NOVA IMS Information Management School

MEGI

Mestrado em Estatística e Gestão de Informação

Master Program in Statistics and Information Management

Forecasting Sovereign Bonds Markets Using Machine Learning

Forecasting the Portuguese Government Bond Using Machine Learning Approach

Tiago Alexandre Rodrigues de Sousa Vieira

Dissertation presented as partial requirement for obtaining the Master's degree in Statistics and Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

FORECASTING SOVEREIGN BONDS MARKETS USING MACHINE LEARNING

Forecasting the Portuguese Government Bond Using

Machine Learning Approach

by

Tiago Alexandre Rodrigues de Sousa Vieira

Dissertation presented as the partial requirement for obtaining a Master's degree in Statistics and Information Management, with a specialization in Risk Analysis and Management

Advisor: Mauro Castelli

October 2020

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude towards my masters tutor, Professor Mauro Castelli. I would have not been able to engage so deeply in this project without all the support provided by him.

For all support, as well as the encouragement to move forward, I would like to thank my mother, father and brothers.

I am extremely grateful to my girlfriend for all time and patience devoted to this dissertation. Your support in the final steps of this project was essential.

Finally, I would like to thank my friends for saying I would not be able to finish this dissertation. Your reverse encouragement was the key. It took time, but as you can see, I made it.

ABSTRACT

Financial markets, due to their non-linear, volatile and complex nature turn any type of forecasting into a difficult task, as the classical statistical methods are no longer adequate. Many factors exist that can influence the government bonds yields and how these bonds behave. The consequence of the behaviour of these bonds are extended over geographies and individuals.

As the financial markets grow bigger, more investors are trying to develop systematic approaches that are intended to predict prices and movements. Machine Learning algorithms already proven their value in predicting and finding patterns in many subjects. When it comes to financial markets, Machine Learning is not a new tool. It is already widely used to predict behaviours and trends with some degree of success.

This dissertation aims to study the application of two Machine Learning algorithms - Genetic Programming (GP) and Long Short-Term Memory (LSTM) - to the Portuguese Government 10Y Bond and try to forecast the yield with accuracy. The construction of the predictive models is based on historical information of the bond and on other important factors that influence its behaviour, extracted through the Bloomberg Portal.

In order to analyse the quality of the two models, the results of each algorithm will be compared. An analysis will be presented regarding the quality of the results from both algorithms and the respective time cost. In the end, each model will be discussed and conclusions will be taken about which one can be the answer to the main question of this study, which is “What will the Yield of the Portuguese Government 10Y Bond be on T+1?”.

The results obtained showed that Genetic Programming can create a model with higher accuracy. However, Long Short-Term Memory should not be ignored because it can also point to good results. Regarding execution time, velocity is a problem when it comes to Genetic Programming. This algorithm takes more time to execute compared to LSTM. Long Short-Term Memory is considerably quicker to get results. In order to take the right decision about which model to choose one must keep in mind the priorities. In case accuracy is the priority, Genetic Programming will be the answer. Nevertheless, when velocity is the priority Long Short-Term Memory should be the choice.

KEYWORDS

Portuguese Government Bonds; Machine Learning; Genetic Programming; Long-Short-Term Memory; Financial Markets

INDEX

FIGURES INDEX

TABLE INDEX

ACRONYMS AND CONCEPTS

Bonds Fixed income instrument that represents a loan made by an investor to a Government or corporate. It can be characterized by different amounts, maturities, guarantees and coupons. Bond details include the end date when the principal of the loan is due to be paid to the bond owner and usually includes the terms for variable or fixed interest payments made by the borrower. The credit rating agencies (like Standard&Poor's, Moody's, and Fitch Ratings) can classify these bonds through the credit quality (risk) of the issuer and the guarantees of the bond. The market prices of bonds are based on their characteristics and they can be quoted by price or yield. [1]

Sovereign Bonds Sovereign bonds are fixed income debt issued by a sovereign Government. These bonds are usually considered the lowest risk available in their respective countries and thus offer the lowest rates in the country. The interest rates are influenced by the monetary policies of the country's central banks, which aim to avoid sudden changes. [2]

Portugal Fixed Rate Government Bond (OT) Security representing a medium to long-term Republic of Portugal loan, with a face value of EUR 0.01, with time to maturity between one and thirty years. OTs are placed through auctions, syndicates or by tap and pay (or not) interest periodically and are redeemed on maturity at face value. [3]

Yield Measure of the return of a bond. [3]

Term Structure of Interest Rates (commonly known as the yield curve) Relationship, at a given point in time, between yields and time to maturity of a set of bonds of the same class of risk. [3]

Agência de Gestão da Tesouraria e da Dívida Pública – IGCP Public entity responsible for the integrated management of cash, funding and the direct debt management, which includes, under the applicable law, the debt of public corporations whose financing is ensured through the state budget. It is also responsible for coordinating the financing of autonomous administrative and financial services and funds. [4]

Primary Dealers Financial intermediaries recognized by IGCP for their capacity to place OT and to ensure liquidity in the secondary market of these securities. [5]

Machine Learning Learning process of computers from experience and automatically improve the efficiency of their own program's execution. A simple but effective rote-learning facility can be provided within the framework of a suitable programming language. [6]

Genetic Programming Genetic Programming is a domain-independent problem-solving approach in which computer programs are evolved to solve, or approximately solve, problems. Genetic algorithms are based on the Darwinian principle of reproduction and survival of the fittest among string structures to form a search algorithm and analogues of naturally occurring genetic operations such as Crossover (sexual recombination) and Mutation. [7] [8]

Long Short-Term Memory LSTM's are a special subset of Recurrent Neural Networks (RNN) that can capture context-specific temporal dependencies for long periods of time. Each LSTM neuron is a memory cell that can store other information. While neurons in normal neural network merely take in their previous hidden state and the current input to output a new hidden state, a LSTM neuron also takes in its old cell state and outputs its new cell state. [9]

1. INTRODUCTION

The fixed income market has a strategic importance to the global economy. The global bond market is valued over 102.8$ trillion, as compared to the $74.7 trillion valuation of the global equity market [10]. In Portugal, the sovereign bonds market represents 145.496€ millions, compared to the 4.069€ millions market value of PSI20 Index (benchmark stock market index of companies that trade on Euronext Lisbon) [11].

As there is a greater variety of bonds than stocks, secondary trades of a bond are not negotiated only in regulated markets but can be dealt through direct trader-to-trader negotiations. This system reduces liquidity and transparency in trades, which increases the risk for buyers and sellers. Although most bonds are available to individual investors, the majority are purchased by large institutional funds and insurance companies with the purpose of hedging risk.

A major challenge confronting speculators, investors, businesses and governments is to accurately forecast price's and yield's movements in financial markets. In their quest to forecast the market prices, they assume that future occurrences are based at least partially on present and past events. The existence of so many price and yield trends in financial markets (a trend is a sustained increase or decrease in price over a substantial period) and the enormous number of correlations among fundamental events and economies affecting the markets, makes this task even more challenging for the human mind.

With the development of Machine Learning, investors are hoping that all these trends, correlations and other market mysteries can be unraveled because of the capability that these new tools provide when it comes to processing all the information that humans were not able to process by themselves.

Machine Learning refers to programming a computer for optimizing a performance criterion using example data or past experiences. A model is defined by parameters and learns with the repeated execution of a computer program. The main goal is to optimize the parameters of the model using the training data and past experiences. The model may be predictive if it makes projections for the future, descriptive if it gets knowledge from data, or both.

There are already a lot of different studies that tried to predict prices and movements of different products and markets (from the USA market to European and Asian markets), using several Machine Learning algorithms [12] [13]. Most of them focus on trying to predict stock markets prices and movements and only few try to forecast bonds markets yields or movements [14].

In Portugal there are already some studies related to forecasting stock prices using different Machine Learning approaches [15]. Most of these studies tried to forecast stock market prices, but none of them tried to predict bond yields, neither of corporate bonds nor Government bonds. The characteristics of stock markets and bond markets are very different, but in their essence, the principles that regulate these markets and investors are very similar.

Portuguese Government bonds represent a market of 145.496€millions, distributed by 22 issues and traded all days through 19 market makers. These bonds are very different between themselves –

their maturity goes from 1 to 30 years, the currency can be different and the coupon could have a fixed or floating rate (see figure 17 in appendix 9.1).

Forecasting with accuracy the exact yield or the movement of the yield of the Portuguese Government 10Y Bond is important for these market makers, but also for the IGCP and for investors. To market markets it is important to forecast this yield or identify the trend in the movement of prices in order to ensure a fair price in the secondary market of these bonds and manage the liquidity of this market. To professional and non-professional investors, investment funds and other institutional investors it is important to understand the movement of the market in order to achieve profit and to manage their own portfolios. To IGCP, predicting the bond yield or movements impacts their strategies on issuing new bonds and their management of the amount of debt and its duration. The impact of forecasting the yield of the Portuguese Government bond is huge in many aspects and impacts within a certain way the life of all Portuguese citizens, in a way that it is their Government debt and the health of their national financial system.

1.1. FORECASTING PROBLEM

When the subject is money, everyone wants to know which will be the prices in the future, as this information is what allows the making of decisions that obtain the best relation regarding cost and opportunity (opportunity costs represent the benefits an individual, investor or business misses out on when choosing one alternative over another [16]). In financial markets this is the most important concept, as it is what differentiates between profit and loss. All participants in financial markets, no matter the type of product being traded or the maturity of the product, want to know/predict which will be the price of the product in T+1 (a future moment, e.g. tomorrow = today +1). A lot of effort is required by the participants in order to achieve this goal and to try to forecast the most accurate yield or movement of the market.

Machine Learning algorithms have already been successfully applied to forecasting prices, improving trading strategies and financial modelling. These algorithms are developed to find the best forecast function, $\hat{f}$, through the identification of hidden relations and patterns in the data either by parameter optimization, creation of expressions, variable selection or all of them combined.

Considering a short-term forecasting problem (one-day head forecast), the objective of this project is to predict the yield of Portuguese Government 10Y Bond in a given day (T+1) using a set of input variables that influence the yield of the bond in the previous day (T).

This forecasting problem can be described as follows:

Given a set of input variables $x_1, x_2, \dots, x_m$:

$$\hat{y}_{t+1} = \hat{f}(x_1, x_2, \dots, x_m)$$

Where $\hat{y}_{t+1}$ is the next value of the time-series and $\hat{f}$ is the forecasting function.

The term structure of interest rates provides the information on bond returns of different maturities. Short term interest rates are directly correlated with monetary policy implementation while long term bonds yield-to-maturity reflect the expectations of investors on the future of economic activity [17]. The tenor of ten years is considered by all participants in financial markets as a benchmark for

sovereign bonds. In order to have an approximation of the future term structure for Portugal this tenor will be used. When the yield on this tenor changes it is very common that its curve follows a similar movement. It could be in a different magnitude, but it will probably be in the same direction. Despite not being available every day, a bond with a constant maturity in the 10-year benchmark, it is possible to calculate a theoretic yield of such bond by interpolating the daily term structure of interest rates. See figure 18 (appendix 9.1).

1.2. RESEARCH OBJECTIVES

Machine Learning was already used to predict stock prices in the national market (PSI-20 Index) but it was not used to predict bonds yield (neither corporate bonds nor sovereign bonds). In order to fulfil the lack of knowledge in this field, this study aims to apply two different algorithms in order to find the best function for forecasting the yield of Portuguese Government 10Y Bond.

The two chosen algorithms, Genetic Programming (GP) and Long Short-Term Memory (LSTM), were already used in many different studies and showed that they could be a good tool for forecasting prices, specially on Stock Market. Both algorithms were already used in forecasting stock market prices. GP was already used to predict national stock market price. However, it was not possible to find any study where LSTM had been applied in the Portuguese market. The present study intends to understand what the behaviour of both algorithms in the bond market is and compare their accuracy. Beyond the comparison between the accuracy of forecasts, this study will explore the cost benefits between them. In the end, the aim is to find the best tool for forecasting the yield of Portuguese Government 10Y Bond to support the activity of all the participants of this market.

- Machine Learning was already used to predict stock prices in the national market (PSI-20 Index) but it was not used to predict bonds yield (neither corporate bonds nor sovereign bonds). In order to fulfil the lack of knowledge in this field, this study aims to apply two different algorithms in order to find the best function for forecasting the yield of Portuguese Government 10Y Bond.

- The two chosen algorithms, Genetic Programming (GP) and Long Short-Term Memory (LSTM), were already used in many different studies and showed that they could be a good tool for forecasting prices, specially on Stock Market. Both algorithms were already used in forecasting stock market prices. GP was already used to predict national stock market price. However, it was not possible to find any study where LSTM had been applied in the Portuguese market. The present study intends to understand what the behaviour of both algorithms in the bond market is and compare their accuracy. Beyond the comparison between the accuracy of forecasts, this study will explore the cost benefits between them. In the end, the aim is to find the best tool for forecasting the yield of Portuguese Government 10Y Bond to support the activity of all the participants of this market.

If these two algorithms cannot forecast with accuracy the yield of the bond, they will, hopefully, at least give the direction of the yield. In this case, they can forecast if the yield of bonds goes up or down, no matter what the proportion of this movement is. This can, at least, give an idea if, in that moment, the price trend is going up or down, allowing participants to take some action regarding that information.

In the past few years Portuguese Government Bonds yields suffered an enormous instability. The PGB 1.95% 15/06/2029 (the benchmark on 2019 to 10y of Portugal) went from 1.8% yield to 0.4% only in the past year. This instability has a long history in the last decade. After the 2012 crisis a lot of movements and trends happened and not all of them were discovered and/or analysed. Taking advantage of Machine Learning, this study aims to find more movements/trends in that period that were not already discovered.

1.3. DISSERTATION DESIGN

Considering the goal of this study, the methodology will be data driven. This methodology will be divided in 5 stages: defining the problem (phase 1), developing an approach to the problem (phase 2), formulating a research design (phase 3), collecting/preparing/analysing data (phase 4) and preparing/presenting the report (phase 5).

In phase 1, the problem is defined based on a real problem observed in financial markets. Looking at financial markets in Portugal the question "What will the yield of the Portuguese government 10Y Bond be on T+1?" is one of the most important questions to many players that play an important role every day.

Phase 2 consists of defining the first steps of the study, considering the main goal. In order to answer the question (problem) defined in the last phase, Machine Learning algorithms are used to predict the yield of the bond. The two algorithms chosen were Genetic Programming and Long Short-Term Memory. Four hypotheses were defined

- Overall, GP models perform best in forecasting and have the best cost/benefit relationship;
- Overall, GP models perform best in forecasting but don't have the best cost/benefit relationship;
- Overall, LSTM models perform best in forecasting and have the best cost/benefit relationship;
- Overall, LSTM models perform best in forecasting but don't have the best cost/benefit relationship;

The main problem of this study and the tools chosen to get the best answer result in more secondary questions, such as:

- Is it possible to forecast the precise yield of a bond?
- How accurate can this forecast be?
- What are the costs in terms of time and resources of this forecast?
- Can algorithms forecast the direction of the yield movement?

In phase 3 the exploratory study starts, and the key variables and size of the sample are defined. It is on this phase that the structure of the equation to introduce in the models is defined, as well as all variables to collect from the sample.

Phase 4 is the main phase. The data is collected (from Bloomberg) and pre-processed to input to the algorithms. The data is split in training and test sets. To configure and apply these algorithms a one code library accessible online is used (for Genetic Semantic Genetic Programming) and a script is written in order to apply LSTM. Concerning the Genetic Programming approach, more specifically Geometric Semantic Genetic Programming (GSGP) the implementation used is freely accessible at http://gsgp.sourceforge.net, a script created and documented by Castelli and Vanneschi [18].

In phase 5 is the evaluation of the results of each algorithm regarding the goals and questions defined. To compare the models derived from both algorithms the Mean Absolute Error is used as measure of the error in both models.

In the last phase, all information is summarized in a written report and presented to the public.

The scheme bellow, figure 1, represents the research methodology and design adopted, as explained in the paragraphs above.

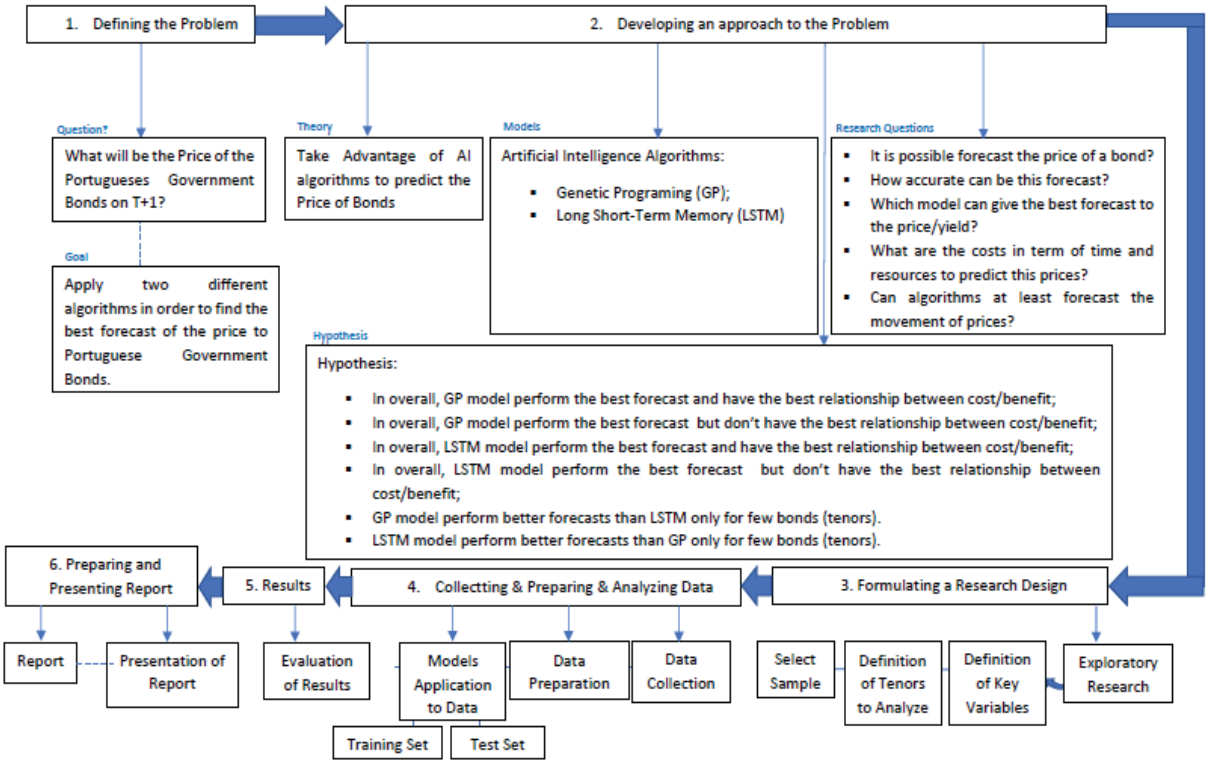

Figure 1- Research Methodology and Desing

2. LITERATURE REVIEW

There is a great amount of literature about forecasting financial markets and it includes a lot of different approaches, from sophisticated Machine Learning algorithms to classical statistical models. During the writing of this dissertation different studies were read and analysed in order to understand which approaches should be followed and in which context should they be used. Different markets, products and resources demand different models.

Many of the most popular classical statistical models currently used by academic researchers have been mostly used to explain the behaviour of term structure interest rates. A partial listing of these interest rate models includes: Merton [18], Vasicek [20] or Cox, Ingersoll, and Ross [21]. Regarding all these models, modelling and forecasting the term structure of interest rates is by no means an easy endeavour. Long and short maturities are known to react quite differently to shocks hitting the economy.

One of the most widely used statistical models for fitting the term structure of interest rates is Nelson's Siegel Model [22]. This model is extensively used by central banks and monetary policy makers [23] [24] Fixed-income portfolio managers use the model to understand how to immunize their portfolios [25] [26]. In academic research, one example of the application of this model is to describe the spot rate curves of Deutsche Mark denominated bonds, in order to calculate the risk structure of interest rates [27]. Fabozzi, Martellini, and Priaulet [28] and Diebold and Li [29] compared this model against others for forecasting the term structure of interest rates and found it performs well, especially for longer forecast horizons. In 2007 Pooter [30] set up a showdown between the random walk, the original Nelson's Siegel model (with and without macro factors) and the no arbitrage models (with and without macro factors). Interestingly, the authors of this last study mentioned that the predictive ability of individual models varies considerably over time with a prime example being the Nelson and Siegel. In this study it was found that models that incorporate macroeconomic variables seem more accurate in periods during which the uncertainty about the future path of interest rates is substantial. Whereas modelling interest rates movements over time is already an arduous task, accurately forecasting future rates is an equally difficult challenge.

Regarding the Machine Learning approach, the suitability of this methodology for predictive analysis makes it particularly attractive for financial forecasting in the context of returns predictability, market movements and risk premia measurement [31]. Throughout the years, Machine Learning methodologies have helped to improve the empirical understanding of different asset prices, from forecasting or modelling term structures interest rates and stock market prices to currency exchange rates.

When it comes to forecasting currency exchange rates, many approaches and algorithms were used by different researchers. All of them tried to achieve an efficient model that is able to provide a reliable prediction regarding the rate [32] [33]. Most of these studies prove that Machine Learning can be a valuable tool for this task, but further work will be necessary in order to achieve a reliable model.

Due to its profitability and easiness of penetration, the stock market has been one of the most explored financial fields when it comes to Machine Learning. Algorithms such as Artificial Neural Networks were widely used for stock market prices prediction in many contexts. Dase R.K. and Pawar

D.D. [34] provide a review of the extended literature presented about the use of it along the last years. In this article we can observe how extensive the application can be for this type of tool. The most successful trading model in stock markets should be the most profitable one. Throughout the years, several reliable models were created by researchers and companies, demonstrating that, whether in bull or bear market, the models significantly outperform [35] [36]. So far, at this moment, it is possible to find a study or even a model for every regulated stock market and most of them show good results, meaning once again that Machine Learning can be a powerful help for forecasting asset prices but can also provide tools to create models that help in predicting financial crisis. A financial crisis can emerge from a series of local and/or regional market shocks and can evolve posteriori into a global financial crisis due to the interconnectedness of the financial markets. In order to prevent this type of events, or at least minimize their effects, this method can be again a way to find patterns and anticipate them [37].

Despite major advances in the term structure of interest rates modelling in the last years, little attention has been paid to the key practical problem of forecasting the yield curve. Interest rate point forecasting is crucial for bond portfolio management, and interest rate density forecasting is important for both derivatives pricing and risk management. Predicting a term structure is one of the most complex tasks in financial markets, due to the complexity of the underlying variables. Alongside with this complexity, the difficulty to join this market makes this field one of the most unexplored in financial markets. Despite all these, some Machine Learning algorithms were already studied in this context, such as Neural Networks [38][39] and the Gaussian Process [40]. Regardless the algorithm, the results of using Machine Learning for this task are quite satisfactory, turning it into a better choice for the medium- and long-term structures associated with the yield curve. With all the research done until the present moment, it is possible to understand that Machine Learning can be a respectable instrument that can allow the better understanding of all relationships between underlying variables and the forecast of behaviour of term structures of interest rates. All this knowledge will allow the stakeholders of financial markets to have a better perspective of the market and to take all necessary actions to manage their portfolios.

Genetic Programming is not a completely new theme. The origins of GP go back at least to the 1950s and since then it has been the theme of a lot of research papers. One of the first descriptions was made by Koza (1992) – “Genetic Programming is a problem-solving approach in which computer programs evolve producing new individuals, based on Darwinian principle, changing the syntax of the parents without taking into account the semantics of the individuals, in order to solve a problem”. One of the most important insights in Genetic Programming was Genetic Semantic [41]. This new perspective provides new insights on the relation between program syntax and semantics, search operators, fitness landscape and allows for principled formal design of semantic search operators for different classes of problems. This new methodology in GP has a strong limitation of producing offsprings that are larger than their parents, creating programs with massive size. To overcome this limitation, Vanneschi, Castelli and Silva [43] introduced a new implementation, making it possible to use these operators in an efficient and useful way. This new approach was explored, and the applications were presented in four real life problems related to the electricity field and pharmacokinetics [44]. This study demonstrates that Genetic Semantic - Genetic Programming (GS-GP) performs better than Standard Genetic Programming (in training data and test data) and with these new operators, GS-GP can be applied in many new problems. Throughout the years Genetic Programming has been applied in many problems regarding financial markets. The most common

one is forecasting prices of stocks [45] [46] [47] [48]. The good results show that GP can be a powerful instrument for forecasting prices. In Portugal, Genetic Programming was applied in the stock market context, in order to forecast the PSI-20 Index [15].

Long Short-Term Memory has been around since 1997 [49] but it only became popular recently because of the evolutions on Machine Learning and of the availability of programming interfaces that can handle this algorithm. The main advantage of Long Short-Term Memory is its ability to learn context-specific temporal dependence. Each unit remembers information for either a long or a short period of time without explicitly using an activation function within the recurrent components. In the last few years, Long Short-Term Memory became one of the most used algorithms in financial markets. Stock markets have been the choice par excellence to be the example of performance in forecasting using LSTM [50] [51].

When it comes to the Portuguese market, the literature is not as extensive as it is in other markets. Most of the literature concerns forecasting the stock market and its volatility using statistical models [52] [53] Despite the reduced literature, there are already some studies that apply Machine Learning to forecasting stock prices using different algorithms such as Genetic Programming [15] or Neural Networks [54]. Concerning the bonds market, it is still a very unexplored field in Portugal, and I could not find any study where a Machine Learning approach for sovereign bond yield prediction was used.

3. FINANCIAL MARKETS AND MACHINE LEARING

3.1. FINANCIAL MARKETS

In a simple way, a bond is a contract that commits the issuer to make a predefined sequence of payments to investor until a specified terminal date. This fixed income instrument represents a loan made by an investor to a government or corporate. It can be characterized by different amounts, maturities, guarantees and coupons. Bond details include the maturity date (when the principal of the loan is due to be paid to the bond owner) and usually include the terms for variable or fixed interest payments made by the borrower, designed by coupons. The credit rating agencies (like Standard & Poor's, Moody's, and Fitch Ratings) can classify these bonds through the credit quality (risk) of the issuer and the guarantees of the bond. The market prices of bonds are based on their characteristics and they can be quoted by price or yield.

The bond yield can be defined as the return investors realize on a bond. The yield is a function of the bond's price and its coupon. An inverse relationship occurs between yields and bond prices. When interest rates rise and exceed the coupon rate of a bond, the price of the bond decreases. The opposite happens when interest rates falls.

The length of time to maturity defines the magnitude of impact in yield and price of the bond. The longer the maturity, the more volatile the fluctuations in price will be. For a basis point change in a bond yield, the magnitude of the respective change in the bond price will be greater for bonds with longer maturities.

New issues of bonds are sold in the primary market, whereby the issuer of the bonds receives the incomes through the selling intermediary – the investment banker. Investors who have acquired bonds could sell them any time before maturity through the secondary markets. In other words, the buying and selling of widely traded bonds takes place in the secondary markets. These markets are important because they provide liquidity to the products, which also determines their prices.

The bond markets do not work exactly like stock market, as there are some distinguishing differences. One distinctive characteristic is that bonds are low-risk assets. In some circumstances the risks can even be ignored. The Internet has provided accessibility to online secondary stock market, allowing trading to individual investors. However, the advent of online bond trading is much slower and not as widespread for various reasons.

3.1.1. Portuguese Government Bonds

Portuguese Government Bonds represent a total debt of €145.496 million, with twenty-two bonds issued [11]. In a total of fifteen fixed rate bonds with maturities between one year and thirty. When it comes to short-term, a total of five treasury bills exist with maturities lower than one year.

The placement of fixed income bonds in the primary market is ensured by a group of financial institutions to which IGCP has granted the status of Primary Dealers (OEVT) or Other Auction Participants (OMP).

The status of OEVT and OMP is granted on the basis of an assessment of the capability of the financial institutions to consistently place and trade Portuguese Government debt instruments in

international, European or domestic markets, ensuring the access to a regular investor base and contributing to the liquidity of these instruments in the secondary market.

In the secondary market, Portuguese Government Bonds are sold through nineteen primary dealers, which include some of the largest investment banks (Goldman Sachs International Bank, J.P. Morgan Securities) and largest banks in the world (Barclays Bank, Banco Santander). Primary Dealers (OEVT) are considered financial intermediaries for their capacity to place bonds and to ensure liquidity in the secondary market of these securities. Dealers quote bid and offer prices for these bonds in an Interdealer platform - MTS Portugal (reference market for public Portuguese debt market makers).

3.2. MACHINE LEARNING

There is no consensus for a formal and worldwide definition of Machine Learning. Nevertheless, it can be described as a computational field that strives to create computer programs that repeatedly improve with their own experiences on data. The main concern of this field is to identify patterns in the data and to use these patterns to make predictions about unseen data or about the future.

Machine Learning involves many other scientific fields, such as: mathematics, statistics, econometrics, probability, artificial intelligence, computational complexity theory, information theory, philosophy, neurobiology, and others.

One of most consensual definitions of Machine Learning was written by Tom Mitchel: “A computer program is said to learn from an experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.” [55].

Machine Learning can be described as the search for a “very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by the learner” [56]. The main result is the capability of the model to predict precisely when presented with new data. When a model is created, it is presented to the training data and it learns some rules based on that specific data. The main outcome of Machine Learning is the capability of the model to predict precisely, based on rules learned from previously seen data, when presented to new data. Machine Learning is simply an attempt to answer questions such as "How does learning performance vary with the number of training examples presented?" and "Which learning algorithms are most appropriate for various types of learning tasks?".

