FEDERAL UNIVERSITY OF RIO GRANDE DO SUL
INSTITUTE OF INFORMATICS
BACHELOR OF COMPUTER SCIENCE

RAFAEL THOMAZI GONZALEZ

Ensemble System Based on Genetic Algorithm For Stock Market Forecasting

Final Paper presented in partial fulfillment of the requirement for the degree of Bachelor of Computer Science.

Supervisor: Prof. Dr. Dante Augusto Couto Barone
Co-supervisor: MSc. Carlos Alberto Padilha

Porto Alegre
2014

ACKNOWLEDGMENTS

First of all, I would like to thank all the members of my family for all the continued support throughout my graduation. In particular, I would like to thank my mother, Débora Thomazi Ramos Gonzalez, for the affection, the understanding and the unfailing encouragement. I would also like to thank my friends who shared with me some of this time and helped me overcome the most difficult moments of this journey.

I also thank Professor Dr. Dante Augusto Couto Barone and MSc. Carlos Alberto Padilha for their insight and advice that helped me a great deal in clarifying my research goals and to stay focused during this process. Without their ideas and guidance this work would not have been possible.

ABSTRACT

Financial time series forecasting is regarded as one of the most challenging applications of time series forecasting. Many researchers have been focusing on this topic due to the potential of yielding significant profits on the invested money in a short time frame. Believing in the predictability of stock markets, traders have been using Technical Analysis tools for a very long time to analyze and predict the behavior of stocks, aiming to make the best investment decisions possible with this information.

In addition, applying machine learning techniques to predict stock market movements has become an area of research that has received a lot of attention in recent years. Popular algorithms such as Artificial Neural Networks and Support Vector Machines have been widely used in this area and they have been reporting satisfactory performances. As an attempt to improve the accuracy of these algorithms, researchers have been proposing techniques to combine them, forming Ensemble Systems. This work presents the design of an Ensemble System based on Genetic Algorithm for forecasting the weekly prices' trend in the Sao Paulo Stock Exchange Index (Ibovespa Index). In order to evaluate the performance of the proposed method, experiments were conducted to compare it with other popular ensemble methods (e.g., Bagging, Boosting and Random Forests).

Finally, the empirical results show that the proposed model outperforms the other ensemble methods. Therefore, this implies that the proposed approach can be used by traders as a promising tool for forecasting stock market prices.

Keywords: Genetic Algorithm. Ensemble System. Financial Market. Technical Analysis. Forecasting.

Sistema de Comitê baseado em Algoritmo Genético
para Predição de Valores do Mercado de Ações

RESUMO

A Previsão de séries temporais financeiras é considerado como uma das mais desafiadoras aplicações de previsão de séries temporais. Muitos pesquisadores estão se concentrando nesse tema devido ao potencial de se obter lucros significativos sobre o dinheiro investido em um curto espaço de tempo. Acreditando na previsibilidade do mercado de ações, os *traders* têm utilizado ferramentas de análise técnica desde a muito tempo para analisar e prever o comportamento das ações, com o objetivo de fazer melhores decisões de investimento com essa informação.

Além disso, a aplicação de técnicas de aprendizado de máquina para prever os movimentos do mercado de ações tornou-se uma área de pesquisa que tem recebido muita atenção nos últimos anos. Algoritmos populares, como Redes Neurais Artificiais e Support Vector Machines têm sido bastante utilizados nessa área e vêm apresentando desempenhos satisfatórios. Na tentativa de melhorar a precisão desses algoritmos, pesquisadores estão propondo técnicas para combiná-los, formando Sistemas de Comitê. Este trabalho apresenta o projeto de um Sistema de Comitê baseado em Algoritmo Genético para prever tendência dos preços semanais no Índice Bovespa. A fim de avaliar o desempenho do método proposto, experimentos foram realizados para compará-lo com outros métodos populares de criação de comitê (por exemplo, *Bagging*, *Boosting* e *Random Forests*).

Finalmente, os resultados empíricos mostram que o modelo proposto supera os outros métodos de comitê. Portanto, isso implica que a abordagem proposta pode ser utilizada por *traders* como uma ferramenta promissora para prever os preços do mercado de ações.

Palavras-chave: Algoritmo Genético. Sistema de Comitê. Mercado Financeiro. Análise Técninca. Previsão.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

IBOV	Acronym for Bovespa Index (Brazil)
GSPC	Acronym for S&P 500 Index (U.S.)
DIA	Acronym for Dow Jones Industrial Average Index (U.S.)
N225	Acronym for Nikkei 255 Index (Japan)
HSI	Acronym for Hang Seng Index (China)
GDAXI	Acronym for DAX Index (Germany)
FTSE	Acronym for FTSE 100 Index (U.K.)
USDBRL	U.S. Dollar – Brazilian Real pair
EURBRL	Euro – Brazilian Real pair
CNYBRL	Chinese Yuan – Brazilian Real pair
EURUSD	Euro – U.S. Dollar pair

INDEX

1 INTRODUCTION

The following sections present an overview of fundamental issues necessary for understanding this project, as well its goals, and how it is organized to achieve them. For more details on the topics discussed, it is recommended to consult the listed references.

1.1 Financial Time Series Forecasting

A time series is an ordered sequence of data points, usually measured in uniform time intervals (NIST/SEMANTECH, 2012). An important property of time series is that data observations are interdependent and, thus, it is essential to maintain the order in which the data was generated (WEI, 2006). Some of the objectives in studying time series include understanding and modeling its evolution, forecasting its future values, and understanding how it impacts, or is impacted, by other factors. Time series arise in many important areas such as economics, finance, engineering, and natural sciences (NIST/SEMANTECH, 2012).

Financial markets are considered to be complex and non-linear dynamical systems (ABU-MOSTAFA; ATIYA, 1996).They are also characterized by data intensity, noise, a high degree of uncertainty, and hidden relationships (HALL, 1994). Moreover, financial markets can be influenced by many factors such as political, economic, and psychological, which can further increase its volatility. Forecasting financial time series is therefore a difficult task. However, predicting is important in the sense that it provides concrete data for market participants to make more informed and accurate investment decisions.

Technical Analysis has been widely used by traders as a tool for predicting the future behavior of the stock prices (MURPHY, 1999). Recently, applications of machine learning techniques to stock market prediction has been receiving a lot of attention, as it has been showing better predictive accuracy. In (SETTY; RANGASWAMY; SUBRAMANYA, 2010), a review is presented of these techniques and its applications in stock markets. There have been many studies using Artificial Neural Networks (ANNs) for financial time series modeling and forecasting. Some successful examples are: (KIMOTO et al, 1990), (YOON; SWALES, 1991), (CHOI; LEE; RHEE, 1995), and (PINTO, 2011). However, ANNs have some disadvantages including the need for the determination of the number of processing elements in the hidden layers, and the value of controlling parameters, which can lead to overfitting (KIM, 2003). In order to overcome these issues, researchers have been using

Support Vector Machines (SVMs), a specific type of learning algorithm developed by Vapnik (1995) that implements the structural risk minimization principle, i.e. it aims to minimize an upper bound of generalization error. This method has become very popular and some applications of SVM to financial forecasting problems have been reported in (KIM, 2003), (CAO; TAY, 2001), (TAY; CAO, 2001). As an attempt to improve the accuracy and stability of a classification system generated by single machine learning algorithms, many researchers have been proposing the combination of multiple classifiers, forming Ensembles (KUNCHEVA, 2004). In recent years, these systems have become a popular topic. Their application in predicting financial time series has also shown successful results such as in (LAHMIRI, 2014) and (LING; YUE; ZHANG, 2013).

Evolutionary Computation (EC) techniques, another subfield of Artificial Intelligence, have been proving to be a powerful tool kit for economic analysis. Researchers have been using these techniques to approach financial problems such as modeling and forecasting financial markets (PROCHNOW, 2013; LI, 2000), creating and optimizing trade strategies (ALLEN; KARJALAINEN, 1999), and portfolio management (SEFIANE; BENBOUZIANE, 2012).

1.2 Objectives

Focusing on financial forecasting, the goal of this project is to design and implement a Genetic Algorithm based Ensemble System. More specifically, an Ensemble System is used to approach a classification problem relevant to stock market forecasting, while a Genetic Algorithm performs feature selection and parameter optimization to generate different subsets for the individual classifiers of the ensemble. Prediction accuracy is not only a major concern in machine learning applications, but it is also of great interest to investors. Thus, the proposed method aims to achieve reasonable accuracy in forecasting the movement direction of the Ibovespa Index's (Sao Paulo Stock Exchange Index) weekly prices.

In addition, this work examines the feasibility of the proposed method by comparing it with other popular ensemble methods, testing them all on the same data set. Finally, all results are compared in order to verify the differences in accuracy rates of each model.

1.3 Report Structure

This work is divided in eight chapters, including the introduction, and is organized as follows:

Chapter 2 presents the concepts of Financial Market, focusing on two different types of market, the Stock Market and the Forex Market, which are the main data sources of this project. The chapter ends with, one of the most used methods of predicting future activities in these markets, the Technical Analysis.

Chapter 3 delves into the principles of Genetic Algorithm, starting with the theoretical foundations of the method, and proceeding through the main issues on designing this kind of algorithm.

Chapter 4 introduces the concept of Ensemble Systems, which are techniques to improve the predictive performance of machine learning algorithms. It also presents three different algorithms to build ensembles: Bagging, Boosting and Random Forest.

Chapter 5 details the proposed method to optimize an ensemble system using Genetic Algorithms. First, it is described how the ensemble is constructed; secondly, it is explained in detail, the design of each part of the proposed Genetic Algorithm.

Chapter 6 explains the experimental procedure. It begins by explaining how the research data was collected and how it was transformed into the features used by the classification algorithm. Then, the general settings of the system used during the experiments, as well as the tools used to accomplish them, are explained. Finally, experimental results are detailed and analyzed.

Chapter 7 and 8 present, respectively, the final remarks of this work, along with suggestions for future projects and the references used to develop it.

2 FINANCIAL MARKETS

The main purpose of the economy is to allocate capital efficiently. To achieve this, capital is invested in sectors that are expected to have high financial returns, while investments are withdrawn from sectors with poor prospects. Based on that, financial markets are intended to ensure adequate distribution of investments (WURGLER, 2000).

The term financial market refers to any marketplace where assets such as equities, bonds, currencies, and derivatives are traded. In financial markets, two forces are constantly acting: Supply and Demand. Because of them, these markets are considered to have a transparent asset pricing (INVESTOPEDIA, 2014). Nowadays, almost every country in the world has an active financial market. There are many different types of markets, which trade different types of financial instruments. The two largest markets are the stock market and the currency market, which together combine to trade trillions of dollars per day (LEVINSON, 2005). These two markets, which are used as input data for this project, are further explained below.

Even though financial markets operate in several different ways, they all serve the same basic functions such as the following (LEVINSON, 2005):

- ***Price setting*:** the price of any good or service is determined by the forces of supply and demand. Financial markets allow for the determination of the relative price of the traded assets through the interaction of buyers and sellers. This process is called price discovery.

- ***Asset valuation*:** based on market prices it is possible to assess the worth of a company or the value of the company's assets. This kind of information is very important when someone is buying or selling an asset. It is also essential prior to purchasing insurance for an asset.

- ***Investing*:** financial markets encourage people to invest their money in different kind of assets, aiming to earn profit in the future.

- ***Raising capital*:** financial markets provide mechanisms for firms to expand their business without having to borrow money from traditional sources. Financial instruments such as stocks and bonds allow firms to acquire cash to grow.

