit would be: $profit \cdot initial\_investment$.
Both strategies share some fundamentals:

- trading operations can be either *LONG* or *SHORT*. The former are used to invest on rising prices, the latter on falling prices;
- only one trading operation can be active at a time;
- closing a LONG operation generates a positive profit if closing price is higher than the opening one and negative if it is lower. The opposite rule is applied to SHORT positions;
- if label is not present, so the day is *Unlabeled* – see section 3.4 – behave as if label would be *HOLD*;
- *BUY* labels suggest that price will rise. If there is not an active trading operation, a LONG one is opened; if a SHORT operation is open, then it is closed.
- *SELL* labels suggest that price will fall. If there is not an active trading operation, a SHORT one is opened; if a LONG operation is open, then it is closed;
- *Stop Loss* trading strategy is applied. It is a common approach to limit loss during real world trading sessions. It forces the closing of an operation if price, within the day, goes in the wrong direction by more than a given threshold, with respect to the price operation was opened at. The proposed solution uses a stop loss threshold of 1%. Hence, if a LONG operation is open and, during the day, prices falls below 99% of opening price, operation is closed. The opposite situation happens if a SHORT operation is active and price rises above 101% of initial price.

Then, the two strategies differ from each other with respect to the length, in days, allowed for trading operations. They have been defined as *Multiple days* and *One day* strategies. In order to make a fair comparison these two strategies have been employed for all simulations, even when classification was performed with other techniques.

5.3.1 Multiple days strategy

As the name suggest, this strategy allows active operations to last more than one day. This happens in the following cases:

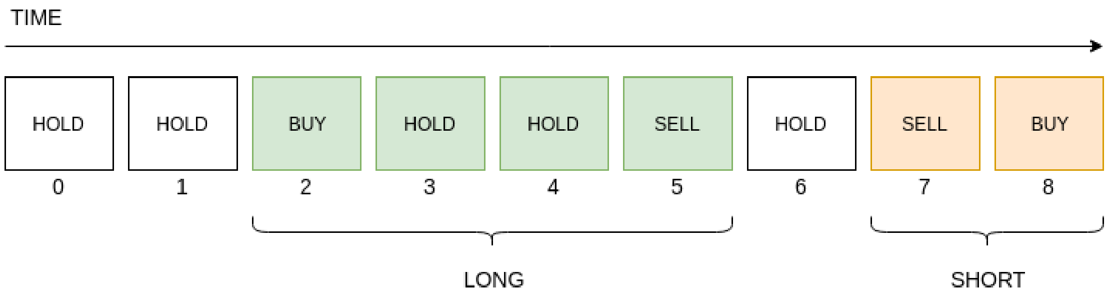
- a LONG operation is active and current day's label is either HOLD or BUY;
- a SHORT operation is active and current day's label is either HOLD or SELL.

In those cases, operation is kept open and the decision is delayed to the day after, as described in Figure 5.2a. This strategy is more sensible to price variations not caught by the model, since days labeled with HOLD class could, in practice, undercover a negative profit.

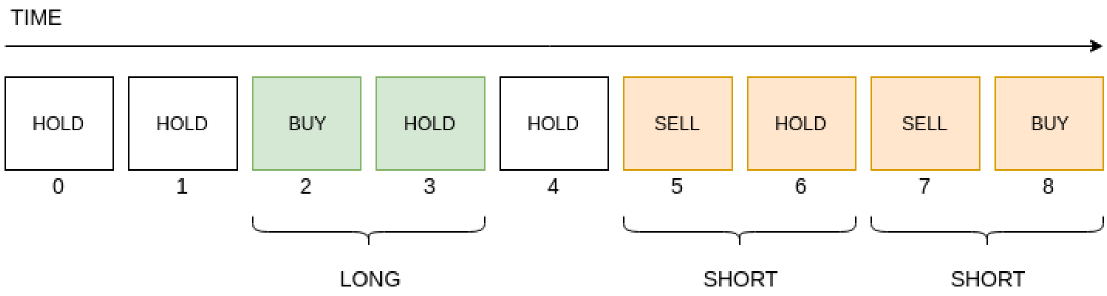
5.3.2 One day strategy

This strategy uses only operations that last a single day. After any operation is opened, the day after it is closed and profit is evaluated, as described in Figure 5.2b.

With respect to multiple days strategy, it is less subject to price variations not caught by the model but with several consecutive enter signals (BUY or SELL), multiple operations are opened. This could potentially affect the final average profit per operation and reduce gains, since each trade in the market is affected by transaction fees.



(a) Trading operations generated with Multiple Days strategy.



(b) Trading operations generated with One Day strategy.

Figure 5.2: A comparison between the two trading strategies. Arrays are the result of classification stage. Each cell is representative of one day and its label is reported inside. Green cells highlight LONG operations while orange ones stand for SHORT operations. It is worth noting how strategies behave differently on day n° 3, when One Day closes the LONG operation.

Chapter 6

Experiments

The novel approach presented in this work to quantitative trading is built upon associative classification. It is a completely different way to tackle time series forecasting with respect to traditional methods since it relies on different assumptions about input data.

In order to prove its efficiency several tests have been done. The automated trading system that uses L^3 classifier has been compared to other automated system that adopt different strategies from traditional time series analysis to forecast future data. For sake of completeness, experiments have explored different configurations for parameters typical of each method – e.g. ARIMA has been tested with different orders:

$$(p, d, q) \in \{[1,0,0], [1,1,0], \cdots\}$$

In the following sections the type of data and the rationale beyond its choice are described. Then models chosen for testing are listed. Also, information on how they have been built, upon which data and how they have been validated are given. Finally, their application in the context of a trading system is described.

6.1 Financial data

As described previously in the document (see section 5.1), stock pre-processing makes use of several technical indicators to build a new input data set. As a consequence, the original raw data set has to contain every stock component to evaluate such indicators.

Such data come from Italian stock exchange market. Stock analyzed are part of Italian first stock market index, the Financial Times Stock Exchange Milano Indice di Borsa or *FTSE MIB*. Stock index are commonly used as market benchmarks, since they include the biggest, most traded stocks in the market.