The use of Machine Learning for predicting time-series, specifically in financial markets, has been successful so far. Nowadays, researchers and companies are trying to develop intelligent algorithms that can capture the hidden patterns inherent to financial markets in order to predict their behaviour more efficiently.

In general, the most widespread approaches used by researchers can be divided in two:

- Statistical and econometrical like Linear Regression (LR), Autoregression Moving Average and ARIMA models. Nevertheless, these approaches involve many assumptions that are non-realistic when it comes to financial markets.

- Predictive models for forecasting market prices based on Machine Learning algorithms such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), Genetic Programming (GP) and Long Short-Term Memory (LSTM).

3.2.1. Genetic Programming

Genetic Programming (GP) uses the principles of the Darwinian evolution and evolutionary computation. This Machine Learning algorithm starts by building a possible solution arranged in a chromosome-like structure and applies data recombination operators to this structure in order to maintain important information. Genetic Programming is often used to learn and discover both the contents and structures of solutions.

One of the most consensual definitions of Genetic Programming was given by Koza [7]: "Genetic Programming is a domain-independent problem-solving approach in which computer programs are evolved to solve, or approximately solve, problems. Genetic Programming is based on the Darwinian principle of reproduction and survival of the fittest and analogues of naturally occurring genetic operations such as Crossover (sexual recombination) and Mutation".

In the next paragraph each phase of the process involved in Genetic Programming is explained. In order to run a Genetic Algorithm four steps are necessary:

1. **Representation space:** In order to create a hypothesis space where the algorithm can be initiated, an initial population of random trees (programs) is generated. A randomly generated tree can be of any size. In practice, Koza [41] defines restrictions to maximum size of the initially generated trees.

2. **Fitness Function:** The fitness function can be defined as the measure of performance. It helps to quantify how accurate a candidate solution is. The function is the instrument that defines in Genetic Programming which candidate solutions or regions of the search space are adequate.

3. **Parameters:** When a Genetic Program is developed it is necessary to control some parameters, as they can affect how the program evolves and, consequently, its performance. Those parameters are:

    - Population Size: The value for this parameter depends on the complexity of the problem. However, generally, GP performs better on large populations.

    - Number of Generations: This parameter indicates the maximum number of generations created by the algorithm.

    - Probabilities of performing the genetic operators: There are two genetic operators: Mutation and Crossover. Traditionally, Crossover is applied more often, usually 90% of probability. Mutation is applied with much less significant probability- around 5%.

    - Selection Method: In this parameter defines how which individual genomes are chosen from a population for future breeding. The most common selection method is tournament selection. When using this method, the tournament size must also be specified.

- Population Initialization: To generate the initial population it is possible to use three methods - full method, growth method or ramped half-half. This last method is commonly used because it provides a more diverse initial population.

4. **Termination Criterion:** The termination criterion controls when the program stops. There are three kinds of termination conditions have been usually employed for Genetic Programming:

    - An upper limit on the number of generations is reached;
    - An upper limit on the number of evaluations of the fitness function is reached;
    - The chance of achieving significant changes in the next generations is excessively low;

How Genetic Programming flows: [58]

1. Start with a randomly generated population, a set of possible candidate solutions to the problem.
2. Calculate the fitness measure ($f(x)$) of each individual in the population.
3. Repeat the following steps until the maximum number of generations is reached:
    a. Randomly select a pair of individuals from the current population. These individuals will be considered the parents for the Crossover operation. Selection is done "with replacement," meaning that the same individual can be selected more than once to become a parent.
    b. With probability CR (Crossover Rate), Crossover the pair at a randomly chosen point (chosen with uniform probability) to form two descendants. If no Crossover takes place, form two descendants that are exact copies of their respective parents.
4. According to a probability MR (Mutation rate), mutate the two descendants at each locus and place the resulting individuals in the new population.
5. Replace the current population with the newly generated population.
6. Go to step 2.

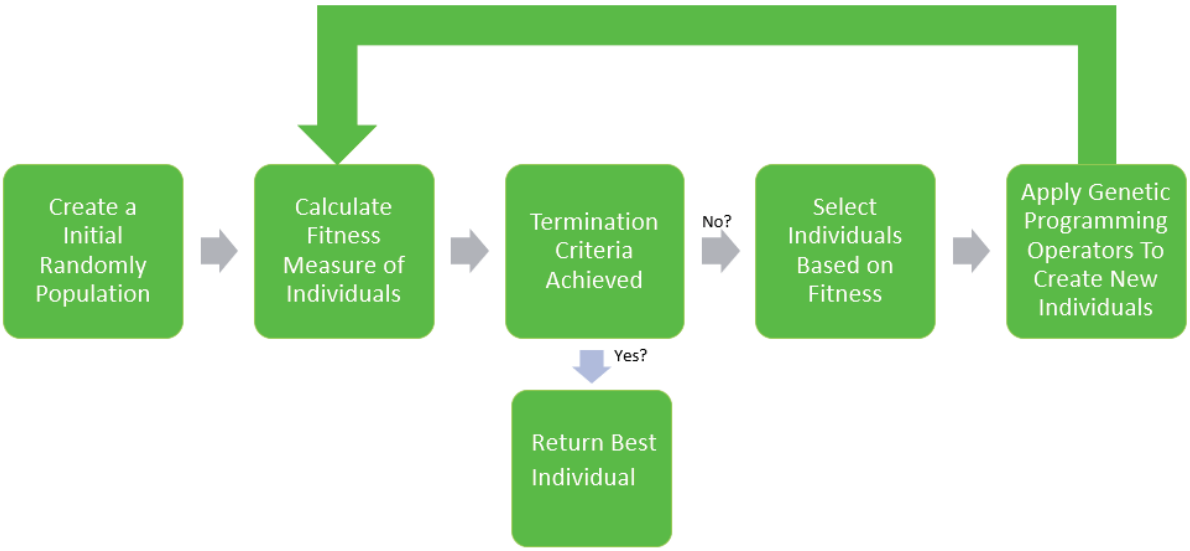

Figure 2- Representation of Genetic Programming Flow

3.2.1.1. Representation of Individuals

Instead of the traditional lines of code, in Genetic Programming the programs are typically represented by syntax trees. The leaves of the tree represent the variables and constants in the program and these are called terminals. At the same time, the arithmetic operations (+, * and max) are internal nodes called functions. The group of acceptable functions and terminals creates the primitive set of Genetic Programming. The primitive set is what defines the search space, where we can find all the individuals that can be generated by the combination of primitives in all possible ways.

Genetic Programming initiates with the selection of an initial population, randomly generated and defined by functions and terminals that are appropriate to the problem field. Therefore, when this algorithm starts, it must make a deduction about a rational set of functions and terminals for the problem. The idea behind this is to evolve programs that are hard to write and about which no one knows ahead of time exactly which functions and terminals will be required. The idea behind the creation of this initial population is to create a blind random search.

The Terminal can involve constants, input variables and/or functions with no arguments. Depending on the problem, the computer program may be boolean-valued, integer-valued, real-valued, complex-valued, vector-valued, symbolic-valued, or multiple-valued. The Function Set, represented by the internal nodes of the trees, consists of standard arithmetic operations, standard programming operations, standard mathematical functions, logical functions, or domain-specific functions. In simple mathematical problems the arithmetic functions can be the simple operators (+,-,*,/). All elements in the function and terminal sets are defined by several arguments, which are called Arity.

In order to apply Genetic Programming, the primitive set requires two proprieties. The possibility to produce solutions with the elements available - a property called sufficiency. Another important property is to ensure that the function set is well defined regarding all probable combinations of

expressions that can happen during the process of evolution – a property called closure. Unfortunately, guaranteeing these properties is not an easy job.

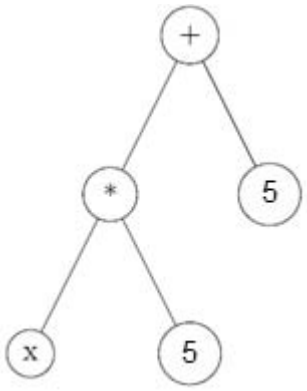

Figure 3- Representation of GP Tree: 5x+5

Primitive Set	Function Set	Arity	Terminal Set	Arity
1	*(+, -, *, /)*	(2,2,2,2)	$(x, y, 5)$	(0,0,0)
2	*(+, -, *, /, sin, cos)*	(2,2,2,2,1,1)	$(x, y, rand(\ \ ), 5)$	(0,0,0,0)
3	*(AND, OR , NOT)*	(2,2,1)	$(x, y, 5)$	(0,0)

Table 1- Three examples of possible primitive set

3.2.1.2. Fitness Function

The fitness function is defined as a measure of how accurate a candidate solution is. This function is the instrument that defines which candidate solutions or regions of the search space are adequate.

Fitness can be measured in numerous ways. For example, in terms of: the amount of error between the actual output and the desired output; the amount of time (fuel, money, etc.) required to bring a system to a desired target state; the accuracy of the program in recognizing patterns or classifying objects; the payoff that a game-playing program produces; the compliance of a structure with user-specified design criteria. Every fitness function depends on the type of problem.

Each error measure has its own strengths and limitations with respect to interpretation of the results. In problems like Regressions, the most common fitness functions are the ones based on error measures, it is the case of the Mean Absolute Error (MAE) and the Mean Squared Error (MSE). Alexei Botchkarev [56] shows that these two measures are some of the most important when the purpose is forecasting with Machine Learning. The formula (1) of these measures is given by:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (1)$$

Where:

- N is the total number of observations;
- y_i is the actual value of variable;
- $\hat{y}_i$ is the predicted value of variable;

3.2.1.3. Initialization the Population

The initial population in Genetic Programming, like in many other algorithms, is randomly generated. However, numerous different approaches exist to generate it. In this type of algorithm, the three simplest methods to generate an initial population are the full and grow methods and a widely used combination of the two known as ramped half-and-half.

In both the full and grow methods, the initial individuals are generated having a pre-specified maximum depth. The depth of a node is the number of edges that need to be traversed to reach the node starting from the tree's root node. The depth of a tree is the depth of its deepest leaf.

In the full method nodes are taken at random from the function set until the maximum tree depth is reached. In the full method nodes are taken at random from the function set until the maximum tree depth is reached. Although, this method generates trees where all the leaves are at the same depth, all initial trees will have an identical number of nodes or the same shape. This can only happen when all the functions in the primitive set have an equal arity.

The grow method, on the contrary, allows for the construction of trees of diversified sizes and shapes. In this method, nodes are selected from the whole primitive set until the maximum depth is achieved. When it is achieved, similarly to the full method, only terminals can be chosen.

The two methods referred above provide a significant varied array of sizes or shapes on their own. Koza [41] proposed a combination called ramped half-and-half. In this method, half of the initial population is generated using the full method and the other half is generated using the grow method. This is done by using a variety of depth limits to make sure that generated trees have a variety of sizes and shapes.

Even though these methods are easy to use and implement, they frequently make it hard to control the statistical distributions of properties such as the sizes and shapes of the generated trees.

3.2.1.4. Genetic Operators

Genetic Operators are used to create new individuals that will come together as the new population. The probabilistic methodology should be used to select the best genetic operator to generate new individuals. The probability of application of each operator is called operator rate. In Genetic Programming the most common operators are Crossover and Mutation.

Generally, Crossover is the most used operator because of its higher probability in contrast to Mutation, that has a lower probability of been applied. The Crossover Rate (CR) is typically above 90% and the Mutation rate (MR) is frequently much smaller, being typically close to 1%. When the sum of the Crossover and Mutation Rates is less than 100% a new operator called reproduction is used with a rate of 100%-(CR+MR).

Crossover

There are many different forms of crossover - homologous, Uniform, Context-Preserving Crossover or Size-Fair Crossover. The most common one is Subtree Crossover.

This operator works by selecting, through a selection algorithm, a common point in the parent programs and then exchange the corresponding subtrees, creating a new individual (child). To allow that the two parents have different forms, one-point crossover studies the two trees from the root nodes and selects the crossover point only from parts of the two trees located in the common region. This operator follows the following steps: [59]

1. Picks two parents using a selection algorithm;
2. Picks a random subtree from each parent and the root of each subtree as the crossover point;
3. Generates two new individuals (children) by exchanging the subtrees selected in their parents;

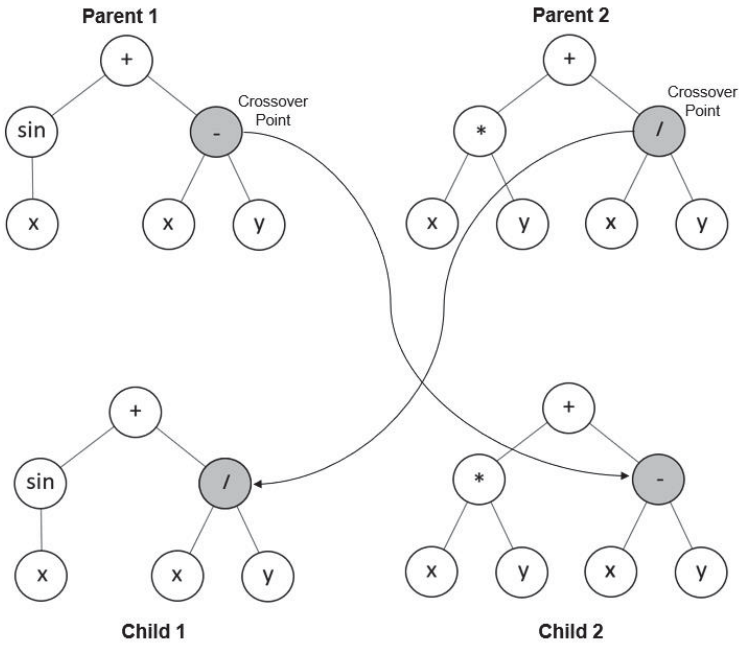

Figure 4-Representation of Crossover Operator

Mutation

Similarly to the crossover operator, there are many forms of mutation - size-fair subtree mutation, Hoist mutation, Shrink mutation, Permutation mutation, Mutating constants at random, Mutating constants systematically. The most frequently used methods for mutation are the Subtree Mutation and Node replacement mutation (also known as point mutation).

One of the first definitions of mutation was introduced by Koza [41], defining mutation as swapping a randomly selected subtree with another arbitrarily created subtree. However, in 1993 Kinnear [57]

introduced a constrain to this operator that prevents the children from being more than 15% deeper than the respective parent.

Basically, in point mutation a node in the tree is randomly selected and randomly changed. The new node needs to have the same number of arguments as the node it is replacing.

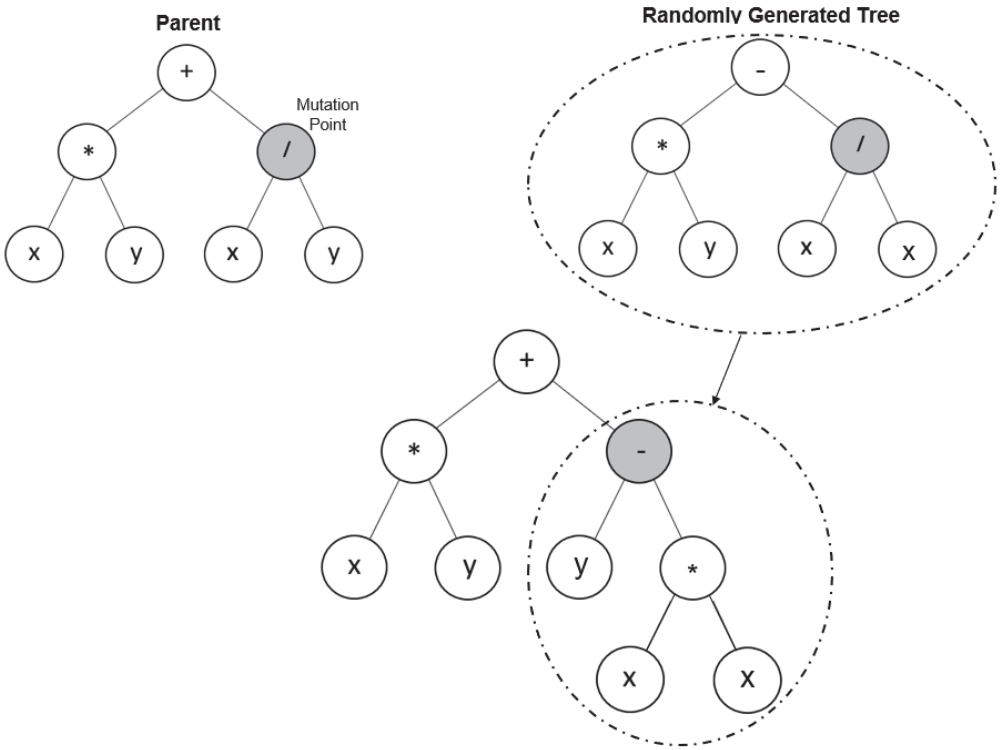

Figure 5- Representation of Mutation Operator

Reproduction

The Reproduction Operator is one of the simplest operators in Genetic Programming, consisting only in copying the designated individual to the next generation without any modification.

3.2.1.5. Selection

The Genetic Operators referred to in the previous chapter are applied to individuals that are probabilistically selected based on fitness. The most employed technique for selecting individuals is tournament selection [59].

In tournament selection, the individuals are chosen randomly from the population. After that, they are compared to each other regarding to their fitness value and only the best are chosen to be the parents for the next generation. When a crossover operator is used, two tournaments are necessary.

One of the problems with this selection method is the fact that a measure of how much better an individual is when compared to other is not considered. This means that a single extraordinarily good program cannot immediately swamp the next generation with its children, leading to a loss of diversity with many consequences. Contrarily, tournament selection increases small differences in fitness to favour the better program even if it is only slightly superior than the other individuals in a tournament. The decision regarding the tournament size parameter needs to be well thought. A

program with a large tournament size highly benefits the fittest individuals, while minor tournament size gives a better change for less fit individuals to be chosen as parents.

3.2.1.6. Geometric Semantic Genetic Programming

Introduced by Moraglio Krawiec and Johnson [42] Geometric Semantic operators are new and promising genetic operators for genetic programming. The concept of semantics in Genetic Programming frequently means a vector of output values obtained by an input data. The semantics of a solution refers to its performance. These operators have proprieties that allow them to induce an unimodal error surface for any supervised learning problem. This can be particularly good in problems where the goal is finding the match between a set of input data and known target values.

The Geometric Semantic Crossover (GSC) is formally defined as:

Definition of Geometric Semantic Crossover (GSC)

Given two parent functions $T_1, T_2: R_n \rightarrow R$, the geometric semantic crossover returns the real function $T_{XO} = (T_1 * T_R) + ((1 - T_R) * T_2)$, where T_R is a random real function whose output values are in the interval [1,0] [43]

Moraglio [42] officially proves that this new operator corresponds to geometric crossover on the semantic space, and thus induces an unimodal fitness landscape. In order for T_R to produce values in the range of [0, 1] the sigmoid function is commonly used:

$$T_R = \frac{1}{1 + e^{(-T_{Rand})}} \qquad (2)$$

Where T_{Rand} is a generated random tree with no constrains.

The other Geometric Semantic Operator is Geometric Semantic Mutation (GSM), which can be defined as:

Definition of Geometric Semantic Mutation (GSM)

Given a parent function $T: R_n \rightarrow R$,the geometric semantic mutation with mutation steps (ms) returns the real function $T_M = T + ms * (T_{R1} - T_{R2})$, where T_{R1} and T_{R2} are random real functions [43].

When this operator was introduced the researchers demonstrated that this operator links to ball mutation on the semantic space and induces a unimodal fitness landscape. T_{R1} and T_{R2}, random generated trees, are being limited to consider values on the interval between zero and one, using the exact same method described for T_R (Geometric Semantic Crossover). Due to the fact that the parameter *ms* is centered in zero, there is a slight perturbation in the individual, corresponding to the difference between the two random trees. However, it is possible to adjust the magnitude of the perturbation produced by this operator in the parameter.

The use of these new genetic operators allows for the modification of the syntactic space of individuals, having the same impact on their semantics. One of the principles that guides supervised learning problems is the fact that the expected output value is known, and that the fitness consists on the distance in the semantic space between any individual and the target point. The Geometric Semantic Operators have a characteristic of inducing a unimodal fitness landscape (error surface), like regressions and classifications problems.

Another important characteristic of these operators is that the geometric properties rests independent from the data on which the individuals are evaluated. This means that geometric semantic crossover creates a progeny that lies between the parents also in the semantic space induced by the test data. This is important because it makes it possible to support control and limit overfitting, allowing for a generalization in the test set (out-of-sample).

However, when Moraglio introduced these operators, they had a limitation. Their use continuously produces progenies that are much bigger than their parents. Consequently, it was producing an exponential growth of the population. In some generations the size of individuals becomes huge and useless to real life problems.

Nevertheless, this problem was solved by Vanneschi and Castelli [43] with a new approach to these operators. They introduced an adjustment on the syntax of genetic individuals that has a precise correspondence on their semantics. This change allows for the use of the operators in real life problems in a time efficient manner.

3.2.2. Long Short-Term Memory

Long Short-Term Memory (LSTM) is one of most powerful Artificial Neural Networks (ANN). Due to its capability of learning long-term dependencies, this algorithm is selected to recognize patterns in sequences of data, such as time series or even text. The main difference between LSTM and other neural networks is the time and sequence rotting, as this algorithm takes in consideration a temporal dimension. [60]

Recurrent Neural Networks (RNN) is a chain of repeating modules of neural networks. In LSTM the traditional hidden layer of recurrent neural networks is substituted by a memory block. This block contains one or more memory cells and a pair of adaptive, multiplicative gates units. In this block the cells can share the same gates, reducing the number of adaptive parameters. Each memory cell has a unit designated as “Constant Error Carousel” (CEC) [61]. This unit solves the problem of nonappearance of new input or error signals. When this happens, the CEC’s local error backflow remains constant, neither growing nor rotting. The CEC is protected from both forward flowing activation and backward flowing error by the input and output gates, respectively. When gates are closed (activation around zero), irrelevant inputs and noise do not enter the cell and the cell state does not perturb the rest of the network.

This algorithm helps maintaining the error, that can be backpropagated through time and layers. This can be done through preserving a constant error, allowing the recurrent networks to continue to learn over numerous time steps, thus opening a channel to connect causes and effects remotely. Since algorithms are frequently confronted by environments where reward signals are sparse and delayed, it became a large problem in Artificial Intelligence and Machine Learning. [60]

LSTMs contain external information from the usual flow of the recurrent network in is gated cell. The information on this cell can be stored. The decision about what to store is taken by the cells themselves, which also decide when reads, writes and erasures are allowed via gateways that open and close. These gateways are like neural network nodes, deciding if the information is blocked or whether it passes, based on strength and importance which they filter with their own sets of weights. Those weights, like the weights that modulate input and hidden states, are adjusted via the recurrent networks learning process. That is, the cells learn when to allow data to enter, leave or be deleted through the iterative process of making deductions, backpropagating error, and adjusting weights via gradient descent. [60]

Long Short-Term Memory Step-by-Step: [62]

1. To initiate the algorithm, it is necessary to decide which information is going to be passed to the cell state. The decision is taken by the sigmoid layer designated by "Forget Gate Layer", it looks at h_{t-1} and x_t (as shown in figure 6 below). The outputs are numbers between 0 and 1 to each cell state C_{t-1}. Here, the numbers close to one (or even one) mean that all corresponding information must be kept. Numbers close to zero (or even zero) represent unnecessary information that must be thrown out.

2. Regarding new information, it is necessary to understand which one are going to be store in cell state. This process has two stages:

    a. First, a sigmoid layer called the "input gate layer" chooses which values going to be update.

    b. Second phase is the creation of a vector, by a Tanh Layer, with new candidates values , $\tilde{C}_t$that should be added to the state.

3. After the two steps above is combining the information that results from those and create an update to the state.

4. With the step above the old cell state its necessary updated the cell state into a new cell state C_t (from previous one C_{t-1}).

5. The output of the flow is based on information of cell state, just need to be filtered:

    a. Its necessary run a sigmoid layer which will decide what information going to be the output.

    b. After that, put the cell state through Tanh Layer (to press the values to be in the interval of minus one and one) and multiply it by the output of the sigmoid gate. Those are the parts that going to be the output.

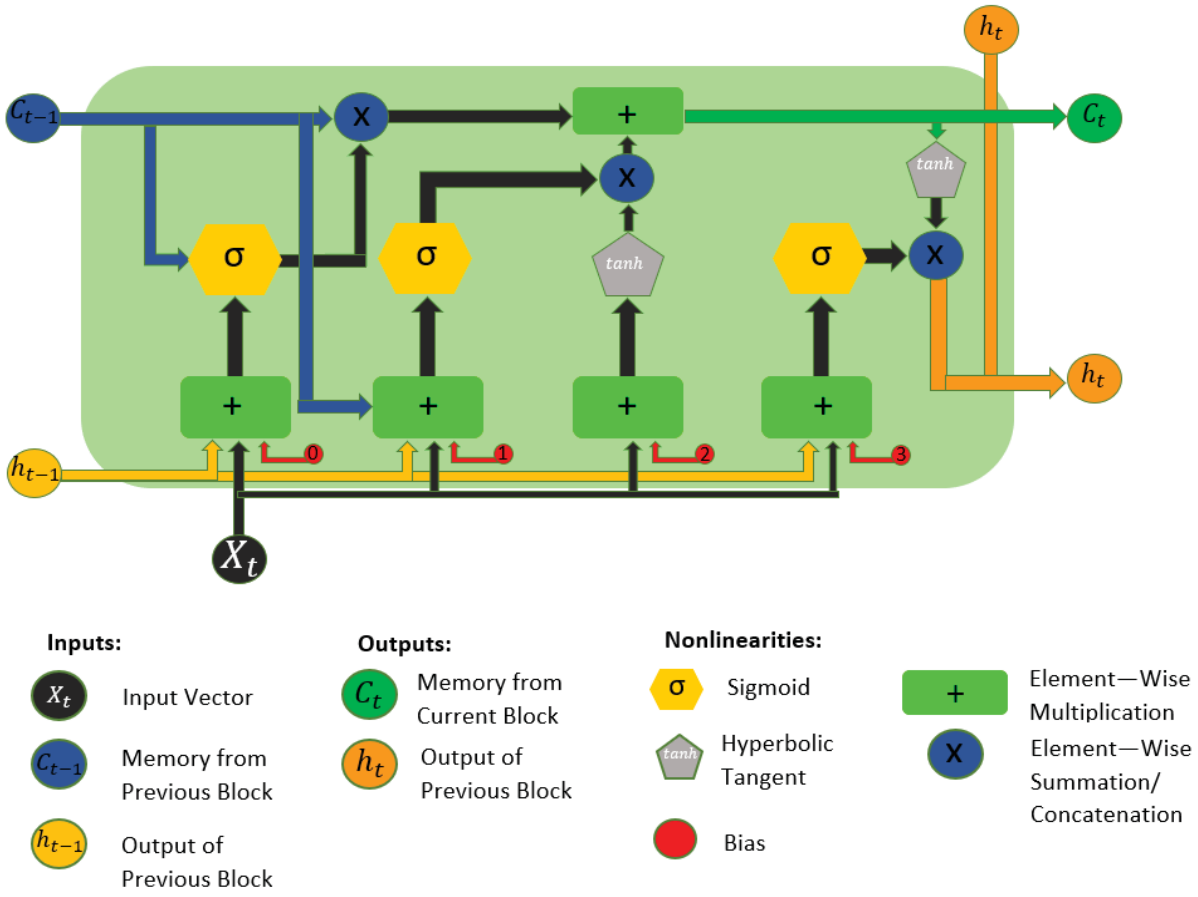

Figure 6-Representation of Long Short-Term Memory Algorithms Flow

4. EXPERIMENTAL STUDY

4.1. METHODOLOGY

4.1.1. Data Description

The 10-year Portugal Government Bonds is one of the fixed maturity securities for which the bond market calculates a daily yield. This tenor is considered by all participants in financial markets as benchmark to sovereign bonds. In order to have an image of the future yield curve for Portugal, this tenor will be used. Despite not being available every day, it is possible to calculate a theoretic yield of a bond with a constant maturity in the 10-year benchmark by interpolating the daily term structure of interest rates.

When it comes to the dataset used, each record (instance) represents data from a single week. The objective is to predict the bond yield at the next day (t + 1) considering a set of variables (reported in table 2) related to the bond yield.

Variable	Description
YLD	The close yield of Portuguese Bond 10Y based on the defined date.
YLD_CHG_NET_1D_NO_BP	One day change in closed yield of Portuguese Bond 10Y expressed in basis points on the defined date.
YLD_HIGH_D	Highest yield on the defined day the Portuguese Bond 10Y reached during the trading day.
YLD_LOW_D	Lowest yield on the defined date the Portuguese Bond 10Y reached during thetrading day.
SPRD_BUND	The close spread of Portuguese Bond 10Y and Germany Bund 10Y based on the defined date.
YLD_BUND	The close yield of Germany Bond 10Y based on the defined date.
FT_BUND	The price of Generic Future Euro Bund Contract 10Y based on the defined date.
SWP_10Y	The close interest rate of a Vanilla Interest Rate Swap 10Y (EuroSwaps 10Y) based on the defined date.
SX5E	The close price of Euro Stoxx 50 Index based on the defined date.
VIX	The close price of Chicago Board Options Exchange Volatility Index based on the defined date.
CPI_GER	The close price of Germany Consumer Price Index based on the defined date. If the doesnt exist any value to that date then it is the price of Germany Consumer Price Index reached the last time.
GDP_GWTH_PT	The close value of Portugal Gross Domestic Product Growth (Quarter on Quarter) based on the defined date. If the doesnt exist any value to that date then it is the value of Portugal Gross Domestic Product Growth (Quarter on Quarter) reached the last time.