2.1 Stock Market

The Stock Market, also known as the Equity Market, is the market in which shares of publicly held companies are issued and traded. This market is one of the most important ways for companies to raise money. It provides companies with access to capital, in exchange for giving investors a slice of ownership in the company's assets and earnings (DAVIDSON, 2009).

There are different types of stocks, each having its own characteristics. Nevertheless, the two main types of stocks are the following (LEVINSON, 2005):

- ***Common or Ordinary Stock*:** this type of stock gives the owner the right to exercise control by electing a board of directors and voting on corporate policy;

- ***Preferred Stock*:** preferred stockholders have priority over common stockholders on earnings and assets in the event of liquidation. Owners of preferred stock also receive dividends before common shareholders, but they don't have voting rights.

Stocks are listed on a stock exchange, which aims to connect stock buyers with stock sellers. The main function of a stock exchange is to ensure fair and orderly trading, offer greater liquidity, and offer investors published information on the prices at which trades have occurred or are being offered (DAVIDSON, 2009). The four major stock exchanges of the world are: New York Stock Exchange (NYSE), Tokyo Stock Exchange, National Association of Securities Dealers Automated Quotation System (NASDAQ) and London Stock Exchange (LEVINSON, 2005).

The movements of the prices in a market, or section of a market, are captured in price indices called stock market indices. These indices are usually computed as a weighted average of market capitalization of selected stocks. They are used to describe the market, and to compare the return on specific investments (LEVINSON, 2005). Some examples of stock indices by region are listed below (BLOOMBERG, 2014):

- ***Americas*:** S&P 500 Index (U.S.), Dow Jones Industrial Average (U.S.), NASDAQ Composite Index (U.S.) and Bovespa Index (Brazil);

- ***Europe*:** FTSE 100 Index (U.K.), DAX Index (Germany) and IBEX 35 Index (Spain);

- ***Asia*:** Nikkei 225 Index (Japan), Hang Seng Index (China) and Shanghai Composite Index (China).

The weekly variation of the Ibovespa Index from September 2013 to September 2014 is shown in Figure 2.1.

Figure 2.1 – Chart of Ibovespa Index from September 2013 to September 2014

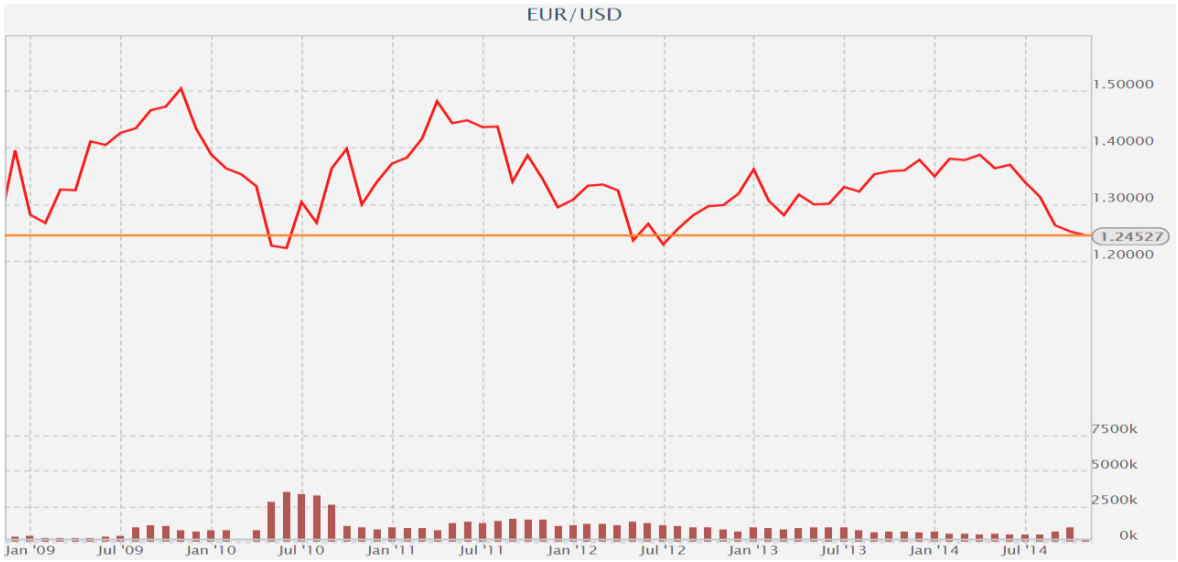
Source: (YAHOO FINANCE, 2014a).

2.2 Forex Market

The Forex market, short for Foreign Exchange market, is where national currencies are traded. Currencies are essentially macroeconomic securities, fluctuating in response to wider-ranging economic and political developments (GALANT; DOLAN, 2007). However, the value of the currency itself can be measured only against an external reference, which, most often, is other currency. The value determined by this reference is called the exchange rate, and is the fundamental price in any economy (LEVINSON, 2005). Figure 2.2 shows the

exchange rate between Euro and U.S. Dollar (EURUSD) from January 2009 to November 2014.

Figure 2.2 – EURUSD variation from January 2009 to November 2014



Source: (ZULUTRADE, 2014).

This market is the largest and most liquid market in the world with an average traded volume that exceeds $2 trillion per day, and includes all currencies in the world (GALANT; DOLAN, 2007). Based on this, the Forex market can directly influence the flow of international investment, and affect domestic interest and inflation rates (LEVINSON, 2005). The currencies are traded in pairs, with names that combine references of both currencies being traded. The most traded currency is the U.S. Dollar (LEVINSON, 2005). Therefore, all major currency pairs involve the U.S. Dollar in one side of the deal (GALANT; DOLAN, 2007). Table 2.1 shows a list with the major currency pairs.

Table 2.1 – Major currency pairs

ISO Currency Pair	***Countries***	***Long Name***	***Nickname***
EUR/USD	Eurozone*/U.S.	Euro-dollar	N/A
USD/JPY	U.S./Japan	Dollar-yen	N/A
GBP/USD	United Kingdom/U.S.	Sterling-dollar	Sterling or Cable
USD/CHF	U.S./Switzerland	Dollar-Swiss	Swissy
USD/CAD	U.S./Canada	Dollar-Canada	Loonie
AUD/USD	Australia/U.S.	Australian-dollar	Aussie or Oz
NZD/USD	New Zealand/U.S.	New Zealand-dollar	Kiwi

** The Eurozone is made up of all the countries in the European Union that have adopted the euro as their currency.*

Source: (GALANT; DOLAN, 2007).

There is no central marketplace for currency exchange. Most trades occur in the interbank markets, among financial institutions which are present in many different countries (LEVINSON, 2005). The Forex market is open 24 hours a day, five days a week and currencies are traded worldwide among the major financial centers such as London, New York, Tokyo, Zürich, Frankfurt, Hong Kong and Sydney (GALANT; DOLAN, 2007).

2.3 Technical Analysis

"*Technical analysis is the study of market action, primarily through the use of charts, for the purpose of forecasting future prices trends*" (Murphy, 1999, p. 1, italics in original). Technical analysis assumes that securities move according to trends and patterns that are sustained over periods of time until a change in the market condition activates another trend. Technical analysis uses historical information - prices and trading volume - to derive technical indicators, which are intended to facilitate the understanding of market movements. The main goal of technical analysis is to forecast the price of the security over some future time horizon, in order to assist traders in making more profitable trades (ROCKEFELLER, 2011).

Technical analysis is based on three premises (MURPHY, 1999):

1. Market price discounts everything;
2. Prices move in trends;
3. History repeats itself.

The first premise is considered the foundation of technical analysis. It states that the market prices reflect not only information about economic factors, but also other factors such as political, psychological and geographical ones (MURPHY, 1999). Thus, price indirectly provides a perspective of the fundamentals, and a study of price action is therefore, all that is required to make predictions. The second premise suggests that asset price movements are believed to follow trends. This means that after a trend has been established, the future price movement is more likely to be in the same direction as the trend than to be against it, although price variations may occur (MURPHY, 1999). Thus, technical analysis essentially looks for patterns in prices that signal continuation or reversals in trend. Figure 2.3 illustrates this premise by showing an uptrend in S&P 500 Index from November 2012 to November 2014. Finally, the third item presents the idea that particular events occur repeatedly in the market.

The repetitive nature of price movements is attributed to market psychology; in other words, market participants tend to provide a consistent reaction for similar market stimuli over time (LI, 2000).

Figure 2.3 – Uptrend in S&P 500 Index from November 2012 to November 2014

Source: (YAHOO FINANCE, 2014b).

For more details on Technical Analysis, the full reading of (MURPHY, 2000) and (ROCKEFELLER, 2011) is recommended. In chapter 6, it is shown how the concepts of Technical Analysis, along with concepts of Genetic Algorithm and Ensemble Systems, are applied in predicting future movements in the stock market.

3 GENETIC ALGORITHMS

In this chapter, the basic concepts necessary for the understanding of Genetic Algorithm will be presented, in order to facilitate the comprehension of the proposed method for financial time series forecasting, which is described in chapter 5.

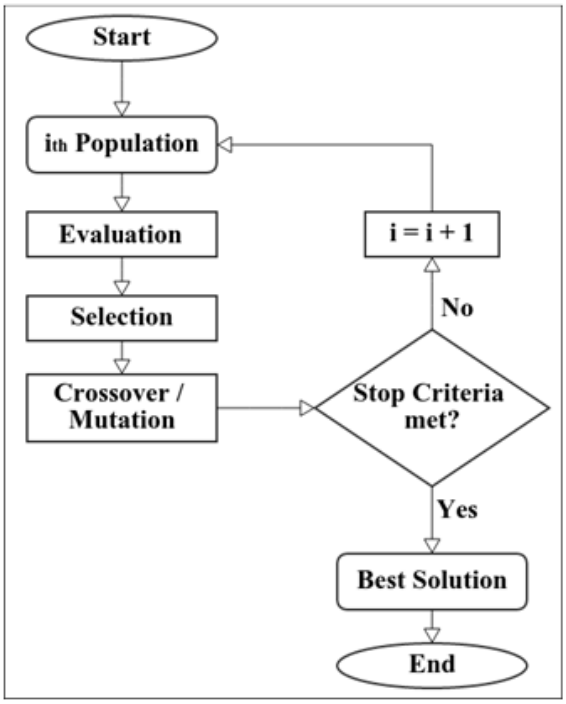
3.1 Concept

Genetic Algorithms (GAs) were first introduced by Holland (HOLLAND, 1975). They are a class of Evolutionary Algorithms, which are general adaptive optimization search methodologies based on a direct analogy to Darwinian natural selection and genetics in biological systems (MELANIE, 1998). GAs have been considered a promising alternative to conventional heuristic methods. The relative insensitivity of GAs to noise and the requirement of no domain knowledge make them a powerful tool for optimization problems. Moreover, associated with the characteristics of exploration (the process of visiting new regions of a search space) and exploitation (the process of visiting those regions of a search space within the neighborhood of previously visited points), GAs are capable of dealing with large search spaces efficiently (MELANIE, 1998). Therefore, they have been widely used in different tasks, including numerical and combinatorial optimization, optimization of neural networks, multi-agent systems, economics, and bioinformatics (WHITLEY et al, 1989; BRAMLETTE, 1991; MICHALEWICZ, 1991).

