USING NEURAL NETWORKS AND GENETIC ALGORITHMS TO PREDICT STOCK MARKET RETURNS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF MASTER OF SCIENCE
IN ADVANCED COMPUTER SCIENCE
IN THE FACULTY OF SCIENCE AND ENGINEERING

By

Efstathios Kalyvas

Department Of Computer Science

October 2001

Contents

Abstract

In this study we attempt to predict the daily excess returns of FTSE 500 and S&P 500 indices over the respective Treasury Bill rate returns. Initially, we prove that the excess returns time series do not fluctuate randomly. Furthermore we apply two different types of prediction models: Autoregressive (AR) and feed forward Neural Networks (NN) to predict the excess returns time series using lagged values. For the NN models a Genetic Algorithm is constructed in order to choose the optimum topology. Finally we evaluate the prediction models on four different metrics and conclude that they do not manage to outperform significantly the prediction abilities of naïve predictors.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright and Ownership

Acknowledgments

I would like to express my thanks and appreciation to my supervisor, Professor David S. Brée, for his valuable advice and guidance, and my gratitude to senior Lecturer Nathan L. Joseph, for his continuous support and assistance. I would also like to thank Rahim Lakka (Ph.D. Student) for his help and enlightening comments.

I need as well to thank Archimandrite Nikiforo Asprogeraka for his psychological and financial support. Last but not least I would like to thank my University teachers Panagioti Rodogiani and Leonida Palio for their help and advice at the initial stage of my postgraduate studies.

Without the help of all these people none of the current work would have been feasible.

«... åéò ôïõò ãïíåßò ìïõ ùöåßëù ôï æåßí åéò ôïõò äÜóêáëïõò ìïõ ôï åõ æåßí.»

Dedication

To my parents Petros and Maria, who believed in me and stood by my side all the way and my sister Sophia and brother Vassilis, my most precious friends.

To Myrto, who made every single moment unique.

Chapter 1

Introduction

It is nowadays a common notion that vast amounts of capital are traded through the Stock Markets all around the world. National economies are strongly linked and heavily influenced of the performance of their Stock Markets. Moreover, recently the Markets have become a more accessible investment tool, not only for strategic investors but for common people as well. Consequently they are not only related to macroeconomic parameters, but they influence everyday life in a more direct way. Therefore they constitute a mechanism which has important and direct social impacts.

The characteristic that all Stock Markets have in common is the *uncertainty*, which is related with their short and long-term future state. This feature is undesirable for the investor but it is also unavoidable whenever the Stock Market is selected as the investment tool. The best that one can do is to try to reduce this uncertainty. *Stock Market Prediction* (or *Forecasting*) is one of the instruments in this process.

1.1 Aims and Objectives

The aim of this study is to attempt to predict the short-term term future of the Stock Market. More specifically prediction of the returns provided by the Stock Market on daily basis is attempted. The Stock Markets indices that are under consideration are the FTSE 500 and the S&P 500 of the London and New York market respectively.

The first objective of the study is to examine the feasibility of the prediction task and provide evidence that the markets are not fluctuating randomly. The second objective is, by reviewing the literature, to apply the most suitable prediction models and measure their efficiency.

1.2 Rationale

There are several motivations for trying to predict the Stock Market. The most basic of these is the financial gain. Furthermore there is the challenge of proving whether the markets are predictable or not. The predictability of the market is an issue that has been much discussed by researchers and academics. In finance a hypothesis has been formulated, known as the *Efficient Market Hypothesis* (EMH), which implies that there is no way to make profit by predicting the market, but so far there has been no consensus on the validity of EMH [1].

1.3 Stock Market Prediction

The Stock Market prediction task divides researchers and academics into two groups those who believe that we can devise mechanisms to predict the market and those who believe that the market is efficient and whenever new information comes up the market absorbs it by correcting itself, thus there is no space for prediction (EMH). Furthermore they believe that the Stock Market follows a Random Walk, which implies that the best prediction you can have about tomorrow's value is today's value.

In literature a number of different methods have been applied in order to predict Stock Market returns. These methods can be grouped in four major categories: i) Technical Analysis Methods, ii) Fundamental Analysis Methods, iii) Traditional Time Series Forecasting and iv) Machine Learning Methods. Technical analysts, known as chartists, attempt to predict the market by tracing patterns that come from the study of charts which describe historic data of the market. Fundamental analysts study the intrinsic value of an stock and they invest on it if they estimate that its current value is lower that its intrinsic value. In Traditional Time Series forecasting an attempt to create linear prediction models to trace patterns in historic data takes place. These linear models are divided in two categories: the univariate and the multivariate regression models, depending on whether they use one of more variables to approximate the Stock Market time series. Finally a number of methods have been developed under the common label Machine Learning these methods use a set of samples and try to trace patterns in it (linear or non-linear) in order to approximate the underlying function that generated the data.

The level of success of these methods varies from study to study and it is depended on the underlying datasets and the way that these methods are applied each time. However none of them has been proven to be the consistent prediction tool that the investor would like to have. In this study our attention is concentrated to the last two categories of prediction methods.

1.4 Organization of the Study

The complementation of the aims and objectives of this study as described earlier takes place throughout five chapters. Here we present a brief outline of the content of each chapter:

In Chapter 2, initially an attempt to define formally the prediction task takes place. In order to be able to predict the market we have to be certain that it is not fluctuating randomly. We search the relevant literature to find out whether there are studies, which prove that the Stock Market does not fluctuate randomly and in order to see which are the methods that other studies have used so far to predict the market as well as their level of success and we present our findings. In the last part of this chapter we select, based on our literature review, the prediction models and the type of data we will use to predict the market on daily basis.

Chapter 3 presents in detail the datasets we will use: the FTSE 500 and S&P 500. Firstly it presents the initial data sets we obtained and covers issues such as: source, descriptive statistics, quality, etc. Secondly it describes the way that we integrate these datasets in order to construct the time series under prediction (*excess returns time series*). In the last part of Chapter 3 two distinct randomness tests are presented and applied to the *excess returns time series*. The tests are: a) the Run and b) the BDS test.

In Chapter 4, we present in detail the models we will apply in this study: the autoregressive (AR) and the feed-forward neural network (NN) models. For each category of model firstly, a description of how they function is given; then the parameters that influence their performance are presented and analysed. Additionally we attempt to set these parameters in such a way that the resulting models will perform optimally in the frame of our study. To accomplish this, we use the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) to define the lag structure of the AR models; for the NN models we choose a number of the parameters based on findings of other studies and use a Genetic Algorithm (GA) to find the optimum topology. Finally we evaluate these models using four different metrics. Three of these are benchmarks that compare the prediction abilities of our models with naï ve prediction models, while the last one is the mean absolute prediction error.

In Chapter 5, two major experiments are reported. These experiments use the models described in the previous chapter. Experiment I applies AIC and BIC and determines the optimum lags, for the AR models. These models are applied to predict the *excess returns time series* and then their performance is evaluated on all four metrics. Experiment II initially applies the GA to find the optimum topology for the NNs models. Then it evaluates the performance of the resulted NN models on all four different metrics. For the adjustment of the parameters of both categories of models, as well as for their evaluation, the same data sets are used to enable a comparison to be made.

Chapter 6, summarizes the findings of this study as well as the conclusions we have drawn. Finally it presents some of our suggestions for future work on the field of Stock Market prediction.

Chapter 2

Stock Markets and Prediction

This chapter attempts to give a brief overview of some of the theories and concepts that are linked to stock markets and their prediction. Issues such as investment theories, identification of available data related to the market, predictability of the market, prediction methodologies applied so far and their level of success are some of the topics covered. All these issues are examined under the ‘daily basis prediction’ point of view with the objective of incorporating in our study the most appropriate features.

2.1 The Stock Market

2.1.1 Investment Theories

An investment theory suggests what parameters one should take into account before placing his (or her) capital on the market. Traditionally the investment community accepts two major theories: the *Firm Foundation* and the *Castles in the Air* [1]. Reference to these theories allows us to understand how the market is shaped, or in other words how the investors think and react. It is this sequence of ‘thought and reaction’ by the investors that defines the capital allocation and thus the level of the market.

There is no doubt that the majority of the people related to stock markets is trying to achieve profit. Profit comes by investing in stocks that have a good future (short or long term future). Thus what they are trying to accomplish one way or the other is to predict the future of the market. But what determines this future? The way that people invest their money is the answer; and people invest money based on the information they hold. Therefore we have the following schema:

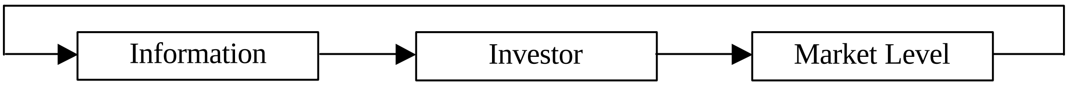


Figure 2.1: Investment procedure.

The factors that are under discussion on this schema are: the content of the 'Information' component and the way that the 'Investor' reacts when having this info.

According to the *Firm Foundation* theory the market is defined from the reaction of the investors, which is triggered by information that is related with the 'real value' of firms. The 'real value' or else the *intrinsic value* is determined by careful analysis of present conditions and future prospects of a firm [1].

On the other hand, according to the *Castles in the Air* theory the investors are triggered by information that is related to other investors' behavior. So for this theory the only concern that the investor should have is to buy today with the price of 20 and sell tomorrow with the price of 30, no matter what the intrinsic value of the firm he (or she) invests on is.

Therefore the *Firm Foundation* theory favors the view that the market is defined mostly by logic, while the *Castles in the Air* theory supports that the market is defined mostly by psychology.

2.1.2 Data Related to the Market

The information about the market comes from the study of relevant data. Here we are trying to describe and group into categories the data that are related to the stock markets. In the literature these data are divided in three major categories [2]:

- *Technical data*: are all the data that are referred to stocks only. Technical data include:
  - The price at the end of the day.

    - The highest and the lowest price of a trading day.
    - The volume of shares traded per day.
- *Fundamental data*: are data related to the intrinsic value of a company or category of companies as well as data related to the general economy. Fundamental data include:
    - Inflation
    - Interest Rates
    - Trade Balance
    - Indexes of industries (e.g. heavy industry)
    - Prices of related commodities (e.g. oil, metals, currencies)
    - Net profit margin of a firm.
    - Prognoses of future profits of a firm
    - Etc.
- *Derived data*: this type of data can be produced by transforming and combining technical and/or fundamental data. Some commonly used examples are:
    - Returns: One-step returns R(t) is defined as the relative increase in price since the previous point in a time series. Thus if y(t) is the value of a stock on day t, R(t)= $\frac{y(t)-y(t-1)}{y(t-1)}$.
    - Volatility: Describes the variability of a stock and is used as a way to measure the risk of an investment.

The study (process) of these data permit us to understand the market and some of the rules it follows. In our effort to predict the future of the market we have to study its past and present and infer from them. It is this inference task that all prediction methods are trying to accomplish. The way they do it and the different subsets of data they use is what differentiates them.

2.2 Prediction of the Market

2.2.1 Defining the prediction task

Before having any further discussion about the prediction of the market we define the task in a formal way.

> "Given a sample of N examples $\{(x_i, y_i), i=1, \ldots, N\}$ where $f(x_i)= y_i$, $\forall i$, return a function g that approximates f in the sense that the norm of the error vector $E=(e_1,\ldots,e_N)$ is minimized. Each e_i is defined as $e_i=e(g(x_i), y_i)$ where e is an arbitrary error function"[2].

In other words the definition above indicates that in order to predict the market you should search historic data and find relationships between these data and the value of the market. Then try to exploit these relationships you have found on future situations. This definition is based on the assumption that such relationships do exist. But do they? Or do the markets fluctuate in a totally random way leaving us no space for prediction? This is a question that has to be answered before any attempt for prediction is made.

2.2.2 Is the Market predictable?

The predictability of the market is an issue that has been discussed a lot by researchers and academics. In finance a hypothesis has been formulated known as the *Efficient Market Hypothesis* (EMH), which implies that there is no way to make profit by predicting the market. The EMH states that all the information relevant to a market is contained in the prices and each time that new information arises the market corrects itself and absorbs it, in other words the market is efficient, therefore there is no space for prediction. More specifically the EMH has got three forms [1]:

- *Weak*: States that you cannot predict future stock prices on the basis of past stock prices.
- *Semi-Strong*: States that you cannot even utilize published information to predict future prices.
- *Strong*: Claims that you cannot predict the market no matter what information you have available.

According to the above the market fluctuations are based on the *'Random Walk'* model. Which more formally stated is equivalent to:

$$y(t)=y(t-1) + rs$$

where y(t) is the value of the market on time t and rs is an *Independent and Identically Distributed* (IID)[1] variable. If we accept the validity of this model we imply that the best prediction that you can have about tomorrow's value is today's value.

[1] IID implies randomness.

Research has been done on the data of stock markets in order to prove that the market is predictable. Hsieh (1991) proved for the S&P 500 that the weekly returns from 1962 until 1989, the daily returns from 1983 until 1989 and the 15 minutes returns during 1988 are not IDD [3]. Tsibouris and Zeidenberg (1996) tested the weak form of EMH by using daily returns of stocks from U.S. stock market (from 1988 until 1990) and they did manage to find evidence against it [4]. White (1993) did not manage to find enough evidence to reject the EMH when he tried to predict the IBM stock returns on daily basis using data from 1972 to 1980 [5].

The conclusion from the results of these studies is that there is no clear evidence whether the market is predictable or not. We have an indication that the daily returns (for the S&P 500) in which we are interested in are not randomly distributed (at least from the period from 1983 until 1989). Therefore the methodology that we use in this study is to test the time series that we are attempting to predict for randomness. If proven non-random we will proceed with the implementation of prediction models. At this point we have to make clear that non-randomness does not imply that no matter what prediction model you will apply you will manage to predict the market successfully; all it states is that the prediction task is not impossible.

2.2.3 Prediction Methods

The prediction of the market is without doubt an interesting task. In the literature there are a number of methods applied to accomplish this task. These methods use various approaches, ranging from highly informal ways (e.g. the study of a chart with the fluctuation of the market) to more formal ways (e.g. linear or non-linear regressions). We have categorized these techniques as follows:

- Technical Analysis Methods,
- Fundamental Analysis Methods,
- Traditional Time Series Prediction Methods
- and Machine Learning Methods.

The criterion to this categorization is the type of tools and the type of data that each method is using in order to predict the market. What is common to these techniques is that they are used to predict and thus benefit from the market's future behavior. None of them has proved to be the consistently correct prediction tool that the investor would like to have. Furthermore many analysts question the usefulness of many of these prediction techniques.

2.2.3.1 Technical Analysis

“Technical analysis is the method of predicting the appropriate time to buy or sell a stock used by those believing in the castles-in-the-air view of stock pricing” (p. 119) [1]. The idea behind technical analysis is that share prices move in trends dictated by the constantly changing attributes of investors in response to different forces. Using technical data such as price, volume, highest and lowest prices per trading period the technical analyst uses charts to predict future stock movements. Price charts are used to detect trends, these trends are assumed to be based on supply and demand issues which often have cyclical or noticeable patterns. From the study of these charts trading rules are extracted and used in the market environment. The technical analysts are known and as ‘*chartists*’. Most chartists believe that the market is only *10 percent logical and 90 percent psychological* [1]. The chartist’s belief is that a careful study of what the other investors are doing will shed light on what the crowed is likely to do in the future.

This is a very popular approach used to predict the market, which has been heavily criticized. The major point of criticism is that the extraction of trading rules from the study of charts is highly subjective therefore different analysts might extract different trading rules by studying the same charts. Although it is possible to use this methodology to predict the market on daily basis we will not follow this approach on this study due to its subjective character.

2.2.3.2 Fundamental Analysis

‘Fundamental analysis is the technique of applying the tenets of the firm foundation theory to the selection of individual stocks”[1]. The analysts that use this method of prediction use fundamental data in order to have a clear picture of the firm (industry or market) they will choose to invest on. They are aiming to compute the ‘real’ value of the asset that they will invest in and they determine this value by studying variables such as the growth, the dividend payout, the interest rates, the risk of investment, the sales level, the tax rates an so on. Their objective is to calculate the intrinsic value of an asset (e.g. of a stock). Since they do so they apply a simple trading rule. *If the intrinsic* *value of the asset is higher than the value it holds in the market, invest in it. If not, consider it a bad investment and avoid it.* The fundamental analysts believe that the market is defined 90 percent by logical and 10 percent by physiological factors.

This type of analysis is not possible to fit in the objectives of our study. The reason for this is that the data it uses in order to determine the intrinsic value of an asset does not change on daily basis. Therefore fundamental analysis is helpful for predicting the market only in a long-term basis.

2.2.3.3 Traditional Time Series Prediction

The Traditional Time Series Prediction analyzes historic data and attempts to approximate future values of a time series as a linear combination of these historic data. In econometrics there are two basic types of time series forecasting: *univariate* (simple regression) and *multivariate* (multivariate regression)[6].

These types of regression models are the most common tools used in econometrics to predict time series. The way they are applied in practice is that firstly a set of factors that influence (or more specific is assumed that influence) the series under prediction is formed. These factors are *the explanatory variables* x_i of the prediction model. Then a mapping between their values x_{it} and the values of the time series y_t *(y is the to-be explained variable)* is done, so that pairs $\{x_{it} , y_t\}$ are formed. These pairs are used to define the importance of each *explanatory variable* in the formulation of the *to-be explained variable*. In other words the linear combination of x_i that approximates in an optimum way y is defined. *Univariate* models are based on one *explanatory variable* (I=1) while *multivariate* models use more than one variable (I>1).

Regression models have been used to predict stock market time series. A good example of the use of multivariate regression is the work of Pesaran and Timmermann (1994) [7]. They attempted prediction of the excess returns time series of S&P 500 and the Dow Jones on monthly, quarterly and annually basis. The data they used was from Jan 1954 until Dec 1990. Initially they used the subset from Jan 1954 until Dec 1959 to adjust the coefficients of the *explanatory variables* of their models, and then applied the models to predict the returns for the next year, quarter and month respectively.