Table 2- Description of Variables on dataset

In this analysis the period between the 1st of January of 2005 and the 31st of March of 2020 was considered, resulting in a total of 5569 daily observations of historical daily data. This interval of time allows for the consideration of data from important periods, with high volatility, like the Financial Crisis of 2008, European Sovereign Debt Crisis and more recently the effect of the Covid-19 virus on Portuguese economy.

The raw data used in the present study was extracted from a Bloomberg Terminal.

Unfortunately, in some periods from 2005 and 2006 Bloomberg does not have data regarding the yield of 10Y Portugal benchmark. Due to this lack of information, linear interpolation was applied to

fill in those periods (on figures 18-19 one can see the evolution of this variable with and without interpolation).

4.1.2. Data Transformation

Understanding the dataset is an essential step to initiate the study. Studying the structure of variables and their behaviour through exploratory analysis is relevant in order to look for interesting and uncommon patterns in data.

In order to begin this exploratory analysis some descriptive statistics were computed (see table 11 in appendix 9.2) for each variable (mean, standard deviation, minimum, maximum and quarters (1,2,3)). Descriptive statistics allow for the presentation of the raw data in a meaningful way, which results in a simpler interpretation of the data.

After analysing the statistics resultant from the table mentioned above, one can see that there are most likely some outliers in many variables. For a start, Spread_Bund has an average of 257,52, whilst the standard deviation is 265,06. This may indicate the presence of dissonant values, that are confirmed by an analysis of the maximum value against the third quarter – 1481,5 versus 345,0. The same happens in variables Highest_Yield_of_Day, Lowest_Yield_of_Day, Bund_Yield, among others. Further analysis were made resorting to visual tools.

Visualization plays an important role in time series analysis and forecasting. Since the present study concerns a time series, it was important to visualize the distribution and evolution of variables (independent and dependent) along the period. In appendix 9.2 one can find the charts with this analysis for each variable. (see figure 20-31)

In the same analysis, the distribution of observations, with and without outliers was computed and designed. The objective of finding outliers is to investigate data objects which are different from or inconsistent with the majority of data. However, in bond market data, outliers are highly intermixed with regular data and it is difficult to judge whether an object is an outlier or not.

One of the most common problems in time series analysis is stationarity. Without an uniform definition that can be unambiguously applied to finite time series, the question of stationarity is victim to all vagaries of statistics, almost to the point of subjectivity. In the most intuitive sense, stationarity means that the statistical properties of the process that generates a time series do not change over time. It does not mean that the series does not change over time, just that the way it changes does not itself change over time. The algebraic equivalent is, thus, a linear function, perhaps, and not a constant one; the value of a linear function changes as x grows, but the way it changes remains constant — it has a constant slope; one value that captures that rate of change. To determine the stationarity of the dependent variable chosen for this study, a chart with the evolution of yield and net change basis point was analysed. One of the most immediate conclusions is the non-stationarity of the yield. The evolution of this variable is volatile through the time.

Since it is proven that the yield is non-stationary, the variable Net Change Basis Point can work as a substitute for it, as it is calculated as the amount that the yield has changed from one day to another in basis points. This variable evolves in a consistent way, around zero. Nevertheless, some statistical tests were applied to it in order to confirm this conclusion.

The Augmented Dickey-Fuller Test was the first test to be performed. This is one of the most common forms of Unit Root tests, where the alternative hypothesis is the stationary of the variable. After applying the test to the yield, there is no statistical evidence (p-value=0.7989, see figure 32 in appendix 9.2) that we can reject the null hypothesis (the variable follows a stationarity evolution). However, when this test is applied to the Net Change Basis Point, the result is different, prevailing statistical evidences that the null hypothesis can be rejected and, as consequence, that the variable follows a stationarity evolution.

The autocovariance function (ACF) is defined as the sequence of covariances of a stationary process. In order to continue the stationarity analysis of the yield" and Net Change Basis Point, this test was applied, resulting in the same conclusions as the previous tests (see figure 33-34 in appendix 9.2).

Seasonality can be described as a general systematic linear or nonlinear component that changes over time and does repeat. This characteristic is important when it comes to analysing time series. One simple approach to test seasonality is to fit a model with allows for seasonality if it is present and if the chosen model has a seasonal component, then the data is seasonal. When this approach was applied to the yield statistical evidence was found (p-value= 5.356377*10^-6, see figure 35 on appendix 9.2) that the additional seasonal component is significant. However, when applied to Net Change Basis Point the conclusion it's the opposite, as there are no statistical evidences that a seasonal component is significant (p-value≈1, see figure 36 on appendix 9.2).

Correlation analysis is another important tool for multivariate analysis in data pre-processing and exploration. Correlation examines relationships between variables. It is employed in this type of analysis before any statistical modelling or data analysis. In figure 40 on appendix 9.2 the correlation matrix for the variables of this study is presented.

Regarding the previous analysis applying adjustments to the dataset structure was necessary in order to present it to algorithms like Genetic Programming and Long Short-Term Memory. The result of those adjustments was a new dataset with a new dependent variable and a reduction on the number of independent variables. In this new dataset the dependent variable is Net Change Basis, as it has a stationary evolution and does not have seasonality. The change in the dependent variable does not change the main goal of the study. The reduction in the number of independent variables was due to the correlation between them. The previous dataset had several variables that were highly correlated, causing complications for the forecasting task. In the table below a description of the new variables is presented.

Variable	Description
YLD_CHG_NET_1D_NO_BP	One day change in closed yield of Portuguese Bond 10Y expressed in basis points on the defined date.
SPRD_BUND	The close spread of Portuguese Bond 10Y and Germany Bund 10Y based on the defined date.
YLD_BUND	The close yield of Germany Bond 10Y based on the defined date.
SX5E	The close price of Euro Stoxx 50 Index based on the defined date.
VIX	The close price of Chicago Board Options Exchange Volatility Index based on the defined date.
CPI_GER	The close price of Germany Consumer Price Index based on the defined date. If the doesnt exist any value to that date then it is the price of Germany Consumer Price Index reached the last time.
GDP_GWTH_PT	The close value of Portugal Gross Domestic Product Growth (Quarter on Quarter) based on the defined date. If the doesnt exist any value to that date then it is the value of Portugal Gross Domestic Product Growth (Quarter on Quarter) reached the last time.

Table 3- Description of Variable on transformed dataset

Following data transformation work, outlier analysis is essential. Hawkins [63] formally defined the notion of an outlier as follows: “An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.”. Nevertheless, the balance between deleting outliers and loosing diversity necessary for Machine Learning algorithms is delicate. In order not to extinguish this diversity, only the extreme values of each variable were considered. It is important to understand that extreme values are a very specialized type of outliers, in other words, all extreme values are outliers, but the reverse may not be true. Through a boxplot analysis extreme values in “Net Change Basis Point” (all values >1.5 and <-1.5) and “GDP Growth Portugal” (all values <-2) were found and then eliminated.

In order to have values for all observations in the dataset interpolation was used for replacing the extreme values eliminated in the previous analysis.

Normalization is a pre-processing technique often applied in data preparation for Machine Learning. The purpose of normalization is to change the numeric values in the dataset to a common scale, without distorting differences in the ranges of values. In Machine Learning, not every dataset requires normalization. It is required only when variables have different ranges. Choi and Moon [64], demonstrate that normalization methods in Genetic Programming decreased the problem difficulties and led to remarkable improvements in performance. Considering the advantages of normalization, this technique was applied to the present dataset.

The result of the previous transformations creates the final dataset as it will be presented to Genetic Programming and Long Short-Term Memory.

4.1.3. Fitness Measure of Models

The root means square error (RMSE) has been used as a standard statistical metric to measure model performance in many studies for different fields. The mean absolute error (MAE) is another useful measure commonly used in model evaluation. Both have been used to assess model performance for many years. However, there is no consensus on the most appropriate metric for model errors [65]. Nevertheless, Cort J. Willmott and Kenji Matsuura [66] prove the existence of statistical evidence that the MAE is the most natural measure of average error magnitude, and that (unlike RMSE) it is an unambiguous measure of average error. Evaluations and inter-comparisons of average model performance error should be based on the MAE. Following this conclusion and the purpose of this study the Mean Absolute Error was chosen as the fitness measure to evaluate the model outcome from Genetic Programming and Long Short-Term Memory.

The Mean Absolute Error (MAE) is calculated as an average of absolute differences between the target values and the predictions. The MAE is a linear score, which means that all the individual differences are weighted equally in the average. However, the same is not true for the RMSE. One important characteristic of this measure is that it penalizes huge errors, thus it is not that sensitive to outliers as the Mean Square Error is. Mathematically, it is calculated using this formula (3):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i) \qquad (3)$$

Where:

- N is the total number of observations;
- y_i is the actual value of variable;
- $\hat{y}_i$ is the predicted value of variable;

Another important thing about the MAE is its gradients with respect to the predictions. The gradient is a step function and it takes -1 when $\hat{y}_i$ is smaller than the target and +1 when it is larger. This is not defined when the prediction is exact, because when $\hat{y}_i$ is equal to Y, we cannot evaluate gradient. It is not defined [67].

4.1.4. Software Methodology

For the initial analysis of the dataset, data transformation and design, the charts presented were made using R Studio Software.

Concerning the Genetic Programming approach, more specifically Geometric Semantic Genetic Programming (GSGP) the implementation freely accessible at http://gsgp.sourceforge.net created and documented by Castelli and Vannesachi [18] was used. This is an open source C++ library and it provides a robust and efficient implementation of geometric semantic genetic operators for Genetic Programming. It implements the standard Genetic Programming algorithm, but with genetic operators entrenched the concept of the semantic. It is versatile and easily adaptable. Its implementation is straightforward, depending on set of configuration parameters. This library was applied to the dataset through the Cygwin software.

Regarding Long Short-Term Memory, a script was written in Phyton language using Spyder through Anaconda Software, with an environment created for this specific purpose

4.2. Genetic Programming Experimental Settings

Regarding Genetic Programming four models were created using different settings. The purpose of this models is to choose the best model to compare with further LSTM models. Each model was run for forty iterations and the average of generations was computed individually. The function set used for Genetic Programming was predefined in the library used, containing only arithmetic operators. The terminal set consisted of six variables, summarized in table 4. The type of mutation used for each model was chosen at random for each mutation event, as in Vanneschi et al. (2014).

“Model 1” started with a basis population of two hundred individuals, that evolved throughout one thousand generations. For initializing the population, the Ramped Half-and-Half method was used, with a maximum initial tree depth equal to six. When it comes to selection, the tournament method was used, with a tournament size of ten. In this model a crossover probability of 70% and mutation probability of 30% were considered.

The main changes when it comes to “Model 2” were regarding the population size, the number of maximum generations and the maximum initial depth. In this model, a population size of five hundred individuals was set. These individuals evolved for one thousand and five hundred generations. The initialization of the population was achieved using the Ramped Half-and-Half method with a maximum initial depth equal to eight. Once again, the selection was performed using

the tournament method, with a tournament size of ten. In this model a crossover probability of 70% and mutation probability of 30% were considered.

The construction of "Model 3" was a mix between the two previous models. In this model a population size of two hundred individuals was considered over two thousand generations. The Ramped Half-and-Half method was still the method chosen for initialization of the population, with a maximum initial depth equal to eight. When it comes to selection, the tournament method was used, with a tournament size of ten. In this model, a crossover probability of 70% and mutation probability of 30% were considered.

"Model 4" is the most different model compared with other models. For this model, a population size of five hundred individuals was set, evolving for two thousand generations. For initializing the population, the Ramped Half-and-Half method was applied with a maximum initial depth equal to six. For selection, the tournament method was chosen, with a tournament size of ten. In this model a

Parameters	Model 1	Model 2	Model 3	Model 4
Terminal Set	$X_1, \ldots, X_6$	$X_1, \ldots, X_6$	$X_1, \ldots, X_6$	$X_1, \ldots, X_6$
Function Set	+,-,*,/	+,-,*,/	+,-,*,/	+,-,*,/
Fitness Function	MAE	MAE	MAE	MAE
Population	200	500	200	500
Population Generations	1000	1500	2000	2000
Probability Crossover	70%	70%	70%	70%
Probability Mutation	30%	30%	30%	30%
Tournament Size	10	10	10	15
Maximum Depth Creation	6	8	8	10
Mutation Step	Randomly	Randomly	Randomly	Randomly
Elitism	Best Individual	Best Individual	Best Individual	Best Individual

Table 4- Genetic Programming settings to each Model

crossover probability of 70% and mutation probability of 30% were considered.

The experimental results were evaluated by plotting the average error of the training and test sets. For each of the forty iterations, the best individual of each generation is stored alongside with the average fitness value per generation, which was then plotted.

4.3. Long Short-Term Memory Experimental Settings

Concerning Long-Short Term Memory a script was written for this specific study. Similarly, to Genetic Programming five models were defined, each one with specific parameters and characteristics. The models were run for forty iterations and the average of epochs was computed individually. On one of the first attempts on this algorithm it was possible to notice how more than three LSTM or GRU layers and more than five hundred epochs created an enormous overfitting of the model to the data. After some attempts, it was possible to conclude that for the purpose and for the data sample of this study one or two layers and no more than twenty-five epochs were enough.

The first model was the simplest model possible. This model had one LSTM layer and only fifteen epochs. After analysing the results of the forty iterations, it was possible to conclude that there was

some space to increase the complexity of the model and to try to improve the results. The second model was created as result of this analysis. In this model the number of layers was increased to two LSTM layers, maintaining the number of epochs.

Later, the type of layer was changed to a GRU layer. Similarly, to the LSTM layer, the GRU has gating units that modulate the flow of information inside the layer, without having separate memory cells. In the LSTM layer, the amount of the memory content that is seen, or used by other units in the network is controlled by the output gate. On the other hand, the GRU exposes its full content without any control.

The process followed was the same as in LSTM. Initially one model was created with only one GRU layer and fifteen epochs. After comparing the results of the model that used LSTM and the model that used GRU, clear evidences were found that the results improve when using the GRU layer, especially in the validation dataset. In order to try to improve the model, two more models were developed. The first one had two GRU layers and fifteen epochs. The second one had two GRU layers, but the number of epochs was increased to twenty-five.

Parameters	Model 1	Model 2	Model 3	Model 4	Model 5
Type of Layer	*LSTM*	*LSTM*	*GRU*	*GRU*	*GRU*
Number of Layers	1	2	1	2	2
Number of Epochs	15	15	15	15	25
Fitness Function	MAE	MAE	MAE	MAE	MAE

Table 5- Long Short-Term settings to each Model

The experimental results are evaluated by plotting the average error of the training and validation datasets. For each epoch the average fitness value is stored, over forty iterations.

5. RESULTS AND DISCUSSION

The results presented in this chapter were obtained using the methodology described and settings presented in previous chapters.

Concerning Genetic Programming, forty iterations have been performed on each of the four models. The results obtained are presented by plotting the average error on the training and test sets. In each generation, the best individual of the population (i.e., the one that has the smallest training error) was chosen and the value of its error on the training and test sets was stored. The reported plots contain the average of all the values collected at the end of each generation. This measure was preferred due to its higher robustness to outliers. In appendix 9.3 (figures 41-54) one can see the mean distribution of each model with and without outliers.

Figures 7 and 8 (also in appendix 9.3.1 and 9.3.2) present, for the dataset taken into account, the training and test errors (MAE) through generations for the all the models considered.

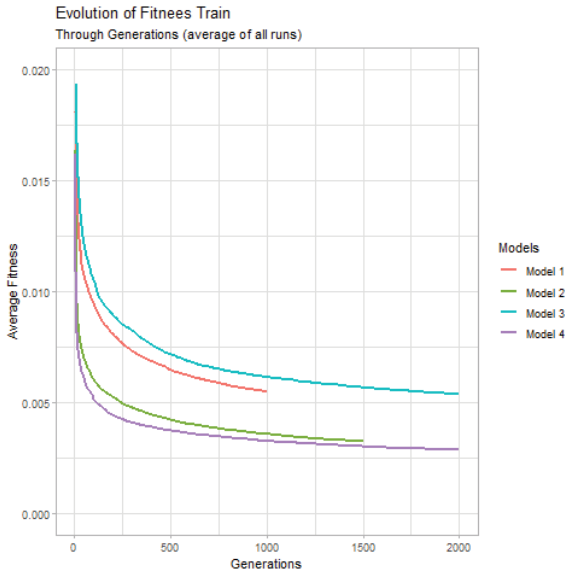

Figure 7-Evolution of Fitness Train through Generations (average of all runs) of Genetic Programming Models

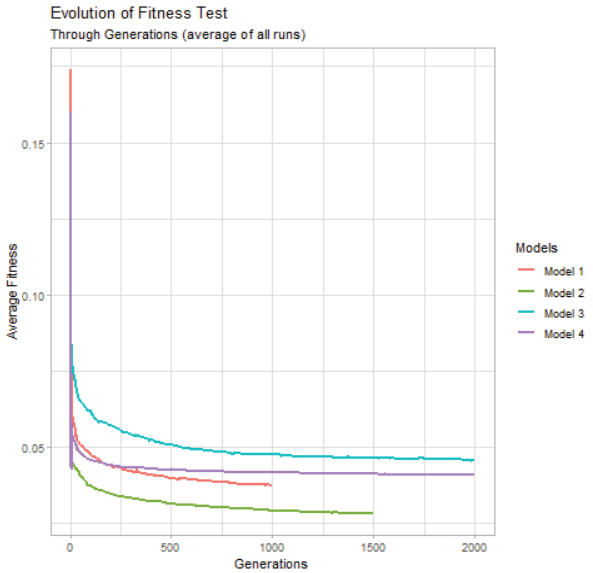

Figure 8-Evolution of Fitness Test through Generations (average of all runs) of Genetic Programming Models

There is no evidence that one model is clearly better than the others. However, more analyses were applied in order to confirm this preliminary conclusion.

Continuing with the visual analysis, in figures 9 and 10 (also in appendix 9.3.1 and 9.3.2) the average of fitness errors on training and test was plotted for each model over forty generations. In the charts below, one can find some evidence that model 2 presents better results when compared to the other three models.

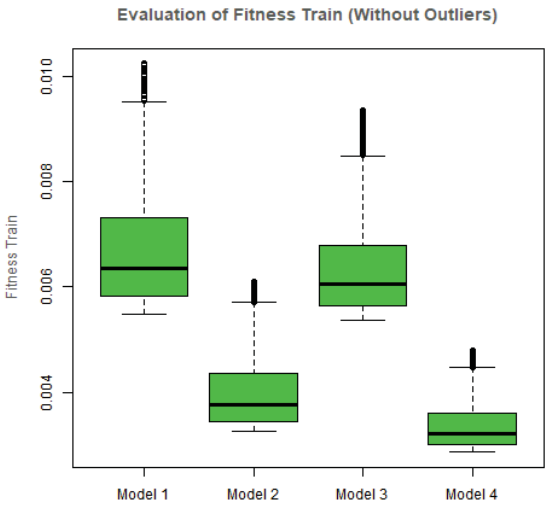

Figure 7- Boxplot of Fitness Train of all models without outliers in Genetic Programming

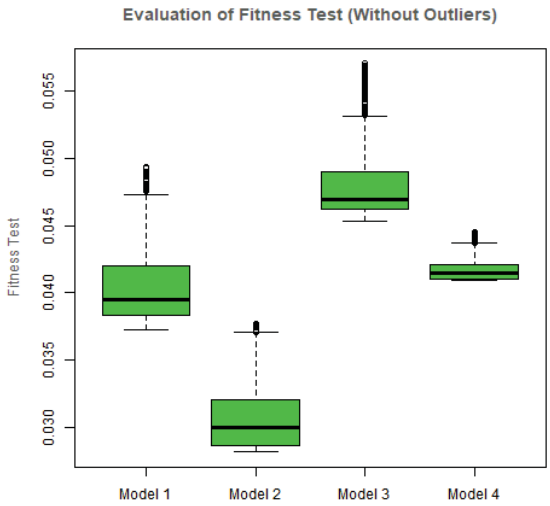

Figure 10-Boxplot of Fitness Test of all models without outliers in Genetic Programming

In order to analyse the statistical significance of the results for each model, the Shapiro Wilk test was firstly used, with α=0.05, to test if the data is normally distributed. This test can be defined as:

"The Shapiro Wilk test is based on a correlation of sample "order statistics" with those of a normal distribution. It can be interpreted as a ratio of two estimates of the variance of the sample, with the estimate in the numerator holding only if the sample is drawn from a normal distribution since coefficients ai are calculated by linear regression to the expected values of standard normal order statistics." [68]

$$W = \frac{(\sum_{i=1}^{N} \propto_i y_i)^2}{\sum_{i=1}^{N} (y_i - m_i)^2} \quad (4)$$

Where

$$m_i = \frac{1}{N} \sum_{i=1}^{N} y_{(i)}$$

Since the p-values for all models after performing the Shapiro Wilk test allow for the rejection of the null hypothesis, it is possible to conclude that there is no statistical evidence of the normal distribution of data.

In order to understand the distribution of fitness on the train and test sets for each pair of models, the Wilcoxon Rank-Sum was applied. This test assumes that the data comes from two matched, or dependent, populations. The data is also assumed to be continuous. Because it is a non-parametric test, it does not require a known probability distribution of the variable being tested. The Wilcoxon Signed Rank test assumes that there is information about the magnitudes and signs of the differences between paired observations. The Wilcoxon Rank-Sum test is used to test if two populations are likely to come from the same distribution. In the present study, this test was applied with α=0.05, considering as null hypothesis the possibility of the samples having equal averages. After applying the test, there is no statistical evidence that the fitness of any two models follows the same distribution.

Having the results of the analyses mentioned before, it is possible to conclude that there is no statistical evidence that any model is better than the others. In the context of the problem, this is not an issue, as the models considered are using similar parameters.

Regarding Long-Short Term Memory, as mentioned before, forty iterations have been performed for each of the five chosen models. In order to be possible to compare the two algorithms chosen for this research, the results obtained for LSTM were presented by plotting the average error on the training and validation sets. The reported plots show the average of all the values collected at the end of each epoch. In annex 9.4 one can see the mean distribution of each model with and without outliers.

In figure 11, the evolution of the average error is shown (collected at the end of each epoch) on the training and validation sets. In a first analysis, the results regarding training fitness for each model are very similar, leaving no space for one to decide about which model performs better. When it comes to the validation set, the results are more heterogeneous between models. As one can see from figure 12, model 3 seems to perform better on the validation set than the remaining models. However, some statistical tests were performed in order to verify this conclusion.

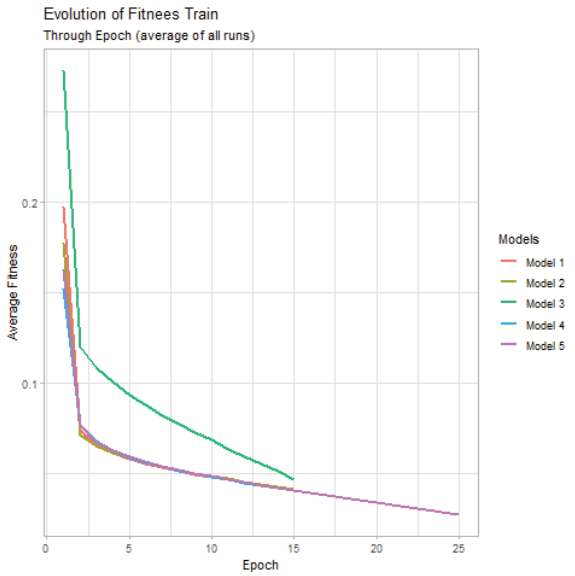

Figure 8-Evolution of Fitness Train through Epochs (average of all runs) of Long Short-Term Memory Models

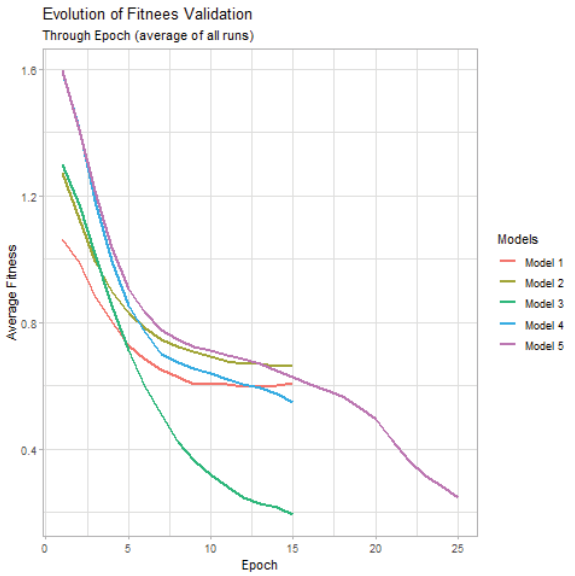

Figure 9- Evolution of Fitness Validation through Epochs (average of all runs) of Long Short-Term Memory Models

When faced with the charts below (figure 13 and 14), representing the distribution of the error (fitness) for each model, one can take similar conclusions to the ones mentioned before. In these charts, it is possible to observe some evidence that model 3 can get better results when compared to other models.

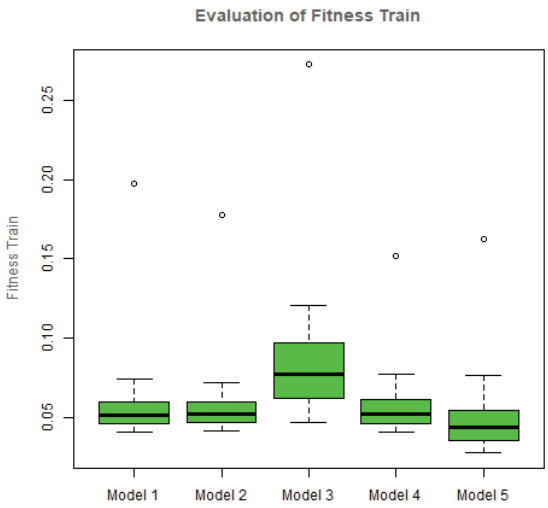

Figure 11- Boxplot to Evaluation of Fitness Train through Epochs (average of all runs) of Long Short-Term Memory Models

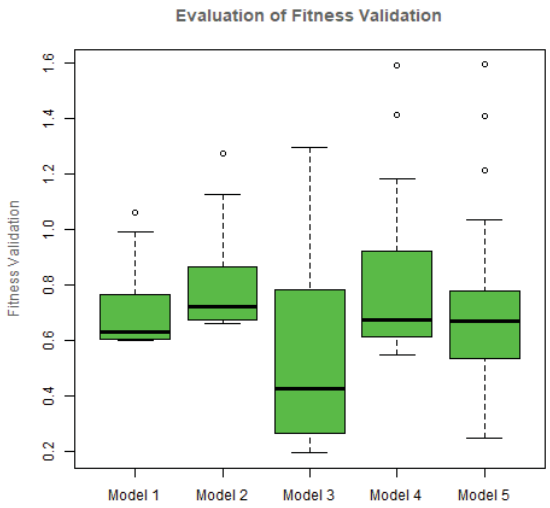

Figure 10- Boxplot to Evaluation of Fitness Validation through Epochs (average of all runs) of Long Short-Term Memory Models

In order to analyse the statistical significance of the results for each model, the same procedure applied to Genetic Programming was applied to the Long-Short Term Memory models.

Firstly, the Shapiro Wilk Test was performed. As seen on table 6 below, considering an $\propto$=0.01, the null hypothesis of the errors following a normal distribution cannot be rejected on the test set for models 3 and 5. This means that in these models, one cannot find statistical evidence that errors on the test set do not follow a normal distribution.

Shapiro Wilk Test		
	Train	Test
Model 1	< 0,001	0,001
Model 2	< 0,001	0,002
Model 3	< 0,001	0,036
Model 4	< 0,001	0,003
Model 5	< 0,001	0,026

Table 6- Shapiro Wilk Test results to Long Short-Term Memory Models

The Wilcoxon Rank-Sum test was applied too. The results of this test can be observed in table 11 on annex 8.3.3. Since on the Shapiro Wilk Test the null hypothesis was not rejected for models 3 and 5, the results of the Wilcoxon Rank-Sum test were used to compare the distributions of errors for these models. It is then possible to verify that, even though these models present a similar distribution of

errors on the training set, when it comes to the validation errors there is statistical evidence that they follow different distributions.

5.1. Comparison between Algorithms

After comparing the models defined for each algorithm, it was also important to compare both algorithms in order to understand and evaluate the results obtained by each approach and understand which one is more competitive. Considering the results obtained from the tests performed and described in the previous chapter, model 2 of Genetic Programming and model 3 of Long-Short Term Memory were chosen to be compared, since they were the models that showed better results for each algorithm.