The simple Genetic Algorithm works with a set of candidate solutions called a population, in which each chromosome (individual) represents a possible solution for a problem. In the first iteration, the initial population can be initialized in several ways, the most common way being the random choice (CANUTO; NASCIMENTO, 2012). The individuals are assessed through a fitness function, which is the function to be optimized (minimized or maximized). Based on the value of this function, chromosomes are selected and some genetic operators (crossover and mutation) can be applied to them with certain probabilities, forming new ones. These operators are successively applied to the population in a loop, being each run of the loop called generation. The main idea is that these individuals evolve, tending to create better ones, until acceptable results are obtained, or some stop condition is met. At the end, the fittest chromosome found during all generations is the GA’s answer to the given problem (EIBEN; SMITH, 2003). Figure 3.1 shows the steps mentioned

above represented as a flowchart. Some of the concepts presented in these steps are key points on the design of GAs and are better explained below.

Figure 3.1 – Flowchart representing the execution of the standard GA



Source: Author.

3.2 Chromosome Representation

Representation is a key issue in GA work because GAs directly manipulate a coded representation of the problem, and because the representation scheme can severely limit the window by which a system observes its world (KOZA, 1992). The binary encoding, which represents chromosomes as fixed-length bit strings, is the most commonly used type of representation, since they are considered easy to implement (MELANIE, 1998). However, for many applications, it is more natural to represent chromosomes using real numbers (MELANIE, 1998). Thus, over the past few years, researchers have been paying attention to real coded GAs because this representation seems to be adequate when tackling real-world optimization problems (BRAMLETTE, 1991; GOLDBERG, 1991; PADILHA et al, 2010). But GA is not limited to these two types of representations. There are other types of representations which are more natural for specifics application problems. Examples of such cases are vectors of integer numbers for function optimization (BRAMLETTE, 1991); ordered lists for the traveling salesman problem (WHITLEY; STARKWEATHER;

FUQUAY, 1989); two-dimensional matrix of integer numbers for the linear transportation problem (MICHALEWICZ, 1991).

3.3 Fitness Function

The evaluation of each individual is performed by means of the Fitness Function, which depends on the specific problem and is the optimization objective of the GA (MELANIE, 1998). The fitness function is a mathematical function that assesses the quality of a chromosome, conveying the goodness of a chromosome in the solution. This function directly affects the survivability of one individual over the generations. For example, in a maximization problem, the higher the fitness value, the better the solution. In a GA application, the formulation of the fitness function is of critical importance and determines the final shape of the hypersurface to be searched. In certain real-world problems, there is also a number of constraints to be satisfied. Such constraints can be incorporated into the fitness function by means of penalty terms which further complicate the search (EIBEN; RUTTKAY, 1996).

3.4 Selection and Genetic Operators

As mentioned above, a new population is generated by probabilistically selecting the fittest individuals from the current population, and then applying the genetic operators: crossover and mutation. Crossover is considered to be the main genetic operator, whereas mutation is viewed as secondary and used sparingly (KOZA, 1992). The following explains how selection can be performed and how these two operators work.

3.4.1 Selection

Simulating the “survival of the fittest” principle, selection methods aim to emphasize the fitter individuals in the current population, expecting that their offspring will have even higher fitness (MELANIE, 1998). Numerous selection schemes have been proposed in literature, some of the most used methods being: *Roulette wheel Selection*, *Tournament Selection* and *Ranking Selection* (MELANIE, 1998). In the *Roulette Wheel Selection*, each individual of the population is represented in proportion to its fitness. Thus, the fittest

individuals occupy a larger portion in the roulette wheel (higher probability to be selected), while those of lower fitness have a relatively smaller portion of the roulette wheel (lower probabilities to be selected) (PADILHA et al, 2010). The roulette is turned a certain number of times, depending on the size of the population, and those who are selected will then be used in the crossover step. The *Tournament Selection* establishes several engagements in order to select individuals, each engagement involving *n* individuals chosen at random from the population. In each combat, the individual with the best fitness value is selected for crossover (MELANIE, 1998). A slightly different approach is the *Ranking Selection*, in which the chromosomes are sorted in order of raw fitness, and then the probability of selecting one individual is computed according to its rank rather than on its absolute fitness. By doing this, this method avoids giving the largest share of offspring to a small group of highly fit individuals (BRAMLETTE, 1991).

Another strategy of constructing the new population is known as *elitism* (or elitist selection). This technique directly transfers some of the fittest individuals of the current population to the next one, unaltered. This is a promising technique since it helps that the best chromosomes do not disappear after the crossover and mutation operations (MELANIE, 1998).

3.4.2 Crossover

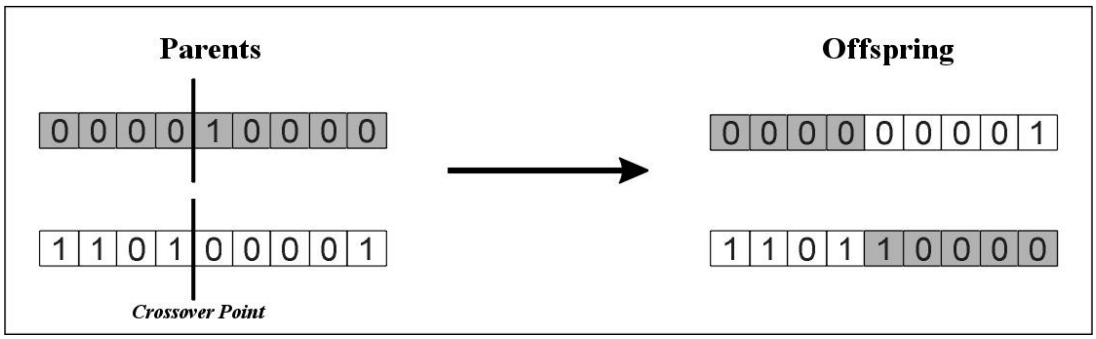
The crossover is an exchange of information between parent chromosomes, creating new solutions (MELANIE, 1998). The idea of performing crossover is that the new individuals may be better than both of the parents, if they receive the best genes from each of the parents. This process occurs with a certain probability, called the crossover rate.

In bit strings, this usually happens through the exchange of determined bit indexes. For example, a simple and often used method is the *One-Point Crossover* (MICHALEWICZ, 1996), in which one point on both parents' strings is selected and all data beyond that point on either string is swapped between the two parents. Figure 3.2 illustrates this technique. Many different crossover algorithms have been devised, often involving more than one cut point. Very similar to the One-Point method, is the *Two-Point Crossover, where* two points are chosen and the genes between these points are exchanged between the two parents (KAYA; UYAR; TEKIN, 2011). Other techniques have been proposed in the literature. In the *Uniform Crossover* method, genes are randomly copied from the first or second parent. Because of

that, the offsprings have approximately half of the genes from one parent and the other half from the other parent; although crossover points can be randomly chosen (MELANIE, 1998).

In real-coded chromosomes, crossover is performed as arithmetical operations between the two parents. *Arithmetical Crossover*, probably the most used operator, works by taking the weighted sum of the two parents for each gene (PICEK; JAKOBOVIC; GOLUB, 2013). A simpler version of this method is the *Average Crossover*, which simply takes the arithmetic average of the two parents (EIBEN; SMITH, 2003). Instead of averaging the values, the *Flat Crossover* generates descendants whose genes are randomly generated in the interval of parents' genes (PICEK; JAKOBOVIC; GOLUB, 2013).

Figure 3.2 – Bit string One-Point Crossover



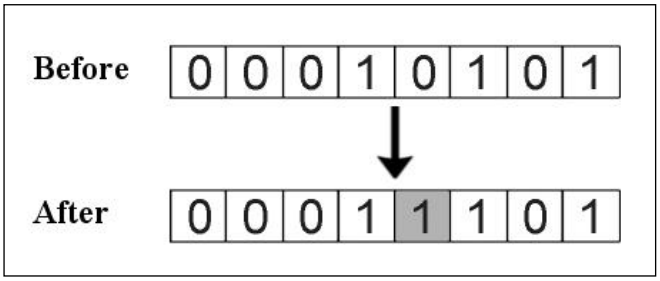
Source: Author.

3.4.3 Mutation

Mutation promotes changes in the value of one or more genes of a selected chromosome (MELANIE, 1998). This operation is very important in keeping the varieties of populations. The role of mutation in GAs is that of restoring lost or unexplored genetic material back into the population to prevent the premature convergence to suboptimal solutions. Like the crossover, it also has a probability of occurrence, called the mutation rate. However, unlike the crossover, this probability is normally very low.

In bit string chromosomes, mutation might be a simple inversion of a gene ('0' to '1' or '1' to '0'). Figure 3.3 illustrates the before and after of this mutation operator. In the floating-point case, mutation can happen by adding or subtracting a small value to a gene. The most used operator in this case, is most likely the *Random Mutation* (MICHALEWICZ, 1996), which replaces selected genes with random numbers generated from the domain of the problem. Another well-known method is the *Non-Uniform Mutation* (MICHALEWICZ, 1996), which is used in this project, and will be better explained in chapter 5.

Figure 3.3 – Bit string Mutation



Source: Author.

3.5 Termination Conditions

Finally, it is important to know how to determine when the result is good enough. There are problems where one can easily define when a solution is optimal or not. However, for many problems, easily recognizing an optimal solution does not exist (KOZA, 1992). Thus, termination conditions are used by GAs to decide whether to continue or stop searching. Each of the enabled termination conditions is checked after each generation to see if it is time to stop. The most commonly used conditions are: a solution that satisfies minimum criteria is found, a fixed number of generations is reached, or a certain number of successive generations haven't produced an improvement in the best solution. At the end, the best solution found is prompted by the GA as the solution for the problem (KOZA, 1992).

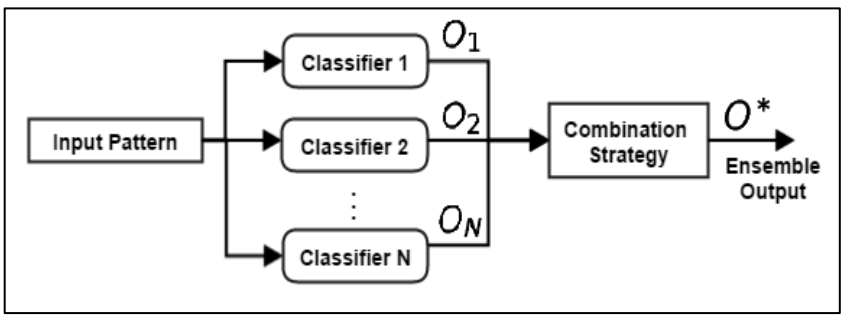
4 ENSEMBLE SYSTEMS

As mentioned in Section 1.1, Ensemble Systems have been widely used as an alternative to enhancing the predictive performance of machine learning algorithms. In this chapter, key factors of designing ensembles systems, as well as three different methods of building ensembles, will be presented.

4.1 Concept

In recent years, researchers have been proposing methods to improve the accuracy and stability of the predictive systems generated by machine learning algorithms. One method that stands out is the combination of multiple learning algorithms, forming Ensembles or Committees (KUNCHEVA, 2004). The main idea of using Ensembles is that different individual models can offer complementary information about unknown instances, improving the quality of the overall classification process in terms of generalization and accuracy (TAN; STEINBACH; KUMAR, 2006). In the regular architecture of a committee, as shown in Figure 4.1, a new input pattern is presented to all classifiers. Then the individual models provide their outputs $(O_1, \dots, O_N)$ and send them to a combination method, which is responsible for providing the final output of the system (O^*).