Afterwards they adjusted their models again using the data from 1954 until 1959 plus the data of the next year, quarter or month. This way as their predictions were shifting in time the set that they used to adjust their models increased in size. The success of their models in terms of correct predictions of the sign of the market (*hit rate*) are presented in the next table:

	Period from 1960-1990	
	S&P 500	**Dow Jones**
Annually	80.6%	71.0%
Quarterly	62.1%	62.1%
Monthly	58.1%	57.3%

Table 2.1: Percentage of correct predictions of the regression models.

Moreover, they applied these models in conjunction with the following trading rule: *If you hold stocks and the model predicts for the next period of time (either month, quarter or year) negative excess returns sell the stocks and invest in bonds, else if the prediction is for positive returns keep the stocks. In case you hold bonds a positive prediction triggers a buying action while a negative prediction a hold action.* Their study took into consideration two scenarios one with and one without transaction costs. Finally they compared the investment strategy which used their models with a *buy and hold* strategy. The results they obtained (for the S&P500, for 1960 to 1990) are the following:

	Change of profits compared to a buy/hold strategy	
	No Transaction Cost	**High Transaction Cost**
Annually	1.9%	1.5%
Quarterly	2.2%	1.1%
Monthly	2.3%	-1.0%

Table 2.2: Comparison of the profits of the regression models with those of a buy/hold strategy.

The results for Dow Jones were similar to those above.

Initially they used four explanatory variables the dividend yields, the inflation rate, change in the industrial production, and the interest rates. They have computed the coefficients of their models and after studying the residuals of those models they discovered that they were not randomly distributed. This fact led them to add more explanatory variables (lagged rates of changes in the business cycle). They did manage to improve their models but still they had non-IID residuals. The final improvement they made was that they have used non-linear *explanatory variables* (lagged values of square returns) in an effort to capture non-linear patterns that might exist in the time series data, the results they had (Table 2.2) indicated that the annual regression did not improve while the quarterly and mostly the monthly regression did.

The conclusions we draw from this case study are the following:

- In order to make profit out of the market a prediction model is not enough, what you need is a prediction model in conjunction with a trading rule.
- Transaction costs play a very important role in this procedure. From table 2.2 it is clear that for the prediction on monthly basis presence of transaction costs cancel the usefulness of their model. It is rational that in our case of daily prediction the presence of the transaction cost will be more significant.
- The improvement they managed to give to their models by adding non-linear explanatory variables raises questions as to whether or not there are non-linear patterns in the excess returns time series of the stock market. And more specifically we observed that as the length of the prediction period was reduced (year, quarter, month) these patterns seem to be more and more non-linear.
- Finally we observe that as the prediction horizon they used was getting smaller the hit rate of their models decreased. Thus in terms of hit rate the smaller the horizon the worst the results.

To sum up, it is possible to apply this methodology to predict the market on a daily basis. Additionally it is widely used by the economists and therefore it is a methodology that we can use for the purposes of the present study.

2.2.3.4 Machine Learning Methods

Several methods for inductive learning have been developed under the common label *"Machine Learning"*. All these methods use a set of samples to generate an approximation of the underling function that generated the data. The aim is to draw conclusions from these samples in such way that when unseen data are presented to a model it is possible to infer the *to-be explained variable* from these data. The methods we discuss here are: The Nearest Neighbor and the Neural Networks Techniques. Both of these methods have been applied to market prediction; particularly for Neural Networks there is a rich literature related to the forecast of the market on daily basis.

2.2.3.4.1 Nearest Neighbor Techniques

The nearest neighbor technique is suitable for *classification tasks*. It classifies unseen data to bins by using their ‘distance’ from the k bin centroids. The ‘distance’ is usually the Euclidean distance. In the frame of the stock market prediction this method can be applied by creating three (or more) bins. One to classify the samples that indicate that the market will rise. The second to classify the samples that indicate fall and the third for the samples related with no change of the market.

Although this approach can be used to predict the market on daily basis we will not attempt to apply it on this study. The main reason is that we will not attempt a classification but a regression task. The classification task has the disadvantage that it flattens the magnitude of the change (rise of fall). On the other hand it has the advantage that as a task it is less noisy comparing to regression. Our intention is to see how well a regression task can perform on the prediction of the market.

2.2.3.4.2 Neural Networks

‘A neural network may be considered as a data processing technique that maps, or relates, some type of input stream of information to an output stream of data‘ [8].

Neural Networks (NNs) can be used to perform *classification* and *regression* tasks. More specifically it has been proved by Cybenko (cited in Mitchel, 1997) that any function can be approximated to arbitrary accuracy by a neural network [9].

NNs are consisted of *neurons* (or *nodes*) distributed across *layers*. The way these neurons are distributed and the way they are linked with each other define the *structure of the network*. Each of the links between the neurons is characterized by a *weight* value. A *neuron* is a processing unit that takes a number of inputs and gives a distinct output. Apart from the number of its inputs it is characterized by a function *f* known as *transfer function*. The most commonly used transfer functions are: the *hardlimit*, the *pure linear*, the *sigmoid* and the *tansigmoid* function[2].

[2] A more detailed description follows in Chapter 4.

There are three types of layers the *input layer*, the *hidden layers*, and the *output layer*. Each network has exactly one input and one output layer. The number of hidden layers can vary from 0 to any number. The input layer is the only layer that does not contain transfer functions. An example of a NN with two hidden layers is depicted in the next figure [10].

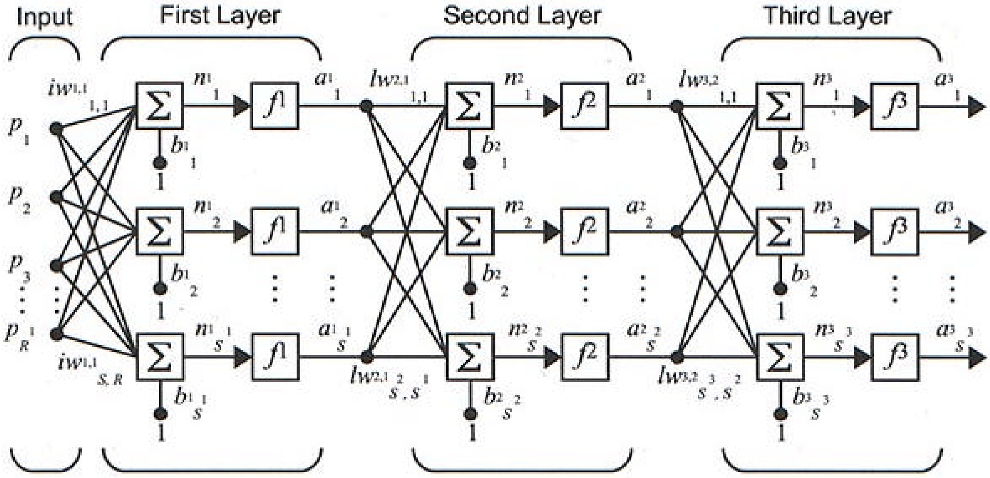


Figure 2.2: NN structure with two hidden layers.

The architecture of this network is briefly described by the string: ‘R-S^1-S^2-S^3’, which implies that the input layer is consisted of R different inputs, there are two hidden layers with S^1 and S^2 neurons respectively and the output layer has S^3 neurons. In our study we will use this notion each time that we want to refer to the architecture of a network.

Once the architecture and the transfer function of each neuron have been defined for a network the values of its weights should be defined. The procedure of the adjustment of weights is known as *training* of the NN. The training procedure ‘fits’ the network to a set of samples (*training set*). The purpose of this fitting is that the fitted network will be able to generalize on unseen samples and allow us to infer from them.

In literature NNs have been used in a variety of financial tasks such as [11]:

- Credit Authorization screening
- Mortgage risk assessment
- Financial and economic forecasting
- Risk rating of investments
- Detection of regularities in security price movements.

Relatively to the present study we found examples of stock market prediction on daily basis [4], [5], [12], [13] using NNs. A brief description of each one of these case studies follows among with our conclusions and comments.

Case Study 1: *“The case of IBM Daily Stock Returns”*

In this study the daily returns of the IBM stock are considered (White) [5]. The data used concern the period from 1972 until 1980. The returns are computed as: r_t=$\frac{p_t - p_{t-1} + d_t}{p_{t-1}}$, where p_t is the value of the share the day t and d_t the dividend paid on day t. Two prediction models were created: an AR model and a feed forward NN. The samples that are used to compute the coefficients of the AR model and train the NN are: [r_{t-5} r_{t-4} r_{t-3} r_{t-2} r_{t-1} | r_t], r_t is the target value. The period from the second half of 1974 until first half of 1978 was used for training (1000 days), while the periods from 1972 until first half of 1974 (500 days) and from the second half of 1978 until the end of 1980 (500 days) for testing the constructed models (*test sets*).

The AR model was r_t= á + â$_1$r$_{t-1}$ + â$_2$r$_{t-2}$ + â$_3$r$_{t-3}$ + â$_4$r$_{t-4}$ + â$_5$r$_{t-5}$ + rs$_t$, where rs$_t$ are the residuals of the models. The NN had a 5-5-1 architecture. Its hidden layer used squashing transfer functions (sigmoid or tansigmoid) and the output layer a linear function. The training algorithm used was the back propagation.

The metric according to which the author made his conclusions was R^2=1-$\frac{\operatorname{var} rs_t}{\operatorname{var} r_t}$. Two experiments took place. In the first one the coefficients of the AR model were calculated on the training set and then r_t and rs$_t$ was calculated on the test sets. For both of the test sets R^2 was calculated:

	1972-1974	**1978-1980**
R^2	0.0996	-0.207

Table 2.3: R^2 for the AR model.

The second data set gave a significant negative result. This means that var rs$_t$ > var r_t, fact that indicates that the prediction model is of no use. While for the first test set R^2 is close to zero this implies that var $rs_t \cong$ var r_t, so the AR model did not manage to capture the patterns in r_t. This fact can be explained in two ways (according to the writer) either there were no patterns, which means that the market is efficient or there are non-linear patterns that cannot be captured by the AR model. In order to check for non-linear patterns that might exist a NN was trained and R^2 was computed again:

	1972-1974	**1978-1980**
R^2	0.0751	-0.0699

Table 2.4: R^2 for the NN model.

These results proved (according to the writer) that the weak form of market efficiency is valid since var $rs_t \cong$ var r_t and there are no patterns linear or non in the residuals produced by the prediction model.

A first comment on this study is that there is no proof or at least an indication whether the AR model used here is the optimum linear model so perhaps there is another linear model (with higher lags than 5) that makes the variance of the residuals smaller and therefore R^2 greater. Secondly the author used a NN to capture the non-linearity that might exist and since he failed he assumed that there is no non-linearity. What if this NN he used is not able to capture it and a more complex network is required? In this case the conclusion that the market is efficient is not valid.

Case Study 2: "Testing the EMH with Gradient Descent Algorithms"

The present case study attempts to predict the sign of the excess returns of six companies traded in New York's stock market (Tsibouris and Zeidenberg) [4]. The companies are: Citicorp (CCI), Jonh Deere (DE), Ford (F), General Mils (GIS), GTE and Xerox (XRX). The prediction is attempted on daily basis. The models created are NN trained with back-propagation techniques. The data considered are from 4 Jan 1988 until 31 Dec 1990. The period from 4 Jan 1988 until 29 Dec 1989 is used to extract data to train the networks, while the returns from 2 Jan 1990 until 31 Dec 1990 are used to test the constructed models. The form of the input data is [r_{t-264} r_{t-132} r_{t-22} r_{t-10} r_{t-5} r_{t-4} r_{t-3} r_{t-2} r_{t-1} | r_t], r_t is the sign of the excess return for day t.

The NNs trained and tested were feed forward networks 9-5-1. All the neurons used the sigmoid transfer function. The evaluation criterion used by the author was the *hit rate* of the model. Hit rate is defined as the percentage of correct predictions of the sign of the return. The results obtained were:

Company	Hit Rate on the Test Set
CCI	60.87%
DE	48.22%
F	60.08%
GIS	53.36%
GTE	53.36%
XRX	54.15%

Table 2.5: The Hit rate of the NN for each one of the stocks considered.

On average the hit rate was 55,01 %. From this statistic the conclusion of the author was that there is evidence against the EMH. Assuming that a naï ve prediction model that would have been based on a random choice of the sign of the return would gave a hit rate of 50%.

As a side note in the study it is referred that an alternative specification using signed magnitudes as inputs and signs and magnitudes as two separate outputs was attempted but it did not perform well.

This study managed to create models that on average outperformed a naï ve prediction model. The way this naive model is defined makes it too lenient. A fairer benchmark would compare the hit rate of the neural network with the hit rate of a prediction model that for the entire test period predicts steadily rise of fall depending on which is met more frequently in the *training set*. Another option could have been to compare the models with the *random walk* model. A second interesting point from this study is that when the NN was trained on the actual returns and not their sign performed worse. The reason for this might be that the *training set* with the actual values is noisier than the one with the signs of the values. Therefore a NN has greater difficulty to trace the real patterns in the input data.

Case Study 3: “Neural Networks as an Alternative Stock Market Model”

This case study investigates the performance of several models to forecast the return of a single stock (Steiner and Wittkemper) [12]. A number of stocks are predicted, all these stocks are traded in Frankfurt’s stock market. They are grouped by the authors in two categories:

Group A: ‘dax-values’	**Group B: ‘ndax-values’**
Siemens	Didier
BASF	PWA
Kaufhof	KHD
Blue Chips	Smaller Companies

Table 2.6: The stocks grouped in two categories dax-values and ndax-values.

The data used consists of the logarithmic value of the daily returns (T) of each one of the stocks above as well as the daily returns of DAX index (D), West LB index (W) and the Index der Frankfurter Werpapierborsen (F). Chronologically they were from the beginning of 1983 until the end of 1986. The training and test sets were defined as follows:

1983 (training set) 250 days	→	1984 (test set) 250 days				
		1984 (training set) 250 days	→	1985 (test set) 250 days		
				1985 (training set) 250 days	→	1986 (test set) 250 days

Figure 2.3: The training and test sets used in the study.

Initially data from 1983 was used to train the models and the data from 1984 to test them. Then the data used shifted in time by a full year, which means that the data from 1984 was used for training while data from 1985 for testing. Finally the models were trained and tested using the data from 1985 and 1986 respectively.

In total nine models were created to predict the returns r_t from each stock, five of them were based of NN and the rest on linear regressions (univariate and multivariate). Three of the networks were feed forward and the rest were *recurrently* structured (the outputs of some of the neurons were used as inputs to others that did not belong to the next layers). More specifically the models were:

Linear Models	
linear regression	$r_t = á + â_1D_{t-1}$
linear regression (a=0)	$r_t = â_1D_{t-1}$
multivariate regression	$r_t = á + â_1D_{t-1} + â_2W_{t-1} + â_3F_{t-1} + â_4T_{t-1}$
linear regression*	$r_t = á + â_1D_{t-1}$

Table 2.7: Linear regression models.

Neural Network Models		
	Structure	**Inputs**
NN 1	1-10-1	D_{t-1}
NN 2	1-5-5-1	D_{t-1}
NN 3	4-10-1	D_{t-1}, W_{t-1}, F_{t-1}, T_{t-1}
NN 4	1(2)-10 (2)-1	D_{t-1}
NN 5	4(8)-10 (2)-1	D_{t-1}, W_{t-1}, F_{t-1}, T_{t-1}

Table 2.8: Neural network models.

The fourth model is not an actual prediction model since its coefficients were always calculated on the *test set* and not on the *training set*. NN 4 and NN 5 are recurrent networks and in their architecture string the numbers in brackets indicate the number of recurrent neurons used. For NN 1, NN 2 and NN 4 the input used was D_{t-1}, while for NN 3 and NN 5 the inputs used were D_{t-1}, W_{t-1}, F_{t-1} and T_{t-1}. All the data used to train and test the NNs where normalized in order to be in the interval [0,1]. The training algorithm was the back propagation (with learning rate 0.0075 with no momentum term). The error function used was the mean absolute error (*mae*):

$$\text{mae}=\frac{1}{n}\sum_{t=1}^{n}\left|r_t - a_t\right| \quad (2.1)$$

where a_t is the prediction that the model gave for the return of day t.

The rank of the models in terms of *mae* was the following:

Model	dax-value mae	ndax-value mae	Total mae	dax-values Rank	ndax-values Rank	Total Rank
linear regression	0.0081259	0.0123370	0.0102314	8	8	8
linear regression (á=0)	0.0081138	0.0123136	0.0102137	7	7	7
linear regression*	0.0080028	0.0120792	0.0100410	5	4	5
multivariate regression	0.0071830	0.0121974	0.0096902	2	6	3
NN 1	0.0080565	0.0121543	0.0101054	6	5	6
NN 2	0.0089707	0.0127085	0.0108396	9	9	9
NN 3	0.0071691	0.0118060	0.0095010	3	2	2
NN 4	0.0078866	0.0120313	0.0099590	4	3	4
NN 5	0.0071732	0.0116660	0.0094196	1	1	1

Table 2.9: The performance of all models in mae terms.

The NNs did better than the linear regression models. Moreover the best results came from a recurrent network. Indeed a strict rank of the models based on the *mae* give us this conclusion. But the differences between the *mae* of most of the models are very small. For instance in the ‘Total *mae*’ the difference between the first and the second model is 0.0000814 while between the first and the third 0,0002706. Although *mae* is scale variant (it depends on the scale of the input data) this type of differences are small even for returns and thus cannot give us a clear rank for the tested models. Having also in mind that the performance of a NN is heavily influenced by the way its parameters are initialised (weight initialisation) at least for the NN models it would be safer to rank them having in mind the mean and the standard deviation of their performance for various initialisations of their weights. Further more this study gave us no indication of how well these models would do if they were applied to predict the market and make profit out of it (or against a naï ve prediction model e.g. the random walk model).

However we can say that at least for the specific experiments described by the table above univariate regression models seem to be steadily worse than the NNs (apart from NN2). Also it seems that NNs with the same number of layers and nodes performed better when they were fed with more input data (NN1 and NN3). Another observation is that networks with the same inputs but different structures (NN1 and NN2) had significant difference in their performance; therefore the topology of the network seems to influence heavily the *mae*.

Case Study 4: "A multi-component nonlinear prediction system for the S&P 500 Index."