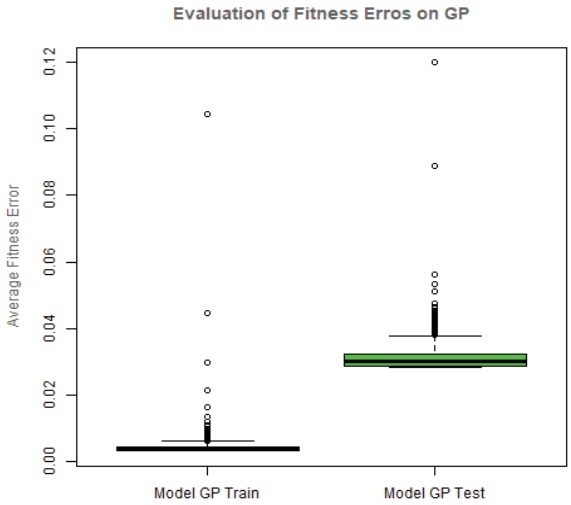

Figure 12- Boxplot with Fitness Errors of Genetic Programming

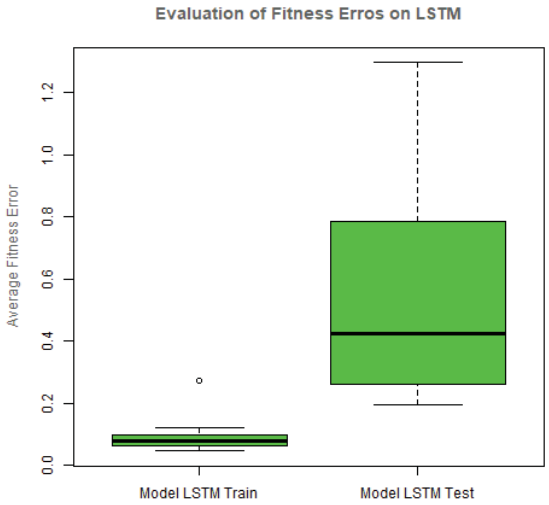

Figure 13- Boxplot with Fitness Errors of Long Short-Term Memory

Table 7 presents the values for the mean absolute error (MAE) of the training and test sets of each model. In this first approach, it is possible to observe a difference in the two algorithms, with model 2 from Genetic Programming clearly performing better. The difference between the amount of error on the train and test sets has a higher magnitude for the model created using LSTM.

Mean Absolute Error		
	Train	Test
Genetic Programming	0,004	0,030
Long-Short Term Memory	0,091	0,562

Table 7- Mean Absolute Error from Algorithms

The differences referred to above and presented in table 7, can be easily visualized in the boxplot on figure 72. In this chart it is possible to see that fitness errors in the model created by Genetic Programming are in a range between 0-0.12, while the errors from the model created by with Long-

Short Term Memory are in a range of 0-1.2 (see figure 72). Even with the first model producing some outliers, those never achieve the higher values of the model created by Long-Short Term Memory.

In order to compare these two models and verify the statistical significance of their results, the same tests that were applied for previous model comparison were used. The p-values obtained for the comparison between the Genetic Programming model and the Long-Short Term Memory model are presented in table 8.

Shapiro Wilk Test		
	Train	Test
Genetic Programming	< 0,001	< 0,001
Long-Short Term Memory	< 0,001	0,036

Table 8- Shapiro Wilk Test to Algorithms Errors

Wilcoxon Sum Rank Test		
	Long-Short Term Memory	
	Train	Test
Genetic Programming	< 0,001	< 0,001

Table 9- Wilcoxon Rank-Sum Test to Algorithms Errors

As seen on the table above, according to the Wilcoxon Rank-Sum Test it was proved that the two models do not follow the same distribution.

These results obtained for the Portuguese 10 Years Yield, indicate the capability of Genetic Programming and Long Short-Term Memory to produce predictive models for sovereign bonds yields.

Execution Time Models (Total of Time in seconds)	
	Train
Genetic Programming	145,96
Long-Short Term Memory	5,22

Table 10- Execution Time of the chosen models from Algorithms

As mentioned in the first chapters of this study, the execution time of each algorithms would be a criterion of comparison between models. For each model, the average time per Generation/Epoch in the total of forty iterations was computed. In table 10 the total execution time of each model in seconds is presented. The difference between models is clear – the model created using Genetic Programming takes approximately 29 times more time than the model created by Long-Short Term Memory.

6. CONCLUSION

Prediction on financial markets is always a complex and difficult task. The uncertainty and volatility are characteristics of sovereign yields that make any prediction almost inconceivable. The task of predicting future values of sovereign yields is the focus of this paper.

A review of the existing literature on the topic revealed that the existing research for yield prediction using a Machine Learning approach was scarce, especially in Portugal where it was not possible to find any papers on the topic. Understanding the future behaviour of financial markets, with a focus on the sovereign bonds market, is tremendously important to all participants. Planning an investment, hedging risk or even managing debt are some examples of responsibilities that depend on this market and consequences of this prediction. It is clear from this discussion that there is a need to address this problem.

Experiments were carried out on fifteen years of historical data with six variables regarding Portuguese Government Sovereign Debt. In this study, the application of an evolutionary algorithm and an artificial neural network were studied, namely Genetic Programming and Long Short-Term Memory, in order to find an answer to the problem proposed on this study.

On the initial phase of this study, both algorithms were individually analysed, testing different combinations of settings and scrutinizing their results. A comparation between algorithms was analysed too.

Concerning the different results from each algorithm and the main objective of this study, it is possible to conclude that Genetic Programming presented evidence that it can provide the finest prediction, when compared with Long Short-term Memory. This conclusion derived from the analysis of the mean absolute error and boxplot for each model both on the train and test sets. It was also supported by the statistical tests Shapiro Wilk and Wilcoxon Rank-Sum.

Analysing the results of the train dataset for the best Genetic Programming model (model 2) and the best LSTM model (model 3), it is possible to see that there is a difference of mean absolute error of 0.087 between the models, with Genetic Programming providing the best results. None of the models' errors followed a normal distribution and they did not follow the same distribution. This indicates, concerning the properties of the dataset and settings defined for the models, that the models created using Genetic Programming provide better results on the training process.

On the test dataset it was necessary to apply the same analyses to compare algorithms. The conclusion was similar to the one taken regarding the train dataset: the difference on mean absolute errors between models is 0.532 on test. Concerning statistical tests, the only difference from the conclusions taken regarding the training set was that in the model created using Long Short-Term, considering a significance level inferior to 0.03 ($\alpha<0.03$), there is no statistical evidence that the errors do not follow a normal distribution.

Experimental results reported in this study have shown that Genetic Programming is more than capable of producing satisfactory results when comparing the artificial neural network Long Short-Term Memory. These results are a clear indication that this algorithm is more than capable of generating appropriate predictive models of Portuguese Government Yield on 10Y.

However, despite of this algorithm producing satisfactory results, one of the concerns of this study was the time consumption of the algorithms. On this issue, considering that each model was run forty times on both algorithms, Genetic Programming needs 140.74 seconds more per iteration than Long Short-Term Memory, on average. Depending on the purpose of the use of the models created by these algorithms this could be a problem. In a trading environment, 140.74 seconds (2.34 minutes) is a long time, as the market has a lot of volatility and a lot of variables can change on that time. Nevertheless, when the purpose of these models is hedging risk, planning investments or managing debt, the difference between execution time of models is residual and can be ignored.

To summarise, this study provides an overview of two different algorithms from the field of Machine Learning - Genetic Programming and Long Short-Term Memory, two of the most used in predictions for financial markets. The focus was to predict Portuguese Government Yield on 10Y on next day (T+1) resorting to models constructed using these two different Machine Learning techniques. The results obtained showed Genetic Programming as the algorithm that can create the model with the finest accuracy. However, Long Short-Term Memory should not be ignored because it can also predict with a good accuracy. Regarding execution time, velocity is a problem of Genetic Programming. This algorithm takes more time to execute compared with other methodologies. Long Short-Term Memory is considerably quicker to get results. To take the right decision about which model to choose, one must take the priorities into consideration. In case of accuracy being the priority, Genetic Programming will be the answer, when velocity is the priority Long Short-Term should be the choice.

7. LIMITATIONS AND FUTURE WORKS

Concerning future working, there are many possible approaches that can be explored. As referred to in this study, Portuguese Government Bonds yield has many consequences for many players in financial markets. Exploring more deeply the prediction of it is truly important.

This study was written as a partial requirement for obtaining the Master's degree in Statistics and Information Management. My formation was not focused on Advanced Analytics methodologies. As a consequence of my lack of knowledge in this area I recognize that there is space to improve the parameter settings and the script of each algorithm in order to obtain more accurate results.

Other future work that could be explored is to compare Genetic Programming to classical (statistical) methodologies. In this study, this algorithm was compared with an artificial neural network but comparing a Machine Learning approach with linear models or any other classical model could give insides about the benefits (or not) of use this type of approach.

Another way of exploring this topic is to compare Genetic Programming and Long Short-Term Memory with other Machine Learning algorithms, such as Support Vector Machine or Random Forests.

Portuguese Government Bonds is a large topic to explore with many questions and approaches to be explored. On this study I just explored a part of this topic.

8. BIBLIOGRAPHY

[1] Adam Hayes, 2019, Bond. Retrieved from: https://www.investopedia.com/terms/b/bond.asp.

[2] Daniel Martin, Barnabás Póczos, Burton Hollifield, 2014, Machine learning-aided modeling of fixed income instruments.

[3] Agência de Gestão da Tesouraria e da Dívida Pública – IGCP, 03 Janeiro 2020, Glossário. Retrieved from: https://www.igcp.pt/pt/menu-lateral/gestao-da-divida-publica/glossario/

[4] Agência de Gestão da Tesouraria e da Dívida Pública – IGCP, 03 Janeiro 2020, Apresentação. Retrieved from: https://www.igcp.pt/pt/menu-lateral/o-igcp-e-p-e/apresentacao/

[5] Agência de Gestão da Tesouraria e da Dívida Pública – IGCP, 03 Janeiro 2020, OEVT and OMP. Retrieved from: https://www.igcp.pt/en/1-4-399/market-participants/oevt-and-omp/

[6] Donald Michie, 1968, "Memo" Functions and Machine Learning, Experimental Programming Unit University of Edinburgh.

[7] John R. Koza, 1997, Genetic Programming, Search Methodologies: Introductory Tutorials In Optimization And Decision Support Techniques pg. 127- 164.

[8] R. Lakshman Naik, D. Ramesh, B. Manjula, Dr. A. Govardhan, 2012, Prediction of StockMarket Index Using Genetic Algorithm, Computer Engineering and Intelligent Systems.

[9] Raghav Nandakumar, Uttamraj K., Vishal, Y. V. Lokeswari, 2018, Stock Price Prediction Using Long Short-Term Memory, International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue:03.

[10] Securities Industry and Financial Markets Association, 2019, SIFMA- Capital Markets Fact Book 2019.

[11] Data obtain through Bloomberg Terminal, 31/03/2020.

[12] Bin weng, 2017, Application of machine learning techniques for stock market prediction, Faculty of Auburn University.

[13] Sneha Soni, 2016, Applications of ANNs in Stock Market Prediction: A Survey, International Journal of Computer Science & Engineering Technology.

[14] Swetava Ganguli, Jared Dunnmon, 2017, Machine Learning for Better Models for Predicting Bond Prices, Cornell University.

[15] André Dinis Oliveira, Novembro 2016, Forecasting the PSI-20 index using a Machine Learning approach, NOVA Information Management School.

[16] Will Kenton, 2019, Opportunity Cost. Retrieved from: https://www.investopedia.com/terms/o/opportunitycost.asp

[17] Arturo Estrella and Frederic S. Mishkin, 1996, The yield curve as a predictor of recessions in the United States and Europe, Current Issues in Economics and Finance Volume 2 Number 7.

[18] Castelli, Mauro & Silva, Sara & Vanneschi, Leonardo, 2014, A C++ framework for geometric semantic Genetic programming. Genetic Programming and Evolvable Machines.

[19] Merton R. C. (1973). Theory of rational option pricing, Bell Journal of Economics and Management Science 4: 141–183.

[20] Vasicek O. (1977). An equilibrium characterization of the term structure, Journal of Financial Economics 5: 177–188.

[21] Cox J., Ingersoll J. and Ross S., 1985, A theory of the term structure of interest rates, Econometrica 53: 385.407.

[22] Charles R. Nelson and Andrew F. Siege, 1987, Parsimonious Modeling of Yield Curves, The Journal of Business Vol. 60, No. 4 (Oct., 1987), pp. 473-489.

[23] Bank of International Settlements, 2005, Zero-coupon yield curves — Technical documentation. BIS Paper, 25. (pp. 1–37).

[24] Laura Coroneo, Ken Nyholm & Rositsa Vidova-Koleva, 2008, How Arbitrage-Free is the Nelson-Siegel Model, Working Paper Series Nº874.

[25] Barrett, W. R., Gosnell, T. F., Jr., & Heuson, A. J., 1995, Yield curve shifts and the selection of immunization strategies. Journal of Fixed Income, 53–64.

[26] Hodges, S. D., & Parekh, N., 2006, Term structure slope risk: Convexity revisited. Journal of Fixed Income, 16(3), 54–59

[27] Dullmann, K., & Uhrig-Homburg, M., 2000, Profiting risk structure of interest rates: An empirical analysis for Deutschemark-denominated bonds. European Financial Management, 6, 367–388.

[28] Fabozzi, F. J., Martellini, L., & Priaulet, P., 2005, Predictability in the shape of the term structure of interest rates. Journal of Fixed Income, 40–53.

[29] Diebold, F. X., & Li, C. (2006). Forecasting the term structure of government bond yields. Journal of Econometrics, 130, 337–364.

[30] Pooter, M.D., Ravazzolo, F., van Dijk, D., 2007, Predicting the Term Structure of Interest Rates: Incorporating Parameter Uncertainty, Model Uncertainty and Macroeconomic Information, Tinbergen Institute Discussion Papers 07-028/4.

[31] Gu, Shihao, Bryan T. Kelly, and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, Chicago Booth Research Paper 18-04, Chicago Booth

[32] Joarder Kamruzzama, Ruhul A. Sarker, 2014, NN-Based Forecasting of Foreign Currency Exchange Rates. Neural Information Processing - Letters and Reviews Vol. 3, No. 2.

[33] Minakhi Routa, Babita Majhib,c, Ritanjali Majhid, Ganapati Pandaea, 2014, Forecasting of currency exchange rates using an adaptive ARMA model with differential evolution-based training.

Journal of King Saud University - Computer and Information Sciences. Volume 26, Issue 1, January 2014, Pages 7-18

[34] Dase R.K., Pawar D.D, 2010, Application of Artificial Neural Network for stock market predictions: A review of literature. International Journal of Machine Intelligence, Volume 2, Issue 2, 2010, pp-14-17.

[35] T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka., 1990, Stock market prediction system with modular neural networks, IJCNN International Joint Conference on Neural Networks

[36] Xiaowei Lin, Zehong Yang, Yixu Song, 2011, Intelligent stock trading system based on improved technical analysis and Echo State Network. Expert Systems with Applications Volume 38, Issue 9, September 2011, Pages 11347-11354.

[37] Sotirios P. Chatzis, Vassilis Siakoulis, Anastasios Petropoulos, Evangelos Stavroulakis, Nikos Vlachogiannakis, 2018, Forecasting stock market crisis events using deep and statistical machine learning techniques. Expert Systems with Applications Volume 112, 1 December 2018, Pages 353-371.

[38] Cheng, W., Wagner, L. and Lin, C.H., 1996, Forecasting the 30-year U.S. Treasury bond with a system of neural networks, Journal of Computational Intelligence in Finance, vol. 4, pp. 10-16.

[39] Din, A. ,2003, Forecasting interest rates using neural network models, Geneva Research Collaboration, Technical Report.

[40] Rajiv Sambasivan, Sourish Das, 2017, A statistical machine learning approach to yield curve forecasting. 2017 International Conference on Computational Intelligence in Data Science (ICCIDS).

[41] Koza, JR., 1992, Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT, Cambridge.

[42] Alberto Moraglio, Krzysztof Krawiec, Colin G. Johnson, 2012, Geometric Semantic Genetic Programming. Parallel Problem Solving from Nature - PPSN XII pp 21-31.

[43] Vanneschi L., Castelli M., Manzoni L., Silva S., 2013, A New Implementation of Geometric Semantic GP and Its Application to Problems in Pharmacokinetics. Genetic Programming. EuroGP 2013. Lecture Notes in Computer Science, vol 7831.

[44] Leonardo Vanneschi, Sara Silva, Mauro Castelli and Luca Manzoni, 2014, Geometric Semantic Genetic Programming for Real Life Applications. Genetic Programming Theory and Practice XI pp 191-209.

[45] M. A. Kaboudan, 1999, Genetic Programming Prediction of Stock Prices. Computational Economics 16: 207–236, 2000.

[46] R. Lakshman Naik, D. Rames , B. Manjula, Dr. A. Govardhan, 2012, Prediction of Stock Market Index Using Genetic Algorithm. Computer Engineering and Intelligent Vol 3, No.7, 2012.

[47] Alaa Sheta, Hossam Faris, Mouhammd Alkasassbeh, 2013, A Genetic Programming Model for S&P 500 Stock Market Prediction. International Journal of Control and Automation Vol.6, No.5 (2013), pp.303-314.

[48] Mauro Castelli, Leonardo Vanneschi, Leonardo Trujillo, Aleš Popovič, 2017, Stock index return forecasting: Semantics-based genetic programming with local search optimizer. Int. J. Bio-Inspired Computation, Vol. 10, No. 3, 2017.

[49] Sepp Hochreiter, Jürgen Schmidhuber, 1997, Long Short-Term Memory. Neural Computation Volume 9 Issue 8 November 15, 1997, p.1735-1780.

[50] Kai Chen, Yi Zhou, Fangyan Da, 2015, A LSTM-based method for stock returns prediction: A case study of China stock market. 2015 IEEE International Conference on Big Data (Big Data).

[51] Hansson, Magnus, 2017, On stock return prediction with LSTM networks. Department of Economics Lund University.

[52] Martins, Graça, Gualter Couto e Piriquito Costa, 2002, Análise da volatilidade do prémio de risco do mercado de capitais português. Estudos de Gestão, VII(1):19-42.

[53] Caiado, Jorge (2004). Modelling and forecasting the volatility of the portuguese stock index PSI-20. Estudos de Gestão, IX(1):3-22.

[54] Isfan, M., Menezes, R., Mendes, D. A., 2010, Forecasting the Portuguese stock market time series by using artificial neural networks. 7th International Conference on Applications of Physics in Financial Analysis Journal of Physics: Conference Series 221.

[55] Tom M. Mitchell, 1997, Machine Learning, McGraw-Hill Science/Engineering/Mat.

[56] Alexei Botchkarev, 2018, Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology, Interdisciplinary Journal of Information, Knowledge, and Management, 2019, 14, 45-79.

[57] Kenneth E. Kinnear, Jr, 1993, Generality and Difficulty in Genetic Programming: Evolving a Sort, Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest, Ed., San Mateo, CA: Morgan Kaufmann.

[58] Mitchell Melanie, 1999, An Introduction to Genetic Algorithms, A Bradford Book The MIT Press Cambridge, Massachusetts Fifth printing 1999.

[59] Poli, Riccardo & Langdon, William & Mcphee, Nicholas., 2008, A Field Guide to Genetic Programming.

[60] A Beginner's Guide to LSTMs and Recurrent Neural Networks. Retrieved from: https://pathmind.com/wiki/lstm#long

[61] Felix Gers, 2001, Long Short-Term Memory in Recurrent Neural Networks, E´Cole Polytechnique Fe´De´Rale De Lausanne.

[62] Christopher Olah, 2015, Understanding LSTM Networks. Retrieved from: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[63] Hawkins D., 1980, Identification of Outliers, Chapman and Hall.

[64] Choi, Sung-Soon & Moon, Byung, 2003, Normalization in Genetic Algorithms, pag.862-873.

[65] Chai, Tianfeng & Draxler, R.R.., 2014, Root mean square error (RMSE) or mean absolute error (MAE)- Arguments against avoiding RMSE in the literature. Geoscientific Model Development.

[66] Willmott, C. & Matsuura, K., 2005, Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance.

[67] Georgios Drakos, 2018, How to select the Right Evaluation Metric for Machine Learning Models: Part 1 Regression Metrics. Retrieved from: https://bit.ly/3aVg2V3

[68] David Ruppert, David S. Matteson, 2015, Statistics and Data Analysis for Financial Engineering.

9. APPENDIX

9.1. PORTUGUESE GOVERNMENT DEBT

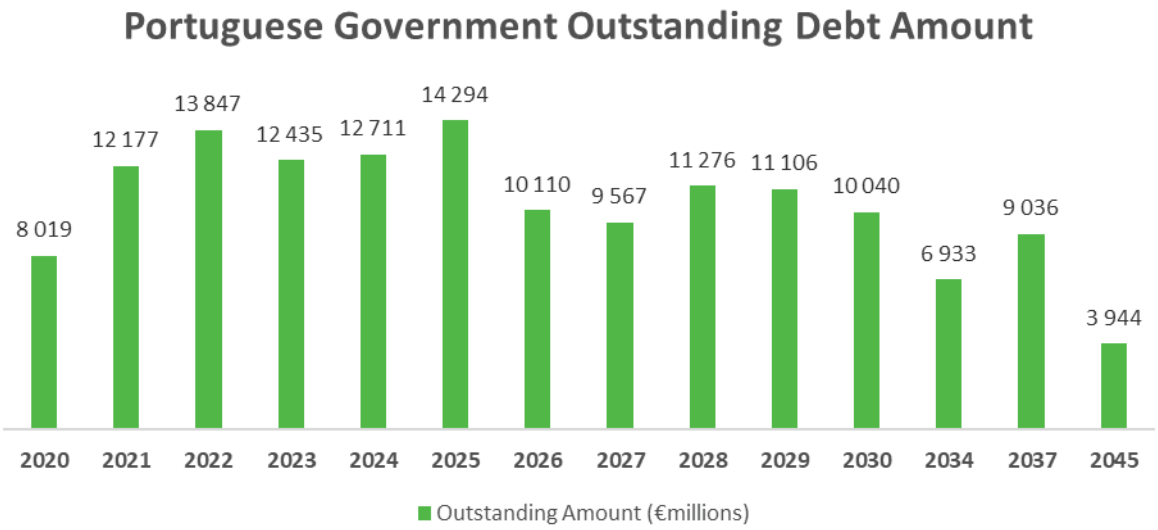

Figure 14- Chart of Portuguese Governmenet Outstanding Debt Amount (25 March 2020)

Figure 15- Evolution of Yield of Portuguese Government Bond 10 Years (With #NA Data)

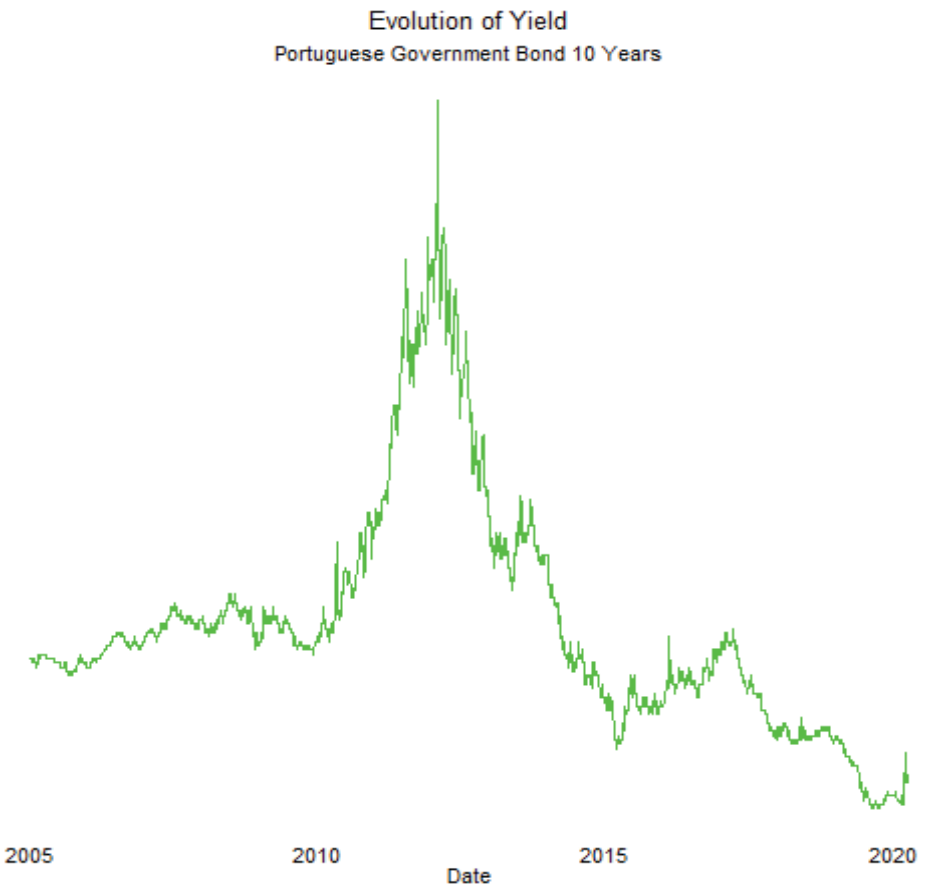

Figure 16- Evolution of Yield of Portuguese Government Bond 10 Years (With Interpolated Data)