The years picked for experiments are 2011, 2013 and 2015. Using non contiguous time periods helps differentiating and consolidate results since markets have had diverse trends and conditions. Yet, FTSE MIB is not immutable: companies and holdings belonging to the index have changed over the considered years. Tables 6.1 summarizes which stock have been analyzed for each year.

Normally, FTSE MIB gathers 40 of the major Italian companies that for capitalization and volumes traded influence the market. Still, some of them were not available for the years collected, hence only a sub set of FTSE MIB companies has been traded by the automated system in simulations. Since stocks differ from year to year, the choice to run experiments and collect performance separately finds one more justification.

Table 6.1: Stock analyzed for each year. They are 27 for year 2011, 29 for year 2013 and 31 for year 2015.

	2011	**2013**	**2015**
1	A2A.MI	A2A.MI	A2A.MI
2	AGL.MI	AGL.MI	AGL.MI
3	ATL.MI	ATL.MI	ATL.MI
4	AZM.MI	AZM.MI	AZM.MI
5	BPE.MI	BPE.MI	BPE.MI
6	BZU.MI	BZU.MI	BZU.MI
7	CPR.MI	CPR.MI	CNHI.MI
8	ENEL.MI	ENEL.MI	CPR.MI
9	ENI.MI	ENI.MI	ENEL.MI
10	EXO.MI	EXO.MI	ENI.MI
11	FCA.MI	FCA.MI	EXO.MI
12	G.MI	G.MI	FBK.MI
13	ISP.MI	ISP.MI	FCA.MI
14	LUX.MI	LUX.MI	G.MI
15	MB.MI	MB.MI	ISP.MI
16	MS.MI	MS.MI	LUX.MI
17	PRY.MI	PRY.MI	MB.MI
18	REC.MI	REC.MI	MONC.MI
19	SPM.MI	SFER.MI	MS.MI
20	SRG.MI	SPM.MI	PRY.MI
21	STM.MI	SRG.MI	REC.MI
22	TEN.MI	STM.MI	SFER.MI
23	TIT.MI	TEN.MI	SPM.MI
24	TOD.MI	TIT.MI	SRG.MI
25	TRN.MI	TOD.MI	STM.MI
26	UCG.MI	TRN.MI	TEN.MI
27	YNAP.MI	UCG.MI	TIT.MI
28		US.MI	TOD.MI
29		YNAP.MI	TRN.MI
30			UCG.MI
31			US.MI

6.2 Time-based trading systems

Here are listed time-dependent forecasting techniques adopted in tested trading systems. It is worth mentioning that such methods work on traditional time series and predict a continuous value, not directly a class label, since they are not classifiers. In order to perform a fair comparison, the same discretization algorithm applied in stock pre-processing for L^3 classifier has been adopted – see sub section 5.1.

Simulations have been implemented with Python programming language. The implementations present in *statsmodels* [46] statistical library have been used to test autoregressive models, linear regression and smoothing techniques.

6.2.1 Autoregressive models

Forecasting has been tested with AR(p), ARMA(p,q) and ARIMA(p,d,q) models. *statsmodels* lets choosing the order of ARIMA with p, d and q values, respectively the autoregressive, differentiation and moving average orders. Hence:

$$AR(2) \Longleftrightarrow ARIMA(2,0,0)$$
$$ARMA(1,2) \Longleftrightarrow ARIMA(1,0,2)$$

The list of tested configurations is reported in Table 6.2.

6.2.2 Linear regression

Linear regression model does not require parameters. The fitting strategy chosen is Ordinary Least Squares.

6.2.3 Smoothing techniques

Two different smoothing strategies have been tested. The first one is Simple Exponential Smoothing (SES). The second one merges SES with Holt-Winter's model (HW) in a combined approach called *Adaptive Exponential Smoothing*.

Adaptive Exponential Smoothing uses Dickey-Fuller unit root test implementation found in *statsmodels* to pick between SES and HW. Dickey-Fuller algorithm tests that the input series is stationary with a given significance level. If the series shows stationarity over a 95% confidence level (significance level is below 5%) SES is fitted, otherwise HW is picked. Whenever it is used, HW includes a seasonality factor of 5, to reflect the weekly frequency of our data, sampled only on business days.

6.3 L^3-based trading systems

As described in chapter 5, L^3 associative classifier has been adapted after a pre-processing stage on time series. L^3 rule extraction algorithm requires two main values: the minimum

Table 6.2: Summary of each configurations tested for ARIMA, Linear Regression and Smoothing techniques.

ARIMA	**Linear regression**	**Simple Exponential Smoothing**	**Holt-Winter**	
(p,d,q) order	Fitting strategy	Decay rate	Decay rate	Trend slope
1,0,0	Ordinary Least Squares	0.2	0.2	0.2
1,0,0		0.4	0.4	0.4
1,1,1		0.6	0.6	0.6
2,0,0		0.8	0.8	0.8
2,1,0				
2,1,1				
3,0,0				

support and minimum confidence thresholds. Among the possible settings of the L^3 classifier, we have selected the default configuration setting, i.e. minsup=1%, minconf=50%, which entails considering rules occurring at least in 1% of the sampling times and holding in more than half of the times.

Tests have been conducted with the original C implementation provided by authors.

6.3.1 Top ranked classification rules

Further analysis has been conducted on first level rules extracted by the classifier. They have been ranked by support and confidence and the top-10 rules have been included in a new simulation. It is known from the theory that such rules are high quality *used* rules. Analyzing their performances individually can help to assess:

- how much state variables and class values are correlated according to the selected rules;
- how well the rule captured relevant information that are likely to be present also in yet unseen data;
- how many relevant information is loss when only top-10 rules are considered.

Simulations with top rules include the following steps:

1. build the L^3 model on training data as usual;
2. extract top-10 rules and discard others in first and second level rule sets;
3. include in first level rule set only one rule at a time from top-10. Then, classify test set with One Day strategy.