Two experiments of daily and monthly prediction of the Standard and Poor Composite Index (S&P 500) excess returns were attempted by Chenoweth and Obradovich [13]. The daily data used starts from 1 Jan 1985 and ends at 31 Dec 1993. The data set consists of a total of 2,273 ordered financial time series patterns. Initially, each pattern consisted of 24 monthly (e.g. Rate of change in Treasury Bills lagged for 1 month) and 8 daily features (e.g. Return on the S&P Composite Index lagged for 1 day). A feature selection procedure[3] resulted in only 6 of the initial features:

- Return on 30 year Government Bonds.
- Rate of Change in the Return On U.S. Treasury Bills lagged for 1 Month.
- Rate of Change in the Return On U.S. Treasury Bills lagged for 2 Months.
- Return on the S&P Composite Index.
- Return on the S&P Composite Index lagged for 1 day
- Return on the S&P Composite Index lagged for 2 days.

The initial training set contained 1000 patterns[4] from 1 Jan 1985 until 19 Dec 1988. The models were trained on this set and then were used to predict the market at the first trading day after the 19 Dec1988, Day_t. The next step was to include a new pattern based on Day_t excess return in the training set (removing the oldest pattern) and retrain the model. That way the training set had always the same size (window size) but it was shifting through time. This training approach that was followed by the authors is based on their belief that you cannot base your prediction model on the way that the market behaved a long period ago because these historical data may represent patterns that no longer exist.

The monthly historic data consisted of an initial training window of 162 patterns formed using data from Jan 1973 to Feb 1987 and actual predictions were made for the 70-

[3] A search algorithm was applied to determine a subset of the existing features that maximized the differences between the classes based on criteria such as Euclidian, Patrick-Fisher, Mahalanobis and Bhattacharyya distance. These classes were created from a clustering algorithm applied on patterns with various numbers of features. This way the patterns that created the 'clearest' clustering were finally selected.

[4] Apart from this case there was another training set initially created that was consisted of 250 patterns. Each one of these training sets was applied to different models.

month period from Mar 1987 to Dec 1992. The initial monthly data set contained 29 features per pattern that was reduced to 8.

Six different models were created, all using feed forward NN trained with a back-propagation technique. Three of the models were used for the prediction on a daily basis and the other three for prediction on a monthly basis.

	Daily Prediction			Monthly Prediction		
	Model 1	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**
Architecture	32-4-1	6-4-1	6-4-1 6-4-1	29-4-1	8-3-1	8-3-1 8-3-1
Training Window	250	250	1000	162	162	162

Table 2.10: The models considered in the study.

Models 1 and 4 were trained and tested on the initial features data sets while models 2 and 5 where trained and tested on fewer features. Each off these models (1,4,2,5) was then combined with a simple trading rule: *if prediction is that the market will appreciate invest in the market else invest in bonds*. Assuming that the transaction costs are zero the annual rate of return (ARR) for each model was calculated.

	Daily			Monthly		
Processing	**Model**	**ARR**	**Trades**	**Model**	**ARR**	**Trades**
Initial features	1	-2.16%	905	4	-1.67%	62
Reduced features	2	2.86%	957	5	-3.33%	56
Reduced features and 0.5 % noise removal (h)	2	5.61%	476	5	-2.97%	52

Table 2.11: The annual return rates provided by models 1, 2, 4 and 5.

For daily prediction feature reduction improved the annualized returns. Furthermore a strategy of removing from the dataset those patterns with a target value close to zero was applied. According to this strategy if the target value of a pattern was greater than -h and smaller than h this pattern was removed from the training set. This type of noise removal improved the performance of the predictor significantly.

For the monthly prediction case the features reduction had the opposite results, while the noise removal improved the performance slightly.

The architecture of the NNs was determined experimentally through the trial and error approach on a small set of training data.

Models 3 and 6 consist of two NNs each. The first of these NN was trained on positive samples (samples that indicate that the market appreciates) while the second was trained on negative samples (samples that indicate that the market depreciates). The way that these NNs were used is shown in the following figure:

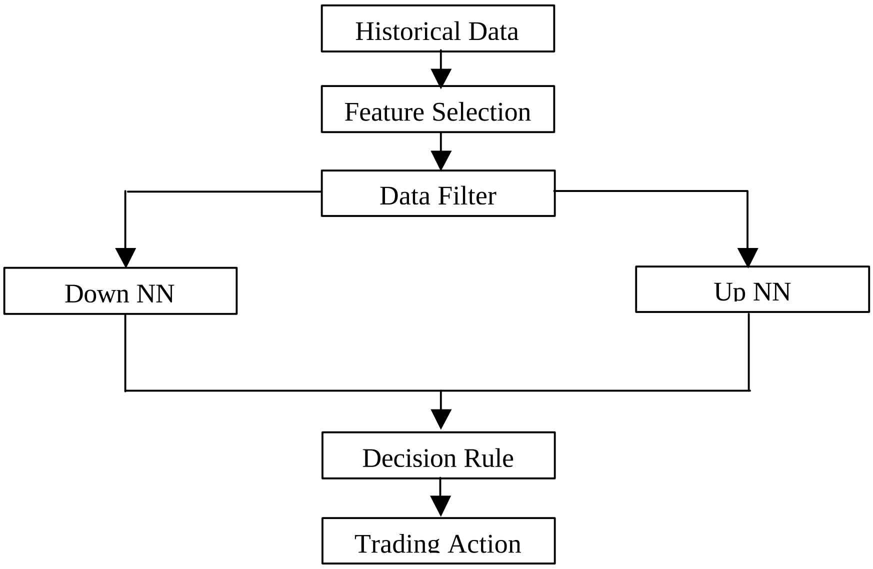


Figure 2.4: The stock market prediction system that uses models 3 and 6.

Firstly the feature space was reduced; later on the data was filtered and dived into two groups those that indicate appreciation of the market and those that indicate depreciation. The NNs were trained separately. Once the nets were trained each unseen sample (from the test set) were through the both NNs. Therefore two predictions were made for the same sample. These predictions were fed to a trading rule that decided the trading action. Three different trading rules were tested.

Rule 1: Maintain current position until a clear buy/sell recommendation is received.
Rule 2: Hold a long position in the market unless a clear sell recommendation is received.
Rule 3: Stay out of the market unless a clear buy/sell recommendation is received.

A number of experiments for different definitions of the ‘clear buy/sell signal’ and different noise clearance levels took place. For daily prediction Rule 2 resulted in an annual return rate of 13.35%, while a buy and hold strategy for the same period gave a return of 11.23%. The predictions based on Rules 1,3 did not manage to exceed the buy and hold strategy.

On the other hand, the prediction on monthly basis for the optimum configuration of the ‘clear buy/sell signal’ and noise clearance level gave annual return of 16.39% (based again on Rule 2). While the annual return rate for a buy and hold strategy was 8.76%.

This case study led us to the following conclusions. Firstly more input features do not necessarily imply better results. By introducing new features to the input of your model you do not always introduce new information but you always introduce new noise. We also have an indication of what features are important on daily basis market prediction. Of course this does not imply by any means that the list of input features used on this study is exhaustive. Furthermore, this study proved how important is the use of the correct trading rule in a prediction system. Therefore it is not enough to create robust prediction models, you also need robust trading rules that, working in conjunction with your prediction model, can give you the ability to exploit the market. Another point that is clear from the study is that by selecting your initial data (e.g. noise removal) you can improve your prediction ability. Lastly the evaluation strategy followed by the current case study is perhaps the optimum way to evaluate a model’s predictive power. The only drawback is that it did not incorporate transaction costs.

All the case studies reported in this section make clear that it is possible to use NNs in the frame of daily basis prediction. The success of NNs varies from one study to the other depending on their parameters settings and the underlying data.

2.3 Defining The Framework Of Our Prediction Task

2.3.1 Prediction of the Market on daily Basis

In this paragraph we attempt to sum up our review to the literature in order to define some basic characteristics of our study. These characteristics concern the exact definition of our prediction task, the models and the input data we are going to use in order to accomplish this task.

The case studies we have seen so far led us to a number of conclusions. Firstly the work of Hsieh [3] and Tsibouris et al [4] gave us clear indications that the market does not fluctuate randomly, at least for the markets and the time periods they are concerned with. On the other hand White's study [5] suggests that since neither the linear model nor the NN manage to find patterns in the data there are no patterns in it.

Secondly, we have indications from the work of Pesaran & Timmerman [7] and Steiner & Wittkemper [12] that there are non-linear relationships in the stock market data; the first two did not study daily data but it is clear from their work that when the prediction horizon decreased from year to quarter and then month the non-liner patters in the data increased.

Thirdly, the work of Chenoweth & Obradovic [13] proved that NNs that use input data with large dimension do not necessarily perform better; on the contrary large dimensionality of the input data led to worse performance. Whereas the experiments of Steiner & Wittkemper [12] indicated that networks with few inputs under perform comparing with others that used more. Therefore too much information or little information can lead to underperformance.

Additionally it became clear that a prediction model has to be used in conjunction with a trading rule, in this case the presence of transaction costs is heavily influential to the profit we can have from the market [7]. The nature of the trading rule is also heavily influential as Chenoweth & Obradovic [13] proved. Their work indicates that using their prediction models with Rules 1 and 3 resulted in useless models (in terms of their ability to beat a buy and hold strategy) while the use of trading Rule 2 allowed them to beat the buy and hold strategy.

Finally, as the work of Steiner & Wittkemper [12] indicated and as far as the specific experiments they did are concerned the NNs performed steadily better comparing to the univariate regression models, whereas they performed closer to multivariate regression models.

None of these studies though compared the prediction ability of the models constructed with the *random walk model*. Also in the cases that NN models were trained and tested the choice of their architecture was not based on a rational methodology. Additionally issues such as validation[5] and variance of the prediction ability of the NN models due to the random way that their weights are initialized were not examined by these studies.

Having in mind the above we attempt to define the basic characteristics of our study. The first decision we have to make is related to the type of time series we want to predict. The most obvious option would be the actual index of the market on daily basis. But is this the most appropriate? The second decision concerns the type of prediction models we are going to use. Is it going to be NNs, traditional time series regression or both? Finally we have to select the kind of input data we will use in conjunction with our models.

2.3.2 Defining the Exact Prediction Task

As already has been stated the most obvious prediction task that one could attempt is the prediction of the time series of the actual value of the market. But is this a good choice? As far as the presented case studies are concerned, none of them adopted this strategy. Instead they select to use the daily return r_t. Some reasons for this are [2]:

- r_t has a relatively constant range even if data for many years are used as input. The prices p_t obviously vary more and make it difficult to create a model compatible with data over a long period of time.
- It is easier computationally to evaluate a prediction model that is based on returns and not in actual values.

Therefore the case of using returns seems to be more eligible.

The return r_t for day t is defined as $\frac{p_t - p_{t-1}}{p_{t-1}}$ where p_t is the actual price of the market on day t. What the return describes, is to what extend (in percentage) the investor manage to gain or loose money once using the stock market as a tool of investment. Thus if p_t is greater than p_{t-1} this implies positive returns therefore gains for the investor.

[5] The role of validation is discussed in details in Chapter 4

Is this approach correct? The gains depend not only on the sign of the return but on its magnitude too. If the alternative for the investor was just to keep his capital without investing it then the sign would be enough, but this is not a realistic scenario. Capitals never 'rest'. A more realistic scenario is to assume that if an investor does not place his capital to the stock market (or to any other investment tool) he would at least enjoy the benefits of the bond market. Therefore we need another way to calculate the excess return of the market by incorporating the 'worst' case of profit if not investing to the market. In such a scenario the excess return would be:

$$R_t = r_t - b_t$$

where b_t is the daily return if investing in bonds. The calculation of b_t will be based on the treasury bill (T-Bill) rates announced by the central bank of each country a certain number of times per year (that varies from one country to the other). This is the type[6] of time series we are trying to predict on this study.

2.3.3 Model Selection

The literature review indicates that for a prediction on daily basis we can use models such as *Traditional Time Series Models* and the *NNs*. In order to have a clearer view for them we list their benefits and drawbacks.

Traditional Time Series Models:

- Widely accepted by economists.
- Not expensive computationally.
- Widely used in the literature.
- Difficult to capture non-linear patterns.
- Their performance depends on few parameter settings.

Neural Networks:

- Able to trace both linear and non-linear patterns.
- More expensive computationally.
- Not equally accepted by economists in respect with the traditional time series approach.
- Their performance depends on a large number of parameter settings.

[6] More specific this time series is going to be transformed using natural logs. This is not a straightforward task thus it is going to be analysed in details in the next chapter.

It is clear that each category of models has its strong and weak points. In our attempt to compare them we did not manage to select one and neglect the other. Instead we are going to use both and compare their efficiency on the attempted task. More specifically, at the first stage we will use *Traditional Time Series prediction models* and we will examine if they manage to capture all the patterns that exist in our data, if not we will use NN models to attempt to capture these patterns. The case studies we have examined clearly indicate that there are non-linear relationships in the data sets used. Thus our intuition is that in our case study too the *Traditional Time Series prediction models* will not be able to take advantage of all the patterns that exist in our data sets.

2.3.4 Data Selection

The evidence we have from the fourth case study is that at least for the NN models the more input features you include the more noise you incorporate without necessarily to offer new information to your model. In this sense the less features you include in your input data the better. On the other hand case study three indicated that networks with structure x-10-1 performed significantly better in case that x=4 that when x=1 or in other words performed better when 3 extra input features were feed into the model. The conclusion we have is that there is a clear trade off between noise and new information when adding features in your input space.

In the present study we will attempt to predict the excess return time series by using only lagged values of the series. In that way we are trying to keep the inserted noise to our data set as low as possible. The cost we pay for this is that perhaps the information fed to our models is not enough to give us the opportunity for good predictions. An additional reason we adopt this strategy is that we want to see how well predictions we can have by using the information that the time series itself carries. The size of the optimum lag is an issue we have to investigate.

Summary

In this chapter we described the main investment theories and the way these theories influence the market. This description allowed us to understand the way that the market is defined. Furthermore we concluded that in order to attempt a prediction task we have to be certain that such a task is feasible. If the market fluctuates randomly then there is no space for predictions. Therefore our first concern should be to get evidence against randomness in the series we would try to predict. Secondly we categorized the available prediction methods and we spotted those that are possible to fit in the frame of our study. For each one of them we presented case studies. Then based on the evidence we found we selected the most appropriate characteristics for the prediction task attempted in our study.

Chapter 3

Data

In this chapter we consider the datasets we use in our study. Initially topics such as the origin of the data, their description in statistical terms as well as their quality are covered. Later on we describe the procedure of their integration in order to create the *excess returns* time series. Furthermore we format these series in such a way that will be compatible with the models we will use. Lastly the *excess returns* time series are tested for randomness.

3.1 Data Understanding

3.1.1 Initial Data Collection

The data considered in this study are obtained from *DataStream International* [14]. We are concerned with the London and the New York stock markets and more specifically with the FTSE-500 and the S&P-500 indices. In order to form the data series we have described in the previous chapter we have obtained the following time series: FTSE-100 index, T-Bill Rates of UK, S&P-500 index, T-Bill Rates of US.

The FTSE-500 data consist of 3275 daily observations of the index from 4 Jan 1988 until 12 Dec 2000 and the respective T-Bill rates for the same period and frequency. The UK T-Bill rates are on an annualized scale with maturity of one month. The S&P-500 data concern the value of the index on daily basis from 4 Jan 1988 until 12 Dec 2000, a total of 3277 observations. The US T-Bill rates cover the same period, frequency and scale but they have a thirteen-week maturity. These characteristics of our initial datasets are summarized in the following table:

Series	From	To	Observations
FTSE 500 Index	04/01/1988	12/12/00	3275
UK T-Bill rates	04/01/1988	12/12/00	3275
S&P 500 Index	04/01/1988	12/12/00	3277
US T-Bill rates	04/01/1988	12/12/00	3277

Table 3.1: Initial datasets.

3.1.2 Data Description

A detailed description of each one of these series follows. A list of descriptive statistics is presented as well as graphs with their values over time.

For the series of FTSE 500, UK T-Bill rates, S&P 500 and US T-Bills we have:

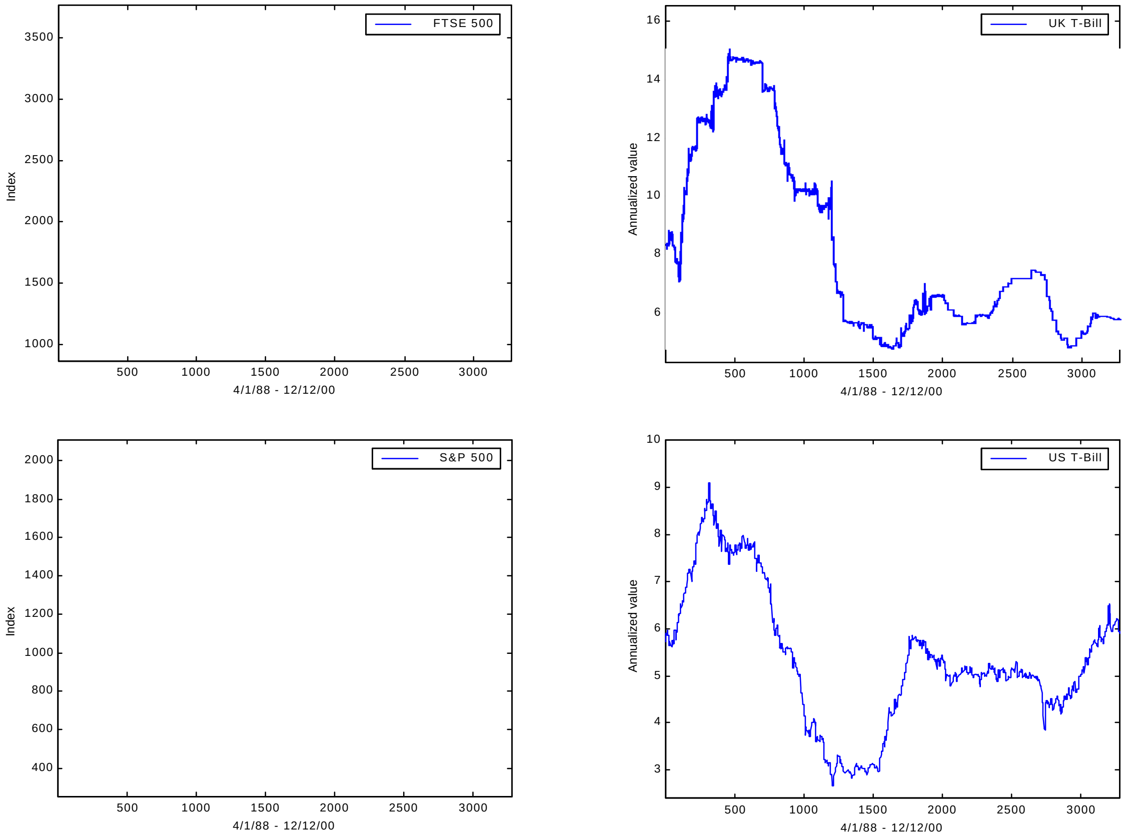


Figure 3.1: Time series of FTSE 500, UK T-Bill rates, S&P 500 and US T-Bills.