9.2. DATA PRE-PROCESSING

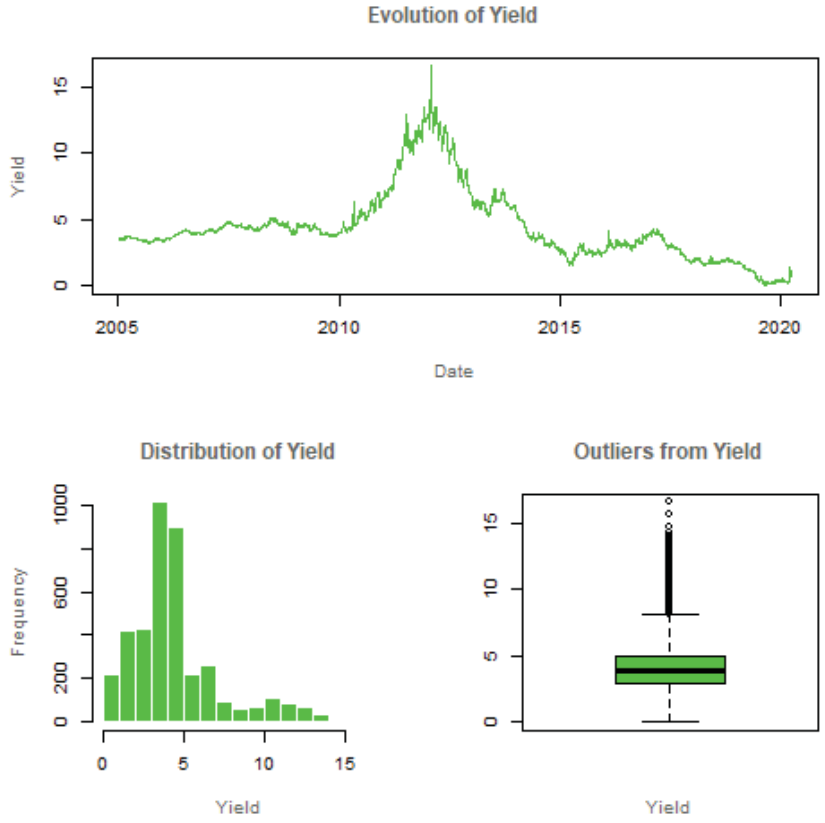

Figure 17- Descriptive Analysis of Yield Variable

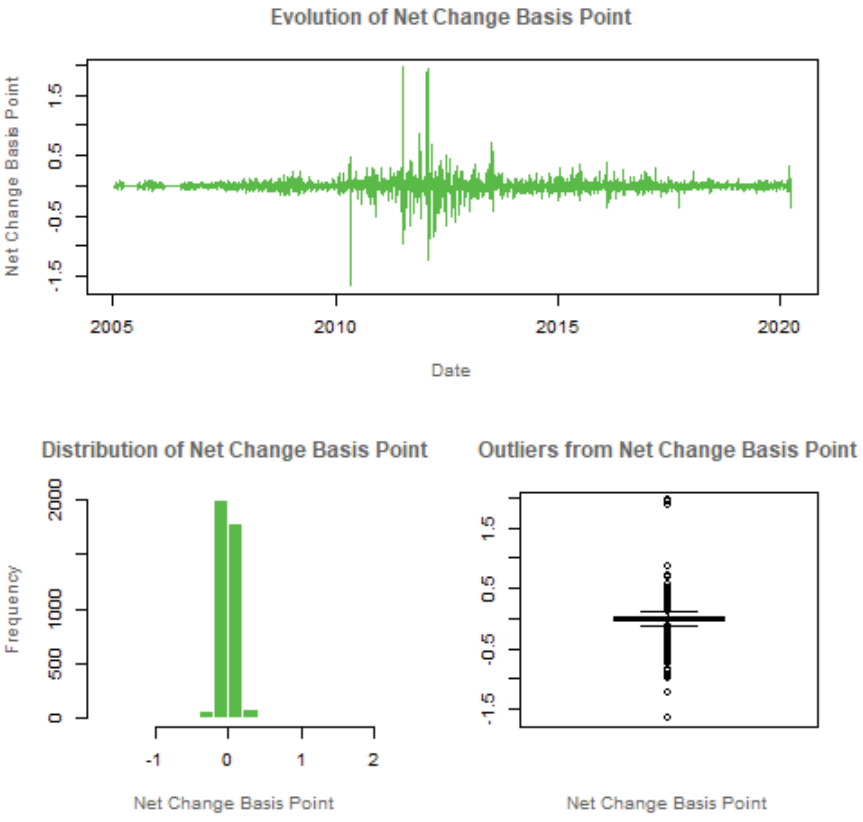

Figure 18- Descriptive Analysis of Net Change Basis Point Variable

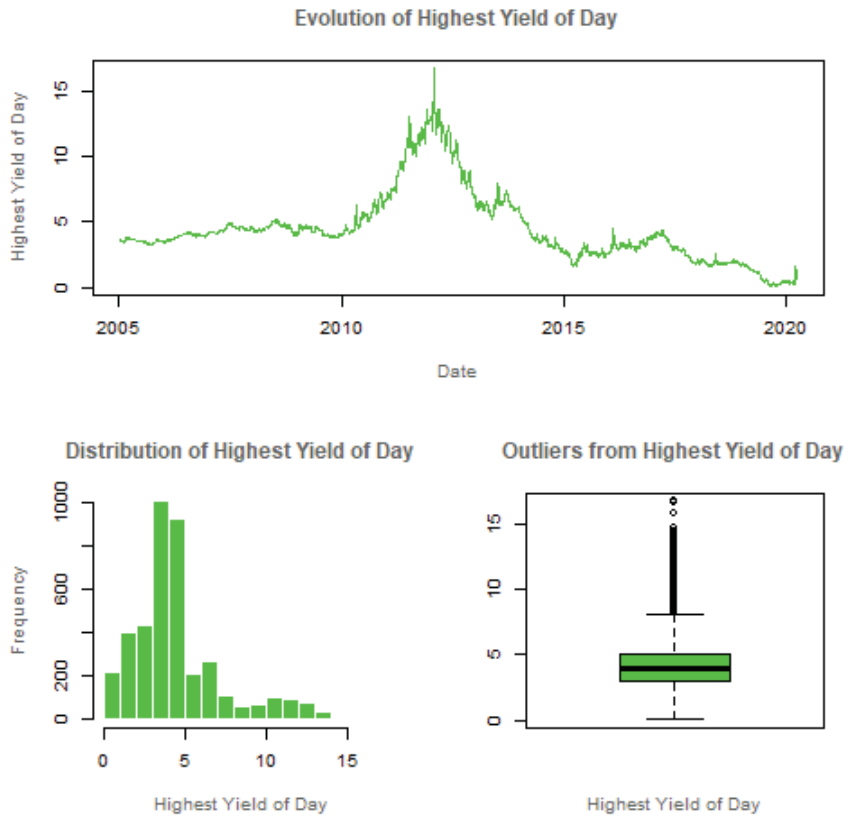

Figure 19- Descriptive Analysis of Highest Yield of Day Variable

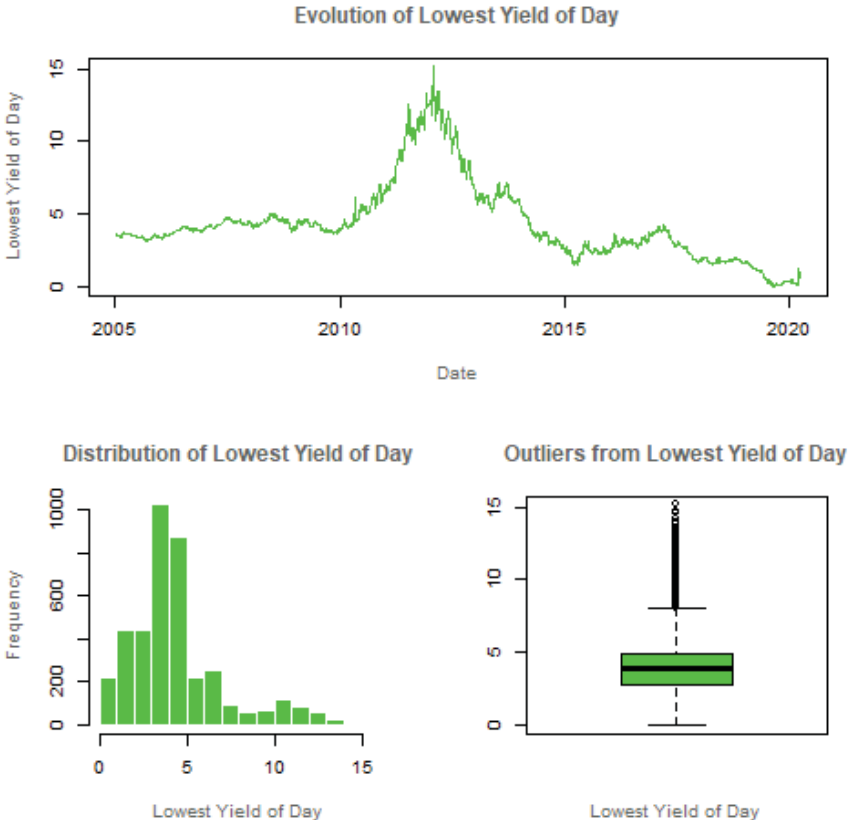

Figure 20- Descriptive Analysis of Lowest Yield of Day Variable

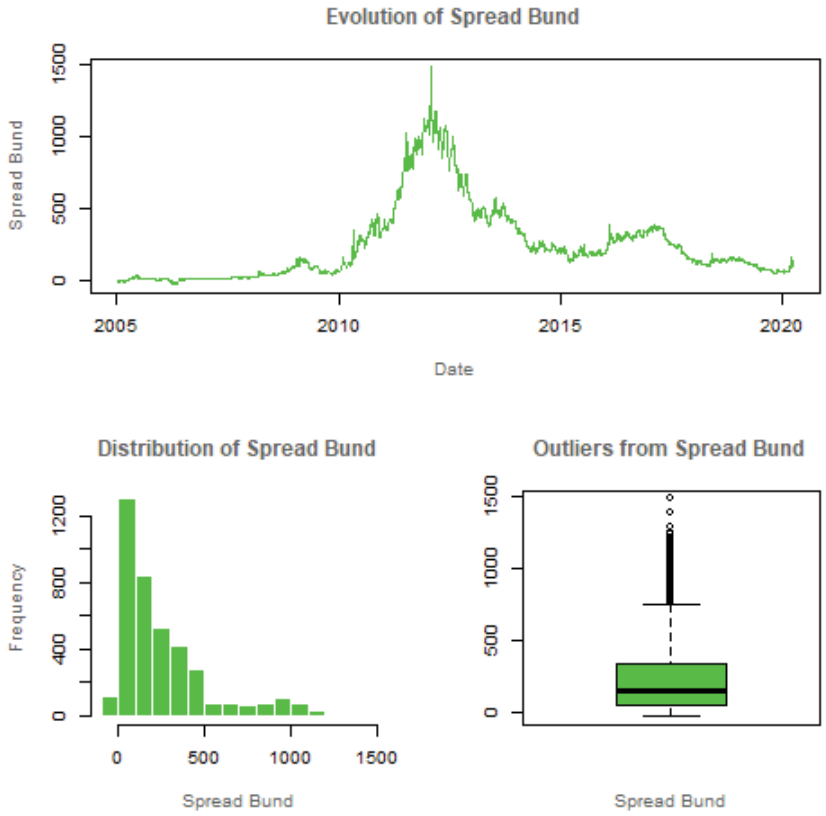

Figure 21- Descriptive Analysis of Spread Bund Variable

Figure 22- Descriptive Analysis Bund Yield Variable

Figure 23- Descriptive Analysis of Bund Future Variable

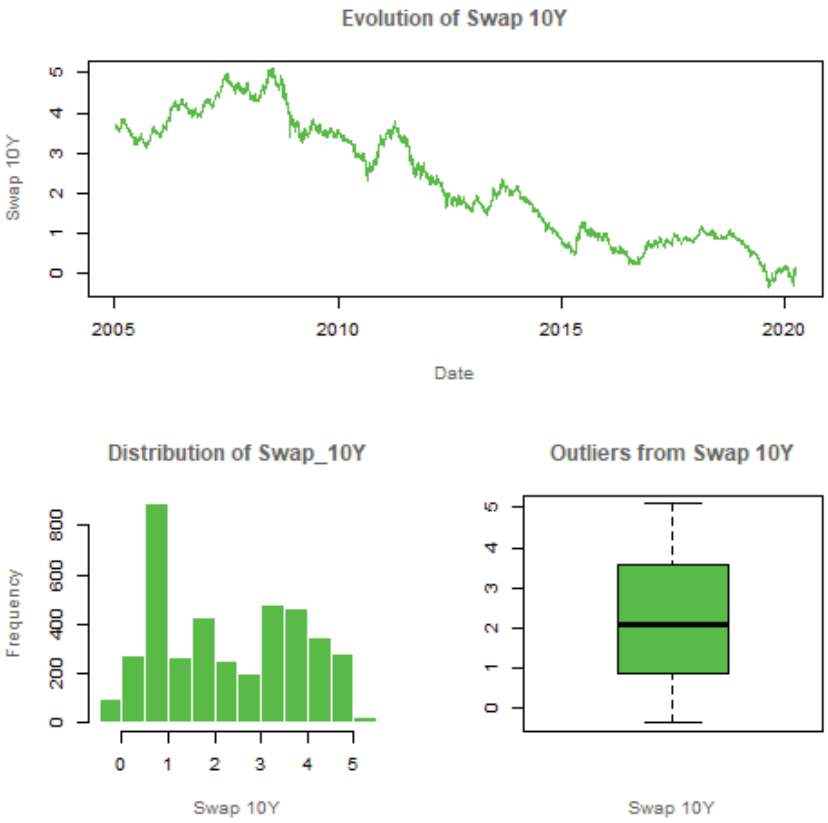

Figure 24- Descriptive Analysis of Swap 10Y Variable

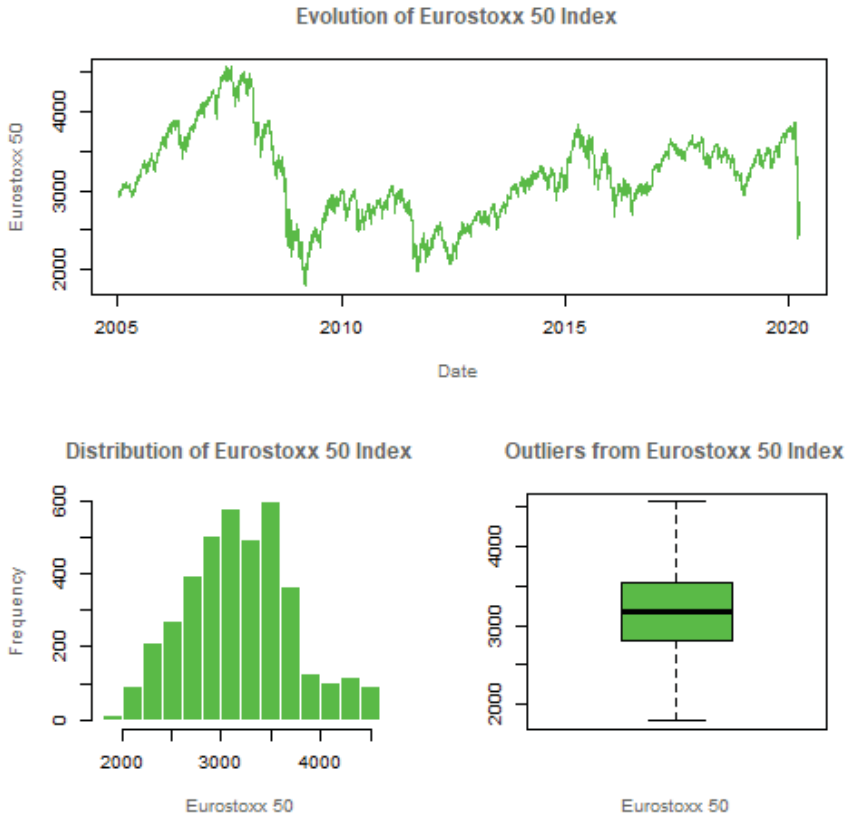

Figure 25- Descriptive Analysis of Eurostoxx 50 Variable

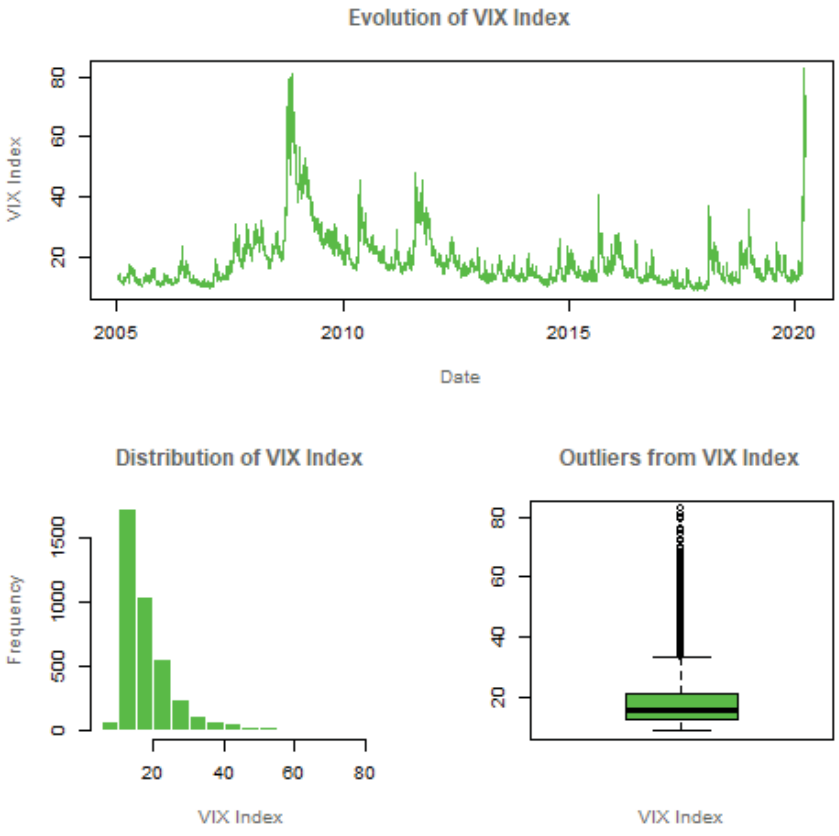

Figure 26- Descriptive Analysis of VIX Index Variable

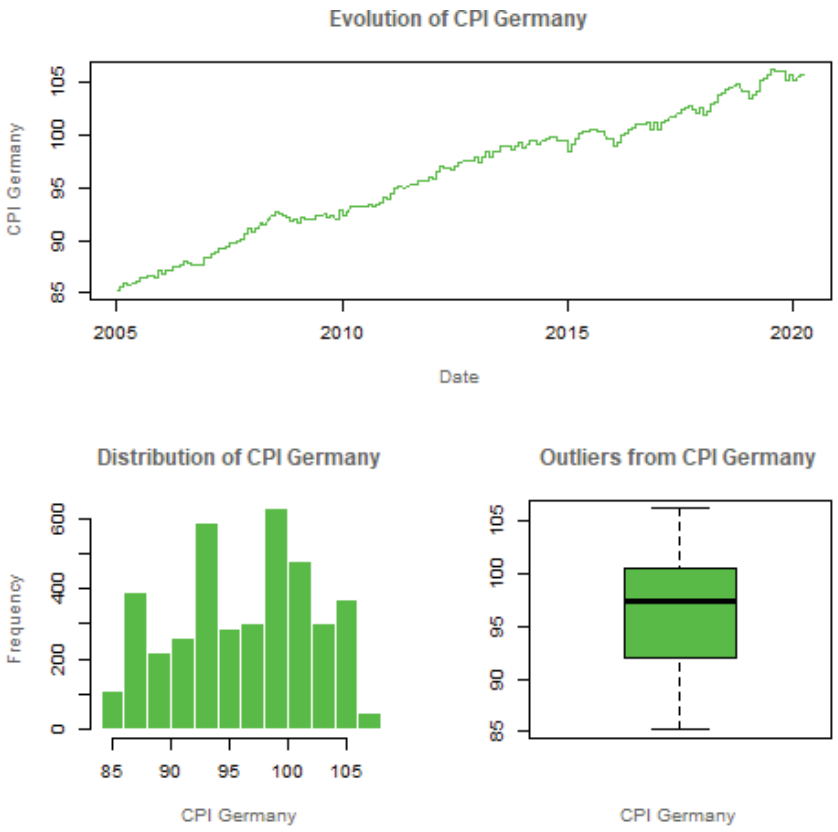

Figure 27- Descriptive Analysis of CPI Germany Variable

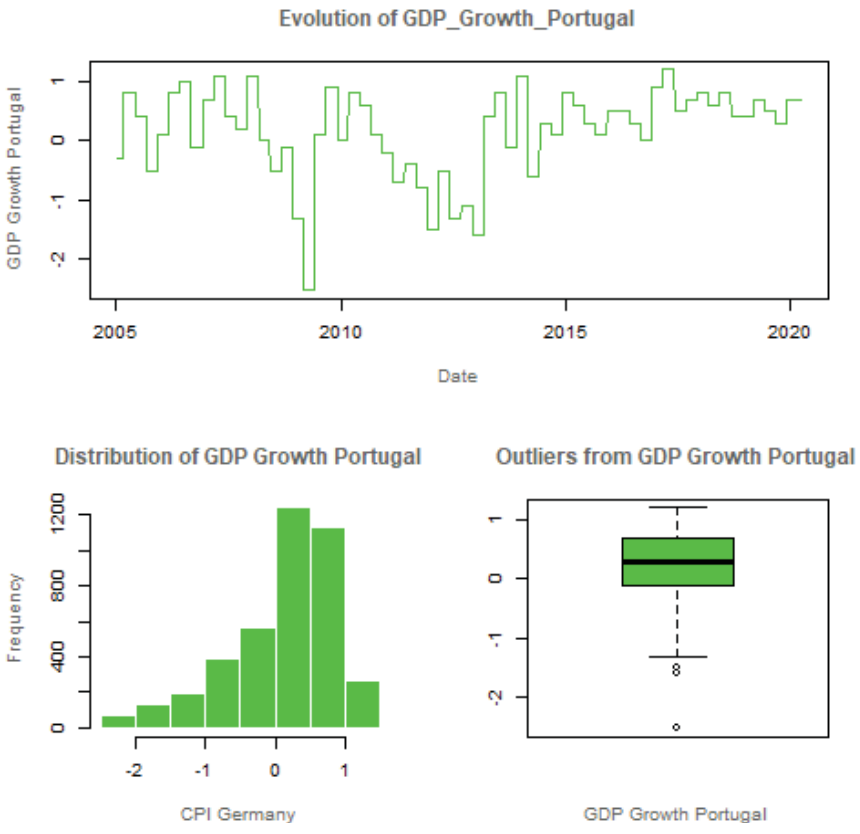

Figure 28- Descriptive Analysis of GDP Growth Portugal Variable

	Bund_Future	Bund_Yield	CPI_Germany	Eurostox_50	GDP_Growth_Portugal	Highest_Yield_of_Day	Lowest_Yield_of_Day	Net_Change_Basis_Point	Spread_Bund	Swap_10Y	VIX_Index	Yield
Mean	139,96	1,95	96,32	3 181,12	0,16	4,49	4,39	0,00	257,52	2,30	18,61	4,44
Std.Dev	19,66	1,53	5,79	542,26	0,75	2,80	2,75	0,12	265,06	1,48	9,33	2,78
Min	109,80	-0,86	85,30	1 809,98	-2,50	0,12	0,06	-1,63	-3,20	-0,33	9,14	0,07
Q1	121,49	0,45	92,10	2 802,55	-0,10	2,84	2,77	-0,03	66,80	0,90	12,89	2,81
Median	140,39	1,75	97,40	3 165,47	0,30	4,00	3,93	0,00	174,90	2,10	15,69	3,96
Q3	159,63	3,33	100,60	3 532,05	0,70	5,13	5,04	0,03	345,00	3,56	21,07	5,10
Max	179,44	4,68	106,20	4 557,57	1,20	16,70	15,17	1,96	1 481,40	5,09	82,69	16,61
Skewness	0,14	0,08	-0,17	0,20	-1,25	1,32	1,28	1,74	1,63	0,12	2,73	1,30
Kurtosis	-1,42	-1,42	-1,06	-0,25	1,54	1,78	1,65	77,27	2,24	-1,36	9,91	1,70

Table 11- Descriptive Statistics to all variables of Original Dataset

```
> tseries::adf.test(Dataset_Original_Without_NA$Yield, k = 0)

        Augmented Dickey-Fuller Test

data:  Dataset_Original_Without_NA$Yield
Dickey-Fuller = -1.479, Lag order = 0, p-value = 0.7989
alternative hypothesis: stationary

> tseries::adf.test(Dataset_Original_Without_NA$Net_Change_Basis_Point, k = 0)

        Augmented Dickey-Fuller Test

data:  Dataset_Original_Without_NA$Net_Change_Basis_Point
Dickey-Fuller = -52.921, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In tseries::adf.test(Dataset_Original_Without_NA$Net_Change_Basis_Point,  :
  p-value smaller than printed p-value
```

Figure 29- Augment Dickey-Fuller Test on Yield and Net Change Basis Point Variables

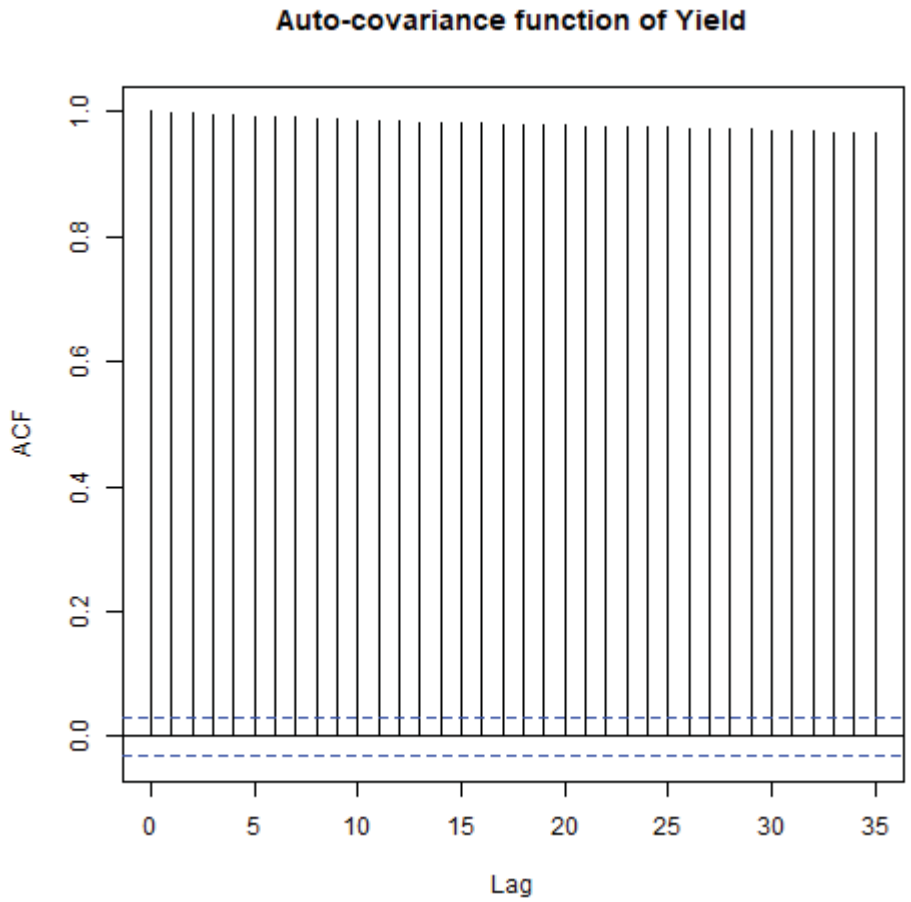

Figure 30- Auto-Covariance Function of Yield Variable

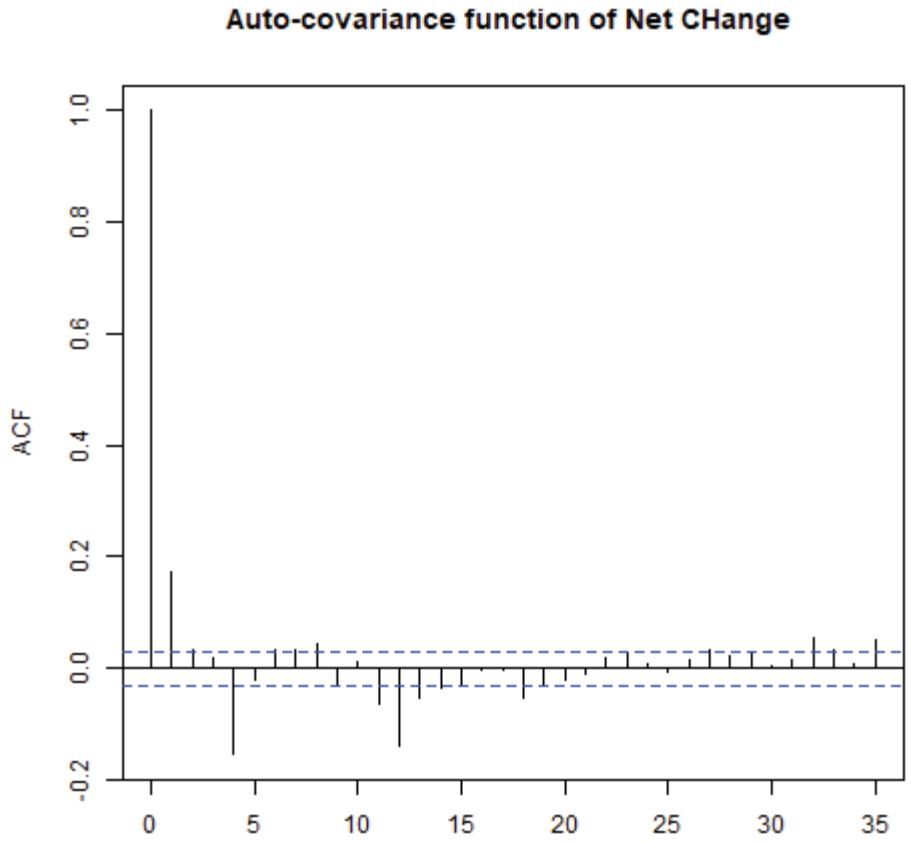

Figure 31- Auto-Covariance Function of Net Change Basis Point Variable

```
> ets(Dataset_Original_Without_NA$Yield)
ETS(A,Ad,N)

Call:
 ets(y = Dataset_Original_Without_NA$Yield)

  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.0869
    phi   = 0.8

  Initial states:
    l = 3.6648
    b = -0.1487

  sigma:  0.1146

     AIC     AICc      BIC
15742.14 15742.17 15779.87
> Fit_Yield<- ets(Dataset_Original_Without_NA$Yield)
> Fit_Yield_2<- ets(Dataset_Original_Without_NA$Yield, model="ANN")
> Deviance_Yield <- 2*c(logLik(Fit_Yield) - logLik(Fit_Yield_2))
> DF_Yield <- attributes(logLik(Fit_Yield))$df - attributes(logLik(Fit_Yield_2))$df
> #P value
> 1-pchisq(Deviance_Yield,DF_Yield)
[1] 5.356377e-06
```

Figure 32- Seasonality Test on Yield Variable

```
> ets(Dataset_Original_Without_NA$Net_Change_Basis_Point)
ETS(A,N,N)

Call:
 ets(y = Dataset_Original_Without_NA$Net_Change_Basis_Point)

  Smoothing parameters:
    alpha = 1e-04

  Initial states:
    l = -7e-04

  sigma:  0.1152

     AIC     AICc      BIC
15775.75 15775.76 15794.62
> Fit_Net_Change <- ets(Dataset_Original_Without_NA$Net_Change_Basis_Point)
> Fit_Net_Change_2 <- ets(Dataset_Original_Without_NA$Net_Change_Basis_Point, model="ANN")
> Deviance_Net_Change <- 2*c(logLik(Fit_Net_Change) - logLik(Fit_Net_Change_2))
> DF_Net_Change <- attributes(logLik(Fit_Net_Change))$df - attributes(logLik(Fit_Net_Chang
e_2))$df
> #P value
> 1-pchisq(Deviance_Net_Change,DF_Net_Change)
[1] 1
```

Figure 33- Seasonality Test on Net Change Basis Point Variable

```
> #The QS statistic looks for positive seasonal autocorrelation in a series to test the hypothesis that there is
 no seasonality in the series
> qs(Dataset_Original_Without_NA$Yield, freq= 5, diff = T, residuals = F, autoarima = T)
Test used:  QS

Test statistic:  0
P-value:  1
```

Figure 34- QS Statistics on Yield Variable

```
> qs(Dataset_Original_Without_NA$Net_Change_Basis_Point, freq= 5, diff = T, residuals = F, autoarima = T)
Test used:  QS

Test statistic:  31.41
P-value:  1.510944e-07
```