Figure 4.1 – Standard architecture of an ensemble system



Source: Author.

Two points are important when designing ensemble systems: the base classifiers and the combination method (TAN; STEINBACH; KUMAR, 2006). In relation to the ensemble components, it is fundamental to have set of individual models whose errors are at least somewhat uncorrelated. The base classifiers should disagree with one another, since the combination of identical classifiers does not improve the accuracy (KUNCHEVA, 2004).

Thus, when combining them, individual failures will be minimized. In other words, the individual models should be diverse among themselves. Once the set of base classifiers has been created, the next step is to choose an effective way of combining their outputs. The choice of the best combination method for an ensemble requires exhaustive training and testing (TAN; STEINBACH; KUMAR, 2006). These two key factors of ensemble system development are further explained below.

4.1.1 Diversity in Ensembles

As mentioned previously, diversity among the base classifiers is the key to achieving high accuracy in ensembles. Diversity can be reached when the base models are built under different circumstances, such as in the following ways:

- ***Different training sets*:** multiple training sets are created by resampling the original data according to some sampling distribution. A classifier is then built from each training set. Diversity can be reached through the use of learning strategies such as *Bagging* (BREIMAN, 1996) and *Boosting* (FREUND; SCHAPIRE, 1996), which build training sets with different instances;

- ***Different input features*:** a subset of input features is chosen to form each training set. The subset can be either chosen randomly, or based on the recommendation of a domain expert. *Random Forest* (BREIMAN, 2001), which uses decision trees as its base classifier, is a well-known ensemble method that uses this approach;

- ***Different parameters settings*:** diversity can be achieved by changing the initial parameter setting of the base classifiers. For instance, an ensemble of decision trees can be constructed by injecting randomness into the tree-growing procedure. Similarly, an artificial neural network can generate different models by either changing the initial weights of the links between the neurons or its network topology;

- ***Different learning algorithms*:** diversity can be achieved by using different types of individual classifiers, also called heterogeneous ensembles. For instance, usually an ensemble that is composed of a decision tree and a neural network, as they have different inductive bias, is likely to be more diverse than ensembles composed only of either neural networks, or decision trees;

4.1.2 Combination Methods

Choosing how the output of the base classifiers will be combined is also a very important and difficult task. There are a great number of combination methods reported in literature. Three of the most used methods for classification problems are: Majority Voting, Weighted Voting and Naive Bayesian (KUNCHEVA, 2004):

- ***Majority Voting*:** in this method each model outputs a class value, and the class with the most votes is the one proposed by the ensemble.

- ***Weighted* Voting:** in this method the base classifiers models are not treated equally. Each classifier is associated with a coefficient (weight), which is usually proportional to its classification accuracy (DIETTERICH, 2000). Thus, the higher the weight of a classifier, the more it influences the learning of the ensemble. Let K be the number of classifiers in the ensemble, and w_i the weight of the ith classifier, the final output of the ensemble ($C^*(x)$) is calculated as follows:

$$C^*(x) = argmax_y \sum_{i=1}^{K} w_i \times \delta(C_i(x) = y) \tag{4.1}$$

- ***Naive Bayesian*:** this method assumes that the classifiers are mutually independent, given a class label (conditional independence). Assuming that the ensemble is formed of K classifiers, and c_i is the class label assigned to the instance x by the classifier C_i, the probability that the ensemble classifies the instance x as c_j is calculated as follows:

$$c_j(x) = \prod_{i=1}^{K} P\left(w_j \middle| C_i(x) = c_i\right) \tag{4.2}$$

Where $P(w_j|C_i(x) = c_i)$ represents the probability that the class c_i assigned by the classifier C_i is equal to w_j. This probability is calculated as follows:

$$P(w_j|C_i(x) = c_i) = \frac{N°\ instances\ labelled\ as\ s_i\ by\ C_i\ whose\ true\ label\ is\ w_j}{N°\ instances\ labelled\ as\ s_i\ by\ C_i} \quad (4.3)$$

At the end, the class c_j with higher probability is assigned to the instance x.

4.2 Ensemble Learning Methods

Three of the most popular methods for producing ensembles are: Bagging, Boosting, and Random Forests (KUNCHEVA, 2004). In order to provide a better overview on how these three techniques work, they are individually presented below. The performance of these three methods is compared with the performance of the proposed method in chapter 6.

4.2.1 Bagging

Bagging, also known as bootstrap aggregating, is based on the idea of data resampling (BREIMAN, 1996). Diversity is promoted in Bagging by using bootstrapped replicas of the training dataset. Each replica represents a subset of the training data generated by randomly drawing with replacement. Each new dataset will have the same number of instances as the original dataset. Because the sampling is done with replacement, some instances may appear more than once in the same training set, while others may be omitted. Because of that, the effective size will be lower than the original dataset, and the datasets will overlap significantly. On average, each bootstrap contains 63.2% of the original training set (TAN; STEINBACH; KUMAR, 2006). Each derived dataset is used to train a classifier, and then for any test instance, the outputs of the individual classifiers are aggregated via simple majority voting. By using this aggregation method, Bagging improves generalization error by reducing the variance of the base classifier. Finally, since every sample has an equal probability of being selected, Bagging does not focus on any particular instance of the training set. Therefore, it is less susceptible to model overfitting when applied to noisy data (BREIMAN, 1996). Figure 4.2 illustrates the standard procedure for the Bagging algorithm.

Figure 4.2 – Standard Bagging procedure

1. *Let K be the number of bootstrap samples.*
2. **for** $i = 1$ *to* K **do**
3. *Create a bootstrap sample* D_i *of size* N.
4. *Train a base classifier* C_i *on the bootstrap* D_i.
5. **end for**
6. $C^*(x) = argmax_y \sum_i^k \delta(C_i(x) = y)$.

$\{\delta(.) = 1$ *if its argument is true and* 0 *otherwise*$\}$

Source: (TAN; STEINBACH; KUMAR, 2006).

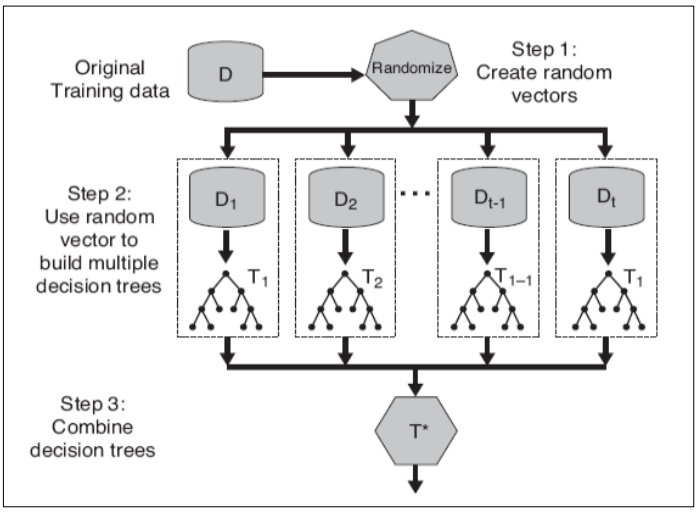
4.2.2 Boosting

Boosting, like Bagging, is also based on data partitioning. It is an iterative procedure used to adaptively change the distribution of training examples so that the individual classifiers will focus on instances that are hard to classify. However, unlike Bagging, Boosting constructs the base classifiers on weighted versions of the entire dataset, which are dependent on previous classification results (DIETTERICH, 2000). Initially, all instances have equal weights so that they are equally likely to be chosen for training. Then weights are changed according to the performance of the classifier. Erroneously classified instances get higher weights, and the next classifier is boosted on the reweighted training set. As the boosting rounds go on, examples that are the hardest to classify tend to become even more prevalent. At the end, a sequence of base classifiers is obtained, which is then combined by majority voting, or by weighted majority voting, in the final decision. There are several variations of Boosting that appear in the literature. AdaBoost (FREUND; SCHAPIRE, 1996), short for “Adaptive Boosting”, is probably the most used algorithm (TAN; STEINBACH; KUMAR, 2006). In each iteration, it adjusts the weights of the training instances to emphasize the examples that were misclassified by the last learned classifier. Instead of using the majority vote scheme, the final output of the ensemble is computed using weighted voting of the prediction of each individual classifier, where the weights depend on the error rate of the classifier on the training set. This approach allows AdaBoost to penalize models that have poor accuracy, e.g., those generated at the earlier boosting rounds.

4.2.3 Random Forest

Random Forest, introduced by Breiman in (BREIMAN, 2001), combines the prediction made by multiple decision trees independently induced, where each tree is constructed based on the value of a random vector sampled independently, and with the same distribution for all trees in the forest. Each tree is generated by randomly drawing, with replacement, a bootstrap resample with the same size of original data set. Then, at each node split, a subset of attributes is selected at random from the original set of attributes, and the best split on this subset is used to split the node. The tree is then grown to its entirety without any pruning. Once the trees have been constructed, the predictions are combined using a majority voting scheme. Since only a subset of attributes needs to be examined at each node, this approach helps to reduce the runtime of the algorithm. Moreover, this randomization helps to improve the generalization error of the ensemble, since it reduces the correlation among the decision trees (TAN; STEINBACH; KUMAR, 2006). Figure 4.3 illustrates the Random Forest algorithm.

Figure 4.3 – Random Forest algorithm

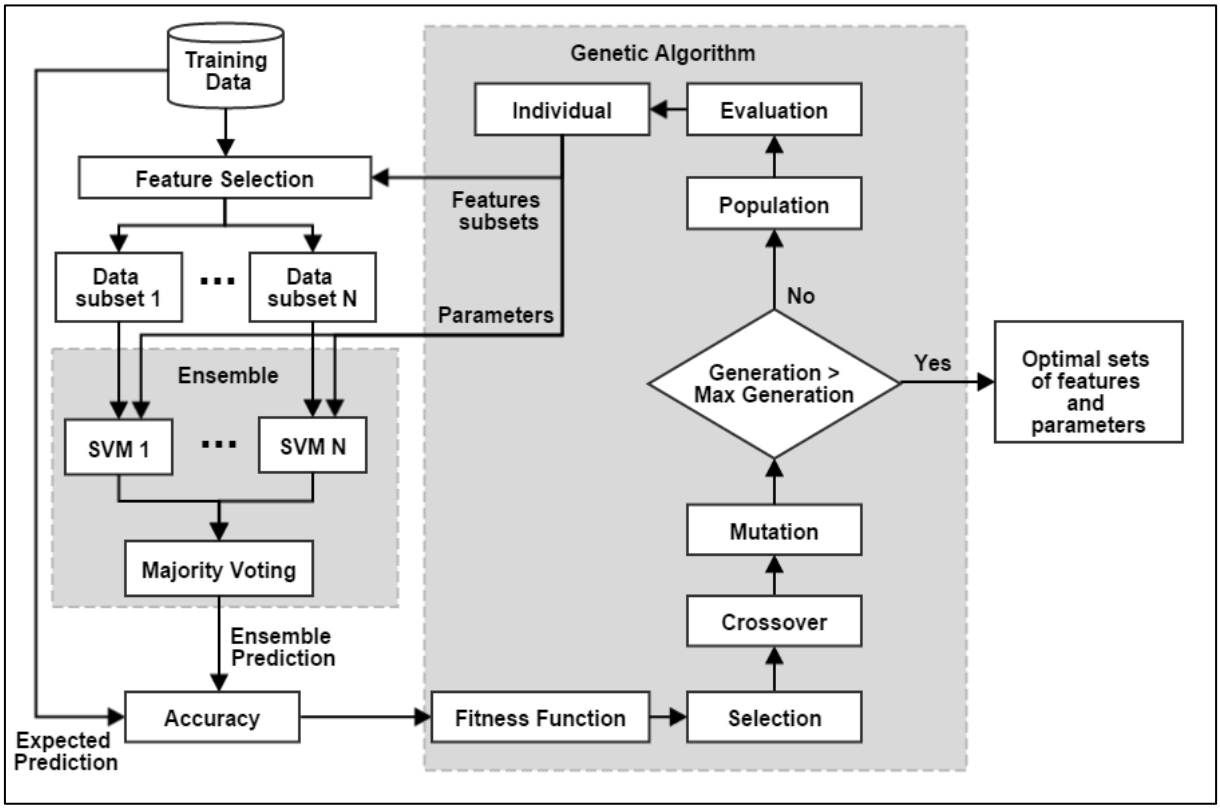


Source: (TAN; STEINBACH; KUMAR, 2006).