Statistic	**FTSE 500**	**UK T-Bills %**	**S&P 500**	**US T-Bills %**
Mean	1860.049814	8.140526718	778.3715777	5.334599939
Standard Error	11.89038006	0.055489607	8.172531339	0.025619592
Median	1659.2	6.5938	542.13	5.12
Mode	1021.4	7.1875	374.6	5.03
Standard Deviation	680.4581006	3.175537887	467.8372685	1.466595797
Sample Variance	463023.2266	10.08404087	218871.7098	2.15090323
Kurtosis	-0.775200553	-0.63554572	-0.34451257	-0.365324091
Skewness	0.680275287	0.902097111	1.015461827	0.372249529
Range	2462.84	10.2656	1639.23	6.43
Minimum	958.79	4.7813	278.41	2.67
Maximum	3421.63	15.0469	1917.64	9.1
Observations	3275	3275	3277	3277

Table 3.2: Descriptive statistics of FTSE 500, UK T-Bill rates, S&P 500 and US T-Bills.

Their plots over time clearly indicate trends. The FTSE and S&P series has a trend upwards while the T-Bill rates have a reversed trend. Thus there is a clear relationship between the value of the stock market and the value of T-Bill rates. In other words these graphs describe that: decrease of the interest rates implies increase in the market. Moreover comparing the type of fluctuations for the FTSE - S&P couple and UK T-Bills – US T-Bills couple we can see that they fluctuate similarly. This is reasonable since the economies of the two countries are traditionally highly correlated.

3.1.3 Data Quality

In all four datasets there are no missing values. However very often in datasets, there exist samples that do not comply with the general behavior or model of the data. Such data samples, which are grossly different from or inconsistent with the remaining set of data, are called *outliers* [15]. Our next step is to investigate the datasets described above for *outliers*.

In order to check for *outliers* we calculate the first (Q1) and the third (Q3) quartile of the distribution our data. The fist and the third quartile of a distribution are defined as its 25-th and 75-th percentiles respectively. A value x_p is called the k-th percentile of a given distribution if $P(X<x_p)=k/100$, where X is a random variable [16]. Therefore the 25-th percentile is a value that splits our dataset in two subsets that each one contains 25% and 75% of the mass of our samples respectively. The 50-th percentile is the *median* of our data.

For each of the sets above we calculate Q1 and Q3. Then we form the quantity Q3-Q1 and we call an *extreme outlier* any value in our dataset that is greater than Q3+3(Q3-Q1) or lower than Q1-3(Q3-Q1). This being the case we have checked all four datasets and we found no *extreme outliers*.

3.2 Data Preparation

3.2.1 Data Construction

In the previous chapter we set the objective to predict the *excess returns* that come from the stock market. The *excess returns* were defined as the difference between the returns from the market and the returns from T-Bills on a daily basis. The stock market returns r_t are defined as $\frac{p_t - p_{t-1}}{p_{t-1}}$ (3.1) where p_t is the index on day t. Moreover the returns from the T-Bill rates on a daily basis can be calculated as $b_t = \frac{1}{100} \frac{rate_{t-1}}{360}$ (3.2), where $rate_{t-1}$ is the annualised value of the T-Bill rate on day t-1 as a percentage. By setting $c_{t-1} = \frac{1}{100} rate_{t-1}$ we transform (3.2) to $b_t = \frac{c_{t-1}}{360}$ (3.3). From (3.1) and (3.3) the excess return is $R_t = r_t - b_t$.

In our study we choose to transform the returns from the market and the T-Bill rates before we calculate the *excess returns*. The transformation we introduce is taking the natural logs of these returns. The problem we face is that although we can calculate the logarithm of b_t we cannot do the same for r_t. This is due to the fact that r_r is not always positive and thus its logarithm cannot be calculated for the negative values. The way we bypass this problem is that we rescale r_t by adding 1 to it. Therefore for the stock market returns we have:

$$1+r_t = 1 + \frac{p_t - p_{t-1}}{p_{t-1}} = \frac{p_t}{p_{t-1}} \quad (3.4)$$

The result of applying logarithms on (3.4) is ln($\frac{p_t}{p_{t-1}}$). Similarly for the T-Bill rates we rescale (3.3) and the we transform it resulting to ln($\frac{c_{t-1}}{360}$+1). The *excess returns* of the stock market is defined as:

$$y(t)= \ln(\frac{p_t}{p_{t-1}})–\ln(\frac{c_{t-1}}{360}+1) =\ln(\frac{\frac{p_t}{p_{t-1}}}{\frac{c_{t-1}}{360}+1}) \quad (3.5)$$

The following table is mapping the relationship between the data series we obtained from DataStream and the symbols we use here.

DataStream Series	Symbol
FTSE 500, S&P 500	p_t
UK T-Bill, US T-Bill	$rate_t$

Table 3.3: The symbols used for each one of the initial time series.

The outcome of this procedure is two time series with the *excess returns* of FTSE and S&P respectively. The next table presents some basic statistics that describe these series.

Statistic	FTSE 500	S&P 500
Mean	0.000118258	0.000372266
Standard Error	0.000139216	0.000166938
Median	0.000270064	0.000469634
Mode	#N/A	-0.000205812
Standard Deviation	0.007965796	0.009554908
Sample Variance	6.34539E-05	9.12963E-05
Kurtosis	2.25058397	5.888557517
Skewness	-0.094864286	-0.522411917
Range	0.095780447	0.127418966
Minimum	-0.041583703	-0.074574421
Maximum	0.054196744	0.052844545
Count	3274	3276

Table 3.4: Descriptive statistics of the excess returns of FTSE and S&P.

Moreover the value of the series against time is presented by the following graphs:

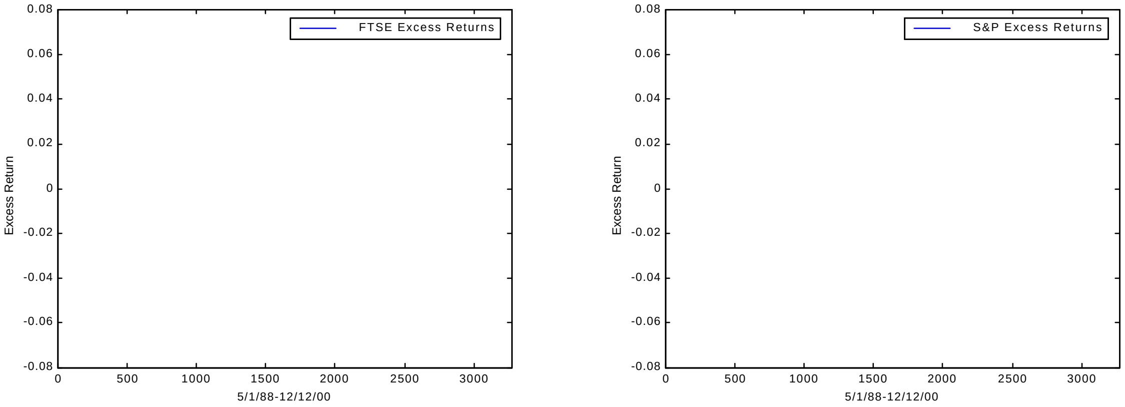


Figure 3.2: The excess returns time series for FTSE and S&P.

From these graphs it is clear that the S&P returns are more volatile, it also appears to have more extreme values. Another observation is that FTSE and S&P excess returns fluctuate in a similar way; there are periods that both series have a ‘narrow’ fluctuation and others that have a ‘wider’ one. This is rational since the FTSE 500 and the S&P 500 indices and the UK and US T-Bills fluctuate similarly.

3.2.2 Data Formation

For the needs of the *traditional time series regression models* and the *neural networks* we divide the *excess returns* time series into subsets. These subsets form two major categories, sets that will be used to define the parameters of the models and sets that will be used to measure their prediction ability. For each category of models we form a different number of subsets.

Traditional time series regression models adjust their *explanatory variables* in a set of data called *in sample* data, and can be used to make predictions on a second set of data called the *out of sample* data. Thus we divide the *excess returns* datasets (Set A) into two subsets:

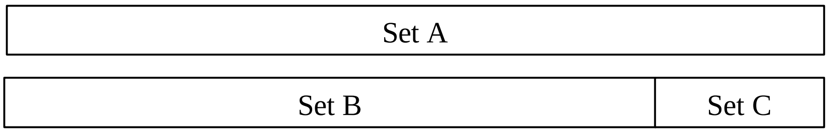


Figure 3.3: Training (B) and Test (C) Sets.

Set B contains approximately 90% of the samples in Set A and Set C the rest 10%.

In a similar way the neural networks are adjusted (trained) on a part of the available data and tested on another part. Again we will use Set B to adjust the parameters of the models and Set C to measure their prediction ability. This way we will be able to make comparisons of the performance of both types of models on the same dataset. In this study we will use the term '*Training set*' for Set B and '*Test set*' for Set C. Additionally, due to the nature of the parameters adjustment of the neural network models we need to divide the *training set* (Set B) into three new subsets:

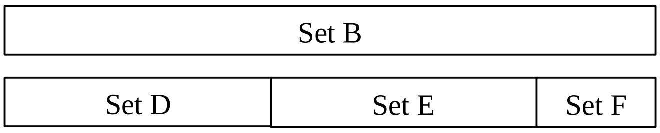


Figure 3.4: Training1 (D), Validation1 (E) and Validation2 (F) sets.

We will use the terms *Training1*, *Validation1* and *Validation2* set for Set D, Set E and Set F respectively. The purposes that each one of these subsets serves are described in details in chapter 4.

Training and *Test* sets have a predefined size while the size of *training1*, *validation1* and *validation2* sets can vary. More specifically for the *Training* and *Test* sets we have that:

Set	**Market**	**Samples**	**From**	**To**
Training	FTSE	2974	5/1/88	6/10/99
	S&P	2976		
Test	FTSE	300	7/10/00	12/12/00
	S&P	300		

Table 3.5: The size of training and test sets for FTSE and S&P.

3.3 Testing For Randomness

3.3.1 Randomness

"Randomness in a sequence of observations is defined in terms of the inability to device a system to predict where in a sequence a particular observation will occur without prior knowledge of the sequence". (Von Mises, sited in [17])

It is clear from this definition that whenever one has a random sequence he is unable to device a system to predict the sequence. Bennett also states "the meaning of randomness is the *unpredictability* of future events based on past events". Therefore it is essential for us to prove or at least to have strong indications that the time series of data produced by the stock markets are not random. Only then it will be possible to create systems that can predict the market.

In the literature there are a number of tests that can be used to prove whether a sequence is random or not. These tests are divided into two major categories *empirical* and *theoretical tests* [18]. In empirical tests we manipulate groups of numbers of a sequence and evaluate certain statistics (Frequency test, Ferial test, Gap test, Run test, Collision test, Serial Correlation test). In the theoretical tests we establish characteristics of the sequence by using number theoretic methods based on the recurrence rule used to form the sequence. Whenever a test fails to prove that a sequence is non-random we are a step closer to accept that this specific sequence is random. A test might fail to prove that a sequence is non-random but second one might prove that the same sequence is non-random.

In this study we use two randomness tests that both belong to the category of empirical tests, 'Run' and 'BDS' test. The results we obtained from both of these tests gave us indications of non-randomness for the data (*Training sets*) on which they were applied to.

3.3.2 Run Test

A run in a sequence of symbols is a group of consecutive symbols of one kind preceded and followed by (if anything) symbols of another kind [16]. For example, in the sequence:

+ + + - + + - - - - + + - -

the runs can be exhibited by putting vertical bars at the changes of the symbol:

+ + + | - | + + | - - - - | + + | - -

In this sequence we have three runs of '+' and three runs of '-'.

We consider now the series of a stock market and we calculate its median[7]. To each one of the *excess returns* we assign the symbol '+' if it is above the median and '-' if it is below. The outcome of this procedure is a new series let's name it S (S will contain only '+'s and '–'s). If the initial sequence contains an odd number of points we neglect the median. This way S will contain m '+'s and m '-'s, thus the length of S will be 2m. We also define as r_+ and r_- the number of runs that contain '+' and '-' respectively and r to be equal to $r_+ + r_-$. If r is even then in S we have $r_+ = r_-$, if r is odd then either $r_+ = r_- + 1$ or $r_- = r_+ + 1$. The run test is based on the intuitive notion that an unusually large or an unusually small value of r would suggest lack of randomness. More specifically it has been proven that for a random series S, r is approximately normally distributed with mean:

$$\mathrm{E(r)=m+1} \quad (3.6)$$

and variance:

$$\mathrm{var(r)}=\frac{\mathrm{m(m+1)}}{\mathrm{2m-1}} \cong \frac{1}{4}(2m-1) \quad (3.7)$$

The tactic we follow here is that for the *Training sets* of the *excess returns* series of both markets we calculate r and we compare it with the distribution that r would follow if the series were random (according to equations 3.6 and 3.7).

A second practice we adopt is that we create 5000 series with the same length as the *excess returns* time series. The series are consisted of random numbers belonging in the interval [0,1]. For this task we use the random number generator of Matlab 5.2. We plot the distribution of r for these 5000 series and then again we compare the runs we found in the stock market series with this distribution.

The *training sets* for FTSE and S&P contain 2974 and 2976 samples respectively. For the FTSE *training set* we found r_{FTSE} =1441 and m_{FTSE} =1487, while for the S&P we found $r_{S\&P}$=1483 and $m_{S\&P}$ =1488. Thus according to (3.6) and (3.7) the number of runs in a sequence of length 2974 and 2976 should follow normal distributions with mean and variance (1488, 743.25) and (1489, 743.75) respectively. The figures below indicate these results.

[7] Median is the number that divides a set into two equal, in terms of mass of elements, subsets.

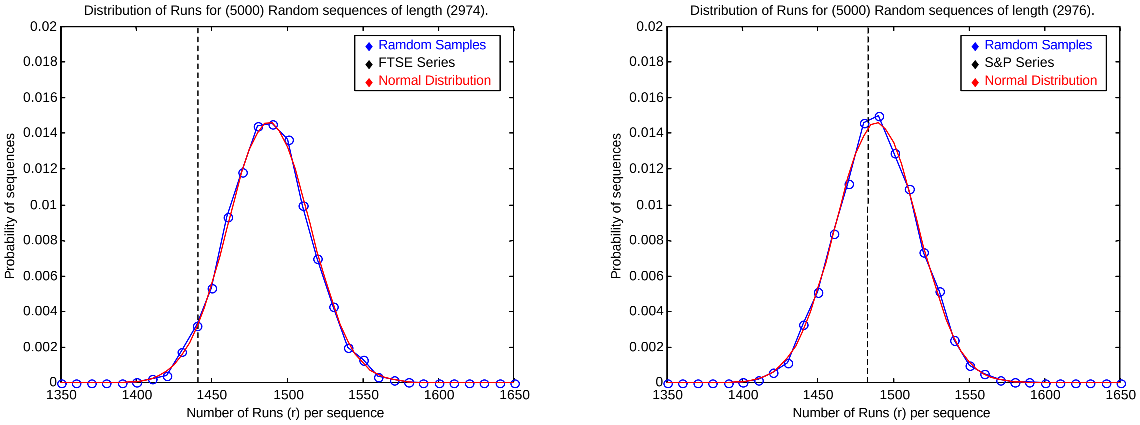


Figure 3.5: The Runs in the FTSE and S&P excess return series.

The dotted lines indicate the number of runs in each set of excess returns (FTSE and S&P), while the red lines describe the distribution of r in random series which have the same length with our *excess return* series (base on equations 3.6 and 3.7). Finally the blue lines describe the distribution of r for the simulation we did using 5000 random series.

Moreover we convert all the distributions to standard normal distributions N(0, 1). The conversion is done according to: $Z = \dfrac{\text{X - E(r)}}{\text{var(r)}}$, where X are the values of the initial distributions. Then we calculate the probability $P(Y \leq \bar{r}_{FTSE})$, where $\bar{r}_{FTSE} = \dfrac{r_{\text{FTSE}} - \text{E(r)}}{\text{var(r)}}$ and Y is a random variable of the standard normal distribution. $P(Y \leq \bar{r}_{FTSE})$ equals to 0.04, which implies that random series with smaller or equal numbers of runs comparing to the FTSE series occur with frequency 4%. Thus we can be confident by 96% that the FTSE series is not a random series. Similarly for the S&P we calculated $P(Y \leq \bar{r}_{S\&P})$ and we found that approximately 45% of the random time series has the same or less runs. Thus we did not find evidence against the randomness of the S&P series.

To sum up the results we obtained from the run test we say that the run test gave us indication for non-randomness in the FTSE excess returns but did not manage to indicate non-randomness for the S&P series.

3.3.3 BDS Test

BDS test originally introduced by Brock, Dechert and Scheinkman in 1987 is “a non-parametric method for testing for serial dependence and non-linear structure in a time series”[19]. This method can be applied to a time series and prove whether the members of the series are Independently and Identically Distributed (IID), the IID consists the null hypothesis for the BDS test. IID implies randomness; therefore if a series is proved to be IID it is random. In this study we use BDS to test for both serial dependence and non-linear structure of time series.