Figure 35- QS Statistics on Net Change Basis Point Variable

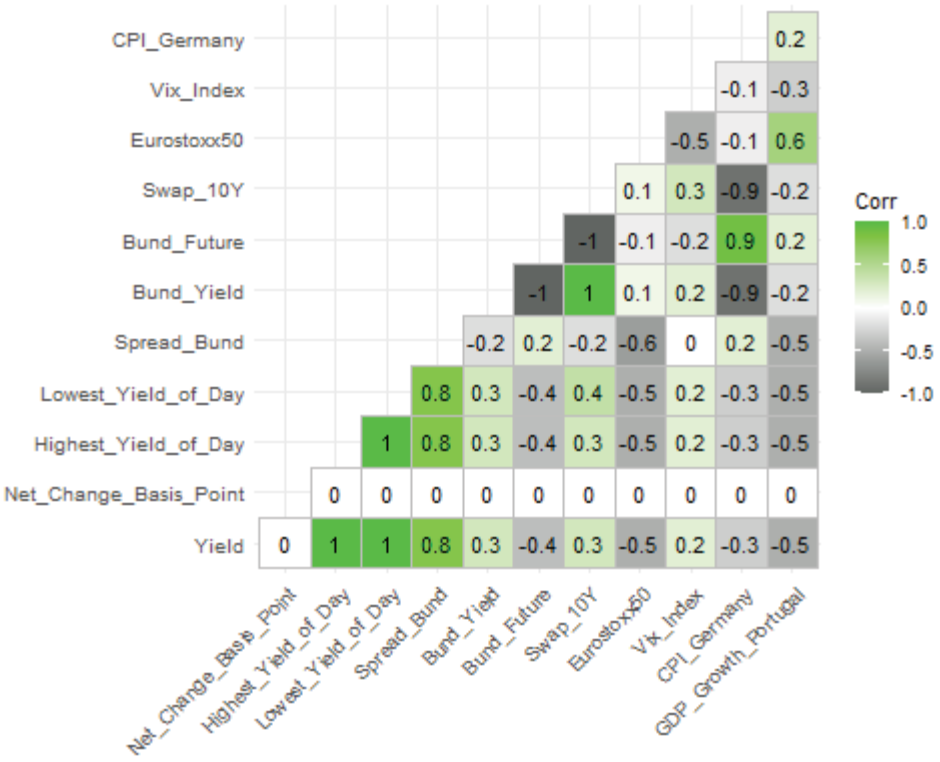

Figure 36- Correlation Matrix of Original Dataset

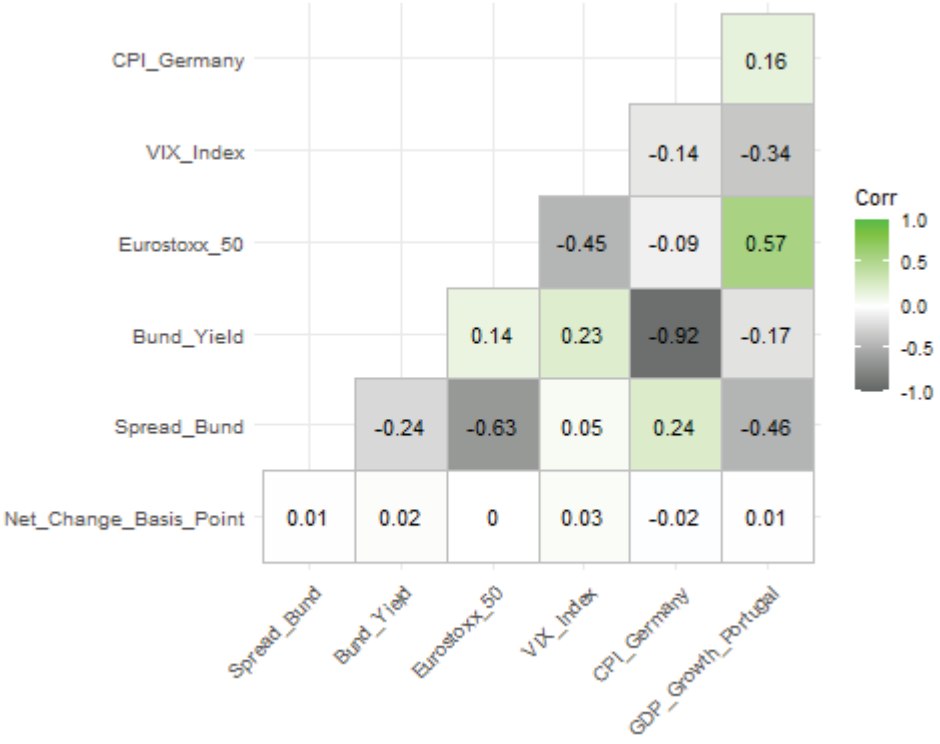

Figure 37- Correlation Matrix of Modified Dataset

9.3. GENETIC PROGRAMMING

9.3.1. Genetic Programming- Fitness Train Models Analysis

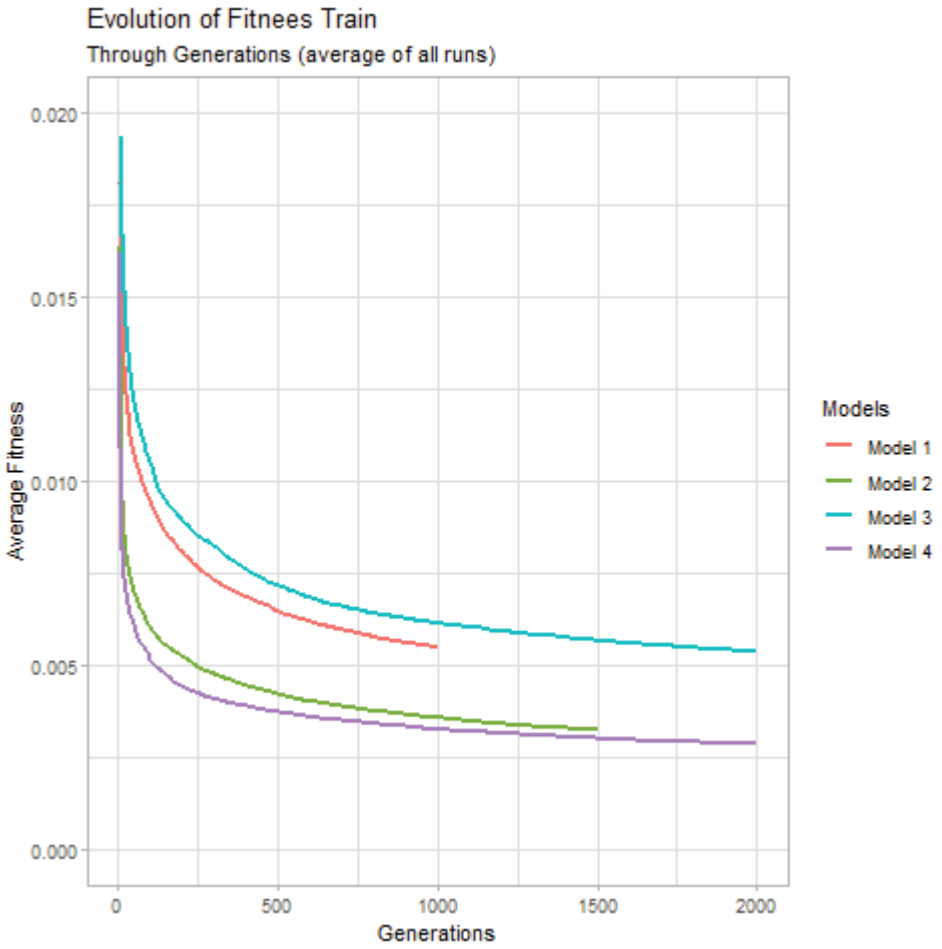

Figure 38- Evolution of Fitness Train through Generations (average of all runs) of Genetic Programming Models

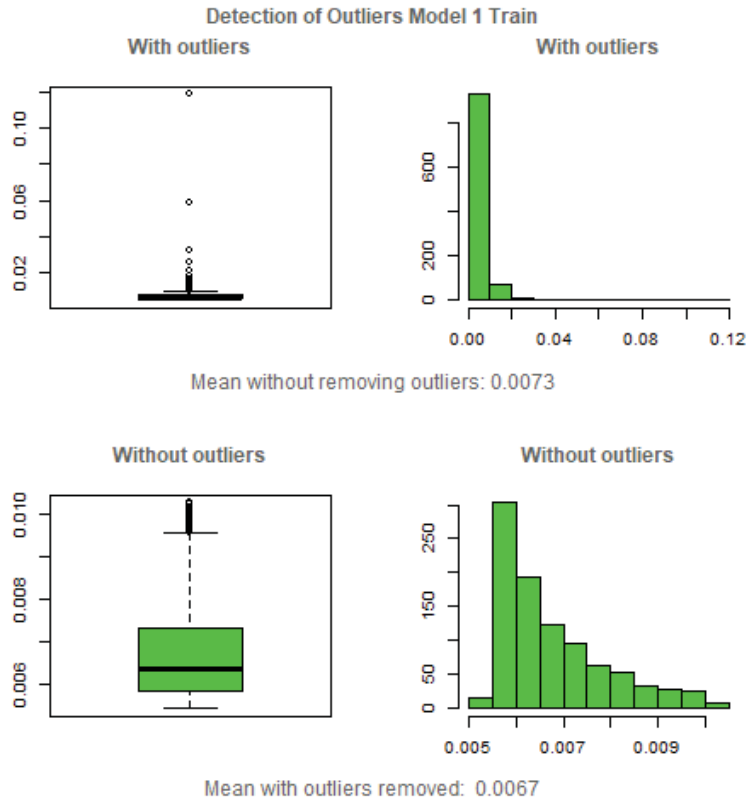

Figure 39- Outliers Analysis on Fitness Train of Model 1, Genetic Programming

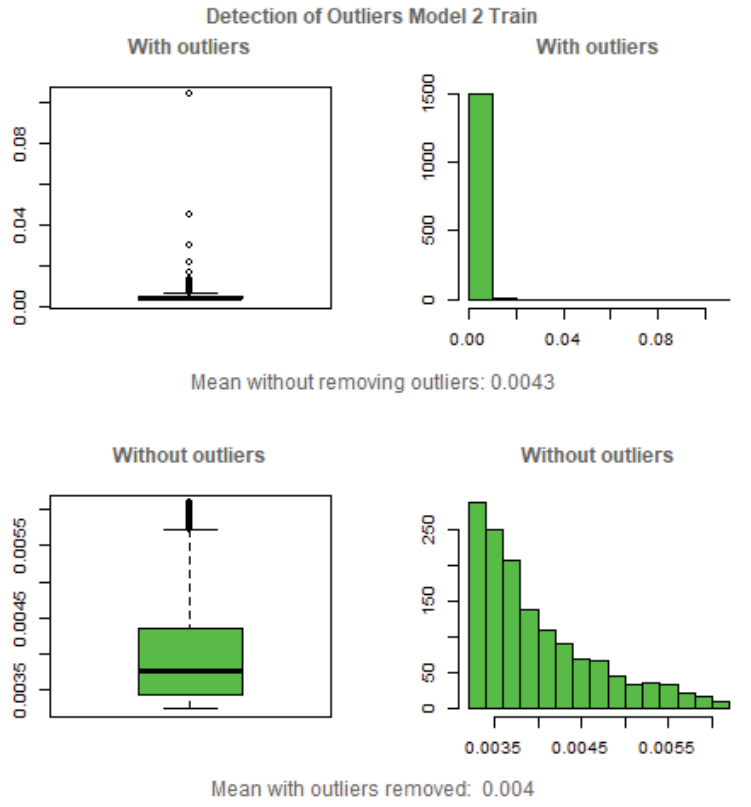

Figure 40- Outliers Analysis on Fitness Train of Model 2, Genetic Programming

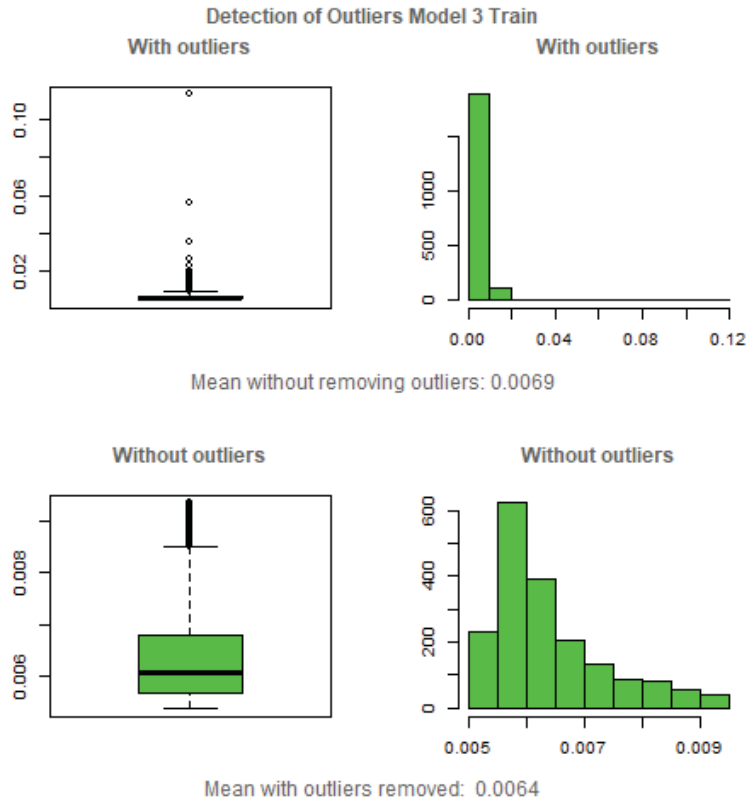

Figure 41- Outliers Analysis on Fitness Train of Model 3, Genetic Programming

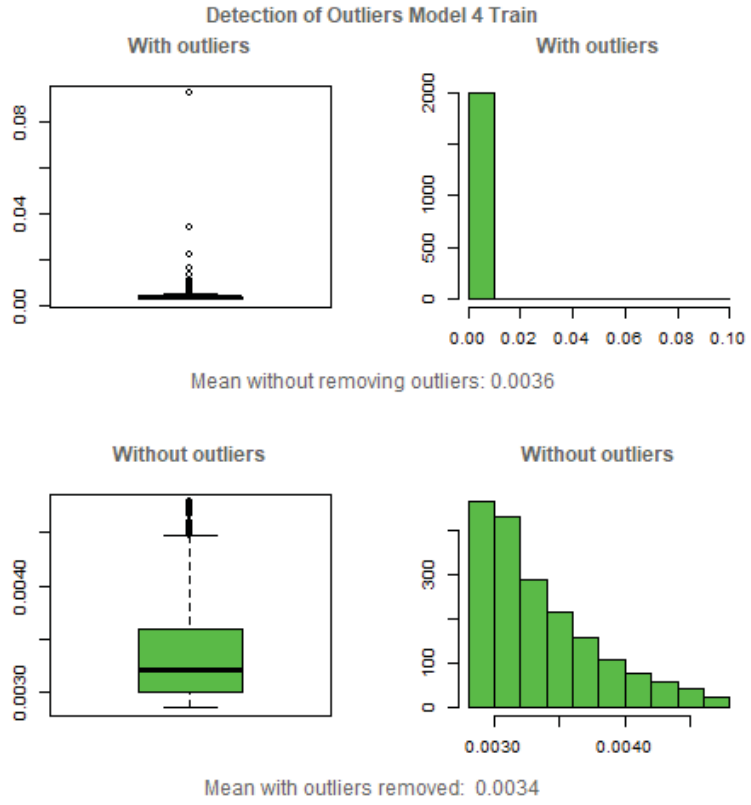

Figure 42- Outliers Analysis on Fitness Train of Model 4, Genetic Programming

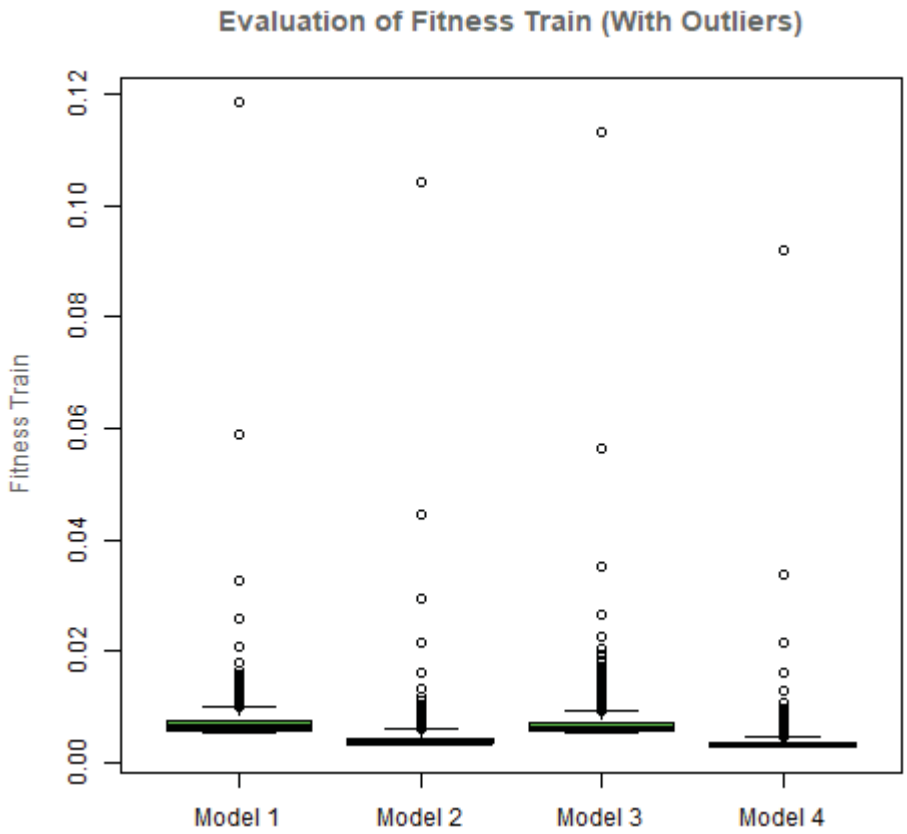

Figure 43- Boxplot with Fitness Train distribution of models (With Outliers), Genetic Programming

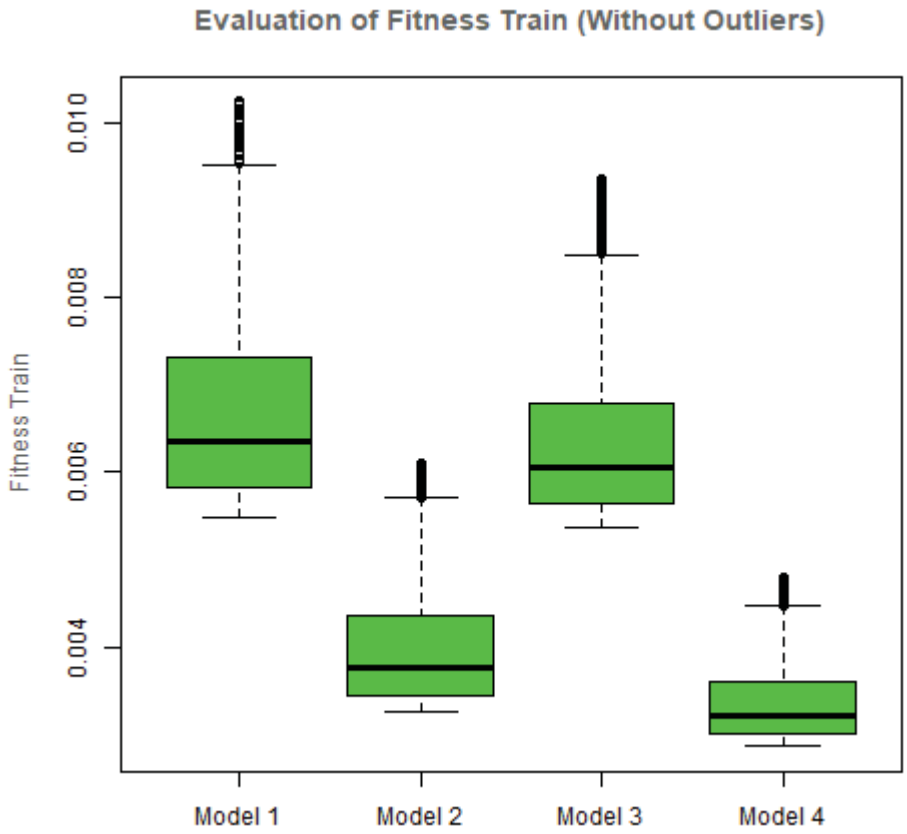

Figure 44- Boxplot with Fitness Train distribution of models (Without Outliers), Genetic Programming

9.3.2. Genetic Programming- Fitness Test Models Analysis

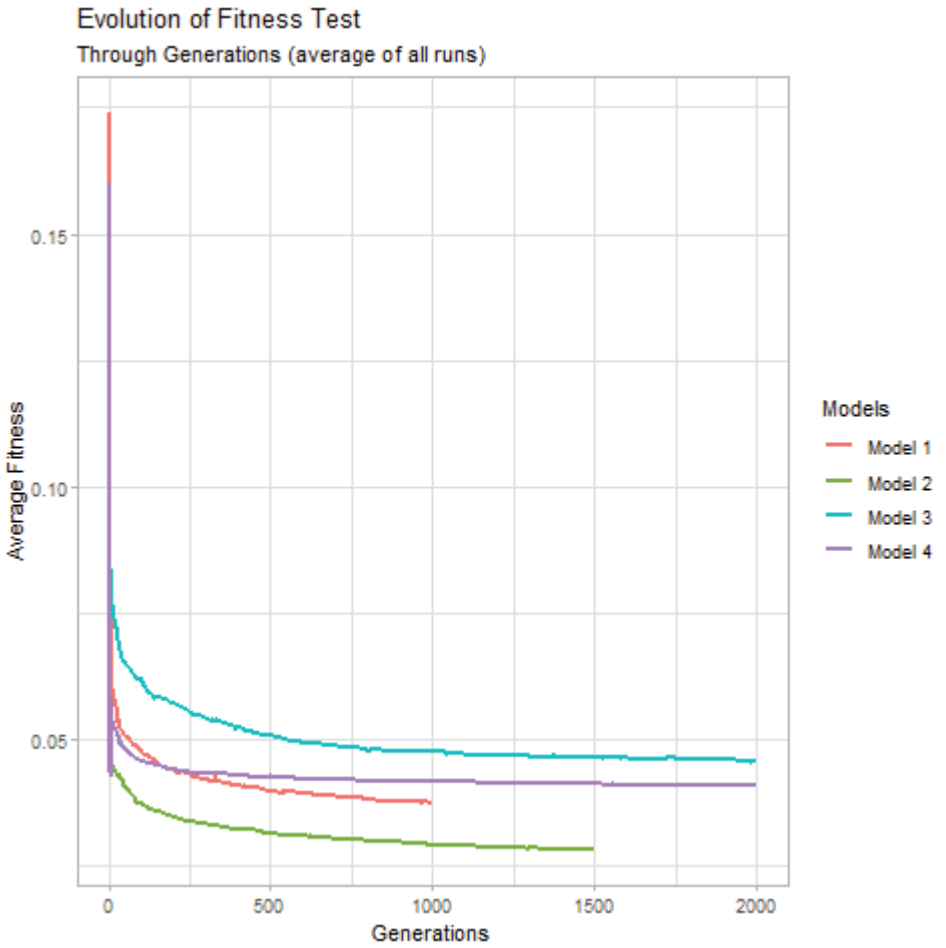

Figure 45- Evolution of Fitness Test through Generations (average of all runs) of Genetic Programming Models

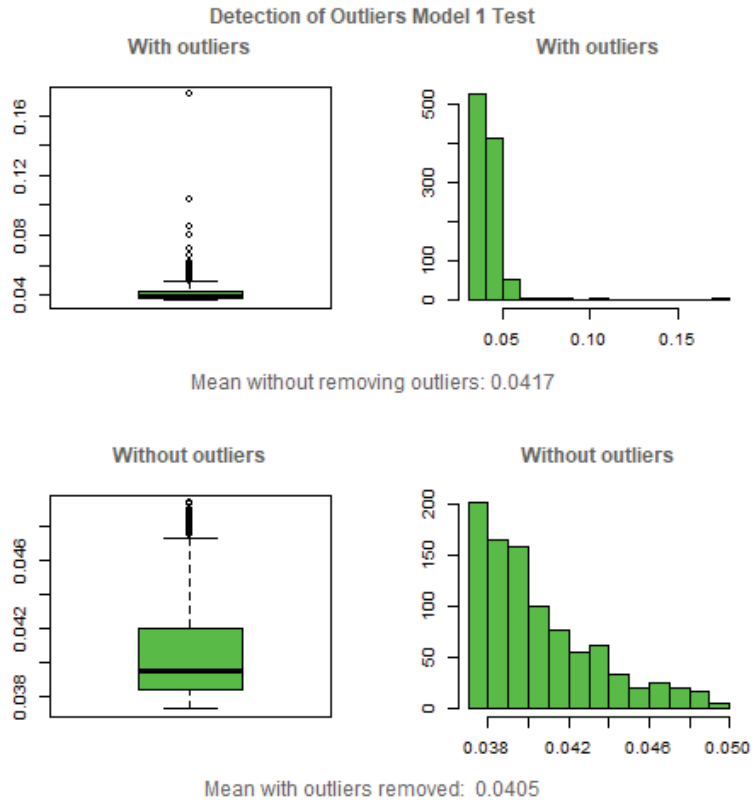

Figure 46- Outliers Analysis on Fitness Test of Model 1, Genetic Programming

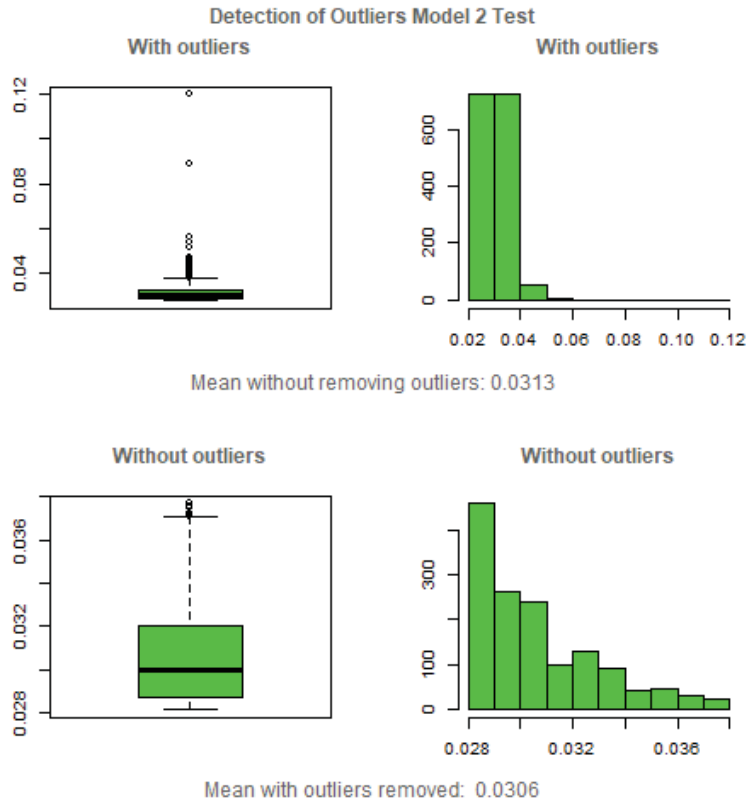

Figure 47- Outliers Analysis on Fitness Test of Model 2, Genetic Programming

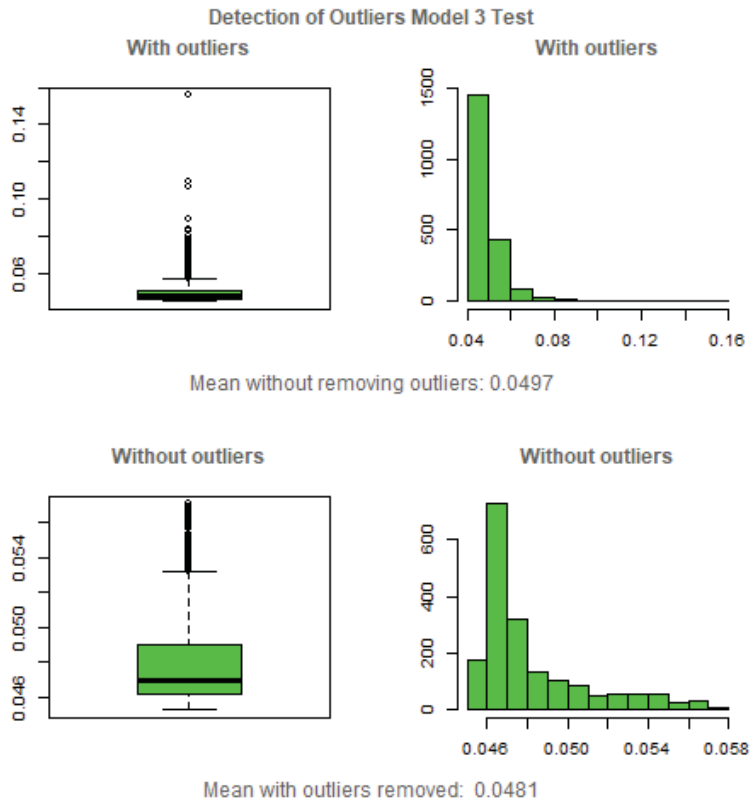

Figure 48- Outliers Analysis on Fitness Test of Model 3, Genetic Programming

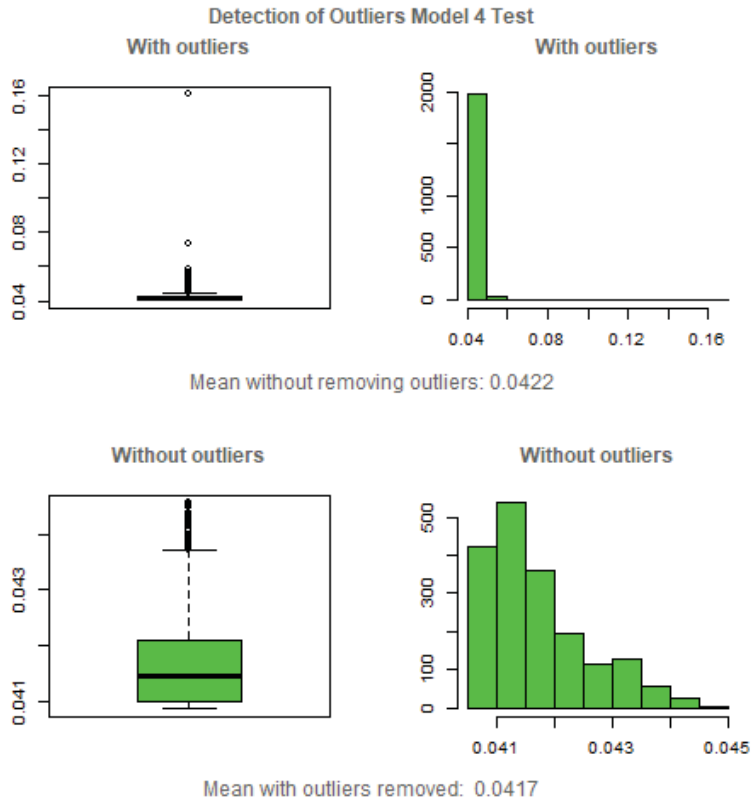

Figure 49- Outliers Analysis on Fitness Test of Model 4, Genetic Programming

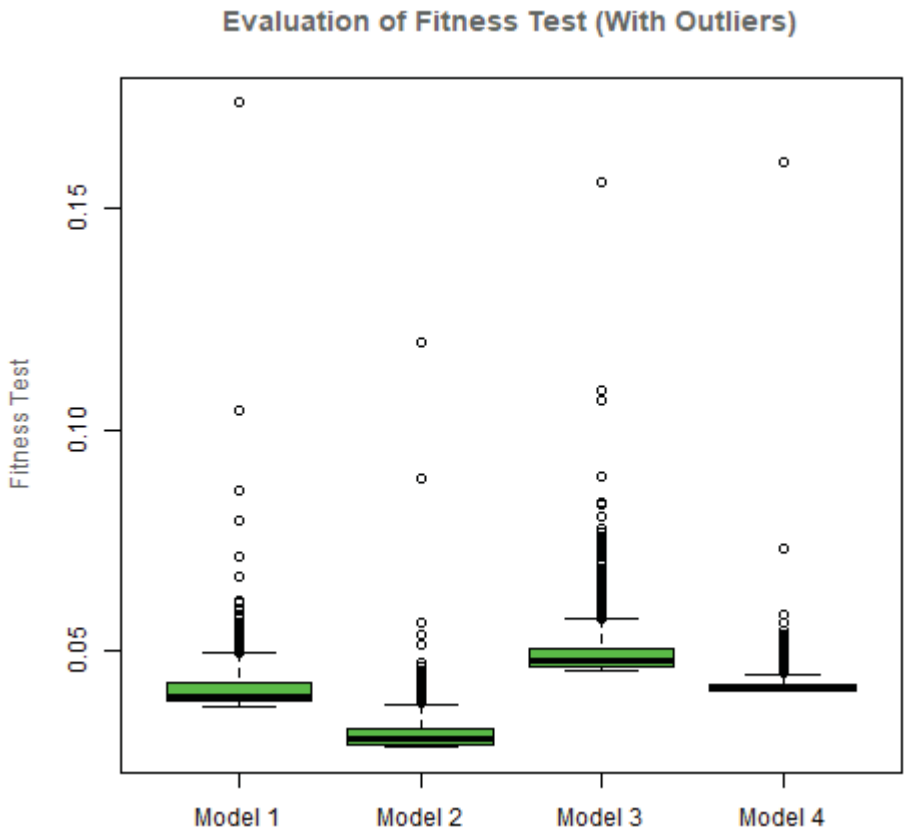

Figure 50- Boxplot with Fitness Test distribution of models (With Outliers), Genetic Programming