6.4 Types of trading sessions

A trading session is run against test set right after the respective classification array has been generated by one of the forecasting methods.

If three labels are present, the trading session will generate – and eventually gain profit from – both LONG and SHORT operations. Instead, if two class labels are present, the session can be either *Bullish* or *Bearish* if respectively only LONG or SHORT positions can be opened.

6.5 Validation strategy

Two different validation strategies, both commonly used on Machine Learning and Data Mining algorithms, have been analyzed. They differ from the definition of which portion of the data set is considered as training and test.

Hold-out is a validation technique that assigns a fixed portion of data to training set and the remaining one to test set. The model is built once with information available in training data and all the test set labels are assigned with that model. In this work training set has been fixed as the initial 66% of the days of the current year.

Expanding window uses instead a dynamic, increasing training set. Once the initial training window has been decided, a model is built and it is used to predict only the first test day following that window. Next, training window is increased by one day and the same operations are repeated. The number of different models that are built is:

$$|M| = |D| - n$$

where $|D|$ is the data set size and n is the number of days in the initial window – in all experiments conducted it has been fixed as the initial 33% of the days of the current year.

Chapter 7

Results

In this chapter experimental results have been reported in structured tables. Simulations have been run for each year and for each validation strategy separately. Considering separate years has allowed trading system to work on different market conditions. Differentiation is also incremented by the fact that time periods are not adjacent.

A similar argument can be done for validation strategies. Hold-out and expanding window consist in two operational approaches quite different. Even in real trading system their practical use would be different. The first one brings low computational complexity of building the model just once, but each prediction is made with a fewer information learned. Expanding window enlarges its knowledge at each prediction, but this could lead to disadvantages: noise can be introduced and much more time is needed to build and then analyze the obtained model. Also for these reasons results from different validation strategies could not be compared directly.

7.1 Tables format

Results are organized as follows: each section contains tables relative to a single year. There have been reported table with different types of sorting. Furthermore, tables results are divided by validation strategy.
Each table presents a bold row to highlight the top performing strategy. They have been taken into account only models that have operated more than 50 trading operations. This choice is manly due to make more robust results analysis and discussion: there have been found models that over-performed by far other techniques but with a very limited amount of operations. Even though metrics like *average profit per operations* in principle make results comparable, techniques that have operated a very limited amount of trade cannot suggest solid conclusions. Techniques that did not operate 50 trades have been reported in tables for sake of completeness and comparison, but they are presented in italic.

Tables contain one row per model. Each row describes:

- *Classifier*: the forecasting technique used as core of the trading system;

- *Operations type*: describes which operations have been used by the system, either SHORT and LONG or only one of the two;
- *Trading strategy*: either Multiple Days or One Day;
- *Total profit*: evaluated as the arithmetic sum of the profits among all stocks;
- *Total operations*: evaluated as the arithmetic sum of the trades operated among all stocks;
- *Average profit per operation*: evaluated as $Total\ profit$ / $Total\ operations$;
- *Average unlabeled ratio per stock*: evaluated as the arithmetic mean among all stocks of the ratio between the number of unlabeled days and the number of days in test set.

Profit and average profit per operation are gross measures. In real world trades, they would be lower due to several costs applied at each transaction.

7.2 Year 2011

Hold-out

In 2011 L^3-based system performed well compared to other time-based technique. Even if it obtained a substantial lower gross profit, between the two models the average profit per operation of L^3 was higher than Adaptive Exponential Smoothing. Additionally, L^3 bullish system got the first place in average profit per operation (2.66%) ranking beating AES. ARIMA, SES and Linear regression could not operate a valuable number of operations. Forecasting using top-10 L^3 rules lead to good results in both the metrics, even if they are lower compared to the model that uses the entire rule set. They also present an high value in average unlabeled ratio per stock column (97.44%). This is mainly due to the fact that rules are quite specific – i.e. contains several conditions that records has to satisfy – hence the number of days which could not be classified is large.

Table 7.1: Year 2011, hold-out: best results in terms of total profit obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**AES**	**LONG-SHORT**	**Multiple days**	**593.64%**	**634**	**0.94%**	**0.00%**
2	L3	LONG-SHORT	Multiple days	300.91%	268	1.12%	7.74%
3	L3-TOP10	LONG-SHORT	One day	83.70%	182	0.46%	97.44%
4	*ARIMA*	*LONG-SHORT*	*Multiple days*	*10.43%*	*4*	*2.61%*	*7.41%*
5	*SES*	*LONG-SHORT*	*Multiple days*	*2.71%*	*8*	*0.34%*	*0.00%*
6	*Linear Regression*	*SHORT*	*Multiple days*	*0.69%*	*1*	*0.69%*	*0.00%*

Table 7.2: Year 2011, hold-out: best results in terms of average profit per operation obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**L3**	**LONG**	**Multiple days**	**210.15%**	**79**	**2.66%**	**15.39%**
2	*ARIMA*	*LONG-SHORT*	*Multiple days*	*10.43%*	*4*	*2.61%*	*7.41%*
3	AES	LONG	Multiple days	181.53%	98	1.85%	0.00%
4	L3-TOP10	SHORT	One day	77.66%	106	0.73%	97.65%
5	*Linear Regression*	*SHORT*	*Multiple days*	*0.69%*	*1*	*0.69%*	*0.00%*
6	*SES*	*LONG-SHORT*	*Multiple days*	*2.71%*	*8*	*0.34%*	*0.00%*

Expanding window

Using expanding window, time-dependent techniques surpassed the best outcome of L^3, both in total profit and average profit per operation. L^3 did beat only ARIMA models. Nevertheless, it showed good positive performances in both metrics (505.41% in total profit and 0.97% in average profit per operation).