Why BDS test? Because it has been used and analyzed by many researchers and it has been applied extensively on finance data. Some examples include: Kosfeld and Rode (1999) [20], Barnes and De Lima (1999) [21], Barcoulas, Baum and Onochie (1997) [22], Afonso and Teixeira (1999) [23], Johnson and McClelland (1998) [24], Koèenda (1998) [25], Robinson (1998) [26] and Kyrtsou, Labys and Terraza (2001) [27]. Also a number of software packages have been developed that implement the test in Matlab, C, Pascal [28].

For a time series $x_1, x_2, x_3, ..., x_T$ the test constructs the vectors Y_i =($x_i, x_{i+1}, ..., x_{i+M-1}$) for i=1,…,T-M+1, where the parameter Ì is called the ‘embedded’ dimension. BDS in order to describe the correlation between the members of a time series the BDS uses the *correlation integral* C_T (å,Ì). The correlation integral is the portion of pairs Y_i, Y_j lying within a distance of å of each other, for a given dimension M.

$$C_T\text{(å,Ì)=}\frac{2}{(T-M)(T-M+1)}\sum_{i=1}^{T-M}\ \sum_{j=i-1}^{T-M+1} X_{ij} \quad (3.8)$$

where X_{ij} equals 1 , if || Y_i - Y_j ||<å and equals 0 , if || Y_i - Y_j ||>=å . ||.|| is the Euclidean Distance or any other metric. It has been proved (by Brock et al [19]) that if Y_i , i=1,…,T-M+1 is IID then

$$\lim_{T \to +\infty} (C_T(e,M) - C_T(e,M)^M) = 0 \quad (3.9).$$

The BDS test is based on the discrepancy between (3.9) and the estimated correlation integrals of a given the time series.

$$BDS_T(e,\mathrm{M}) = \sqrt{\mathrm{T}}\,\frac{C_T(e,M) - C_T(e,1)}{s_{\mathrm{T}}(e,M)} \quad (3.10)$$

It has also been proved that if the null hypothesis of IID holds then $BDS_T(e,\mathrm{M})$ asymptotically follows standard normal distribution (Brock et at [19]). Where $s_{\mathrm{T}}(e,M)$ is the standard sample deviation of $\sqrt{\mathrm{T}}\ [C_T(e,M) - C_T(e,1)]$.

Before applying the BDS to a time series we have to define two parameters, the distance å and the embedded dimension M. Hsieh (1989) suggests for å the values 0.50, 0.75, 1.00, 1.25, 150; Girerd-Potin and Tamasco (1994) use the values 0.25, 0.50, 0.75, 1.00, 2.00; whereas Brock, Hsieh and LeBaron (1992) indicate å equal to 0.25, 0.50, 1.00, 1.50, 2.00 (sited in Kyrtsou et al [27]). For the embedded dimension M the values from 2~10 are suggested and commonly used, (Hsieh, sited in Kosfeld et al [20]). In our study we apply the BDS test for å: 0.25, 0.5, 0.75, 1, 1.25, 1.5 and we assign to M values from 2~10.

We applied the BDS test on the *Training sets* of both FTSE and S&P, the results we obtained are described by the following tables:

å	**M**	**BDS**	**å**	**M**	**BDS**
0.25	2	3.9348	1.00	2	4.4823
	3	6.3502		3	7.5251
	4	8.4794		4	9.2272
	5	9.2951		5	10.5554
	6	6.9667		6	11.9176
	7	6.0857		7	13.9070
	8	-3.1901		8	15.7533
	9	-9.0164		9	17.7098
	10	-7.4865		10	19.6577
0.50	2	3.7778	1.25	2	5.0746
	3	6.9087		3	8.0995
	4	8.7623		4	9.8059
	5	10.3623		5	11.0114
	6	11.8521		6	12.1681
	7	14.3864		7	13.7664
	8	16.3714		8	15.2015
	9	16.9425		9	16.6403
	10	13.2973		10	18.0410
0.75	2	4.0192	1.50	2	5.7753
	3	7.1819		3	8.8488

	4	8.8148		4	10.5193
	5	10.1926		5	11.5957
	6	11.5726		6	12.5416
	7	13.5498		7	13.8322
	8	15.5559		8	14.9446
	9	17.5544		9	16.0614
	10	19.5031		10	17.0744

Table 3.6: The BDS statistic for FTSE *excess returns* calculated for various values of å and M.

Since the BDS statistic follows a standard normal distribution N(0,1) in case that the series is random, the probability of the BDS to be out of the interval [-3.2905, 3.2905] is less than 0.1 %. From the table above it is clear that none of the values of BDS is in this interval. Therefore the BDS test gives us a strong indication that there is serial dependence in the FTSE *Training Set*.

å	**M**	**BDS**	**å**	**M**	**BDS**
0.25	2	4.7884	1.00	2	5.1170
	3	7.0203		3	7.3578
	4	8.7744		4	8.7767
	5	10.7293		5	10.7505
	6	12.3894		6	12.7289
	7	13.7319		7	14.9321
	8	9.9366		8	17.0988
	9	5.6920		9	19.6354
	10	-5.1686		10	22.3334
0.50	2	4.5101	1.25	2	5.3932
	3	6.6254		3	7.6109
	4	8.3061		4	8.9667
	5	10.5839		5	10.6594
	6	12.5269		6	12.3456
	7	15.0612		7	14.1469
	8	17.4929		8	15.8594
	9	20.5513		9	17.7409
	10	25.0099		10	19.5920
0.75	2	4.6522	1.50	2	5.7625
	3	6.8370		3	7.8769
	4	8.3712		4	9.1629
	5	10.5478		5	10.5516
	6	12.6793		6	11.9329
	7	15.1086		7	13.3657
	8	17.5403		8	14.7237
	9	20.6471		9	16.1866
	10	24.1886		10	17.5113

Table 3.7: The BDS statistic for S&P *excess returns* calculated for various values of å and M.

Similarly for the S&P case we get the BDS statistic even more clearly out of the interval [-3.2905, 3.2905]. Thus again we obtained strong evidence against the randomness of the series.

Summary

In this chapter we gave a description of the data we have collected in order to construct the *excess returns* time series we will attempt to predict. We also indicated the procedure we followed to construct this series. Lastly we applied two randomness tests and we proved that the FTSE and S&P *excess returns* series are not random therefore we proved that the task of predicting these series is not impossible.

Chapter 4

Models

In this chapter a detailed description of the models we will use to predict the market takes place. Furthermore selection of their optimum parameters is attempted. In order to achieve the optimum parameters setting we introduce methods such as Information Criteria and Genetic Algorithms.

4.1 Traditional Time Series Forecasting

4.1.1 Univariate and Multivariate linear regression

As has already been stated Traditional Time Series Forecasting models are widely used in econometrics for time series prediction. These models are capable of mining linear relationships between factors that influence the market and the value of the market. Therefore they assume that there is a function f such as [6]:

$$y_t= f(x_{1t}, x_{2t},\ldots, x_{kt})= \acute{a} + \hat{a}_1x_{1t} + \hat{a}_2x_{2t} + \ldots + \hat{a}_kx_{kt} + rs_t \quad \text{for } t=1,\ldots,N \quad (4.1)$$

where:

- x_i, are called *explanatory variables*
- y, is the *explained variable*
- $\hat{a}_i$ for i=1,..,k are the *coefficients*
- y_1, …, y_N, is the time series we are trying to predict.
- and rs_t is an *independent and identically distributed (IID) noise component*.

Based upon this assumption Traditional Time Series Forecasting models are attempting, given a sample of N examples $\{(x_{1t},\ldots,x_{kt}, y_t), t=1,\ldots,N\}$, to return a function g that approximates f in a sense that the norm of the error vector $E=(e_1,\ldots,e_t)$ is minimized. Each e_t is defined as $e_i=e(g(x_{1t},\ldots, x_{kt}), y_t)$ where e is an arbitrary error function. Function g is defined as:

$$\bar{y}_t = g(x_{1t}, x_{2t},\ldots, x_{kt}) = \bar{a} + \bar{b}_1 x_{1t} + \bar{b}_2 x_{2t} + \ldots + \bar{b}_k x_{kt} \text{ for } t=1,\ldots,N \quad (4.2)$$

where:

- $\bar{a}$, $\bar{b}_i$ for i=1,..,k are the *estimators of the coefficients*
- $\bar{y}_t$, is the prediction for y_t.

The error function e is usually either the mean square error (*mse*) function or the mean absolute error (*mae*) function:

$$\text{mse}=\frac{1}{N}\sum_{t=1}^{N}(y_t-\bar{y}_t)^2 \quad (4.3)$$

$$\text{mae}=\frac{1}{N}\sum_{t=1}^{N}| y_t-\bar{y}_t | \quad (4.4)$$

In our case study we will use with our regression model the *mse* function.

The *estimators of the coefficients* $\bar{a}$, $\bar{b}_i$, i=1,..,k given that the error function of the model is the *mse* function are calculated as follows:

The *mse* equals to

$$Q=\frac{1}{N}\sum_{t=1}^{N}(y_t-\bar{y}_t)^2=\frac{1}{N}\sum_{t=1}^{N}(y_t-\bar{a}-\bar{b}_1 x_{1t}-\bar{b}_2 x_{2t}-\ldots-\bar{b}_k x_{kt})^2,$$

our objective is to select $\bar{a}$, $\bar{b}_i$, i=1,..,k in a way that Q is minimized. But Q is minimized relatively to $\bar{a}$, $\bar{b}_i$, i=1,..,k at the points that its first partial derivatives are zero. Therefore the *estimators of the coefficients* come from the solution of the following system of equations:

$$\frac{\partial Q}{\partial \bar{a}}=0,\ \frac{\partial Q}{\partial \bar{b}_i}=0, \text{ for } i=1,\ldots,k$$

In (4.2) for k=1 we have a *univariate* regression model while for k>1 we get a *multivariate* regression model.

Furthermore having in mind that in our case we will use only lagged values to predict the returns of the market (4.2) is transformed to:

$$\bar{y}_t = \sum_{i=1}^{k} \bar{a} + \bar{b}_i \, y_{t-i} \text{ , for t=1,…,N (4.5)}$$

Equation (4.5) is known as *autoregressive model* of order k (or AR(k))[29]. We will use this model and we will adjust its coefficients on 90% of the data samples that we have and then we will measure its ability to generalize on the remaining 10% of our data.

For an autoregressive model apart from the error function the only parameter we have to define is the size of the lag k we are going to use. In order to do so we will use two different *Information Criteria* the *Akaike* and the *Bayesian Information Criterion*. These methods will allow us to select the value of k that offers the most information to our AR model without incorporating redundant complexity.

4.1.2 Use of Information Criteria to define the optimum lag structure

Information criteria are based on a principle proposed by Box and Jenkins (1976), the *principle of parsimony*, which states that given several adequate models, the one with the smallest possible number of parameters should be selected (sited in [30]). This principle can be formalized as a rule, in which the closeness of fit is traded-off against the number of parameters. In the time series literature several information criteria have been proposed, with the model which is selected being the one for which the criterion is *minimized*.

If N is the number of observations used to calculate the *estimators of the coefficients* of an AR model, Var($å_t$) is the variance of the residuals of the prediction model ($å_t$= y_t- $\bar{y}_t$) and k the number of *explanatory variables*, then the Akaike Information Criterion (AIC) is equal to:

$$\text{AIC=N log(Var}(å_t))\text{+2k (4.6)}$$

This gives a non-linear trade-off between the residuals variance and the value of k, since a model with a higher k will only be preferred if there is a proportionally larger fall in var($å_t$). Geweke and Meese (1981) have suggested the Bayesian Informaion Criterion defined by (sited in [30]):

$$\text{BIC}=\text{N}\log(\text{Var}(å_t))+\text{k}\log(\text{N}) \quad (4.7)$$

The BIC gives more weight to k than AIC, so that an increase in k requires a larger reduction in var($å_t$) under BIC than under AIC.

In our study we will calculate both BIC and AIC to select the optimum value for k. This way we will be able to define all the parameters related with our AR model and then use it as a prediction tool.

4.1.3 Evaluation of the AR model

The effectiveness of the AR model will be measured on a set of unseen data (~10% of all available data). The issue that arises is the choice of metrics we will use to evaluate its performance. Two types of metrics will be used in this study: *the mean absolute prediction error,* and *benchmarks that will compare the predictive power of our model with the Random Walk Model.* Both metrics are widely used in econometrics to describe the effectiveness of prediction models.

The mae is described by equation (4.4). It is quite clear that the closest to zero the *mae* is the better our model or else the closest our predictions $\bar{y}_t$ to the actual values y_t.

In order to use the second metric and compare the AR model with the Random Walk (RW) model we will use a coefficient suggested by Henry Theil (1966), the *Theil coefficient*. *Theil coefficient* (or *inequality coefficient*) is defined as [31]:

$$\text{Theil}=\frac{\sqrt{\sum_{t=1}^{N}(y_t-\bar{y}_t)^2}}{\sqrt{\sum_{t=1}^{N}y_t^2}} \quad (4.8)$$

It is the fraction of the *mse* of our model in respect to the *mse* of the RW. The prediction of the RW model for day t is in terms of returns 0% (the same price as day t-1). That is why the *mse* of the RW equals to the denominator of the fraction above. In case that Theil is less than one we have a model that outperforms the RW, while in case that Theil is close to one our model is as good as the RW.

Equation (4.8) is proposed for series of returns but in our case does not manage to depict properly the RW on the *excess returns* series we use, which is:

$$y_t = \ln(\frac{p_t}{p_{t-1}}) - \ln(\frac{c_{t-1}}{360}+1)$$

If we want to be precise and depict the *mse* of the RW model on the actual prices of the market, then we would have that the prediction of the price of the market for day t (p_t) is equal to the price of day t-1 (p_{t-1}):

$$\overset{=}{p}_t = p_{t-1}, \text{ where } \overset{=}{p}_t \text{ is the prediction of the RW for day t.}$$

From the last two equations we have that the prediction of the *excess returns* on day t according to the RW on prices is:

$$\overset{=}{y}_t = -\ln(\frac{c_{t-1}}{360}+1) \quad (4.9)$$

Using (4.9) the *Theil Coefficient* would be:

$$\text{Theil}=\frac{\sqrt{\sum_{t=1}^{N}(y_t-\bar{y}_t)^2}}{\sqrt{\sum_{t=1}^{N}(y_t-\ln(\frac{c_{t-1}}{360}+1))^2}} \quad (4.10)$$

A third approach is to assume that the *excess returns* time series itself follows a random walk. In this case the Theil that would compare our model with this type of RW would be:

$$\text{Theil}=\frac{\sqrt{\sum_{t=1}^{N}(y_t-\bar{y}_t)^2}}{\sqrt{\sum_{t=1}^{N}(y_t-y_{t-1})^2}} \quad (4.11)$$

The RW on the denominator this time indicates that the prediction of the return on day t is going to be equal with the return on day t-1.

In conclusion the metrics we will use to evaluate the AR model are:

Metric	Equation	Description
Mae	(4.4)	Mean absolute error
Theil A	(4.10)	Comparison of the AR model with the RW on prices p_t
Theil B	(4.11)	Comparison of the AR model with the RW on excess returns y_t
Theil C	(4.8)	Comparison of the AR model with the model which states that the price of the market tomorrow will be such that will allow us to have the same profits that we would have by investing in bonds.

Table 4.1: The metrics we will use to evaluate the AR models.

For simplicity reasons we will refer to equations (4.10), (4.11) and (4.8) as TheilA, TheilB and TheilC respectively.

4.1.4 Checking the residuals for non-linear patters

It is clear from the way it is defined that an AR model is capable of finding linear patterns that exist in the data set A={(x_{1t},…,x_{it}, y_t), t=1,…,N}, under the assumption that such patterns do exist. But an AR model has no ability to trace non-linear patterns that might exist in A.

Having this in mind we will apply the BDS test in order check for non-linear patterns in the residuals å$_t$ produced by the in sample observations. In case that BDS proves that non-linearity does not exist then the AR model should manage to capture all patterns that exist in our sample. If not, we have to apply other models capable of capturing not only the linear but also the non-linear patterns in A. This being the case we will use feed forward *Artificial Neural Networks*.

4.1.5 Software

The software we will use in order to calculate the *estimators of the coefficients* of the AR model as well as the optimum lag structure is Microfit 4.0 [32]. Microfit is a package for the econometric analysis of time series data. This package offers the possibility to create *multivariate* regression models and apply Akaike and Bayesian Information Criteria. As output from Microfit we will have the residuals of the AR model based on the in sample data and the predictions of the model on the out of sample (unseen) data. Then we will use code in Matlab to apply the BDS test on the residuals å$_t$ [28], calculated on the in sample data, and estimate the mae and the Theil coefficients (TheilA, TheilB and TheilC) on the unseen data.

4.2 Artificial Neural Networks

4.2.1 Description

The concept of Artificial Neural Networks (ANN) has a biological background. ANNs imitate loosely the way that the neurons in human brain function. An ANN is consisted of a set of interconnected processing units, *neurons*. According to an illustrative definition a NN is:

> '… an interconnected assembly of single processing elements, *units* or *nodes*, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or *weights*, obtained by a process of adaptation to, or *learning* from, a set of training patterns' [33].

ANNs provide a general, practical method for learning real-valued, discrete-valued, and vector valued functions from examples. Thus, in our project we will attempt to create ANNs that will learn the function that generates the time series we are trying to predict.

A brief description of the concepts related to NNs follows.

4.2.1.1 Neurons

A *neuron* is a processing unit that takes a number of inputs and gives a distinct output. The figure below depicts a single neuron with R inputs p_1,p_2, …, p_R, each input is weighted with a value w_{11}, w_{l2} , …, w_{lR} and the output of the neuron *a* equals to *f*(w_{11} p_1 + w_{12} p_2 + … + w_{1R} p_R).

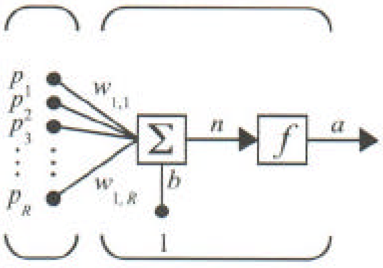


Figure 4.1: A simple neuron with R inputs.