5 PROPOSED METHOD

In this chapter, the proposed model building process is presented in detail. In this study, a hybrid data mining methodology, a Genetic Algorithm based Ensemble System (*GAENSEMBLE*) model, is proposed to forecast stock market variations. In this approach, GA is used for feature selection and parameter optimization simultaneously, in order to generate different subsets of inputs for each base classifier. An Ensemble System is then applied to classify stock market movement direction. First of all, the structure of the ensemble is described. Then details about key points of the GA's design are presented. In order to better understand the proposed method, in Figure 5.1 its overall architecture is shown.

Figure 5.1 – Overall architecture of the proposed method



Source: Author.

5.1 Ensemble System

As mentioned in section 4.1, an effective ensemble should consist of a set of classifiers that are not only highly accurate, but ones that also make their errors in different parts of the input space as well. The proposed ensemble is composed of a set of N Support Vector

Machines (SVM), a better explanation of this type of algorithm is given below. In order to increase the diversity of the ensemble, each SVM is constructed using different sets of attributes and parameters. After all of the SVMs have been trained, their outputs are combined by Majority Voting.

5.1.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) was first proposed by Vladimir Vapnik (VAPNIK, 1995). The basic idea of SVM is to use a linear model to implement nonlinear class boundaries through some nonlinear mapping of the input vector into a high-dimensional feature space. Thus, a linear model constructed in the new space can represent a nonlinear decision boundary in the original space. In the new space, an optimal separating hyperplane is constructed. Therefore, SVM is known as the algorithm that finds a special kind of linear model, i.e. the maximum margin hyperplane, which gives the maximum separation between the decision classes. The margin of the classes is defined by the support vectors, which are the marginal examples of a given class in the training data, i.e. input examples closer to the other class. All other training examples are irrelevant for defining the binary class boundaries.

For the linearly separable case, a hyperplane separating the binary decision classes in the two-attribute case can be represented as the following equation:

$$y = w_0 + w_1 x_1 + w_2 x_2 \tag{5.1}$$

Where y is the outcome, x_i are the feature values and w_i are the weight values that represent the hyperplane, and should be learned by the algorithm. The maximum margin hyperplane can be described as the following equation in terms of the support vectors:

$$y = b + \sum_{i=0} \alpha_i y_i x(i) \cdot x \tag{5.2}$$

Where $x(i)$ is the ith support vector, y_i is the class value of training example $x(i)$, $\cdot$ represents the dot product and the vector x represents a test instance. In this equation, b and α_i are parameters that determine the hyperplane. From the implementation point of view, finding the support vectors and determining the parameters b and α_i are equivalent to solving a linearly constrained quadratic programming.

As mentioned above, SVM transform the inputs into the high-dimensional feature space in order to construct a linear model to implement nonlinear class boundaries. For the

nonlinearly separable case, a high-dimensional version of Equation 5.2 is represented as follows:

$$y = b + \sum_{i=0} \alpha_i y_i K(x(i), x) \tag{5.3}$$

The function $K(x(i), x)$ is defined as the kernel function. There are different kernels for generating the inner products to construct SVMs with different types of nonlinear decision surfaces in the input space. Common examples of kernel functions are the Polynomial kernel $K(x, y) = (xy + 1)^d$ and the Gaussian Radial Basis Function (RBF) $K(x, y) = exp(-\|x - y\|^2/2\sigma^2)$, where d is the degree of the polynomial kernel, and σ is the bandwidth of the Gaussian radial basis function kernel (LIMA; NETO; MELO, 2009).

For the separable case, the coefficient α_i in Equation 5.3 has a lower bound equal to zero. For the non-separable case, SVM can be generalized by placing an upper bound C on the coefficients α_i in addition to the lower bound (WITTEN; FRANK, 2005). The coefficient C, also known as the penalty factor, controls the trade-off between achieving a low error rate on the training data and the model complexity. For small values of C, the number of training errors increases, since it generates a larger-margin separating hyperplane. Conversely, large values of C will lead to a smaller-margin hyperplane, which ends up penalizing non-separable points more severely (VAPNIK, 1995).

Lima et. al. (2009), show that the Gaussian Radial Basis Function is the SVM kernel that allows for higher diversity among the most popular ones, because its Gaussian width parameter promotes a more detailed tuning. Thus, this kernel is used in all SVMs of the proposed ensemble. Finally, for each SVM there are two parameters to be optimized: σ and C.

5.2 Genetic Algorithm

When using SVM, two problems are confronted: how to choose the optimal input feature subset, and how to set the best kernel parameters. These two problems are crucial for obtaining a good predictive performance because the feature subset choice influences the appropriate kernel parameters and vice versa (FRÖHLICH; CHAPELLE, 2003). Therefore, obtaining the optimal feature subset and SVM parameters must occur simultaneously

As mentioned in chapter 3, GAs have been shown to be powerful algorithms capable of dealing with hard optimization problems. Based on this, it is proposed a GA-based feature

selection and parameter optimization algorithm, which aims to overcome the issue mentioned above, and by doing so, improves the performance of the SVM ensemble. The proposed GA has the same architecture as the standard GA mentioned in section 3.1. The main contributions are in the chromosome design, the implementation of the genetic operators, and in the fitness function.

5.2.1 Feature Selection and Parameters Optimization

Feature selection methods aim to reduce the dimensionality of the attributes of a dataset, looking for the best ones. The feature subset selection can be defined as the process that chooses the best attribute subset according to a certain criterion, excluding the irrelevant or redundant attributes. There is a vast number of works in the literature using feature selection in ensemble systems, GA being one of the most used methods, such as in (ZHAO, 2008), (CANUTO; NASCIMENTO, 2012), (OPITZ, 1999). In the context of ensemble systems, feature selection methods are used to provide different subsets of features for the base classifiers, aiming to increase the diversity of the ensemble, and to reduce redundancy among the attributes of a pattern. This is because the individual classifiers will classify the same input patterns, but these patterns will have been built using different subsets of features. In addition, by reducing the dimensionality of the base classifiers, feature selection also helps to reduce the overall complexity of the ensemble.

In addition to the feature selection, proper parameter settings can improve the SVM classification accuracy and avoid either over-fitting, or under-fitting, on the training data. As mentioned in section 4.1.1, diversity can also be promoted in ensemble systems by creating base classifiers with different parameter settings. At this level, GA tries to find the best values of σ and C for each SVM in the ensemble, aiming to increase the accuracy of the final classifier.

5.2.2 Chromosome Design

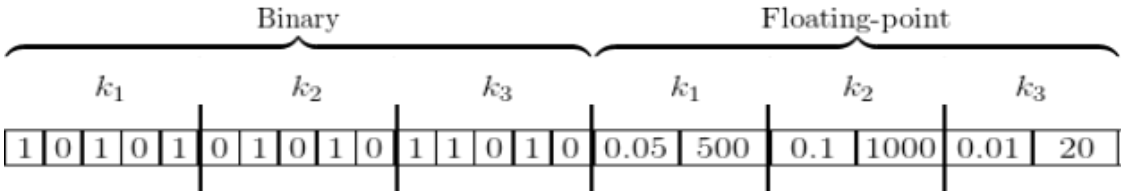
A hybrid representation of the chromosome is used to represent a possible solution for this problem, which means that individuals will be coded as binary, and partially as floating-point values. As mentioned before, the RBF kernel function is used for the SVMs classifiers in this project. Thus, for each individual classifier in the ensemble, it is required to optimize the subset of input features, and the two parameters of the SVM (σ and C).

Assuming that there are F features in the dataset, and the ensemble is composed of K classifiers, the chromosome will have a length of $(F \times K) + (2 \times K)$ genes. The first part will be coded as a bitstring of length $F \times K$, and the second part as a floating-point array of length $2 \times K$.

The binary part represents the subset of features selected for each classifier, which '1' corresponds to a selected feature and '0' to a non-selected one. The first K bits represent the feature subset for classifier k_1, followed by K bits for classifier k_2, and so on. Similarly, the real part contains the two parameters to be optimized for each SVM of the ensemble in which the first two values (σ_1 and C_1) represent the parameters for the classifier k_1, followed by other two values for classifier k_2, and so on.

Figure 5.2 shows an example of an individual that represents an ensemble composed of three SVMs (k_1, k_2 and k_3) and a dataset with 5 attributes. In this example, the attributes 1, 3 and 5 are assigned to classifier k_1, attributes 2 and 4 are assigned to classifier k_2 and, attributes 1, 2 and 4 are assigned to classifier k_3. Also, classifier k_1 has $\sigma_1 = 0.05$ and $C_1 = 500$, classifier k_2 has $\sigma_2 = 0.1$ and $C_2 = 1000$, and classifier k_3 has $\sigma_3 = 0.1$ and $C_3 = 20$.

Figure 5.2 – Hybrid Chromosome



Source: Author.

5.2.3 Selection

The proposed GA implements the *K-Tournament Selection* method mentioned in section 3.4.1. This process is repeated twice so that the two parents used for crossover are selected. It has been shown that this method obtains better results than other selection methods (BACK, 1996). Also, the complexity of this method is less than that of the *Ranking Selection* (HERRERA; LOZANO, 2000), since there is no need to sort the population based on the fitness of the individuals. In this project K is fixed to 3.

As explained in section 3.4.1, the *elitism* assures that the best individuals always survive unaltered to the next generation. Thus, this technique is also used in this project by

copying the best individuals of the population in generation t to the population in generation $t + 1$.

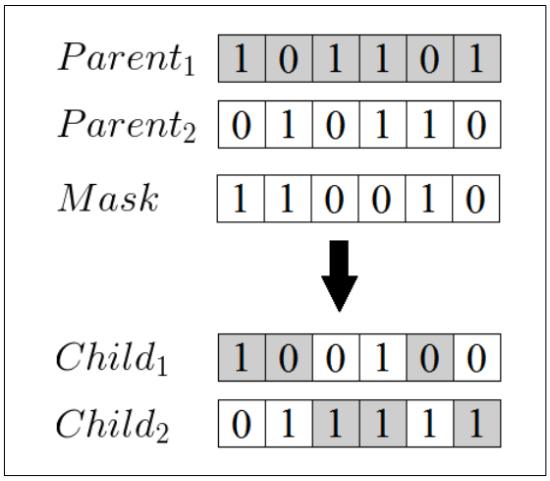
5.2.4 Crossover Operators

As mentioned before, crossover is a process where new individuals are created from information contained within the parents, given a crossover probability p_c. In the proposed GA, the crossover is done by using two parents to produce two offspring. Since the individuals are coded using two different types of representation, it is used two different crossover methods. For the binary part, the *Scattered Crossover* technique is used (PADILHA; NETO; MELO, 2012), while for the real-coded part, the *BLX-Alpha-Beta Crossover* technique is used (PICEK; JAKOBOVIC; GOLUB, 2013). The following sections will explain these methods in further detail.