Table 7.3: Year 2011, expanding window: best results in terms of total profit obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**SES**	**LONG-SHORT**	**Multiple days**	**1420.89%**	**1102**	**1.29%**	**0.00%**
2	AES	LONG-SHORT	Multiple days	1155.55%	1241	0.93%	0.00%
3	Linear regression	LONG-SHORT	Multiple days	717.82%	715	1.00%	0.00%
4	L3	LONG-SHORT	Multiple days	505.41%	801	0.63%	6.94%
5	ARIMA	LONG-SHORT	Multiple days	92.61%	153	0.61%	0.91%

Table 7.4: Year 2011, expanding window: best results in terms of average profit per operation obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**SES**	**LONG-SHORT**	**Multiple days**	**456.04%**	**255**	**1.79%**	**0.00%**
2	Linear regression	LONG	Multiple days	677.65%	448	1.51%	0.00%
3	AES	LONG	Multiple days	847.09%	733	1.16%	0.00%
4	L3	LONG	Multiple days	415.06%	426	0.97%	15.48%
5	ARIMA	LONG-SHORT	Multiple days	78.98%	95	0.83%	2.25%

7.3 Year 2013

Hold-out

Year 2013 confirms results obtained in 2011: L^3 overcome time-dependent techniques except for AES. Also here L^3 bullish variant loses against LONG-SHORT model of AES in total profit, but if compared on average profit per operation L^3 wins. L^3 top-10 rules ran definitely worse, with a profit near zero.

Table 7.5: Year 2013, hold-out: best results in terms of total profit obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**AES**	**LONG-SHORT**	**Multiple days**	**240.88%**	**546**	**0.44%**	**0.00%**
2	L3	LONG	Multiple days	133.69%	116	1.15%	19.86%
3	*SES*	*LONG-SHORT*	*Multiple days*	*94.49%*	*15*	*6.30%*	*0.00%*
4	*Linear regression*	*LONG-SHORT*	*Multiple days*	*50.81%*	*27*	*1.88%*	*0.00%*
5	*ARIMA*	*LONG-SHORT*	*Multiple days*	*11.09%*	*2*	*5.55%*	*55.17%*
6	L3-TOP10	LONG	One day	-0.11%	130	0.00%	98.73%

Table 7.6: Year 2013, hold-out: best results in terms of average profit per operation obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	*SES*	*LONG-SHORT*	*Multiple days*	*54.92%*	*6*	*9.15%*	*0.00%*
2	*ARIMA*	*LONG-SHORT*	*Multiple days*	*11.09%*	*2*	*5.55%*	*0.00%*
3	**AES**	**LONG**	**Multiple days**	**134.98%**	**53**	**2.55%**	**0.00%**
4	*Linear regression*	*LONG-SHORT*	*Multiple days*	*50.81%*	*27*	*1.88%*	*0.00%*
5	L3	LONG	Multiple days	133.69%	116	1.15%	19.86%
6	L3-TOP10	LONG	One day	-0.11%	130	0.00%	98.73%

Expanding window

As for 2011, L^3 model did go short on total profit and average profit per operation against Linear regression, AES and SES. Despite that, its own performances are positive (346.6% of gross profit obtained with 408 operations).

Table 7.7: Year 2013, expanding window: best results in terms of total profit obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**AES**	**LONG-SHORT**	**Multiple days**	**782.08%**	**1077**	**0.73%**	**0.00%**
2	Linear regression	LONG	Multiple days	666.96%	298	2.24%	0.00%
3	SES	LONG	Multiple days	609.19%	537	1.13%	0.00%
4	L3	LONG	Multiple days	346.62%	408	0.85%	16.96%
5	*ARIMA*	*LONG-SHORT*	*Multiple days*	*135.75%*	*23*	*5.90%*	*0.94%*

Table 7.8: Year 2013, expanding window: best results in terms of average profit per operation obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
11	*ARIMA*	*LONG-SHORT*	*Multiple days*	*135.75%*	*23*	*5.90%*	*0.94%*
166	**Linear regression**	**LONG**	**Multiple days**	**666.96%**	**298**	**2.24%**	**0.00%**
69	AES	LONG-SHORT	Multiple days	440.81%	239	1.84%	0.00%
76	SES	LONG-SHORT	Multiple days	148.25%	88	1.68%	0.00%
172	L3	LONG	Multiple days	346.62%	408	0.85%	16.96%

7.4 Year 2015

Hold-out

In 2015 L^3 bullish model over performed also AES in gross profit (203.93% against 170.36%) and did run just below it in average profit per operation (1.59% against 2.04%). Even though L^3 top-10 rules achieved positive results (68% gross profit, 0.44% average profit per operation) they performed worse compared to L^3 with complete rule set.

Table 7.9: Year 2015, hold-out: best results in terms of total profit obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**L3**	**LONG**	**Multiple days**	**203.93%**	**128**	**1.59%**	**22.67%**
2	AES	LONG	Multiple days	170.36%	480	0.35%	0.00%
3	L3-TOP10	LONG	One day	68.00%	156	0.44%	98.62%
4	*ARIMA*	*LONG*	*Multiple days*	*3.57%*	*1*	*3.57%*	*0.00%*
5	*SES*	*SHORT*	*Multiple days*	*0.00%*	*0*	*0.00%*	*0.00%*
6	*Linear regression*	*SHORT*	*Multiple days*	*0.00%*	*0*	*0.00%*	*0.00%*

Table 7.10: Year 2015, hold-out: best results in terms of average profit per operation obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	*ARIMA*	*LONG*	*Multiple days*	*3.57%*	*1*	*3.57%*	*0.00%*
2	**AES**	**LONG**	**Multiple days**	**153.11%**	**75**	**2.04%**	**0.00%**
3	L3	LONG	Multiple days	203.93%	128	1.59%	22.67%
4	L3-TOP10	LONG	One day	68.00%	156	0.44%	98.62%
5	*SES*	*SHORT*	*Multiple days*	*0.00%*	*0*	*0.00%*	*0.00%*
6	*Linear regression*	*SHORT*	*Multiple days*	*0.00%*	*0*	*0.00%*	*0.00%*

Expanding window

As for 2011 and 2013, also in 2015 SES, Linear Regression and AES took major benefits from expanding window validation strategy. L^3 based system obtained results comparable only to ARIMA models. Nonetheless, they are still positive: 310.49% of gross profit and 0.61% of average profit per operation.