Each *neuron* apart from the number of its inputs is characterized by the function *f* known as *transfer function*. The most commonly used transfer functions are: the *hardlimit*, the *pure linear*, the *sigmoid* and the *tansigmoid* function.

hardlimit	**purelinear**	**sigmoid**	**tansigmoid**
$f(x)=\begin{cases}1, x\geq 0\\0, x<0\end{cases}$	*f(x)=x*	*f(x)=* $\frac{1}{1+e^{-x}}$	*f(x)=* $\frac{2}{1+e^{-2n}}-1$
f(x) Î {0,1}	*f(x) Î (-¥ ,+¥)*	*f(x) Î [0,1]*	*f(x) Î [-1,1]*

Table 4.2: The most commonly used Transfer functions.

The preference on these functions derives from their characteristics. *Hardlimit* maps any value that belongs to (-∞,+∞) into two distinct values {0,1}, thus it is preferred for networks that perform classification tasks (multiplayer perceptrons MLP). *Sigmoid* and *tansigmoid*, known as squashing functions, map any value from (-∞,+∞) to the intervals [0,1] and [-1,1] respectively. Lastly *purelinear* is used due to its ability to return any real value and is mostly used at the neurons that are related with the output of the network.

4.2.1.2 Layers

As has already been referred the neurons of a network are distributed across layers. Each network has got exactly one input layer, zero or more hidden layers and one output layer. All of them apart from the input layer consist of neurons. The number of inputs to the NN equals to the dimension of our input samples, while the number of the outputs we want from the NN defines the number of neurons in the output layer. In our case the output layer will have exactly one neuron since the only output we want from the network is the prediction of tomorrow’s excess return. The mass of hidden layers as well as the mass of neurons in each hidden layer is proportional to the ability of the network to approximate more complicated functions. Of course this does not imply by any means that networks with complicated structures will always perform better. The reason for this is that the more complicated a network is the more sensitive it becomes to noise or else, it is easier to learn apart from the underlying function the noise that exists in the input data. Therefore it is clear that there is a trade off between the representational power of a network and the noise it will incorporate.

4.2.1.3 Weights Adjustment

The power of NN models lies in the way that their *weights* (inter unit-connection strengths) are adjusted. The procedure of adjusting the weights of a NN based on a specific dataset is referred as the training of the network on that set (*training set*). The basic idea behind training is that the network will be adjusted in a way that will be able to learn the patterns that lie in the *training set*. Using the adjusted network in future situations (unseen data) it will be able based on the patterns that learnt to generalize giving us the ability to make inferences. In our case we will train NN models on a part of our time series (*training set*) and we will measure their ability to generalize on the remaining part (*test set*). The size of the test set is usually selected to be 10% of the available samples [9].

The way that a network is trained is depicted by the following figure. Each sample consists of two parts the input and the target part (*supervised learning*). Initially the weights of the network are assigned random values (usually within [-1 1]). Then the input part of the first sample is presented to the network. The network computes an output based on: the values of its weights, the number of its layers and the type and mass of neurons per layer.

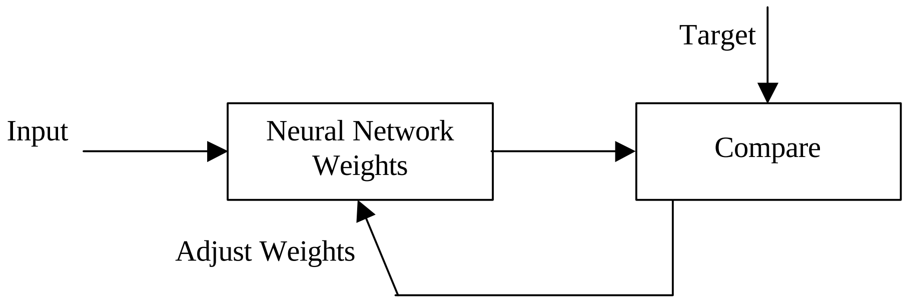


Figure 4.2: The training procedure of a Neural Network.

This output is compared with the target value of the sample and the weights of the network are adjusted in a way that a metric that describes the distance between outputs and targets is minimized.

There are two major categories of network training the *incremental* and the *batch* training. During the *incremental* training the weights of the network are adjusted each time that each one of the input samples are presented to the network, while in *batch* mode training the weights are adjusted only when all the training samples have been presented to the network [9]. The number of times that the training set will be feed to the network is called *number of epochs*.

Issues that arise and are related to the training of a network are: what exactly is the mechanism by which weights are updated, when does this iterative procedure cease, which metric is to be used to calculate the distance between targets and outputs? Answers to these questions are given in the next paragraphs.

Error Function

The *error function* or the *cost function* is used to measure the distance between the targets and the outputs of the network. The weights of the network are updated in the direction that makes the error function minimum. The most common error functions are the *mse* (4.3) and the *mae* (4.4). In our case study the networks we will be trained and tested using the *mse* function.

Training Algorithms

The mechanism of weights update is known as *training algorithm*. There are several training algorithms proposed in the literature. We will give a brief description of those that are related with the purposes of our study. The algorithms described here are related to *feed-forward* networks. A NN is characterized as *feed-forward* network “if it is possible to attach successive numbers to the inputs and to all of the hidden and output units such that each unit only receives connections from inputs or units having a smaller number”[34]. All these algorithms use the gradient of the *cost function* to determine how to adjust the weights to minimize the *cost function*. The gradient is determined using a technique called backpropagation, which involves performing computations backwards through the network. Then the weights are adjusted in the direction of the negative gradient.

Gradient descent

In this paragraph we will describe in detail the way that the weights of a feed forward network are updated using the backpropagation gradient descent algorithm. The following description is related with the *incremental training* mode.

Firstly we introduce the notion we will use. If E_N is the value of the error function for the sample N and $\vec{w}$ the vector with all the weights of the network then the *gradient* of E_N in respect to $\vec{w}$ is:

$$\nabla E_N(\vec{w}) = \left[\frac{\partial E_N}{\partial w_{11}}, \frac{\partial E_N}{\partial w_{12}}, \ldots, \frac{\partial E_N}{\partial w_{mn}}\right] \quad (4.12)$$

where w_{ji} is the weight that is related with the neuron j and its input i. “*When interpreted as a vector in weight space, the gradient specifies the direction that produces the steepest increase in E_N. The negative of this vector therefore gives the direction of the steepest decrease*”[9]. Based on this concept we are trying to update the weights of the network according to:

$$\overset{\rightarrow}{w'} = \vec{w} + \text{Ä}\vec{w} \quad (4.13)$$

where

$$\text{Ä}\vec{w} = -\text{ç}\nabla E_N(\vec{w}) \quad (4.14)$$

Here ç is a positive constant called the *learning rate*; the greater ç is the greater the change in $\vec{w}$.

We as well introduce the following notion:

- x_{ji}, is the i-th input of unit j, assuming that each neuron is assigned a number successively.
- w_{ji}, the weight associated with the i-th input to neuron j.
- $net_j = \sum_i w_{ji} x_{ji}$ (the weighted sum of inputs of neuron j)
- á_j, the output computed by node j.
- t_j, the target of output unit j.
- ó, the sigmoid function
- *outputs*, the set of nodes in the final layer of the network
- *Downstream(j)*, the set of neurons whose immediate inputs include the output of neuron j.

If we make the assumption that we have a network with neurons that use the sigmoid transfer function (ó) then we will try to calculate the *gradient* $\frac{\partial E_N}{\partial w_{ji}}$. Using the chain rule we have that:

$$\frac{\partial E_N}{\partial w_{ji}} = \frac{\partial E_N}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_N}{\partial net_j} x_{ji} \quad (4.15)$$

Given equation (4.15), our remaining task is to derive a convenient expression for $\frac{\partial E_N}{\partial net_j}$. We will consider two cases: (i) the case where neuron j is an output neuron for the network and (ii) and the case that j is an internal neuron (belongs to a hidden layer).

Case 1: Since net_j can influence the network only through á_j. Therefore we can have that:

$$\frac{\partial E_N}{\partial net_j} = \frac{\partial E_N}{\partial a_j}\frac{\partial a_j}{\partial net_j} \quad (4.16)$$

The error function we are using is: E$_N$= $\frac{1}{2}\sum_{k\in outputs}(t_k - a_k)^2$ (4.17), which is a variation of the *mse* function (4.3). This is done due to the fact that if you calculate the derivative of (4.17) $\frac{1}{2}$ is reduced while if we used the *mse* (4.3) $\frac{1}{N}$ would not have been reduced and we would have the factor $\frac{2}{N}$. Now by considering just the first term in equation (4.16)

$$\frac{\partial E_N}{\partial a_j} = \frac{\partial}{\partial a_j}\frac{1}{2}\sum_{k\in outputs}(t_k - a_k)^2$$

The derivatives $\frac{\partial}{\partial a_j}(t_k - a_k)$ will be zero apart from the case that k=j. Thus the above equation is transformed to:

$$\frac{\partial E_N}{\partial a_j} = \frac{\partial}{\partial a_i}\frac{1}{2}(t_j - a_j)^2 = \frac{1}{2}2(t_j - a_j)\frac{\partial (t_j - a_j)}{\partial a_j} = -(t_j - a_j) \quad (4.18)$$

For the second term in (4.16) we have that since we assumed that all the transfer function are sigmoids á$_j$=ó(net$_j$). Thus the derivative $\frac{\partial a_j}{\partial net_j}$ is just the derivative of function ó(net$_j$), which is equal to ó(net$_j$)(1– ó(net$_j$)). Therefore,

$$\frac{\partial a_j}{\partial net_j} = \frac{\partial s\,(net_j)}{\partial net_j} = \acute{a}_j(1 - \acute{a}_j) \quad (4.19)$$

Combining (4.16), (4.18) and (4.19) we get that:

$$\frac{\partial E_N}{\partial net_j} = -(t_j - a_j)\ \acute{a}_j(1 - \acute{a}_j) \quad (4.20)$$

Then from (4.13), (4.14) and (4.15):

$$w_{ji} = w_{ji} + \text{ç}\,(t_j - a_j)\ \acute{a}_j(1 - \acute{a}_j)\ x_{ji} \quad (4.21)$$

This is the way that the w$_{ji}$, where j is a node in the output layer, are updated for a given sample N.

Case 2: In this case j is an internal neuron; therefore the derivation of the training rule must take in account the indirect ways in which w_{ji} can influence E_N. At this point we have to notice that net_j can influence the network outputs only through the neurons in *Downstream*(j). Thus, we can have:

$$\frac{\partial E_N}{\partial net_j} = \sum_{k\in Downstream(j)} \frac{\partial E_N}{\partial net_k} \frac{\partial net_k}{\partial net_j}$$

If we set $\text{ä}_h = -\frac{\partial E_N}{\partial net_h}$ then we have:

$$\text{ä}_j = \sum_{k\in Downstream(j)} d_k \frac{\partial net_k}{\partial net_j}$$

$$= \sum_{k\in Downstream(j)} d_k \frac{\partial net_k}{\partial a_j} \frac{\partial a_j}{\partial net_j}$$

$$= \sum_{k\in Downstream(j)} d_k \; w_{kj} \frac{\partial a_j}{\partial net_j}$$

$$= \sum_{k\in Downstream(j)} d_k \; w_{kj} a_j (1 - a_j)$$

$$= a_j (1 - a_j) \sum_{k\in Downstream(j)} d_k \; w_{kj} \quad (4.22)$$

From (4.13), (4.14), (4.15) and (4.22) we have:

$$w'_{ji} = w_{ji} + \text{ç}\ \text{ä}_j\ x_{ji} \quad (4.23)$$

In conclusion we have that after a single sample N is presented to the network the error function is computed according to (4.17). Then the weights of the network are adjusted starting from the output layer and moving towards the first hidden layer according to:

1. For each output neuron k calculate:

$$\text{ä}_k \leftarrow (t_k - a_k)\ \text{á}_k(1 - \text{á}_k) \quad (4.24)$$

2. For each hidden neuron h calculate:

$$\text{ä}_h \leftarrow \text{á}_h(1 - \text{á}_h) \sum_{k\in Downstream(h)} d_k \; w_{kh} \quad (4.25)$$

3. Update each network weight w_{ji}:

$$w_{ji} = w_{ji} + \text{Ä}w_{ji} \quad (4.26)$$

where:

$$Äw_{ji}=çä_{j}x_{ji} \quad (4.27)$$

The same computations have to be done in the case of batch training, the only difference is that the gradient descent is calculated for all the training samples and it is summed up and then the adjustments of the weights take place based on the total of gradient descents.

The parameters that have to be set and are related to these two *training algorithms* are: the *error function*, the *learning rate* and the number *epochs.* Relatively to the learning rate it is clear that the larger the learning rate the bigger the step. If the learning rate is made too large the algorithm will become unstable and will not converge to the minimum of the error function. If the learning rate is set too small, the algorithm will take a long time to converge. A second issue is related with the number of epochs since we have to cease training before we overfit the network to the specific dataset (training set) canceling in that way its ability to generalize on unseen data.

Gradient descent with momentum

The gradient descent with momentum works in a similar way with the gradient descent but adjusts the weights of the network not based on (4.27) but according to [9]:

$$Äw_{ji}(n)=çä_{j}x_{ji} + ì\, Äw_{ji}(n-1) \quad (4.28)$$

$Äw_{ji}(n)$ indicates the change of $\vec{w}$ at the n-th iteration and ì is a term called *momentum*. The *momentum* takes values in [0,1]. When the momentum constant is close to zero a weight change is mainly based on the gradient descent. While when it is close to 1 the change in the weights is heavily influenced by their last change. The addition of ì $Äw_{ji}(n-1)$ term in (4.28) has as result not to permit to the training algorithm to get stuck to a shallow local minimum of the error function. It also has the effect of gradually increasing the step size on the search regions where the gradient is unchanging, thereby speeding convergence.

The g*radient descent with momentum* algorithm works in *incremental* as well as in *batch mode* and the parameters related to it are: the *error function*, the *learning rate* value, the *momentum* value, and the number of *epochs*.

Gradient Descent with variable learning rate

With the standard *gradient descent algorithm* the learning rate parameter is set to a constant value. The performance of the algorithm is sensitive to the proper setting of the *learning rate*, as it has been stated above. Even the optimum learning rate value it might be too small for parts of the training or too large for others. The solution to this problem is to set a variable *learning rate*. The rate is initialized having a specific value (*lr_init*). Then if the error decreases in a stable way the learning rate is increased by a constant value (*lr_incr*). This way the algorithm converges more rapidly. While if the error fluctuates by being increased and decreased in an unstable way then the *learning rate* is decreased by a constant value (*lr_decr*)[10].
The g*radient Descent with variable learning rate* can be applied only to batch training mode and the parameters related to it are: the error function, the *lr_init, lr_incr, lr_decr,* and the number of *epochs*.

Resilient Backpropagation

Multilayer networks typically use on their hidden layers squashing functions (sigmoid or tansigmoid). These functions are mainly used due to their ability to squash an infinity input range to a finite output space ([0,1] or [-1,1]). Squashing functions are characterized by the fact that their slope (or else their derivative) approach zero as the input gets large or more specifically when its absolute value gets large. This causes a problem when the algorithms described above are used since the gradient descent can have very small magnitude and therefore even in the case that converges it does so very slowly.

The *resilient backpropagation* algorithm updates the weights of the network based only on the sign of the gradient descent and not on its magnitude. The sign is used to define the direction to which the update will be done. The magnitude of the change is initialized for all weights to a value (*delta_init*). Each time that for two successive iterations the derivative of the error function in respect to a specific weight has the same sign, the magnitude for that weight is increased by a value (*delta_incr*). While in the case that for two successive iterations the sign of the derivative is not the same, the magnitude for the weight is decreased by a value (*delta_decr*). This way the training will converge even if the derivative of the error function in respect to the weights is too close to zero [10].

The *resilient backpropagation* algorithm works only in batch mode and the parameters related to it are; the *error function*, the *delta_init*, *delta_incr*, *delta_decr* and the number of *epochs*.

Stop Training

A significant decision related with the training of a NN is the time on which its weight adjustment will be ceased. As we have explained so far over-trained networks become over-fitted to the training set and they are useless in generalizing and inferring from unseen data. While under-trained networks do not manage to learn all the patterns in the underlying data and due to this reason under perform on unseen data. Therefore there is a tradeoff between over-training and under-training our networks.

The methodology that is used to overcome this problem is called *validation* of the trained network. Apart from the *training set* a second set, the *validation set*, which contains the same number of samples is used. The weights of the network are adjusted using the samples in the *training set* only. Each time that the weights of the network are adjusted its performance (in terms of error function) is measured on the *validation* set. During the initial period of training both the errors on *training* and *validation* sets are decreased. This is due to the fact that the network starts to learn the patterns that exist in the data. From a number of iterations of the training algorithm and beyond the network will start to overfit to the training set. If this is the case, the error in the validation set will start to rise. In the case that this divergence continues for a number of iterations the training is ceased. The output of this procedure would be a not overfitted network.

After describing the way that a NN works and the parameters that are related to its performance we select these parameters in a way that will allow us to achieve optimum performance in the task we are aiming to accomplish. The methodology will follow in order to define these parameters is described in the next paragraph.

4.2.2 Parameters Setting

4.2.2.1 Neurons

The properties related to a neuron are the transfer function it uses as well as the way it processes its inputs before feeding them to the transfer function. The NNs we will create use neurons that preprocess the input data as follows: If $x_1,\dots, x_N$ are the inputs to the neuron and $w_1,\dots, w_N$ their weights the value fed to the transfer function would be $\sum_{i=1}^{N} x_i w_i$. In order to define the transfer functions of the neurons in our networks we will use the work of Cybenko (1988) who proved that any function can be approximated to arbitrary accuracy by a network with three layers of neurons. More specifically he defined that this network would have linear transfer functions in its output layer and squashing transfer functions in the other two hidden layers (sited in [9]). Therefore the neurons in the output layer will use the *purelinear* function while the neurons in the hidden layers the *tansigmoid* function. We select the *tansigmoid* and not the *sigmoid* since the excess returns time series contains values in [-1,1], thus the representational abilities of a *tansigmoid* function fit in a better way the series we attempt to predict comparing to those of the *sigmoid's*.