5.2.4.1 Scattered Crossover

In this recombination method, a bit mask is randomly created of the same length as the parent chromosomes, and the elements of this mask decide the outcome of the crossover. Thus, the genes of the first child are selected from the first parent, where the mask is a 1, and from the second parent where the mask is a 0, and vice versa to form the second one. Figure 5.3. illustrates this method.

Figure 5.3 – Scattered Crossover



Source: Author.

5.2.4.2 BLX-Alpha-Beta Crossover

This operator, also called *Blend Alpha Beta Crossover*, generates a new offspring by selecting a random value from the interval between the two genes of the parents. This operator works by increasing the interval in direction of the parent with better fitness by the factor α, and into the direction of the parent with worse fitness by the factor β. Let $P_1 = (c_1^1, \dots, c_n^1)$ and $P_2 = (c_1^2, \dots, c_n^2)$ be two parents with n genes. Assuming that P_1 has the higher fitness and that $c_i^1 \leq c_i^2$ for all $i \leq n$, a new offspring is generated by sampling a random value in the range $[c_i^1 - \alpha I, c_i^2 + \beta I]$ at each position i. Here, $I = c_i^2 - c_i^1$, $\alpha = 0.75$ e $\beta = 0.25$.

5.2.5 Mutation Operators

Mutation aims to introduce some randomness into the population, helping to prevent local minima in populations that are converging too fast. Similar to the crossover step, mutation is also composed of two different methods, and is applied given a certain probability (p_m). The binary part of the chromosome is mutated by randomly flipping some bits of the string, as explained in section 3.4.3, while the real-coded part is mutated using the technique called Non-Uniform Mutation.

5.2.5.1 Non-Uniform Mutation

This operator performs a uniform search in the initial stages of the evolution, and a very localized search in the final stages, i.e. the size of the gene generation interval becomes lower with the passing of generations. This property keeps the population from stagnating at the beginning of the algorithm, and favors the local tuning in the end. It has been shown that Non-Uniform Mutation is very suitable for a wide variety of problems (HERRERA; LOZANO, 2000).

For an individual $C = (c_1, \dots, c_n)$, the muted individual $C' = (c_1', \dots, c_n')$ is generated by the following equation:

$$c_i' = \begin{cases} c_i + \Delta(t, b_i - c_i) & if\ u \leq 0.5 \\ c_i - \Delta(t, c_i - a_i) & if\ u > 0.5 \end{cases} \tag{5.4}$$

Where u is a uniformly distributed random number in the interval [0, 1], a_i and b_i are the lower and the upper bounds for c_i, respectively. The function $\Delta(t, y)$ described below takes value in the interval [0, y]:

$$\Delta(t, y) = y(1 - r^{(1-t/g_{max})^b}) \tag{5.5}$$

Where r is a uniformly distributed random number in the interval [0, 1], t is the generation, g_{max} is the maximum number of generations, and b is a parameter that determines the degree of dependence of the mutation with regards to the number of iterations. In this project b is fixed to 5 (MICHALEWICZ, 1996).

5.2.6 Fitness Function

Although diversity is an important aspect when designing ensemble systems, it is also important to search for accuracy. Ensembles should be composed of diverse and highly accurate classifiers. Based on this, the objective of the GA is to find the optimal set of features and parameters for each SVM of the ensemble that maximizes the classification accuracy. Thus, the fitness function of the proposed GA is simply calculated using the classification accuracy of the SVM ensemble in the training data set.

5.2.7 Termination Criteria

The proposed GA proceeds with the next generation until the process reaches the maximum number of generations, defined as g_{max}. After reaching g_{max}, the fittest individual of the last population is returned as the final solution to the given problem.

6 EXPERIMENTAL PROCEDURES

The previous chapters introduced some very important concepts for understanding the whole experimental procedure. Presented in this chapter are the forecasting results obtained by the proposed model, introduced in chapter 5, and by the other ensemble methods presented in chapter 4. Firstly, it will be explained how the financial market data was collected and organized, followed by the feature extraction process. Then the tools used to perform the experiments, and the parameters of the system that have to be defined are presented. Finally, at the end of this chapter, the achieved performances of each model in forecasting the weekly prices' movements of Bovespa Index are presented and discussed.

6.1 Research Data

As previously mentioned, this project aims to develop an algorithm to predict whether the next-week price of the Bovespa Index will be higher or lower than the current price. Bovespa Index is the main indicator of the Brazilian stock market's average performance. This index is calculated and disseminated by the Brazilian stock exchange, known as BM&FBOVESPA. It is calculated as a weighted average of a theoretical portfolio which contains the more actively traded, and more representative stocks of the Brazilian stock market. This portfolio represents roughly 70% of the exchange's total capitalization, and 80% of its trades (BM&FBOVESPA, 2014).

As globalization has been deepening the interactions between global financial markets, nowadays, almost no market is isolated. Economic data, political perturbation, and any other oversea affairs could cause dramatic fluctuations in domestic markets. Therefore, in order to enhance the accuracy of the proposed method, in this project it is proposed to also use global financial data, such as major world stock indices and exchange rates as input data. Thus, the final research dataset is composed of the following financial data:

- ***World Stock Indices*:** Ibovespa Index (Brazil), S&P 500 Index (U.S.), Dow Jones Industrial Average (U.S.), FTSE 100 (U.K.), DAX Index (Germany), Nikkei 225 (Japan), Hang Seng Index (China);

- ***Exchange rates*:** USDBRL, EURBRL and CNYBRL.

The historical dataset used in this project was downloaded from (QUANDL, 2014). For the stock indices, it contains the open, high, low and closing prices, and the trading volume of each week. For the exchange rates, it contains only the closing price. The total number of samples is 778 trading weeks, covering from 01 January 2000 to 23 November 2014.

Since the price value alone lends little insight into future price movements, the goal becomes to develop features that provide information on not only past and current price movements, but also its future behavior. This process, called feature extraction, is described below.

6.2 Feature Extraction

Technical Analysis has been helping traders and investors better understand price trends, and the mechanics of price movements. Since this project attempts to forecast the direction of weekly price changes of a stock market index, technical indicators are used as input variables for the classifier. A total of 7 different technical indicators are used in the present study to extract information from the financial time series and hence, they act as the features that are used for stock market movement prediction. As determined by the review of domain experts and prior research, they are summarized below (MURPHY, 1999; ROCKEFELLER, 2011):

- ***Rate of Change (ROC)*:** it measures de difference between the current price and the price n days ago. It is calculated as follows:

$$ROC_n = \frac{C_t}{C_{t-n}} \tag{6.1}$$

  Where C_t is the closing price at time t;

- ***Ratio*:** it is the ratio between two ROCs calculated over different time intervals. Particularly, ROC_n/ROC_m for $m > n$ is informative because it lends insight into how the change in price is changing over time;

- ***Disparity*:** it displays the distance of the current price from the moving average of n days (MA_n). It is calculated as follows:

$$Disparity_n = \frac{C_t}{MA_n} \tag{6.2}$$

- ***Price Oscillator*:** it measures the difference between two moving averages of a security's price (MA_n/MA_m for $m > n$);

- ***Stochastic Oscillator*:** it compares where a stock's price closed relative to its price range over a given time period. It is calculated as follows:

$$\%K = \frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \tag{6.3}$$

$$\%D = \left(\sum_{i=0}^{n-1} \%K_{t-i}\right) \Big/ n \tag{6.4}$$

$$Oscilator = \%K - \%D \tag{6.5}$$

  Where HH_t and LL_t mean the highest high and the lowest low in the last n days;

- ***MACD Histogram*:** it indicates changes in the strength and direction of a trend in security's price. It is calculated as follows:

$$MACD = EMA_{12} - EMA_{26} \tag{6.6}$$

$$signal = EMA_9 \; of \; MACD \tag{6.7}$$

$$MACD \; histogram = MACD - signal \tag{6.8}$$

  Where EMA_n means the Exponential Moving Average of n days;

- ***Parabolic SAR ("Stop And Reverse")*:** it trails price as the trend extends over time, aiming to show the direction of a stock's trend and potential points where this trend has a higher-than-normal probability of switching directions. The calculation of SAR is complex, so it is recommended reading the references for a better understanding.

The first two technical indicators, ROC and $Ratio$, are applied to all stock indices and exchanges rates, while the other five are only applied to the Ibovespa Index. The features are created using all the following periods:

- ***ROC*:** $n = 1, 4, 8$;
- ***Ratio*:** ROC_1/ROC_m for $m = 4, 8, 16$;
- ***Disparity*:** $n = 8, 16$;
- ***Price Oscillator*:** $n = 8$ and $m = 16$;
- ***Stochastic Oscillator*:** $n = 14$.

The feature set is then scaled into the range of [-1, 1]. The goal of linear scaling is to independently normalize each feature component to the specified range. It ensures the larger value input attributes do not overwhelm smaller value inputs, and also helps reduce prediction errors.

Since this study aims to predict whether the weekly price will go up or down, this problem can be seen as a binary classification task. Therefore, the instances are categorized as "0" or "1" in the research data. "0" means that the next week's index value is lower than the current value, while "1" means that the next week's index is higher.

6.3 The R Project

The R Project is a programming language and software environment for statistical computing and graphics (RPROJECT, 2014). Besides being an open source system, R is highly extensible through the use of free packages submitted by users, which include specific functions for specific areas of study (HORTON; KLEINMAN, 2011).

Some of these packages were very useful for the development of this project. For instance, in the input data collection step, the *Quandl* package (MCTAGGART; DAROCZI, 2014) is used, which connects directly with the API provided by the website (QUANDL, 2014). After that, the *TTR* package (ULRICH, 2013) is used to create the feature set by calculating the technical indicators mentioned previously. In order to implement the Genetic Algorithm, the *GA* package (SCRUCCA, 2013) is used, as it makes creating customized objective functions and genetic operators possible. Finally, the package *caret* (KUHN, 2014),

which provides a set of functions for training and testing predictive models, is used to implement the SVMs Ensemble of the proposed method and the other methods mentioned in chapter 4, as well as to carry out the experiments.

While the detailed presentation of the R Project is not the focus of this project, it is suggested to read its documentation, available in (RPROJECT, 2014), for a better understanding this tool.

6.4 System Parameters

In order to perform the experiments with the proposed method, some parameters should be defined, as detailed in chapters 3 and 5. Therefore, several experiments were performed with different combinations of parameters to find the best one. The set of parameter values that generated the best results is presented in Table 6.1. This project also exploited the high parallelization capability of Ensemble Systems, so each SVM of the ensemble is trained on a separate *thread*.

Table 6.1 – System parameters values

	Parameter	*Value*
Genetic Algorithm	Population Size	25
	Crossover rate	80%
	Mutation rate	10%
	Number of generations	50
	Elitism	1 individual
Ensemble System	Size	10
	SVM kernel	RBF

6.5 Experimental Results

As already mentioned, the proposed method uses a genetic algorithm to select features and parameters for an ensemble system. In order to analyze the performance of this technique, an empirical analysis is performed. This analysis compares the proposed method (*GAENSEMBLE*) with Bagging (using SVM equipped with RBF kernel as base classifier), AdaBoost, and Random Forest. Moreover, a standalone SVM also equipped with RBF kernel

is used to investigate whether applying an ensemble approach leads to performance gains or not.