Table 7.11: Year 2015, expanding window: best results in terms of total profit obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**SES**	**LONG-SHORT**	**Multiple days**	**1109.43%**	**1145**	**0.97%**	**0.00%**
2	Linear regression	LONG	Multiple days	901.82%	491	1.84%	0.00%
3	AES	LONG-SHORT	Multiple days	734.06%	1273	0.58%	0.00%
4	L3	LONG	Multiple days	310.49%	506	0.61%	14.42%
5	ARIMA	LONG	Multiple days	38.79%	59	0.66%	0.00%

Table 7.12: Year 2015, expanding window: best results in terms of average profit per operation obtained by each technique.

	Classifier	Operations type	Trading strategy	Total profit	Total operations	Average profit per operation	Average unlabeled ratio per stock
1	**SES**	**LONG-SHORT**	**Multiple days**	**387.28%**	**101**	**3.83%**	**0.00%**
2	Linear regression	LONG	Multiple days	901.82%	491	1.84%	0.00%
3	AES	LONG	Multiple days	710.91%	757	0.94%	0.00%
4	ARIMA	LONG	Multiple days	38.79%	59	0.66%	0.00%
5	L3	LONG	Multiple days	310.49%	506	0.61%	14.42%

7.5 Analysis of time series method parameters

This section reports four charts depicting how typical parameters of each model have influenced performances. ARIMA and Adaptive Exponential Smoothing configuration are characterized by multiple parameters, hence each chart uses as independent variable only one parameters and fixes the others as constants. The two dependent variables, reported on Y-axis, are total profit (on the right) and average profit per operations (on the left). The results presented include only expanding window validation strategy, where many time-based models have shown better values in these metrics. The year 2015 has been taken as reference and, for each model, bullish sessions have been considered.

Furthermore, for comparative purposes each chart displays a green dotted line that represents the average profit per operation achieved by L^3 bullish model, the same year.

ARIMA

In Figure 7.1 ARIMA models with a different p autoregressive order have been compared. Only the model ARIMA(3,0,0) has achieved a profit per operation higher than L^3, even if with a number of operations lower than 50.

Simple Exponential Smoothing

Figure 7.2 shows SES performances given by different decay rate. The total profit presents a decreasing trend for an increasing decay, suggesting that forecasting is more accurate when time step further in the past are considered. It is also worth noting that for $\alpha = 0.8$ the gross profit is lower than L^3.

Adaptive Exponential Smoothing

Holt-Winter's model parameters explored in this work are α, the decay rate and β, the slope assigned to the trend factor of the model. Here, two charts have been reported, starting from the top-performing configuration $\alpha = 0.2, \beta = 0.2$: in Figure 7.3 α is fixed, while in Figure 7.4 β is. Both the charts highlights an important fact: between the 8 possible configurations, half of them has achieved a worse result compared to L^3 standard configuration.

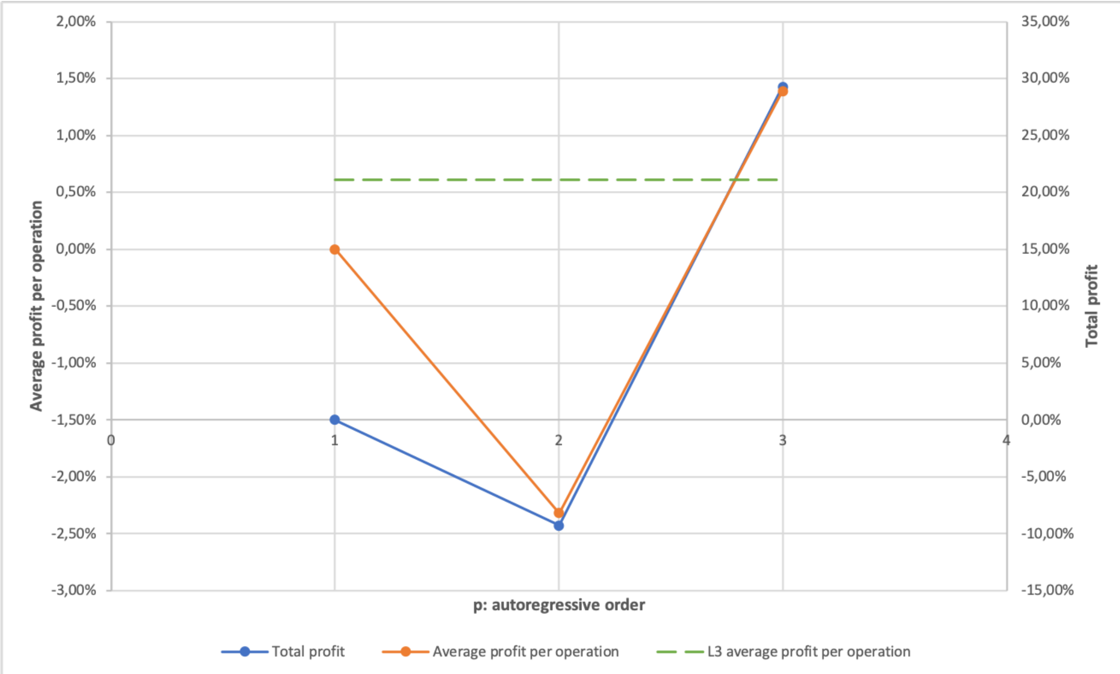


Figure 7.1: ARIMA(p,0,0) metrics in function of p autoregressive factor.