4.2.2.2 Layers

The NNs that interest us has the following structure: x-y-z-1 where x, y can be any integer greater than one, while z can be any non-negative integer. So far we have fully defined the characteristics of the output layer and for the hidden layers the properties of their nodes. What remains open is the number of hidden units per layer as well as the number of inputs. Since there is no rational way of selecting one structure and neglecting the others we will use a search algorithm to help us to choose the optimum number of units per layer. The algorithm we will use is a *Genetic Algorithm (GA)*. The GA will search a part of the space defined by x-y-z-1 and will converge towards the network structures that perform better on our task. Detailed description of the way that a

GA works as well as how it will be used in the frame of our study will be presented in following paragraphs.

4.2.2.3 Weights Adjustment

Error function

The error function we will use is the *mse* function. We select the *mse* function in both cases (AR model or a NN model) to minimize the same cost function. This way the comparison of the models will be more representative. In fact in the case of NNs we will not use the *mse* as it is described by (4.3) but a slight variation described by (4.17) due to reasons described in paragraph 4.2.2.3. This change does not alter the cost function since both (4.3) and (4.17) have minima at the same spots.

Training algorithm

The training algorithms we have referred to so far will be tested on a specific architecture. This architecture will be used as benchmark in order to see which of them converges faster. In total six different training algorithms will be tested:

- Gradient descent (incremental mode)
- Gradient descent (batch mode)
- Gradient descent with momentum (incremental mode)
- Gradient descent with momentum (batch mode)
- Gradient descent with variable learning rate
- Resilient Backpropagation

Then we will use the fastest of these algorithms in all our experiments. Given the fact that the performance of the algorithms above is related with a number of parameters the result that we will have might not be the optimum solution. But we will try to get an indication of which of these algorithms given a small variation in their parameters converges faster on our problem.

Stop Training

For deciding where to stop the training we will use as validation set the ‘*Validation1 set*’. Therefore we will train a network on the ‘*Training1 set*’ and validate its performance on the ‘*Validation1 set’*. This procedure will give us the number of *epochs* below which the specific network will not be over-trained (let’s assume *k*). Then we will train the network for *k epochs* on the new set that will come from the concatenation of the *Training1* and *Validation1* sets (‘*Training set’*).

4.2.3 Genetic Algorithms

4.2.3.1 Description

The basic principles of Genetic Algorithms (GAs) were proposed by Holland in 1975 (sited in [35]). Genetic Algorithms are inspired by the mechanism of natural selection where stronger individuals are likely the winners in a competing environment. They have been applied with success to domains such as: optimization, automatic programming, machine learning, economics, immune systems, ecology, population genetics, evolution and learning, social systems [36]. The Genetic Algorithms are defined as:

> “… search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. While randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance”[35].

In the present study we will use GAs to search for the optimum topology (architecture) of a NN given a specific fitness criterion.

4.2.3.2 A Conventional Genetic Algorithm

A genetic algorithm has three major components. The first component is related with the creation of an *initial population* of *m* randomly selected individuals. The *initial population* shapes the first *generation*. The second component inputs *m* individuals and gives as output an evaluation for each of them based on an objective function known as *fitness function*. This evaluation describes how close to our demands each one of these m individuals is. Finally the third component is responsible for the formulation of the next generation. A new generation is formed based on the fittest individuals of the previous one. This procedure of evaluation of generation N and production of generation N+1 (based on N) is iterated until a performance criterion is met. The creation of offspring based on the fittest individuals of the previous generation is known as *breeding*. The *breeding* procedure includes three basic genetic operations: *reproduction*, *crossover* and *mutation*.

Reproduction selects probabilistically one of the fittest individuals of generation N and passes it to generation N+1 without applying any changes to it. On the other hand, *crossover* selects probabilistically two of fittest individuals of generation N; then in a random way chooses a number of their characteristics and exchanges them in a way that the chosen characteristics of the first individual would be obtained by the second an vice versa. Following this procedure creates two new offspring that both belong to the new generation. Finally the *mutation* selects probabilistically one of the fittest individuals and changes a number of its characteristics in a random way. The offspring that comes out of this transformation is passed to the next generation [36].

The way that a conventional GA works by combining the three components described above is depicted in the following flowchart [37]:

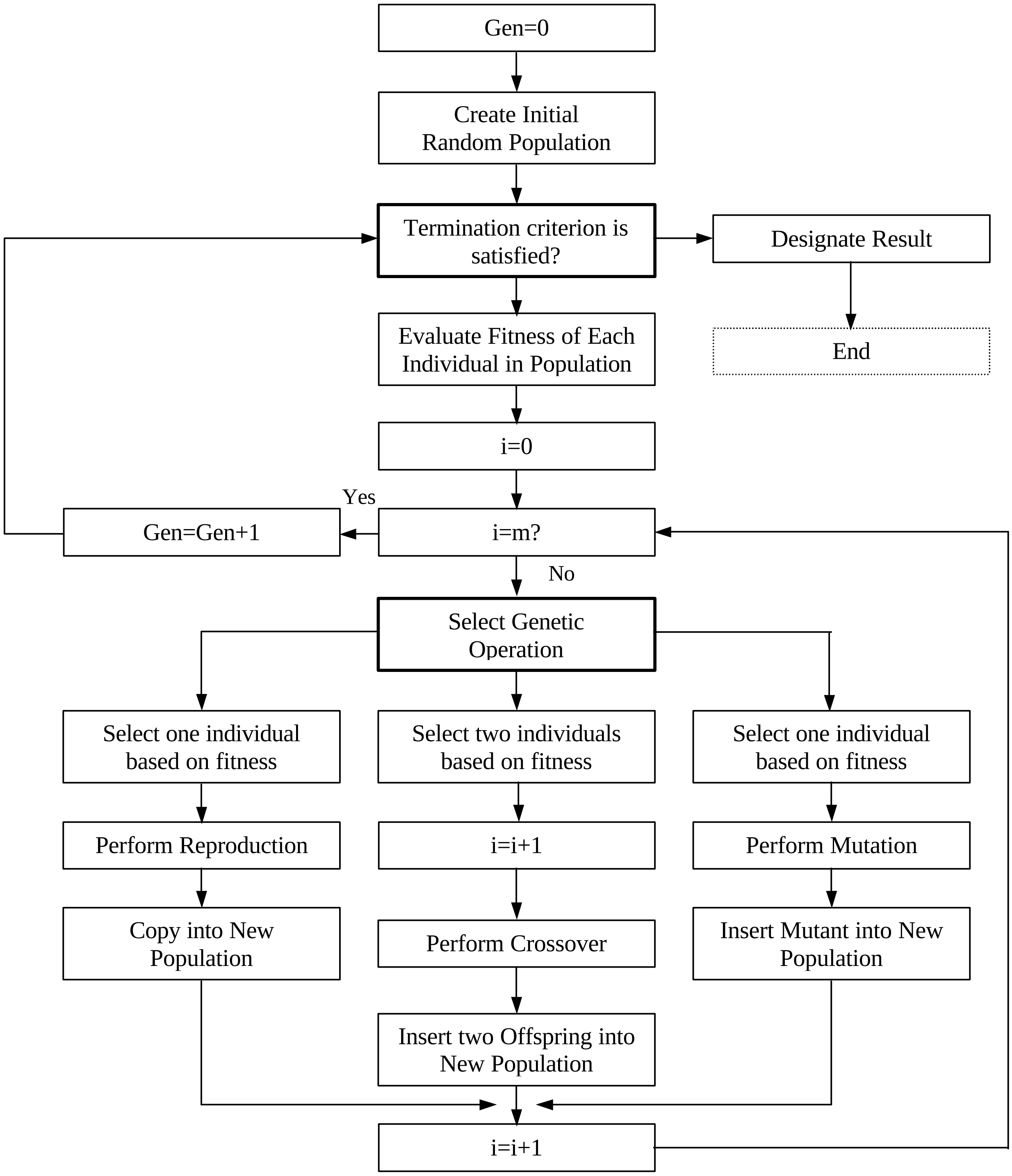

Gen=0
Create Initial Random Population
Termination criterion is satisfied?
Designate Result
End
Evaluate Fitness of Each Individual in Population
i=0
Yes
Gen=Gen+1
i=m?
No
Select Genetic Operation
Select one individual based on fitness
Select two individuals based on fitness
Select one individual based on fitness
Perform Reproduction
i=i+1
Perform Mutation
Copy into New Population
Perform Crossover
Insert Mutant into New Population
Insert two Offspring into New Population
i=i+1

Figure 4.3: A conventional Genetic Algorithm.

As it has been stated each one of the individuals have a certain number of characteristics. For these characteristics the term *genes* is used. Furthermore according to the biological paradigm the set off all *genes* of an individual form its *chromosome*. Thus each individual is fully depicted by its *chromosome* and each generation can be fully described by a set of m *chromosomes*.

It is clear from the flowchart of the GA that each member of a new generation comes either from a *reproduction, crossover* or *mutation* operation. The operation that will be applied each time is selected based upon a probabilistic schema. Each one of the three operations is related with a probability $P_{reproduction}$, $P_{crossover}$, and $P_{mutation}$ in a way that

$$P_{reproduction} + P_{crossover} + P_{mutation}=1$$

Therefore the number of offspring that come from reproduction, crossover or mutation is proportional to $P_{reproduction}$, $P_{crossover}$, and $P_{mutation}$ respectively [37].

Relatively to the way that the selection of an individual (or two in the case of crossover) according to its fitness is done, again the selection is based on a probabilistically method. The selection is implemented by a scheme known in literature as *roulette wheel* [36,37,38]. In GAs the higher the fitness value of an individual the better the individual. Based upon this fact a *roulette wheel* is created by the following steps [38]:

- Place all population members in a specific sequence
- Sum the fitness off all population members F_{sum}.
- Generate a random number (r) between 0 and F_{sum}.
- Return the first population member whose fitness value added to the fitness of the preceding population members, is greater than or equal to (r).

In case we want to select two individuals (crossover) we create the *roulette wheel* twice, the first time using all fitness values and the second time using all apart from the fitness value that corresponds to the chromosome selected from the firstly created *roulette wheel*. This guarantee us that we will not crossover between the same chromosomes, which would mean that the crossover operation would be equivalent to a reproduction operation twice on the same chromosome.

Another issue that is related to a GA is the nature of the termination criterion. This can be either a number of evolution cycles (generations), or the amount of variation of individuals between two successive generations, or a pre-defined value of fitness [38].

4.2.3.3 A GA that Defines the NN's Structure

In this paragraph we describe the way we use Genetic Algorithms in order to search a space of NN topologies and select those that match optimally our criteria. The topologies that interest us have at most two hidden layers and their output layer has one neuron (*x-y-z-1*). Due to computational limitations it is not possible search the full space defined by '*x-y-z-1*'. What we can do is to search the space defined by *xMax-yMax-zMax-1*, where *xMax*, *yMax* and *zMax* are upper limits we set for *x*, *y* and *z* respectively.

Initial Generation

Firstly we have to define the *genes* and the *chromosome* of an individual. In our case we have three *genes,* which describe the number of inputs and the number of neurons in each one of the two hidden layers. The values that these *genes* can have are:

- *x*: integer values from 1 to *xMax*
- *y*: integer values from 1 to *yMax*
- *z*: integer values from 0 to *zMax*

z equal to zero implies absence of the second hidden layer. We have preferred not to consider NNs that have no hidden layers (*y*=0 and *z*=0) because they depict linear models such as the ones we have already considered with the AR models. Having in mind the above we define a *chromosome* as the triplet '*x y z*'.

The initial population consists of *m* randomly selected *chromosomes*. The way that this population is initialized does not permit replicates, which implies that we cannot have chromosome 'a b c' more than once in the initial generation. Thus each time that a new chromosome is generated it is compared with the ones created so far, if it has a replicate it is neglected and a new one is generated if not it becomes a member of the first generation. This approach concerns only the first generation and it is adopted because it allows us to start our search from a more diverse position.

Fitness Function

Once the first generation has been defined a fitness mechanism has to be used in order to evaluate each one of the *m chromosomes* of a generation. The present GA allows us to use a number of different ways to calculate the fitness of a chromosome. It uses four different functions either *TheilA*, or *TheilB, or TheilC*, or the *mae*. Assuming that the chromosome we want to evaluate is 'x y z' the steps that describe this evaluation procedure are:

- Create a NN using the parameters described in 4.2.2 and architecture 'x-y-z-1'.
- Train it and stop its training according to the methodology analyzed in 4.2.2.
- Measure its performance either on *TheilA*, or *TheilB,* or *TheilC* or the *mae*.

This way all the members of the generation will have a value that depicts how well they perform based on one of the metrics mentioned above. This initial form of fitness function is not suitable for the GA for two reasons. Firstly because the fittest the individual the closest the value to zero, which implies that the *roulette wheel* selection will not work properly in case it will be based on these values. Secondly these values for various chromosomes differ by small quantities therefore if the probability of selecting chromosome A over B is linearly proportional to them, this would have as result to flatten our selection. So if this would be the case chromosome A will have approximately the same probability to be selected comparing to B even though their fitness values are 'significantly' different for our domain.

In order to overcome the first problem we reverse the values by applying the following linear transformation [38]:

$$f_i = -g_i + \max\{g_i\} \quad i=1,\ldots,m \quad (4.29)$$

The greater the value of f_i the best chromosome i. Based on f_i we can use the *roulette wheel* selection but still there is a problem with the fact that the magnitudes of f_i are very close. In literature two ways have been proposed to overcome this type of problem the *Power Law Scaling* and the *Sigma Truncation* [38]. In *Power Law Scaling* you apply the following transformation: $Fit_i=f_i^k$ (4.30) where k is a problem dependent constant or variable. In *Sigma Truncation* we have that: $Fit_i= f_i - (\bar{f_i} - c\,\mathrm{var}(f_i))$ (4.31) where c is a small integer and $\bar{f_i}$ is the mean of values f_i for i=1,…,m [38].

In the current GA we adopt a variation of the second approach. Due to the nature of f_i (values close to zero) we did not manage to find a value for k that would globally (for *TheilA*, *TheilB*, *TheilC* and *mae*) give us fitness values with "clear" differences such that the *roulette wheel* would premium the best chromosomes. The transformation we use is:

$Fit_i = f_i - \frac{2}{3}\bar{f_i}$. This way the individuals that have f_i less than 2/3 of the mean will have a negative Fit_i value. The individuals that form the new generation come only from those with a positive Fit_i value. Therefore the *roulette wheel* is formed only from the fitness values (Fit_i) that belong to individuals with positive Fit_i.

Breeding

Each one of the chromosomes of a new generation is created by either a *reproduction*, or *crossover,* or *mutation* operation. The selection of operation is done probabilistically with the method described in paragraph 4.2.3.2.

The *reproduction* selects a chromosome from generation N based on a *roulette wheel* created on the *fitness function* (Fit_i) and pass it to generation N+1.

Crossover operation selects the chromosomes of two of the fittest individuals C_1 and C_2 where C_1='a_1 b_1 c_1' and C_2='a_2 b_2 c_2'. It then chooses randomly the genes on which *crossover* will be done and produces two offspring that are both passed to the next generation. For example if it is indicated that the *crossover* will be done on *genes* a and c we would have the offspring C_1^*=' a_2 b_1 c_2' and C_2^*='a_1 b_2 c_1'.

Lastly *mutation* selects probabilistically a chromosome that belongs to one of the fittest individuals C='a b c' and changes (mutates) a random number of *genes*. For example if in C *genes* a and b are mutated then a new offspring C^* is created where C^*='a^* b^* c' where a^* is a random number between 1 and xMax and b^* a random number between 1 and yMax.

Termination Criterion

The termination criterion we have chosen for the algorithm is related with the number of generations formed. Thus when the algorithm reaches a specific number of generations (*MaxGen*) it stops and returns a structure with all chromosomes of the individuals considered clustered in generations along with their g_i value.

4.2.4 Evaluation of the NN model

The procedure of selecting the optimum NN will result a number of networks, which are expected to have the highest possible performance. This set of networks will be evaluated on unseen data (10% of the available data) and more specific on the same dataset that the AR models will be evaluated on (‘*Test set*’). The metrics we will use are again: the mae, TheilA, TheilB and TheilC. This way the comparison of the NN and the AR models is going to be feasible.

4.2.5 Software

The software we use to train, validate and test the *feed-forward* NNs we consider in this study is the “*Neural Networks Toolbox*” of *Matlab 5.2* [10,39]. The Genetic Algorithm is implemented on *Matlab 5.2* as well. It is created based on the principles described in paragraphs 4.2.3.2 and 4.2.3.3. The code that implements the Genetic Algorithm is included in the CD-ROM, which is sited at the end of this study.

Summary

In this chapter we have presented the models we will use to predict the excess returns time series. We have also attempted to select rationally their parameters and in some cases to describe a methodology that would allow us to do so. For the AR models we decided to use the AIC and the BIC to define the best lag structure we can have; while for the NNs we will test a number of different training algorithms and we will define their optimum structure using a GA. All the models will be evaluated on the same dataset in terms of four different metrics the *mae,TheilA ,TheilB and TheilC.*

Chapter 5

Experiments and Results

In Chapter 3 we proved that both FTSE and S&P *excess return* time series do not fluctuate randomly. In this chapter we describe the experiments we undertook to predict these *excess returns*. We also report the results we obtained along with the parameters used in order to obtain them. Two experiments took place, using the autoregressive (AR) and the neural network (NN) models. In each experiment a rational method was used to select the best model and its performance was evaluated on unseen data based on various metrics.

5.1 Experiment I: Prediction Using Autoregressive Models

5.1.1 Description

In this experiment we applied AR models to predict the *excess returns* time series for both FTSE and S&P. The only parameter we had to define with the AR models was the lag structure. To do so we used the Akaike and the Bayesian Information Criterion. For the lag structure each of these criteria indicated, we constructed the relative model and adjusted its parameters on the *Training Set* (*Training1 Set+Validation1 Set*) and then we measured its performance on the *Test Set*.