In order to perform the experiments, the data set is divided into two parts. About 80% of the data (596 instances) is used for training and model building, while 20% (148 instances) is reserved for out-of-sample evaluation and comparison of performances. In order to obtain a better estimation of training accuracy rates, a 10-fold cross-validation (10-CV) method is applied to all methods. In 10-CV, the training dataset is partitioned into 10 subsets. Of these 10 subsets, 9 subsets are used as training data and a single subset is retained as the testing data. This cross-validation process is then repeated 10 times (the number of folds). The advantage of 10-CV is that all instances are used for both training and testing, and each instance is used for testing only once per fold. Thus, all training accuracy results presented in this section refer to the mean over the 10 different test sets.

Except the Random Forest, none of the methods used for comparison employ any technique of feature selection. Therefore, in order to also optimize these methods, a grid search is performed to choose the best set of parameters for each method. All the results presented in this section then refer to the best model created. The parameters of each method that need to be optimized are presented below:

- ***AdaBoost*:** *iter* (number of boosting iterations to perform) and *maxdepth* (maximum depth of trees);
- ***Random Forest*:** *mtry* (number of features randomly sampled as candidates at each split);
- ***Bagging and standalone SVM*:** C and σ.

Figure 6.1 presents a chart comparing the training and testing accuracy (%) of each model. In a general analysis of this chart, it is possible to state that the proposed method has on average a higher accuracy rate than all the other methods both in the training and testing set. For the training data, the *GAENSEMBLE* outperforms Bagging by 2.79%, AdaBoost by 0.13%, Random Forest by 1.48%, and SVM by 2.32%. For the testing set, the proposed method also presented the highest accuracy rate, outperforming Bagging by 6.08%, AdaBoost by 4.06%, Random Forest by 4.93%, and SVM by 3.38%. Figure 6.2 presents the accuracy standard deviation of the cross-validation process. It is important to analyze this information,

because it gives insights about the model stability. Thus, given that the proposed method has the smallest standard deviation, it can be said that it is also the most stable method.

Figure 6.1 – Classification accuracies in the training and testing set

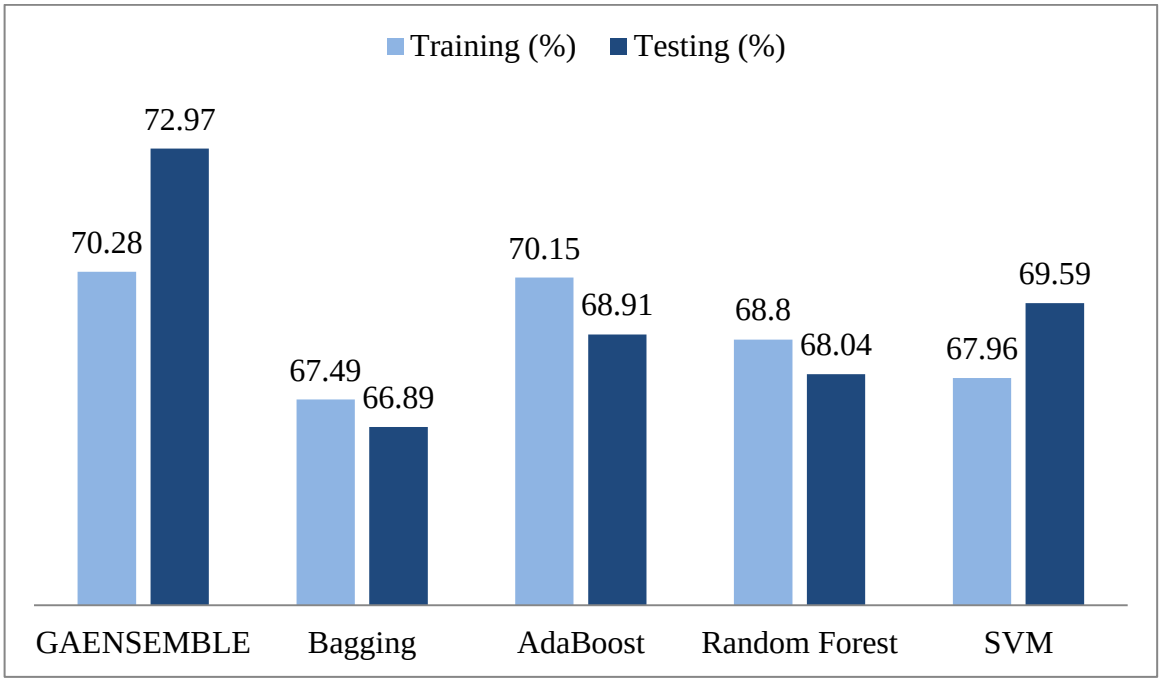
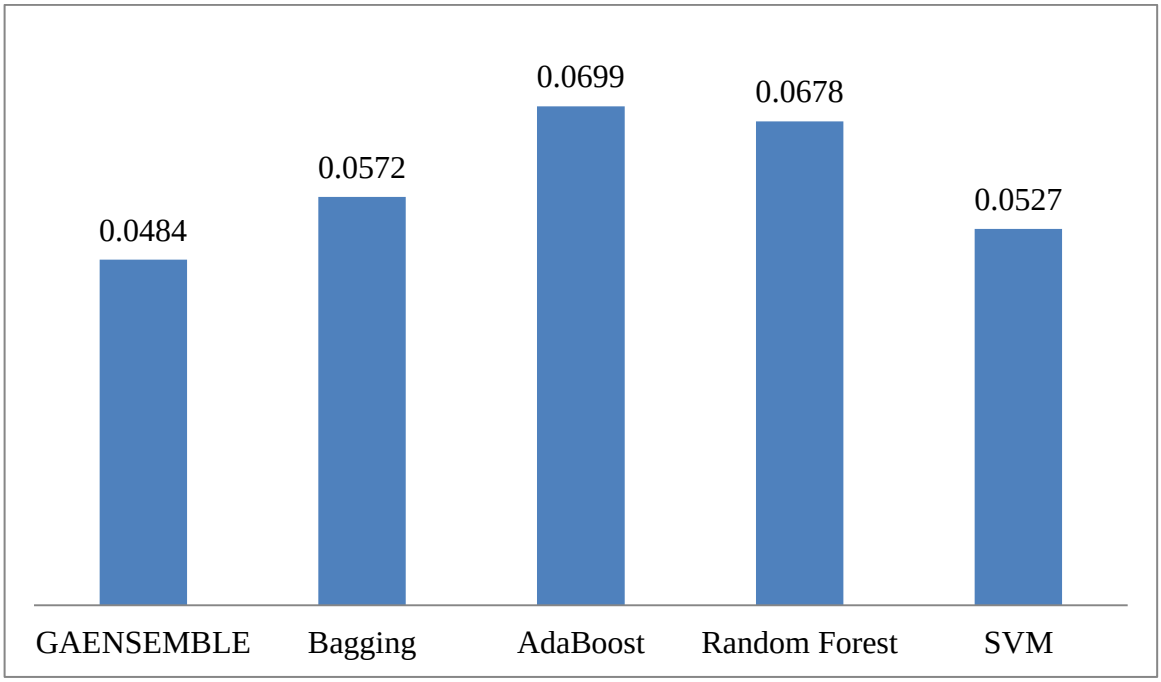


Figure 6.2 – Cross-Validation standard deviations



In addition, it is applied the McNemar's statistical test to further analyze the performance of each method on the testing set. This test is performed to examine whether differences of performances are statistically significant or not. This test is a nonparametric test

for two related samples and may be used with nominal data (BOSTANCI; BOSTANCI, 2012). Table 6.4 shows the results of the McNemar's test (*p*-values) when comparing the proposed method with the other methods. Adopting a significance level of 0.05, a p-value less than 0.05 indicates that there is a significant difference among the two compared methods. As can be seen from Table 6.2, the accuracies on the testing set can be considered significantly different, which confirms that the *GAENSEMBLE* outperforms the other models.

Table 6.2 – Results of McNemar's test (p-values) for the pairwise comparison

	Bagging	*AdaBoost*	*Random Forest*	*SVM*
GAENSEMBLE	0.0116	0.0325	0.0297	0.0415

Finally, performing feature selection and parameter optimization simultaneously generates an enormous search space due to the large number of features combinations and the large range of parameters values. Because of this dimensionality, optimizing the ensemble system using the GA spent lots of time. Table 6.3 presents the time taken, in seconds, to optimize each method in the training data. As expected, the *GAENSEMBLE* took longer than any other method.

Table 6.3 – Training runtimes

	Runtime (s)
GAENSEMBLE	*15650.71*
Bagging	*52.18*
AdaBoost	*8664.91*
Random Forest	*257.46*
SVM	*35.33*

7 CONCLUSIONS AND SUGGESTIONS

This study proposed the use of machine learning algorithms, together with Technical Analysis tools, to forecast stock market movements. As a result, an Ensemble System based on Genetic Algorithm was designed to predict the weekly movement direction of Bovespa Index. In this method, a GA was applied to enhance the classification accuracy of the SVM Ensemble by performing feature selection and parameter optimization. Along with this, Technical Indicators were used to extract features as relevant as possible from historical data of Stock and Forex markets.

In order to prove the feasibility of the proposed method, experiments to compare it with well-known ensemble learning methods such as Bagging, Boosting and Random Forest were performed. As demonstrated in the experimental results, the proposed method outperforms the other ones in predicting whether the weekly price of Bovespa Index will rise or fall. Thus, this work concluded that the *GAENSEMBLE* method provides a promising alternative for financial time-series forecasting.

Finally, it is known that there is not a perfect method for forecasting financial markets because of its complexity and volatility. However, the proposed method proved to be accurate enough so that investors can consider using it together with other forecasting techniques to identify better investment opportunities. Moreover, traders could use it aiming to reduce the risk and to increase the profitability of their trades.

7.1 Future Researches

The comparison of the results obtained by the proposed methods with the ones from the other ensemble methods, demonstrated a large potential for applying it in the field of financial time series forecasting. Therefore, aiming to improve the overall classification performance of the proposed method, future researches can investigate the following possibilities:

- Exploring new Technical Indicators or other types of Financial data that might generate more informative features, which would further increase the performance of the proposed method;

- Performing a final tuning of the parameters of each SVM through a grid search on a subset of parameters generated depending on the final solution of the GA;
- Analyzing financial series with more historical data, so that the proposed method can learn more patterns from past data;
- Developing an algorithm that better utilize the power of parallelization provided by Genetic Algorithms and Ensemble Systems, speeding up the training process.

REFERENCES

ABU-MOSTAFA, Yaser; ATIYA, Amir. **Introduction to financial forecasting.** Applied Intelligence, v. 6, n. 3, p. 205-213, 1996.

ALLEN, Franklin; KARJALAINEN, Risto. **Using genetic algorithms to find technical trading rules.** Journal of Financial Economics, v. 51, n. 2, p. 245-271, 1999.

BACK, Thomas. **Evolutionary Algorithms in Theory and Practice.** New York: Oxford University Press, 1996.

BLOOMBERG. **World Stock Indexes**. Available in: <http://www.bloomberg.com/markets/stocks/world-indexes>. Visited on: November 2014.