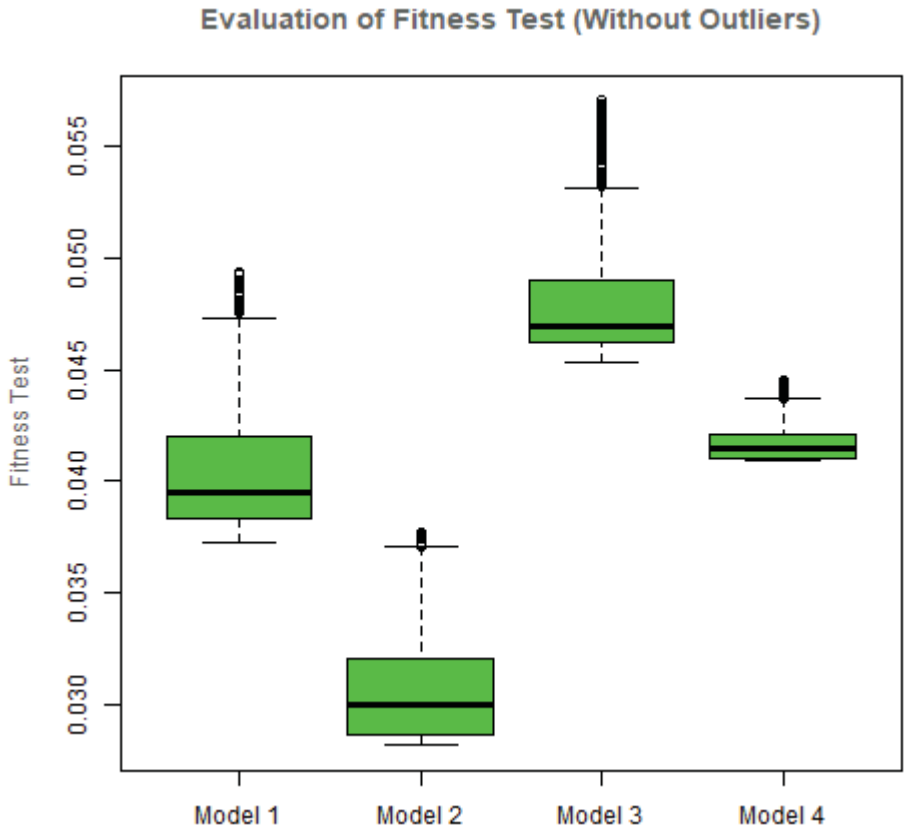

Figure 51- Boxplot with Fitness Test distribution of models (Without Outliers), Genetic Programming

9.3.3. Genetic Programming- Execution Time Analysis

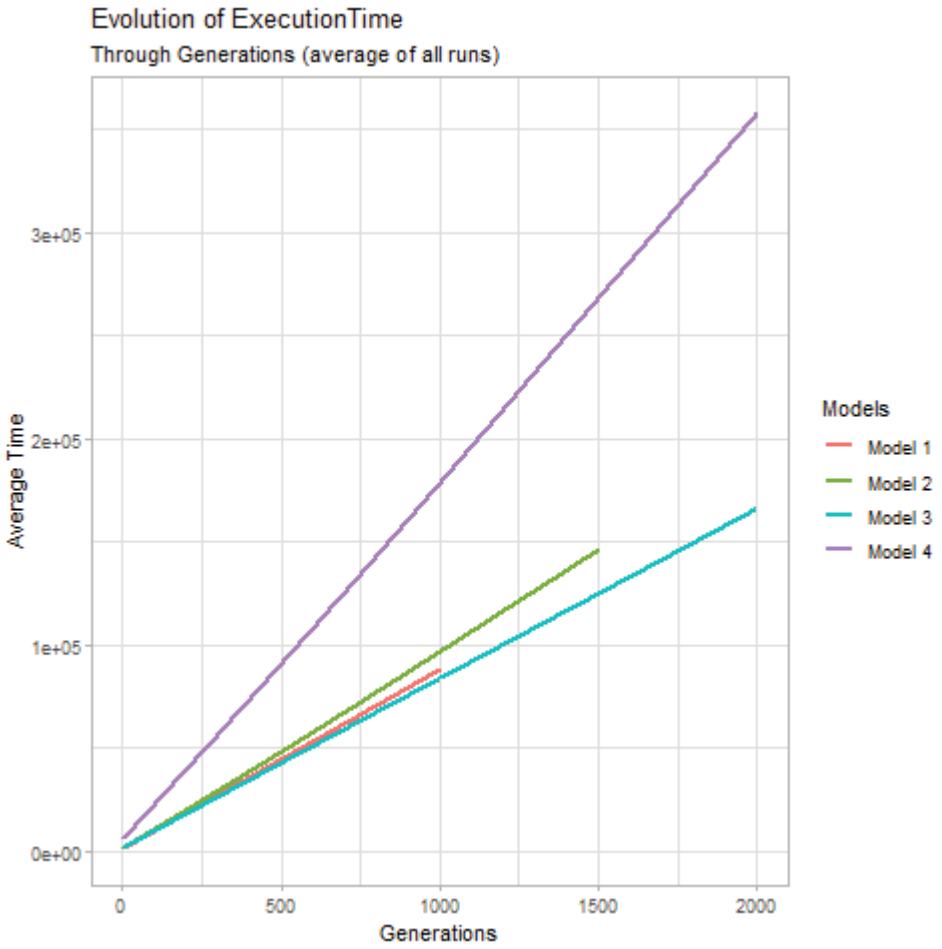

Figure 52- Evolution of Execution Time through Generations (average of all runs) of Genetic Programming Models

9.4. LONG SHORT-TERM MEMORY

9.4.1. Long Short-Term Memory- Fitness Train Models

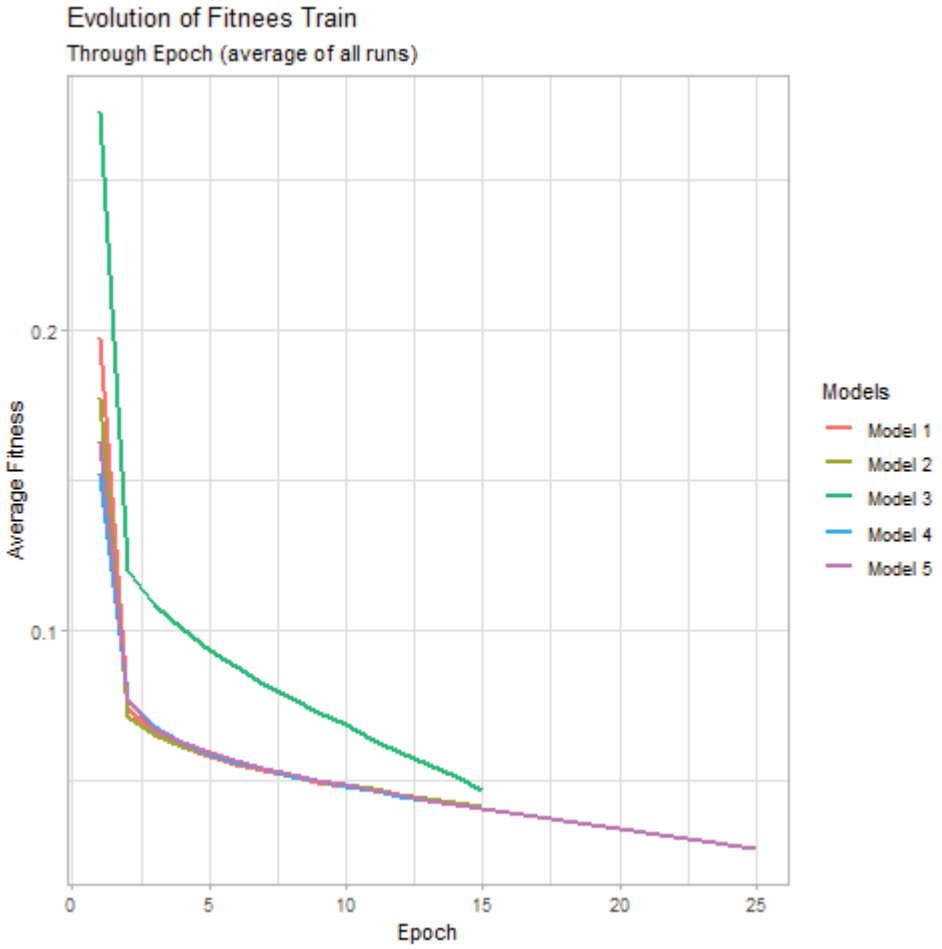

Figure 53- Evolution of Fitness Train through Epochs (average of all runs) of Long Short-Term Memory Models

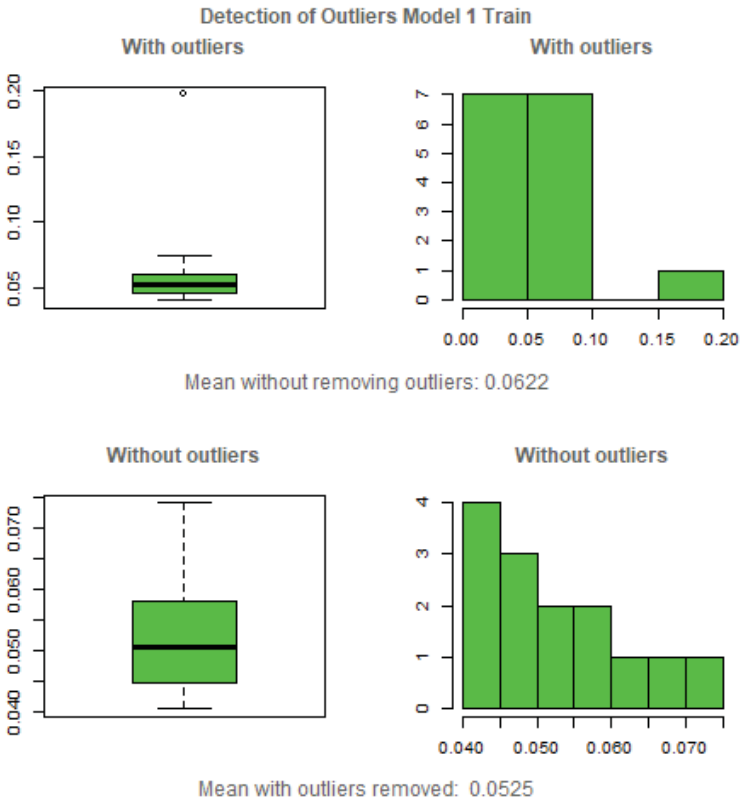

Figure 54- Outliers Analysis on Fitness Train of Model 1, Long Short-Term Memory

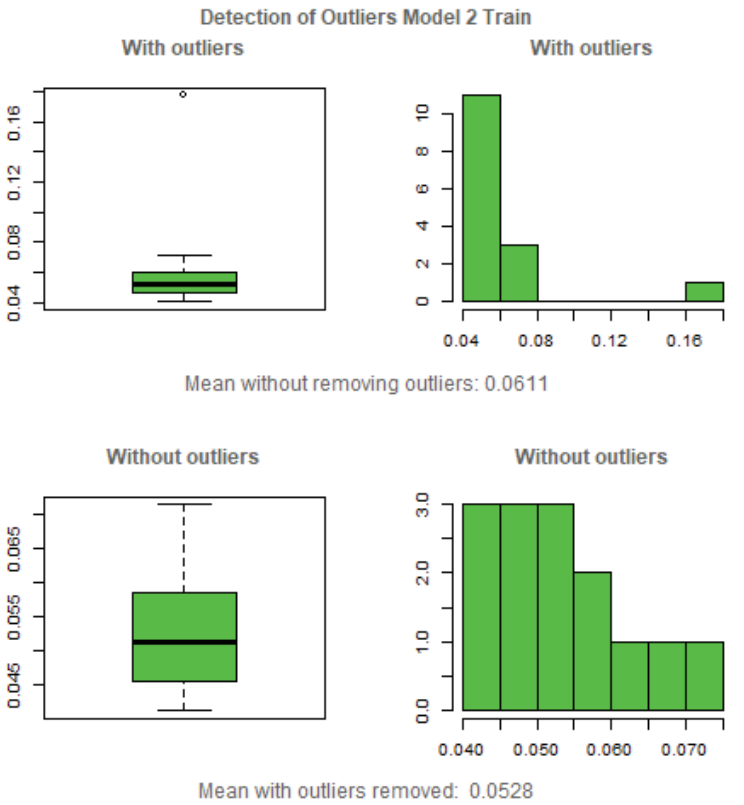

Figure 55- Outliers Analysis on Fitness Train of Model 2, Long Short-Term Memory

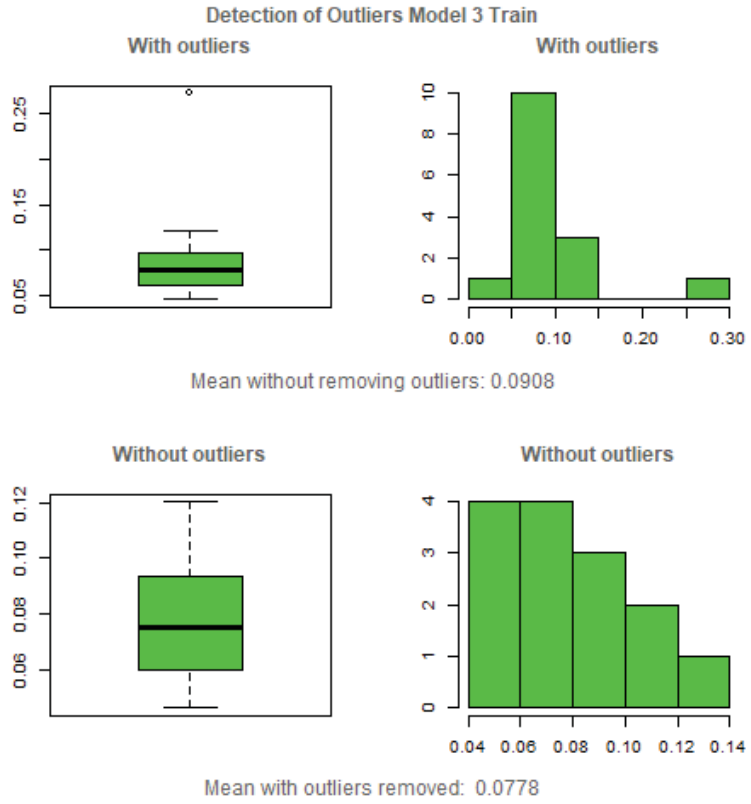

Figure 56- Outliers Analysis on Fitness Train of Model 3, Long Short-Term Memory

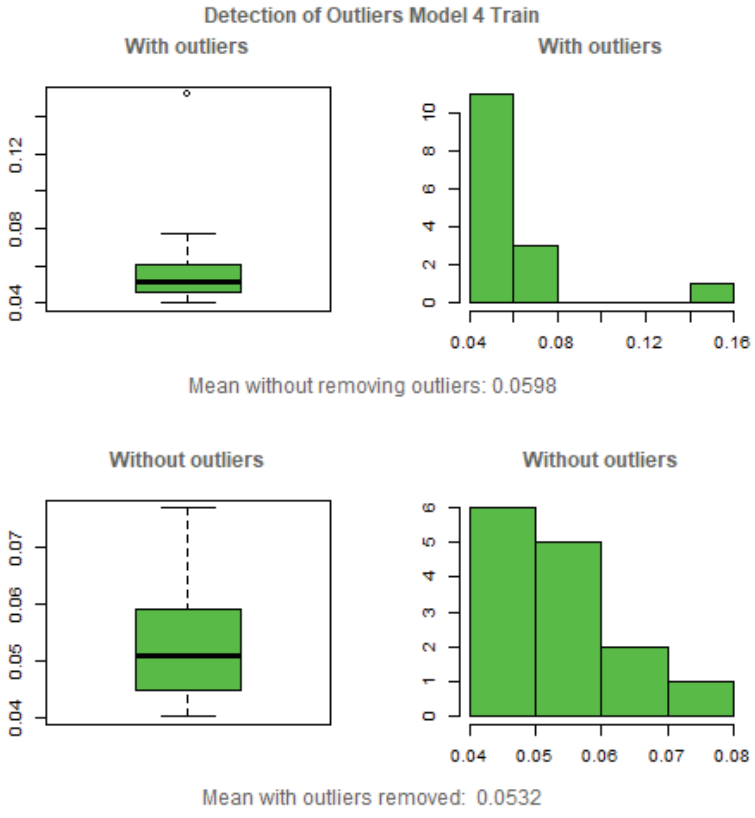

Figure 57- Outliers Analysis on Fitness Train of Model 4, Long Short-Term Memory

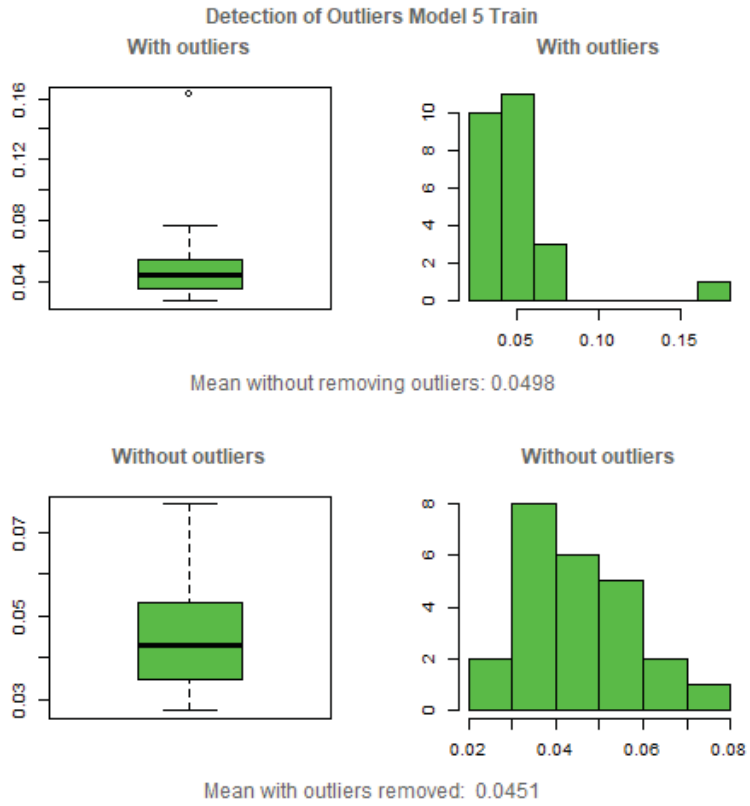

Figure 58- Outliers Analysis on Fitness Train of Model 5, Long Short-Term Memory

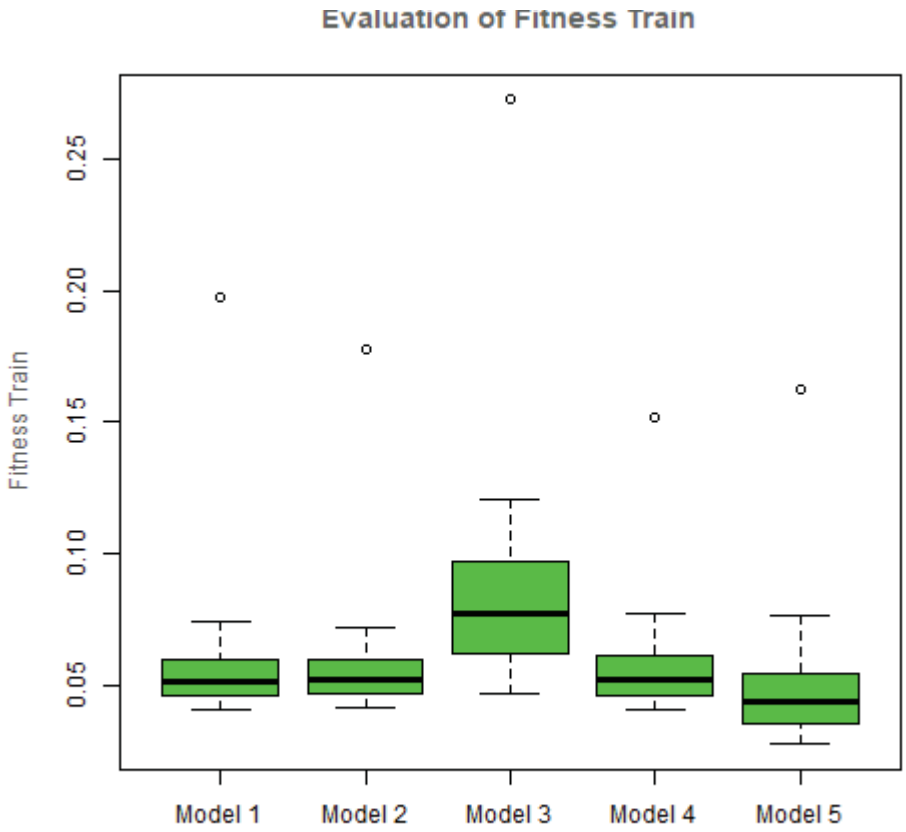

Figure 59- Boxplot with Fitness Train distribution of models, Long Short-Term Memory

9.4.2. Long Short-Term Memory- Fitness Validation

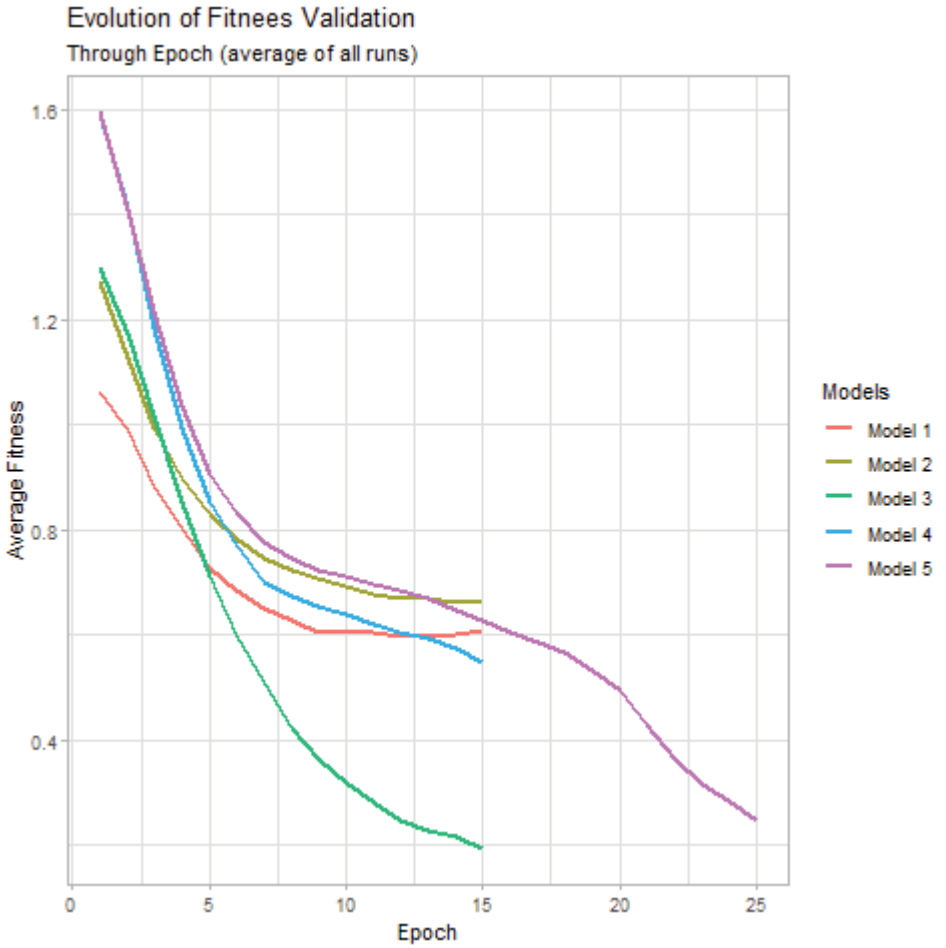

Figure 60- Evolution of Fitness Validation through Epochs (average of all runs) of Long Short-Term Memory Models

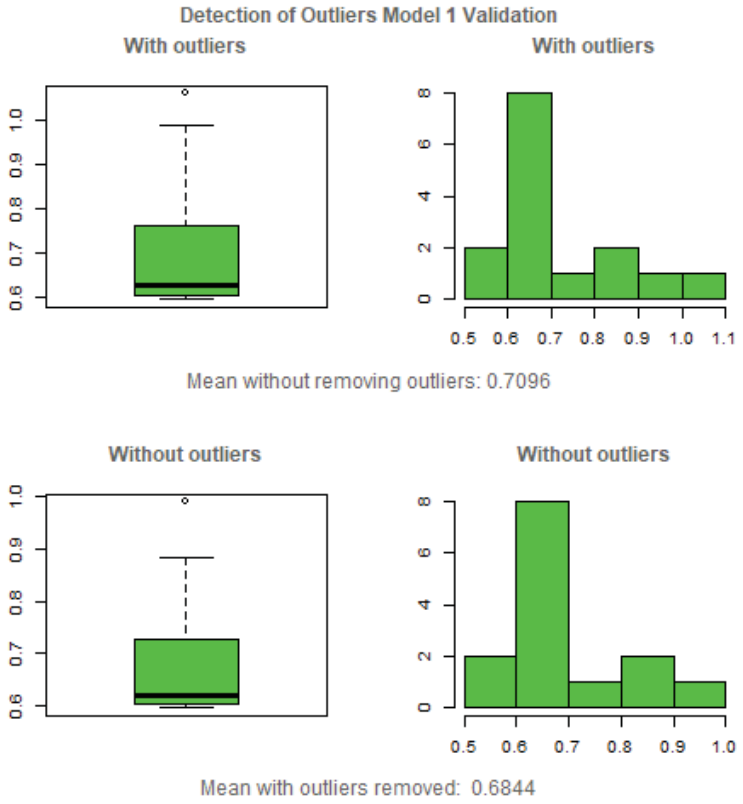

Figure 61- Outliers Analysis on Fitness Validation of Model 1, Long Short-Term Memory