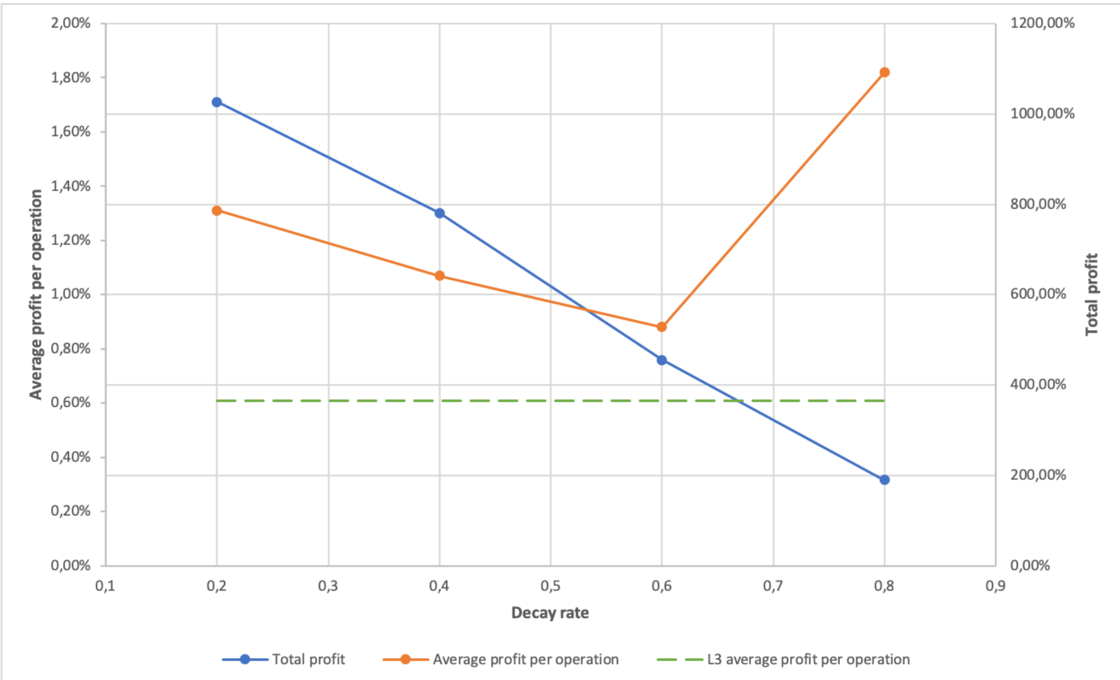


Figure 7.2: Simple Exponential Smoothing metrics in function of α decay rate.

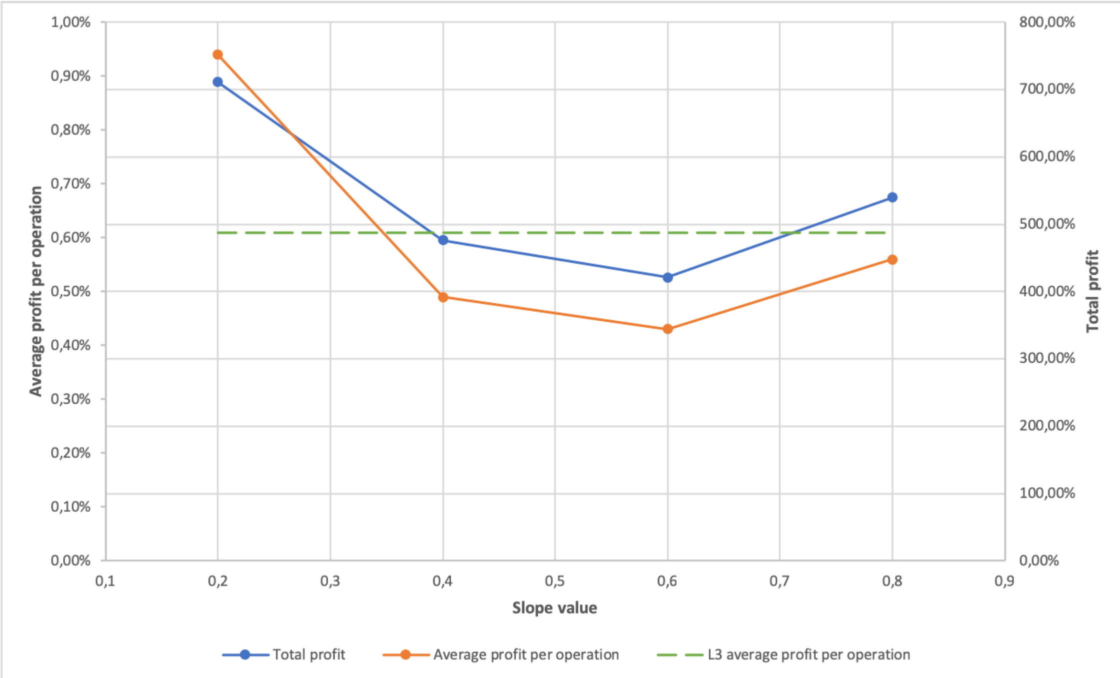


Figure 7.3: AES($\alpha = 0.2$, β) metrics in function of β slope value.