BM&FBOVESPA: The New Exchange. **Bovespa Index - Ibovespa**. Available in: <http://www.bmfbovespa.com.br/indices/ResumoIndice.aspx?Indice=Ibovespa&Idioma=en-us>. Visited on: November 2014.

BOSTANCI, Betul; BOSTANCI, Erkan. **An Evaluation of Classification Algorithms Using Mc Nemar's Test**. Proceedings of the 17th International Conference on Bio-Inspired Computing: Theories and Applications, 2012. p. 15-26.

BRAMLETTE, Mark F. **Initialization, Mutation and Selection Methods in Genetic Algorithms for Function Optimization.** Proceedings of the 4th ICGA, San Diego, CA, 1991. p. 100-107.

BREIMAN, Leo. **Bagging predictors**. Machine Learning, v. 24, n. 2, p. 123-140, aug. 1996.

BREIMAN, Leo. **Random Forests**. Machine Learning, v. 45, n. 1, p. 5-32, oct. 2001.

CANUTO, Anne; NASCIMENTO, Diego. **A Genetic-Based Approach to Features Selection for Ensembles Using a Hybrid and Adaptive Fitness Function**. Proceedings of the IJCNN, Brisbane, Australia, 2012. p. 1-8.

CAO, L. J.; TAY, F. E. H.. **Financial forecasting using support vector machines.** Neural Computing And Applications, v. 10, v. 2, p. 184-192, 2001.

CHOI, J.H.; LEE, M.K.; RHEE, M.W. **Trading S&P 500 stock index futures using a neural network.** Proceedings of the Annual International Conference on Artificial Intelligence Applications on Wall Street, New York, NY, 1995. p. 63-72.

DAVIDSON, Alexander. **How the Global Financial Markets Really Work**: The Definite Guide to Understanding International Investment and Money Flows. 1ª ed. Philadelphia: Kogan Page Limited, 2009.

DIETTERICH, Thomas. **Ensemble Methods in Machine Learning**. First International Workshop on Multiple Classifier Systems. New York: Springer-Verlag, p. 1-15, 2000.

EIBEN, Agoston; RUTTKAY, Zsófia. **Self-adaptivity for constraint satisfaction: Learning penalty functions**. Proceedings of the 3rd IEEE Conference on Evolutionary Computation, Nagoya, Japan, 1996. p. 258-261.

EIBEN, Agoston; SMITH, James. **Introduction to Evolutionary Computing**. Berlin Heidelberg New York: Springer-verlag, 2003.

FREUND, Yoav; SCHAPIRE, Robert. **Experiments with a new boosting algorithm**. Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 1996. p. 148-156.

FRÖHLICH, Holger; CHAPELLE, Olivier. **Feature selection for support vector machines by means of genetic algorithms.** Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, CA, 2003. p. 142-148.

GALANT, Mark; DOLAN, Brian. **Currency Trading for Dummies**. 1ª ed. Indianapolis: Wiley Publishing, Inc., 2007.

GOLDBERG, David E. **Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking**. Complex Systems, v. 5, n. 2, p. 139-167, 1991.

HALL, J. W. Adaptive selection of US stocks with neural nets. In: Deboeck GJ. (Ed.). **Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets.** New York: Wiley, p. 45-65, 1994.

HERRERA, Francisco; LOZANO, Manuel. **Gradual distributed real-coded genetic algorithms.** IEEE Transactions on Evolutionary Computation, n. 4, v. 1, p. 43-63, 2000.

HOLLAND, John H. **Adaptation in Natural and Artificial Systems**. Ann Arbor: The University Of Michigan Press, 1975.

HORTON, Nicholas J., KLEINMAN, Ken. **Using R for Data Management, Statistical Analysis, and Graphics.** 1ª ed. [S.l.]: CRC Press, 2011.

INVESTOPEDIA. **Financial Market**. Available in: <http://www.investopedia.com/terms/f/financial-market.asp>. Visited on: November 2014.

KAYA, Yilmaz; UYAR, Murat; TEKIN, Ramazan. **A Novel Crossover Operator for Genetic Algorithms: Ring Crossover**. Computing Research Repository Journal, 2011.

KIM, Kyoung-jae. **Financial time series forecasting using support vector machines.** Neurocomputing, v. 55, n. 2, p. 307-319, 2003.

KIMOTO, T. et al. **Stock market prediction system with modular neural network.** Proceedings of the International Joint Conference on Neural Networks, San Diego, CA, 1990. p. 1-6.

KOZA, John R. **Genetic Programming: On the Programming of Computers by Means of Natural Selection**. 1ª ed. Cambridge: MIT Press, 1992.

KUHN, Max. **caret: Classification and Regression Training.** 2014. Available in: <http://CRAN.R-project.org/package=caret>. Visited on: November 2014.

KUNCHEVA, Ludmila I. **Combining Pattern Classifiers: Methods and Algorithms.** [S.I.]: Wiley-Interscience, 2004.

LAHMIRI, Salim. **Intelligent Ensemble Systems for Modeling NASDAQ Microstructure: A Comparative Study.** Artificial Neural Networks in Pattern Recognition. Cham: Springer International Publishing, p. 240-251, 2014.

LEVINSON, Marc. **Guide to Financial Markets**. 4ª ed. London: The Economist, 2005.

LI, Jin. **FGP: A Genetic Programming Based Financial Forecasting Tool**. 2000. 190f. Thesis (Doctor of Philosophy in Computer Science) - Department of Computer Science, University of Essex, Colchester, 2000.

LIMA, Naiyan; NETO, Adrião; MELO, Jorge. **Creating an Ensemble of Diverse Support Vector Machines Using Adaboost.** Proceedings of the 2009 International Joint Conference on Neural Networks, Atlanta, Georgia, 2009. p. 2342-2346.

LING, Yun; YUE, BaoLong; ZHANG, Hua. **A New Wrapped Ensemble Approach for Financial Forecast.** Journal of Intelligent Systems, v. 23, n. 1, p. 21-32, 2013.

MCTAGGART, Raymond; DAROCZI, Gergely. **Quandl: Quandl Data Connection.** 2014. Available in: <http://CRAN.R-project.org/package=Quandl>. Visited on: November 2014.

MELANIE, Mitchell. **An Introduction to Genetic Algorithms**. Cambridge: Mit Press, 1998.

MICHALEWICZ, Zbigniew. **A Genetic Algorithms for the Linear Transportation Problem**. IEEE Trans. on Systems, Man, and Cybernetics, n. 21, v. 2, p. 445-452, 1991.

MICHALEWICZ, Zbigniew. **Genetic Algorithms + Data Structures = Evolution Programs**. 3ª ed. New York: Springer-verlag, 1996.

MURPHY, John J. **Technical Analysis of the Financial Markets**. 1ª ed. New York: New York Institute of Finance, 1999.

OPITZ, David. **Feature selection for ensembles.** Proceedings of the 16th National Conference on Artificial intelligence, Orlando, FL, 1999. p. 379-384.

PADILHA, Carlos et al. **An genetic approach to Support Vector Machines in classification problems**. Proceedings of the IJCNN 2010, Barcelona, Spain, 2010. p. 1-4.

PADILHA, Carlos; NETO, Dória; MELO, Jorge. **Random Subspace Method and Genetic Algorithm Applied to a LS-SVM Ensemble.** Proceedings of the 22nd international conference on Artificial Neural Networks and Machine Learning, Lausanne, Switzerland, 2012. p. 164-171.

PICEK, Stjepan; JAKOBOVIC, Domagoj; GOLUB, Marin. **On the Recombination Operator in the Real-Coded Genetic Algorithms**. Proceedings of the IEEE Congress on Evolutionary Computation, Cancún, Mexico, 2013. p. 3103-3110.

PINTO, Rafael Coimbra. **Online Incremental One-Shot Learning of Temporal Sequences.** 2011. 106 f. Thesis (Master) - Graduate Program in Computer Science, Institute of Informatics, Federal University Of Rio Grande do Sul, Porto Alegre, 2011.

PROCHNOW, Fabio. **Genetic Programing for Time Series Forecasting Applied to Financial Markets.** 2013. 61 f. Final Paper (Bachelor) - Bachelor of Computer Science, Institute of Informatics, Federal University Of Rio Grande do Sul, Porto Alegre, 2013.

QUANDL. Available in: <https://www.quandl.com>. Visited on: November 2014.

ROCKEFELLER, Barbara. **Technical Analysis for Dummies**. 2ª ed. Indianapolis: Wiley Publishing, Inc., 2011.

RPROJECT. **The R Project for Statistical Computing.** Available in: <http://www.r-project.org/>. Visited on: November 2014.

SCRUCCA, Luca. **GA: A Package for Genetic Algorithms in R.** Journal Of Statistical Software, 53(4), p. 1-37. 2013. Available in: <http://www.jstatsoft.org/v53/i04/>. Visited on: November 2014.

SEFIANE, Slimane; BENBOUZIANE, Mohamed. **Portfolio Selection Using Genetic Algorithm.** Journal of Applied Finance & Banking, v. 2, n. 4, p.143-154, 2012.

SETTY, Venugopa; RANGASWAMY, T. M.; SUBRAMANYA, K. N. **A Review on Data Mining Applications to the Performance of Stock Marketing.** International Journal of Computer Applications, v. 1, n. 3, p. 24-34, 2010.

TAN, Pang-ning; STEINBACH, Michael; KUMAR, Vipin. **Introduction to Data Mining**. Boston: Pearson Education, Inc., 2006.

TAY, F. E. H.; CAO, L. J. **Improved financial time series forecasting by combining Support Vector Machines with self-organizing feature map.** Intelligent Data Analysis, v. 5, n. 4, p. 339-354, 2001.

ULRICH, Joshua. **TTR: Technical Trading Rules.** 2013. Available in: <http://CRAN.R-project.org/package=TTR>. Visited on: November 2014.

VAPNIK, Vladimir. **The Nature of Statistical Learning Theory.** New York: Springer-verlag New York, 1995.

WHITLEY, Darrell; STARKWEATHER, Timothy; FUQUAY, D'Ann. **Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator**. Proceedings of the 3rd ICGA, Fairfax, VA, 1989. p. 133-140.

WITTEN, Ian; FRANK, Eibe. **Data Mining: Practical Machine Learning Tools and Techniques.** 3ª ed. San Francisco: Morgan Kaufmann Publishers, 2005.

WURGLER, Jeffrey. **Financial markets and the allocation of capital**. Journal of Financial Economics, n. 58, p. 187-214, 2000.

YAHOO FINANCE. **IBOVESPA**. Available in: <http://finance.yahoo.com/echarts?s=^BVSP>. Visited on: November 2014.

YAHOO FINANCE. **S&P 5000**. Available in: <http://finance.yahoo.com/echarts?s=^GSPC>. Visited on: November 2014.

YOON, Y.; SWALES, G. **Predicting stock price performance: a neural network approach.** Proceedings of the 24th Annual Hawaii International Conference on System Sciences, Hawaii, 1991. p. 156-162.

ZHANG, Byoung-Tak; KIM, Jung-Jib. **Comparison of selection methods for evolutionary optimization.** Evolutionary Optimization, v. 2, n. 1, p. 55–70, 2000.

ZHAO, Tianzhong et al. **Classifier Ensemble Based-on AdaBoost and Genetic Algorithm for Automatic Image Annotation.** Proceedings of the 2008 IEEE International Conference of Information and Automation, Zhangjiajie, China, 2008. p. 1469-1473.

ZULUTRADE. **Rate Charts.** Available in: <http://www.zulutrade.com/charts>. Visited on: November 2014.