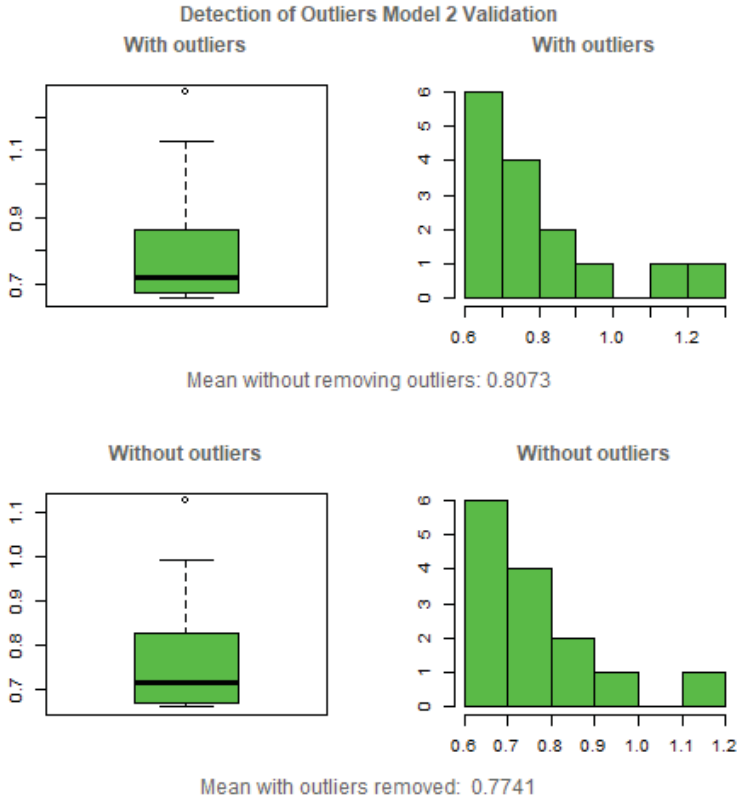

Figure 62- Outliers Analysis on Fitness Validation of Model 2, Long Short-Term Memory

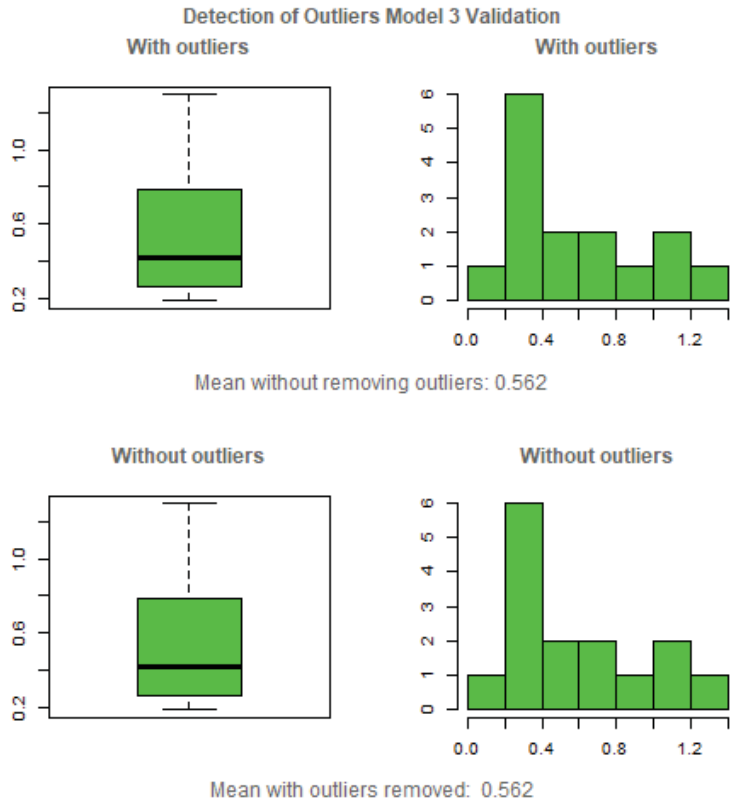

Figure 63- Outliers Analysis on Fitness Validation of Model 3 Long Short-Term Memory

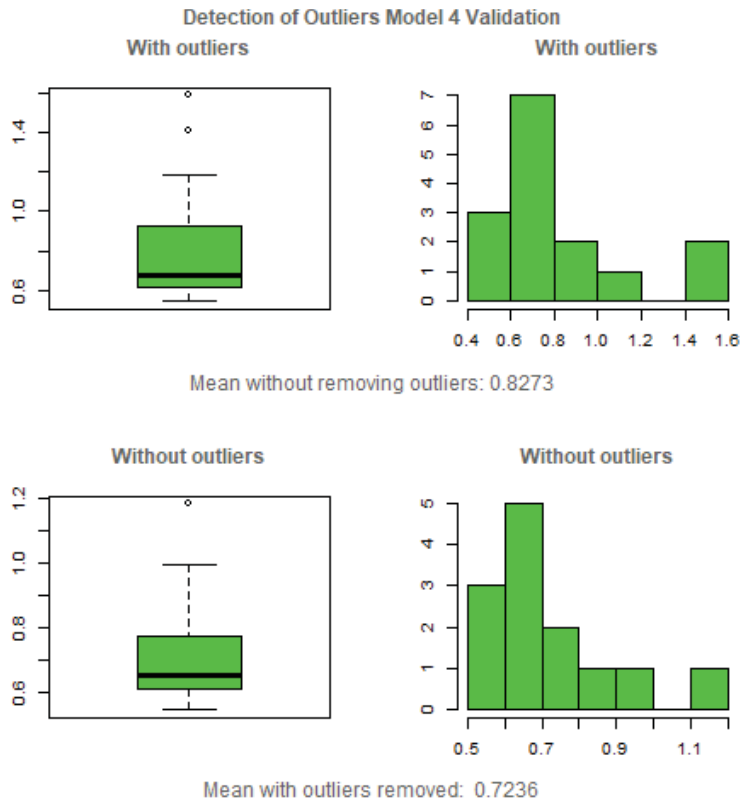

Figure 64- Outliers Analysis on Fitness Validation of Model 4, Long Short-Term Memory

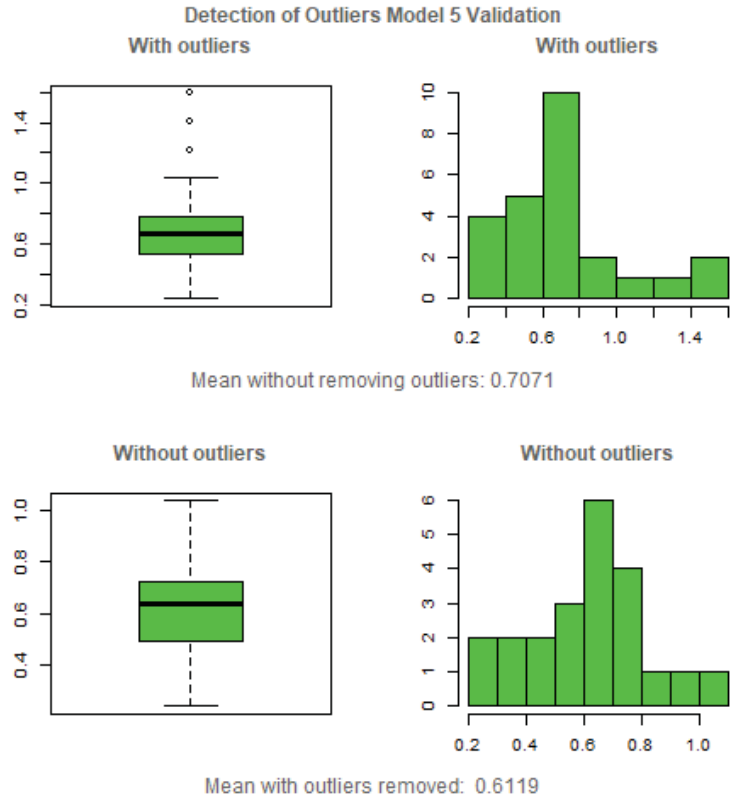

Figure 65- Outliers Analysis on Fitness Validation of Model 5, Long Short-Term Memory

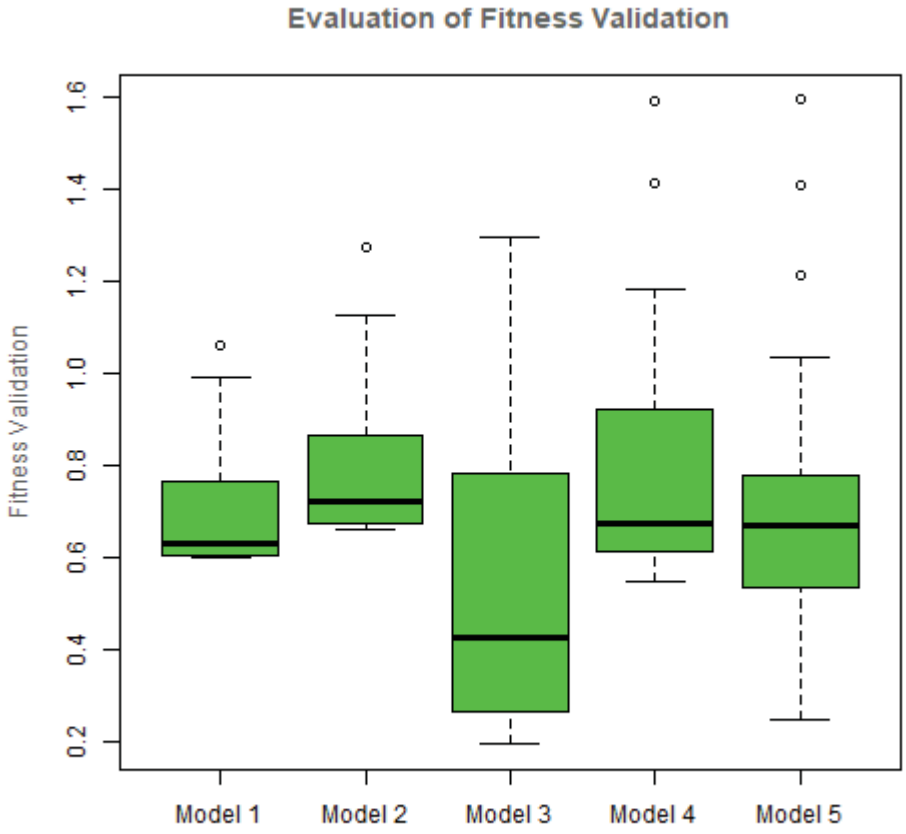

Figure 66- Boxplot with Fitness Validation distribution of models, Long Short-Term Memory

9.4.3. Long Short-Term Memory- Evolution by Model

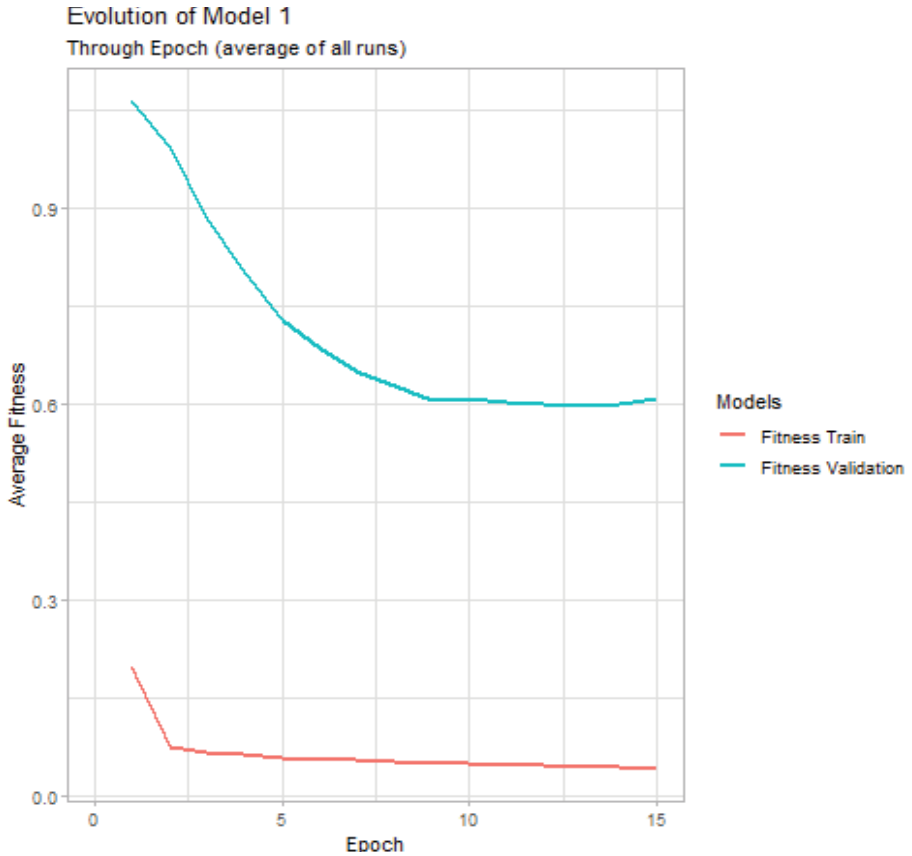

Figure 67- Evolution of Fitness Error in Model 1, Long Short-Term Memory

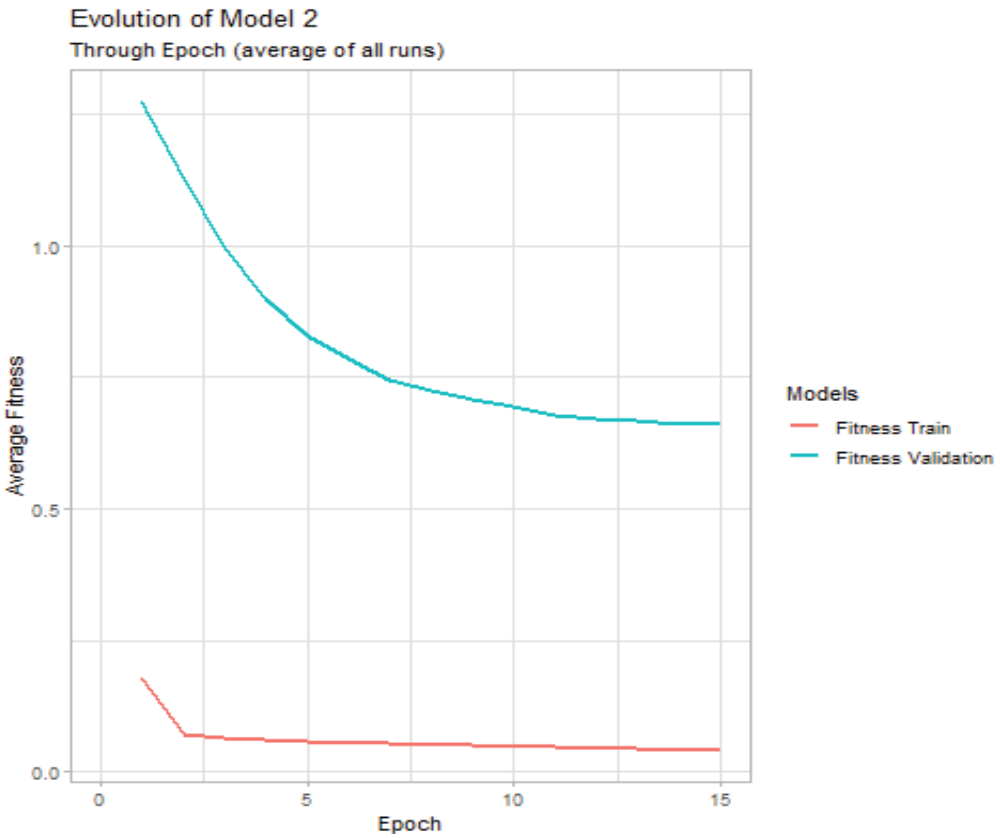

Figure 68- Evolution of Fitness Error in Model 2, Long Short-Term Memory

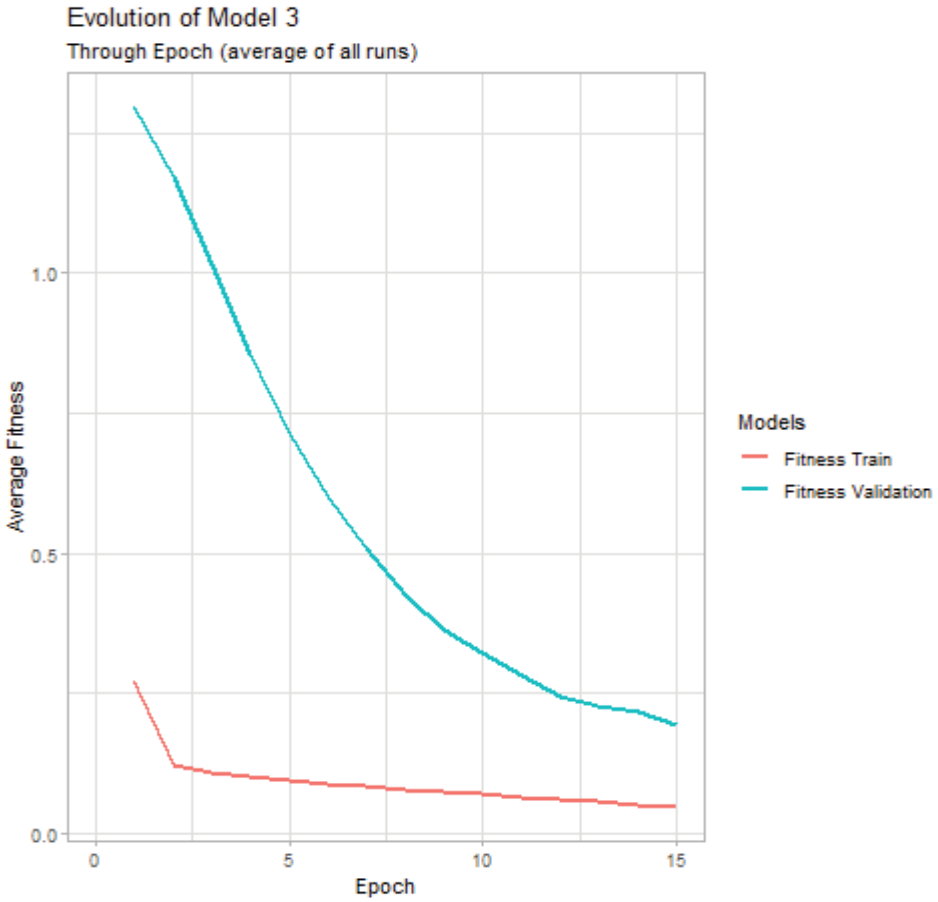

Figure 69- Evolution of Fitness Error in Model 3, Long Short-Term Memory

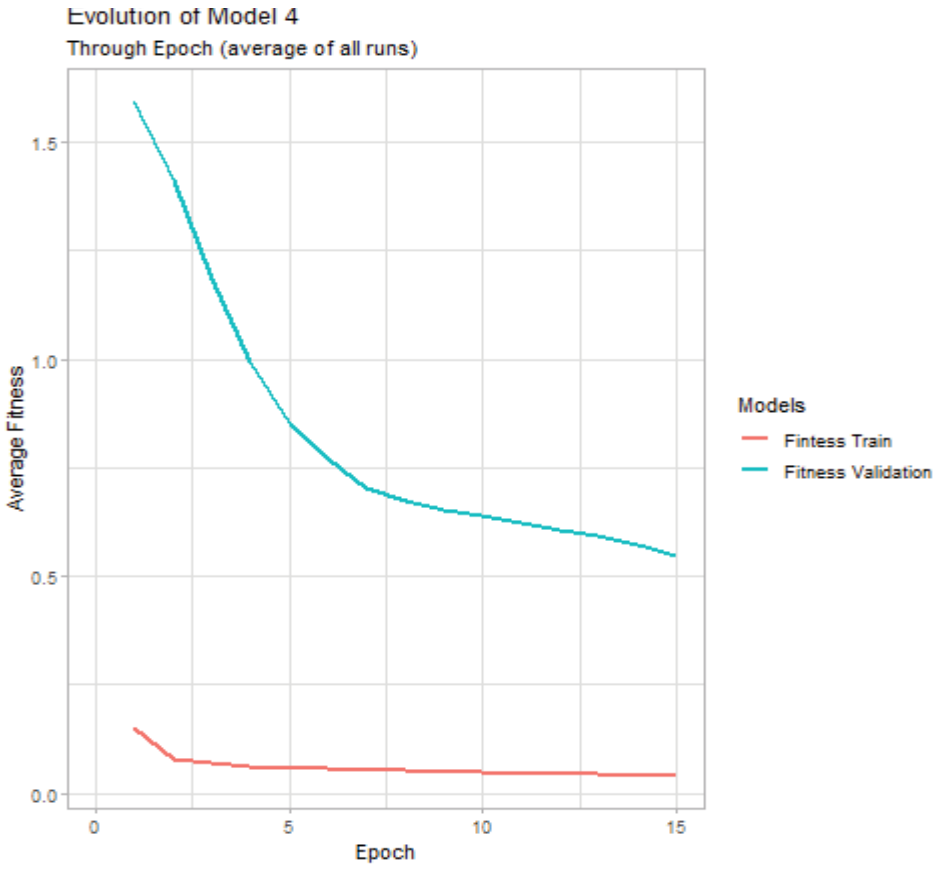

Figure 70- Evolution of Fitness Error in Model 4, Long Short-Term Memory

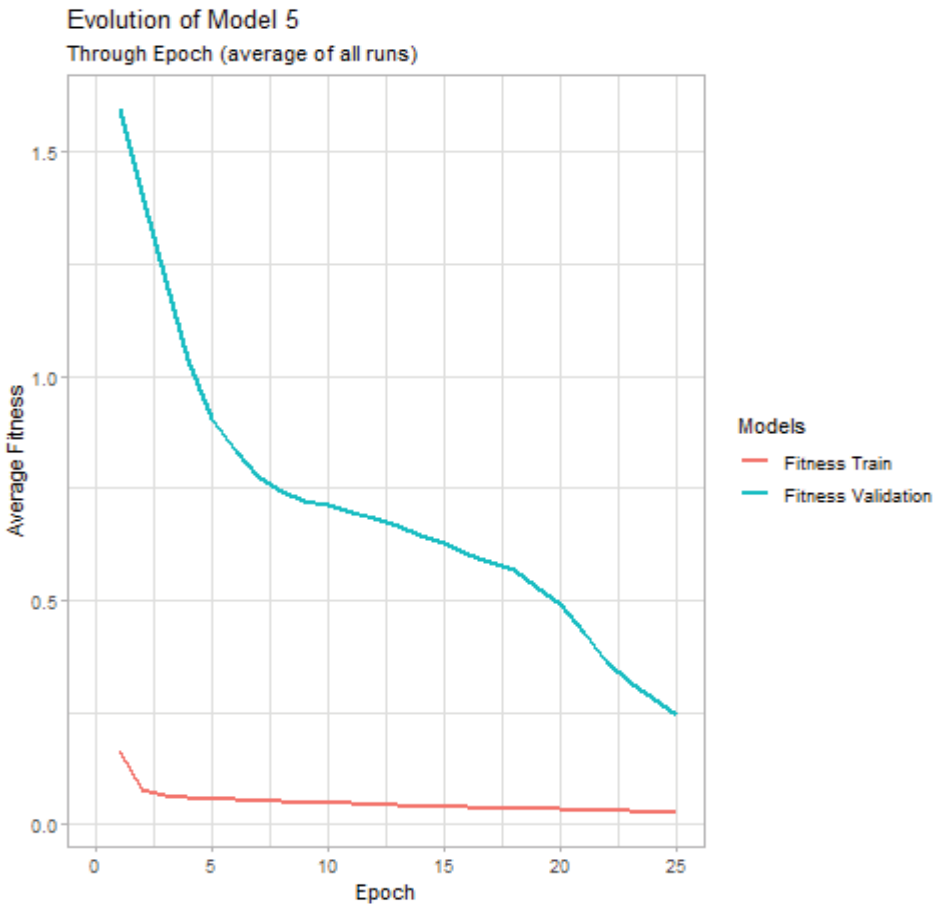

Figure 71- Evolution of Fitness Error in Model 5, Long Short-Term Memory

9.4.4. Long Short-Term Memory- Statistical Test

Wilcoxon Rank-Sum Test										
	Model 1		Model 2		Model 3		Model 4		Model 5	
	Train	Test	*Train*	Test	*Train*	Test	*Train*	Test	*Train*	Test
Model 2	0,903	0,026	-	-	-	-	-	-	-	-
Model 3	0,002	0,074	0,003	0,030	-	-	-	-	-	-
Model 4	0,903	0,486	0,935	0,285	0,003	0,030	-	-	-	-
Model 5	0,057	0,699	0,033	0,083	< 0,001	0,106	0,053	0,319	-	-

Table 12- Wilcoxon Rank-Sum Test to Long Short-Term Memory Models Errors

9.5. COMPARATION BETWEEN MODELS

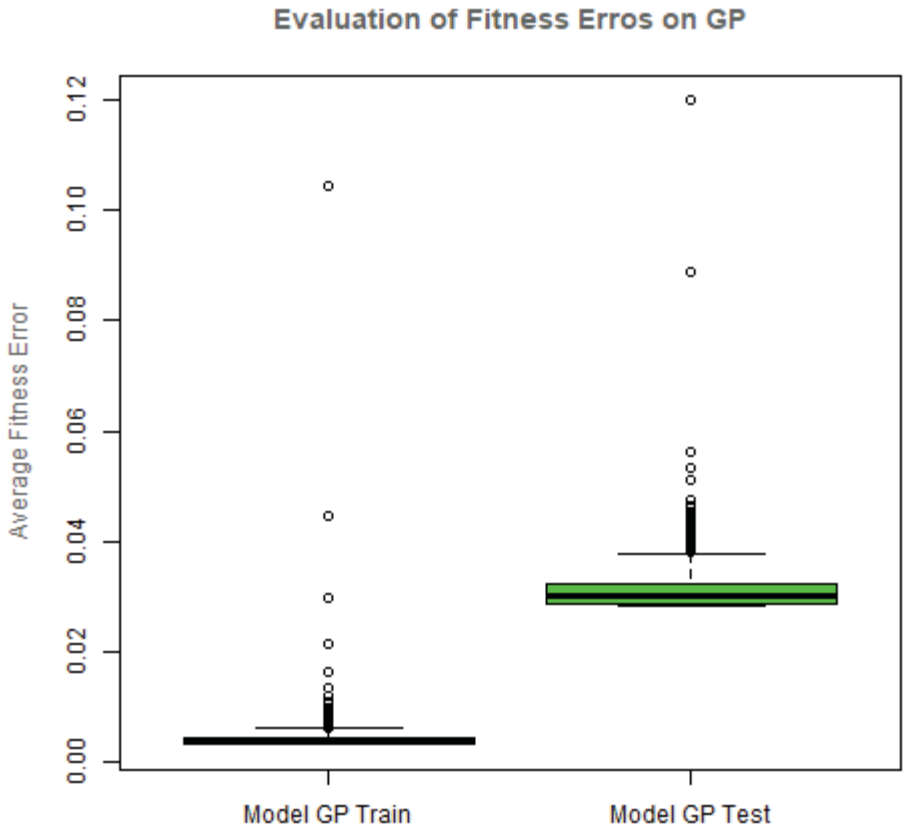

Figure 72- Boxplot with Fitness Erros of Genetic Programming

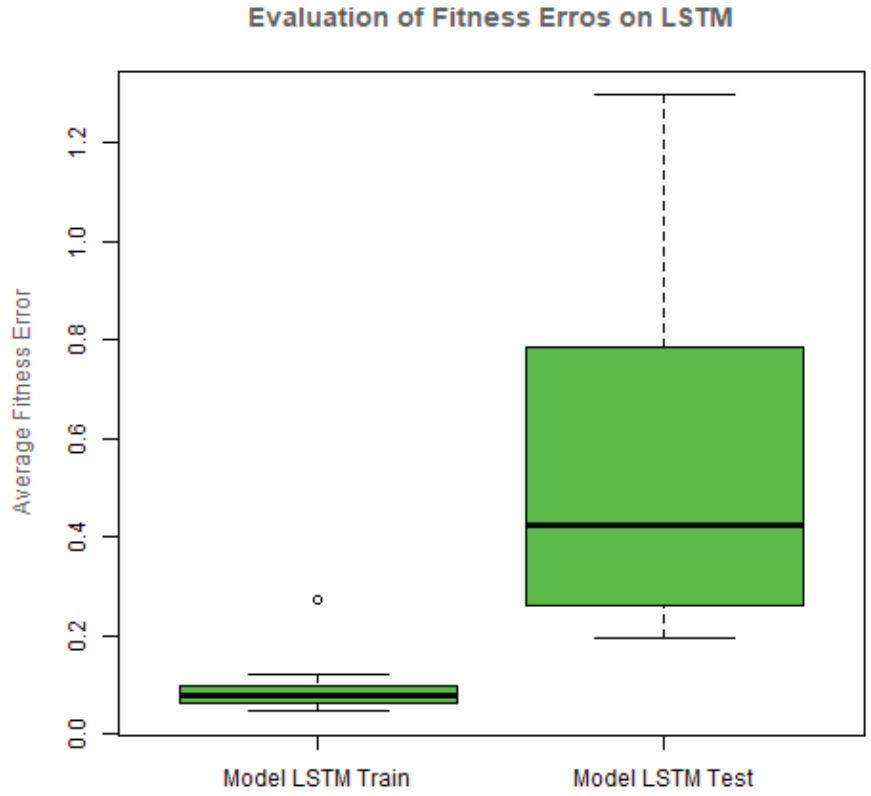

Figure 73- Boxplot with Fitness Erros of Long Short-Term Memory