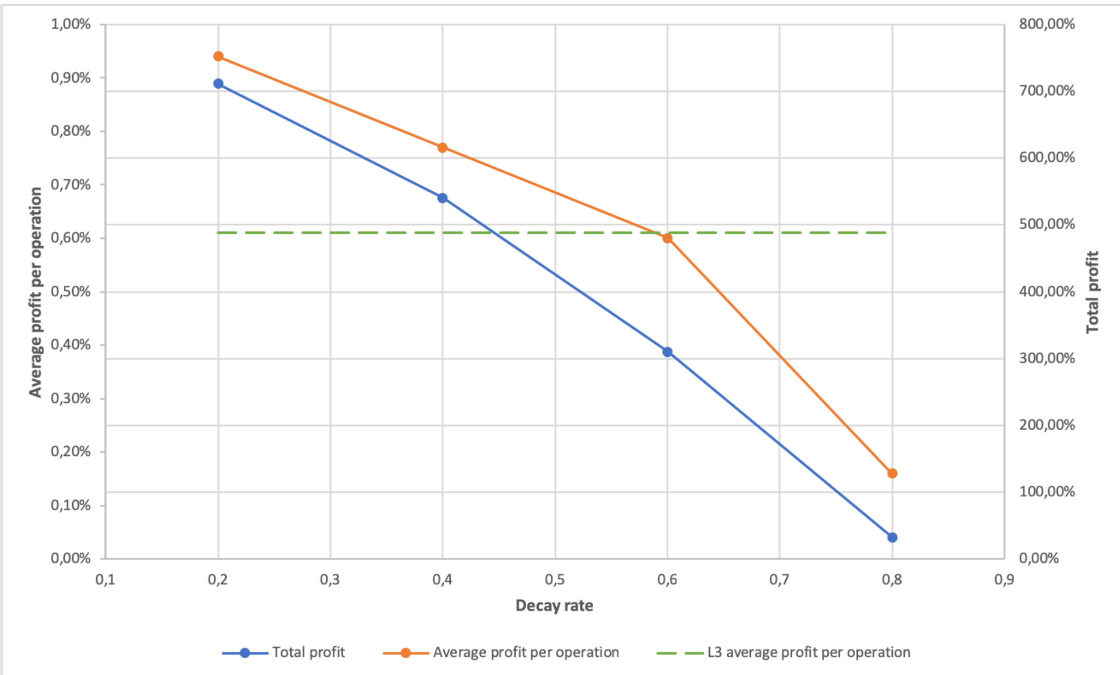


Figure 7.4: AES(α, $\beta = 0.2$) metrics in function of α decay rate.

Chapter 8

Conclusions and future works

This text presented a novel methodology for quantitative trading on stock exchange markets. The main objective was to compare its performances with systems built upon traditional time series forecasting models.

Experimental results have highlighted many facts worth discussing. L^3-based trading technique has performed well on all the three years tested. Nevertheless, the comparison with traditional time series techniques did not have always the same winner, regarding gross profit and average profit per operation performances.
Two main behaviors can be discovered, distinguishable by analyzing separately the two validation strategies adopted: the training done on fixed or expanding windows. Trading systems powered by traditional time series techniques performed better than L^3 models when forecasting was done with an expanding training window. Instead, when training was done only on two-thirds of the year L^3 performed decisively better. It is our opinion that, in the first case, traditional models suffered the temporal leap between furthest forecasts and the day the model was built on. The lack of knowledge of nearest past days has lead to wrong price predictions and, in turn, to wrong classification labels. L^3 rules instead remained valuable indications for the system. Using an expanding training set, traditional techniques performed better than L^3 in average. Both these considerations seem to suggest still that future price movements are strongly influenced by variations in recent days.
Nonetheless, the two classes of methods are totally different in terms of side information granted by the model. Time series forecasting techniques only suggest a possible direction, L^3 models forecast the direction *and* adds rules that are open to human interpretation. Since L^3 based systems still performed well with both expanding and fixed training, we argue that the additional information they provide could be decisive in real world situations. We think hereby that a potential trader that want to integrate one of the solutions has two possible choices. One featured with a *black-box* model that provides good results if updated continuously with recent data; the other choice instead provides an understandable model that is consequently easier to inspect and fine tune.

We think that domain experts would eventually choice the second solution mainly for

two reasons. Real world automated trading systems typically are configured by experts a few times per calendar year. Typically, decision taking rules are evaluated on historical data and, once they are supposed to be profitable, they are inserted in the system to forecast future trends. This approach could not be possible anymore if the model has to be trained on daily frequency: it would become time consuming for a human operator. As a consequence, the model still would be constructed once and the possibility to analyze it would drive the decision towards L^3 systems.
Secondly, time series forecasting techniques are prone to configuration errors. One clear example is given by Adaptive Exponential Smoothing, the only technique that has obtained better results than L^3 systems, even with a fixed training window. In fact, experiments show that the same model, with the same type of operations allowed and the same trading strategy, has slight different configurations that performed way worse than L^3 standard configuration. Imagining a real world application this is a point worth to consider.

Finally, conclusions on L^3 top-quality rules could be done. In two years out of the three, results showed that it is not advisable to run trading sessions having pruned all the rules but the top-10. We think that even if classification noise is reduced, too much information get lost discarding the majority of rules.

We strongly believe that the final points this work has lead to are worth exploring, with deeper and diverse analysis. The main limitation of this work was due to the amount of market data tested and its nature. Future works could easily extend this study evaluating more calendar years. Probably the best choice would be to test both years before 2011 – when the great financial crisis, started from housing market, was at its peak – and after 2015 to collect performances on latest markets trends. Furthermore, this study can be consolidated by analyzing stocks belonging to different – and bigger – stock market indices, like S&P500 or Dow Jones. Finally, one more possible differentiation could be done performing experiments on different financial securities, like Forex or Futures.

Bibliography

[1] J. J. Murphy, *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin, 1999.

[2] H. Markowitz, “Portfolio selection”, *The journal of finance*, vol. 7, no. 1, pp. 77–91, 1952.

[3] Yahoo. (2018). Yahoo Finance, [Online]. Available: https://it.finance.yahoo.com/.

[4] L. A. Teixeira and A. L. I. De Oliveira, “A method for automatic stock trading combining technical analysis and nearest neighbor classification”, *Expert systems with applications*, vol. 37, no. 10, pp. 6885–6890, 2010.

[5] J. W. Wilder, *New concepts in technical trading systems*. Trend Research, 1978.

[6] W. Blau, “True strength index”, *Tech Anal Stocks Commod (traders. com)*, vol. 11, no. 1, pp. 438–446, 1991.

[7] J. E. Granville, *New key to stock market profits*. Prentice-Hall, 1963.

[8] R. E.A. C. Paley and N. Wiener, *Fourier transforms in the complex domain*. American Mathematical Soc., 1934, vol. 19.

[9] L. R. Rabiner and B. Gold, “Theory and application of digital signal processing”, *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1975.

[10] G Clark, S Parker, and S Mitra, “A unified approach to time-and frequency-domain realization of FIR adaptive digital filters”, *IEEE transactions on acoustics, speech, and signal processing*, vol. 31, no. 5, pp. 1073–1083, 1983.

[11] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[12] C Chafield, “The analysis of time series: theory and practice”, *Chapman and Hull, London*, 1975.

[13] P. J. Brockwell, R. A. Davis, and M. V. Calder, *Introduction to time series and forecasting*. Springer, 2002, vol. 2.

[14] Sewaqu, *Linear Regression*. [Online]. Available: \url{https://commons.wikimedia.org/w/index.php?curid=11967659}.

[15] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval", in *European conference on machine learning*, Springer, 1998, pp. 4–15.

[16] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines", *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[18] X. Yin and J. Han, "CPAR: Classification based on predictive association rules", in *Proceedings of the 2003 SIAM International Conference on Data Mining*, SIAM, 2003, pp. 331–335.

[19] E. Baralis, S. Chiusano, and P. Garza, "A lazy approach to associative classification", *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 156–171, 2008.

[20] E. Baralis, L. Cagliero, T. Cerquitelli, P. Garza, and F. Pulvirenti, "Discovering profitable stocks for intraday trading", *Information Sciences*, vol. 405, pp. 91–106, 2017.

[21] T.-l. Chen and F.-y. Chen, "An intelligent pattern recognition model for supporting investment decisions in stock market", *Information Sciences*, vol. 346, pp. 261–274, 2016.

[22] T. Geva and J. Zahavi, "Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news", *Decision support systems*, vol. 57, pp. 212–223, 2014.

[23] K. Żbikowski, "Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy", *Expert Systems with Applications*, vol. 42, no. 4, pp. 1797–1805, 2015.

[24] T. Hill, M. O'Connor, and W. Remus, "Neural network models for time series forecasts", *Management science*, vol. 42, no. 7, pp. 1082–1092, 1996.

[25] W. Leigh, R. Purvis, and J. M. Ragusa, "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*, vol. 32, no. 4, pp. 361–377, 2002.

[26] M. M. Mostafa, "Forecasting stock exchange movements using neural networks: Empirical evidence from Kuwait", *Expert Systems with Applications*, vol. 37, no. 9, pp. 6302–6309, 2010.

[27] O. Castillo and P. Melin, "Simulation and forecasting complex economic time series using neural networks and fuzzy logic", in *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, IEEE, vol. 3, 2001, pp. 1805–1810.

[28] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang, and S.-P. Guo, “Forecasting stock indices with back propagation neural network”, *Expert Systems with Applications*, vol. 38, no. 11, pp. 14 346–14 355, 2011.

[29] F. A. de Oliveira, C. N. Nobre, and L. E. Zárate, “Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index–Case study of PETR4, Petrobras, Brazil”, *Expert Systems with Applications*, vol. 40, no. 18, pp. 7596–7606, 2013.

[30] J. L. Ticknor, “A Bayesian regularized artificial neural network for stock market forecasting”, *Expert Systems with Applications*, vol. 40, no. 14, pp. 5501–5506, 2013.

[31] X. Zhong and D. Enke, “Forecasting daily stock market return using dimensionality reduction”, *Expert Systems with Applications*, vol. 67, pp. 126–139, 2017.

[32] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock market index using fusion of machine learning techniques”, *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, 2015.

[33] C.-F. Tsai, Y.-C. Lin, D. C. Yen, and Y.-M. Chen, “Predicting stock returns by classifier ensembles”, *Applied Soft Computing*, vol. 11, no. 2, pp. 2452–2459, 2011.

[34] P.-C. Chang, C.-H. Liu, C.-Y. Fan, J.-L. Lin, and C.-M. Lai, “An ensemble of neural networks for stock trading decision making”, in *International Conference on Intelligent Computing*, Springer, 2009, pp. 1–10.

[35] Y. Chen and Y. Hao, “A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction”, *Expert Systems with Applications*, vol. 80, pp. 340–355, 2017.

[36] W.-C. Chiang, D. Enke, T. Wu, and R. Wang, “An adaptive stock index trading decision support system”, *Expert Systems with Applications*, vol. 59, pp. 195–207, 2016.

[37] M. Pulido, P. Melin, and O. Castillo, “Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange”, *Information Sciences*, vol. 280, pp. 188–204, 2014.

[38] Y. Kara, M. A. Boyacioglu, and Ö. K. Baykan, “Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange”, *Expert systems with Applications*, vol. 38, no. 5, pp. 5311–5319, 2011.

[39] Y.-K. Kwon and B.-R. Moon, “A hybrid neurogenetic approach for stock forecasting”, *IEEE transactions on neural networks*, vol. 18, no. 3, pp. 851–864, 2007.

[40] M.-J. Kim, S.-H. Min, and I. Han, “An evolutionary approach to the combination of multiple classifiers to predict a stock price index”, *Expert Systems with Applications*, vol. 31, no. 2, pp. 241–247, 2006.

[41] Y. Kim, W. Ahn, K. J. Oh, and D. Enke, "An intelligent hybrid trading system for discovering trading rules for the futures market using rough sets and genetic algorithms", *Applied Soft Computing*, vol. 55, pp. 127–140, 2017.

[42] X. Zhang, Y. Hu, K. Xie, W. Zhang, L. Su, and M. Liu, "An evolutionary trend reversion model for stock trading rule discovery", *Knowledge-Based Systems*, vol. 79, pp. 27–35, 2015.

[43] L. Dymova, P. Sevastjanov, and K. Kaczmarek, "A Forex trading expert system based on a new approach to the rule-base evidential reasoning", *Expert Systems with Applications*, vol. 51, pp. 1–13, 2016.

[44] S. Ho, M Xie, and T. Goh, "A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction", *Computers & Industrial Engineering*, vol. 42, no. 2-4, pp. 371–375, 2002.

[45] Z. Tang, C. De Almeida, and P. A. Fishwick, "Time series forecasting using neural networks vs. Box-Jenkins methodology", *Simulation*, vol. 57, no. 5, pp. 303–310, 1991.

[46] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python", in *Proceedings of the 9th Python in Science Conference*, SciPy society Austin, vol. 57, 2010, p. 61.