5.1.2 Application of Akaike and Bayesian Information Criteria

For the FTSE *Training Set* we applied the Akaike and the Bayesian Information Criterion. We calculated the value that AIC and BIC gave us for lag structures from 1 to 24. The values we obtained are presented in the following figure:

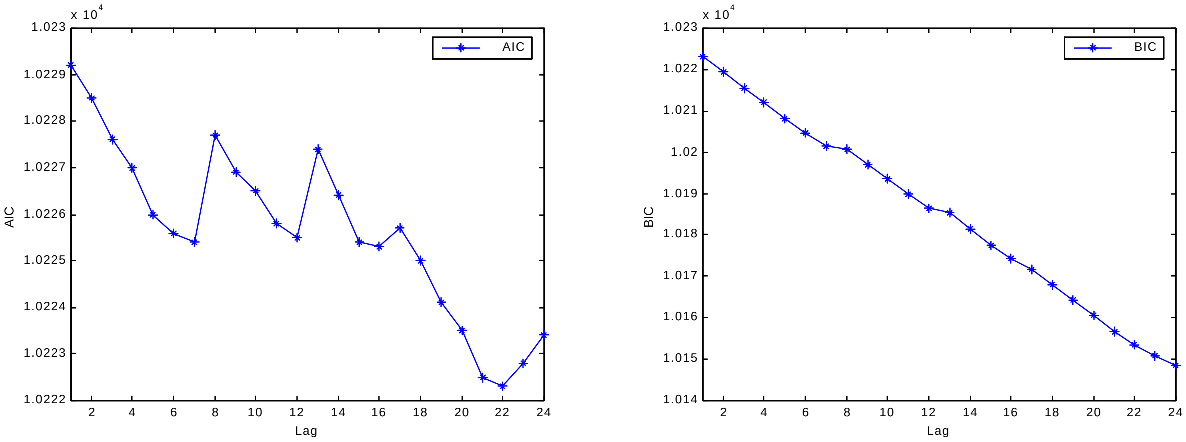


Figure 5.1: AIC and BIC for the FTSE data.

Both criteria are at their maximum for lag structure of one. Thus we attempted to predict the FTSE excess returns using AR model with lag one.

Similarly we calculated the AIC and the BIC for the S&P for lag structures between 1 and 24 and we obtained the following results:

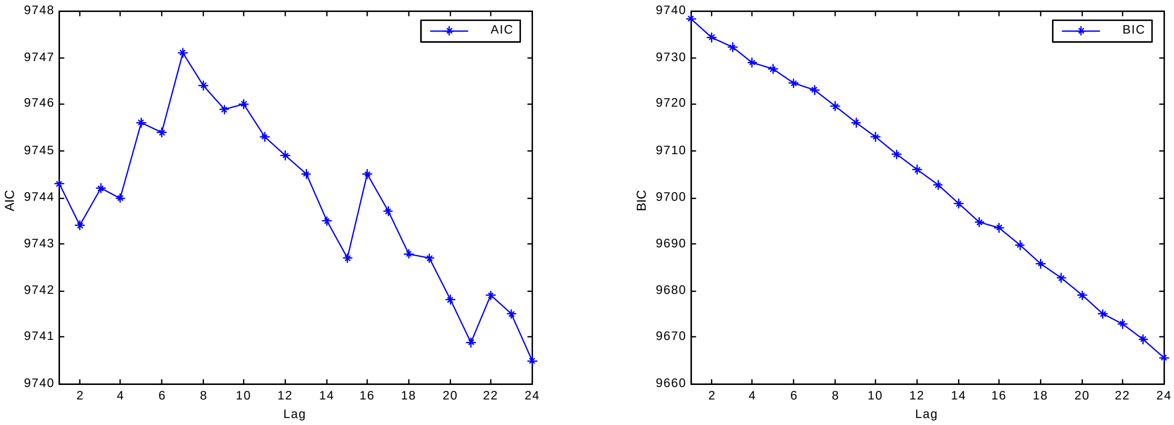


Figure 5.2: AIC and BIC for the S&P data.

For the S&P data the information criteria indicated different lag structures. AIC indicated lag 7 while BIC indicated lag 1. This difference, as we explained in chapter 4, is due to the fact that the BIC is stricter in adding extra parameters to the model comparing to the AIC. The strategy we adopted was that we created two AR models using lags of seven and one respectively then we adjusted their parameters on the *Training Set* and tested their performance on the *Test set*.

5.1.3 AR Model Adjustment

The adjustment of the coefficients of the AR model for the FTSE (using lag 1) gave us the following model:

$$AR_{FTSE}(1)\text{: } y_t = \quad 0.0001128+0.097838\ y_{t-1} \qquad (5.1)$$

The model was adjusted using the first 2974 observations of the FTSE *excess returns* series.

On the other hand, the AR models we obtained from the S&P data for lags of 1 and 7 are described by the following equations:

$$AR_{S\&P}(1)\text{: } y_t = \quad 0.0004262 - 0.0071121\ y_{t-1} \qquad (5.2)$$

$$\begin{aligned} AR_{S\&P}(7)\text{: } y_t = \quad & 0.0005265 - 0.0046695 y_{t-1} \\ & -0.016224 y_{t-2} - 0.043845 y_{t-3} \\ & -0.020708 y_{t-4} - 0.047547 y_{t-5} \\ & -0.021831 y_{t-6} - 0.044306 y_{t-7} \qquad (5.3) \end{aligned}$$

Both of these AR models were adjusted on the first 2976 observations of the S&P *excess returns* time series (*Training Set*).

5.1.4 Evaluation of the AR models

In this paragraph we present the evaluation of the performance of the AR models based on four different metrics: TheilA, TheilB, TheilC and mae. The evaluation of the all AR models is based on the *Test Set*. Equation (5.1) gave us the following Theils and mae:

Metric	**FTSE Lag 1**
TheilA	1.00015406818713
TheilB	0.72262111020458
TheilC	1.00013360604923
Mae	0.00859109183946

Table 5.1: Evaluation of $AR_{FTSE}(1)$

From these results it is clear that the AR model manages to beat only the random walk model based on the *excess returns* (TheilB). Theils A and C indicate that there are naï ve models that can perform as well as the AR model. Therefore using this linear model we did not manage to have robust predictions.

For the S&P dataset (*Test Set*) we obtained for each one of the models (5.2) and (5.3) the following results:

Metric	S&P Lag 1	S&P Lag 7
TheilA	1.00092521883907	1.00022106510965
TheilB	0.70577780771898	0.70076320655560
TheilC	1.00082560834712	1.00024229730378
Mae	0.01037401820067	0.01030333781570

Table 5.2: Evaluation of $AR_{S\&P}(1)$ and $AR_{S\&P}(7)$

Both AR models with lag one and seven, in this case too, did not manage to give robust predictions. The performance of $AR_{S\&P}(7)$ and $AR_{S\&P}(1)$ according to all metrics is almost the same. Therefore the inclusion of the extra number of lags did not help the model to give better predictions.

A comparison of the predictions for the FTSE and S&P datasets indicates that we obtained different results for TheilB and for the *mae*. TheilB for FTSE was 0.722 while for S&P we obtained a significantly better result 0.700. On the contrary according to the *mae* the AR models performed better on the FTSE than they did on the S&P. The last comparison is valid, although mae is a scale variant meric[8], due to the fact that the magnitude of the values of the points in both *excess return* series we are trying to predict is similar.

In conclusion, the application of the AR models in our task did not manage to help us have better predictions comparing to those of naï ve prediction models. Additionally from all three benchmarks we have used the ones that involved TheilA and TheilC proved to be the harsher. The random walk model on the *excess returns* proved to be a model easy to beat (TheilB).

[8] Its value depends on the scale of the data we are trying to predict

5.1.5 Investigating for Non-linear Residuals

As we have shown the AR models did not help us to overcome the predictive power of naï ve models. In this paragraph we investigate the residuals of the AR models on the *Training Set* in order to find out whether there are remaining patterns in them that the AR models did not manage to capture due to their linear character. In order to do so we applied the BDS test on the residuals of equations (5.1), (5.2) and (5.3).

The BDS test on the residuals of (5.1) for the parameters selected in paragraph 3.3.3 gave us the following results:

å	**M**	**BDS**	**å**	**M**	**BDS**
0.25	2	4.2945	1.00	2	5.1469
	3	6.7116		3	8.1120
	4	8.3880		4	9.6382
	5	9.6352		5	10.8927
	6	7.3070		6	12.3159
	7	2.2999		7	14.3000
	8	-2.8054		8	16.1296
	9	-9.3721		9	17.9941
	10	-7.7163		10	20.0585
0.5	2	4.5195	1.25	2	5.7956
	3	7.4705		3	8.6833
	4	9.2221		4	10.1793
	5	10.8393		5	11.3325
	6	12.6673		6	12.5669
	7	15.5165		7	14.1998
	8	17.8999		8	15.6098
	9	18.6337		9	16.9470
	10	17.3384		10	18.4779
0.75	2	4.7781	1.5	2	6.6440
	3	7.7361		3	9.5223
	4	9.2349		4	10.9874
	5	10.5312		5	11.9804
	6	11.9115		6	13.0024
	7	13.9996		7	14.3276
	8	16.0357		8	15.4195
	9	18.3490		9	16.3955
	10	20.7299		10	17.5797

Table 5.3: The BDS test on the residuals of $AR_{FTSE}(1)$.

According to BDS test the BDS statistic for a random time series follows standard normal distribution N(0,1). Therefore since in a N(0,1) 99% of the samples belong to the interval [-2.5758, 2.5758] the magnitudes of the BDS statistic presented in the table above give us very strong indications that the residuals of this AR model are not IID; thus, they contain patterns.

For the residuals of the $AR_{S\&P}(1)$ we obtained the following values for the BDS statistic:

å	M	BDS	å	M	BDS
0.25	2	4.8240	1.00	2	5.1482
	3	7.0039		3	7.3134
	4	8.5678		4	8.5837
	5	10.2717		5	10.6012
	6	11.9532		6	12.6366
	7	13.5690		7	14.8745
	8	9.8014		8	17.0307
	9	5.5463		9	19.4555
	10	-5.1100		10	22.2355
0.5	2	4.4807	1.25	2	5.4220
	3	6.5671		3	7.5899
	4	8.1318		4	8.7670
	5	10.3735		5	10.5390
	6	12.2892		6	12.3065
	7	14.7803		7	14.1371
	8	17.0889		8	15.8366
	9	20.0027		9	17.5773
	10	24.2057		10	19.5444
0.75	2	4.6547	1.5	2	5.7821
	3	6.7947		3	7.8644
	4	8.2262		4	8.9259
	5	10.4272		5	10.4560
	6	12.5967		6	11.9403
	7	15.0374		7	13.3968
	8	17.4382		8	14.7473
	9	20.4227		9	16.0366
	10	23.9908		10	17.5164

Table 5.4: The BDS test on the residuals of $AR_{S\&P}(1)$.

These again indicate lack of randomness in the residuals of model (5.2). Moreover we observe that for the S&P data the BDS statistic has more extreme values that indicate with greater certainty that there are patterns in the underlying data.

Finally the results of the BDS test on the residuals of (5.3) are the ones presented in the next table:

å	M	BDS	å	M	BDS
0.25	2	4.8351	1.00	2	4.9570
	3	7.1520		3	7.1002
	4	8.8854		4	8.5542
	5	10.6666		5	10.4192
	6	14.5545		6	12.3562
	7	17.7491		7	14.5539
	8	14.0296		8	16.6036
	9	19.6583		9	19.0126
	10	-5.1917		10	21.7998
0.5	2	4.4540	1.25	2	5.1761
	3	6.5743		3	7.3170
	4	8.1465		4	8.7586
	5	10.2581		5	10.4384
	6	12.1726		6	12.0562

	7	14.5305		7	13.8607
	8	16.7316		8	15.4371
	9	19.4631		9	17.1770
	10	22.8129		10	19.0589
0.75	2	4.6998	1.5	2	5.5171
	3	6.8104		3	7.5818
	4	8.3812		4	9.0058
	5	10.3980		5	10.4667
	6	12.4742		6	11.7427
	7	14.7956		7	13.1882
	8	17.0902		8	14.4146
	9	20.0824		9	15.7054
	10	23.6145		10	17.1291

Table 5.5: The BDS test on the residuals of AR$_{S\&P}$(7).

These results also lead us to the same conclusions as the one obtained for AR$_{S\&P}$(1). Furthermore we observe that all BDS tests undertaken indicate that the bigger the lag structure (M) the clearer the higher the certainty for correlations in the underlying data.

Therefore our conclusion from the study of the residuals of the AR models is that these specific models did not manage to capture the patterns that according to the BDS test do exist in the residuals. Having in mind that the AR models are capable of capturing only linear patterns in the second part of this chapter we present another experiment we undertook using Neural Network models, which are capable of capturing linear but also non-linear patterns.

5.2 Experiment II: Prediction Using Neural Networks

5.2.1 Description

In the second experiment we attempted to predict the *excess returns* time series using neural network models. The results of Experiment I gave us extra motivation to apply this type of models since the AR models not only did not manage to give better predictions than the naï ve prediction models, but it seems that they did not manage to capture the patterns in the underling data.

Similarly to the first experiment our aim was to select those NN models that perform optimally in the frame of our study. The parameters that influence the performance of a NN model are numerous and more complex than those that influence the performance of an AR model. In chapter 4 we selected some of these parameters rationally and we proposed the methods which we will use to define the rest. Here we present an experiment we undertook in which we applied these methods and we obtained a set of NNs that serve the purposes of our study optimally.

The experiment consisted of three phases (Figure 5.3). In the first phase a genetic algorithm (GA) searched the space of NNs with different structures and resulted a generation with the fittest of all networks searched based on a metric which was either: TheilA or TheilB or TheilC or MAE. The GA search was repeated three times for each metric. Then the best three networks were selected from each repetition of the GA search and for each one of the metrics. The output of the first phase was a set of thirty-six network structures.

In the second phase for each one of the thirty-six resulting network structures we applied the following procedure. We trained (on *Training1 set*) and validated (on *Validation1 set*) the network. Then we used the indicated number of epochs from the validation procedure and based on it we retrained the network on the *Training1* plus the *Validation1 set*. Finally we tested the performance of the network on unseen data (*Validation2 set*). This procedure was repeated 50 times for each network structure for random initializations of its weights. From the nine networks for each performance statistic, we selected the most stable in terms of standard deviation of their performance. Thus the output of the second phase was a set of four network structures.

During the third phase for each one of these four networks we applied the following procedure 50 times. We trained each network on the first half of the *Training Set* and we used the remaining half for validation. Then, using the indicated epochs by the validation procedure, we retrained the network on the complete *Training Set*. Finally we tested the network on the *Test Set* calculating all four metrics. The performance for each network on each metric was measured again in terms of standard deviation and mean of its performance over 50 times that it was trained, validated and tested. The following figure depicts all the phases of Experiment II:

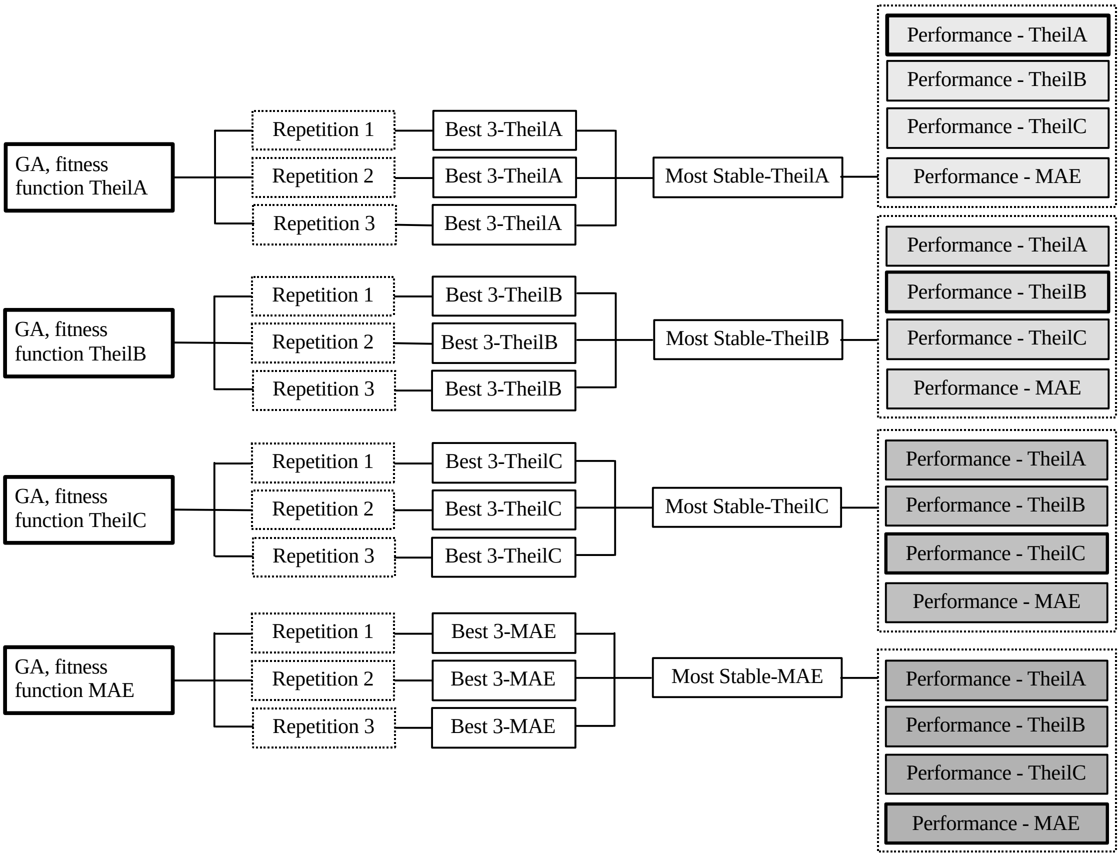


Figure 5.3: Experiment II

Experiment II was repeated for both FTSE and S&P data. For this scenario the final outcome was a set of 8 networks (4 for FTSE and 4 for S&P) evaluated on each one of TheilA, TheilB, TheilC and MAE.

The next paragraphs present in details the parameters we used for the experiment along with the results we obtained.

5.2.2 Search Using the Genetic Algorithm

The first set of parameters for Experiment II is related to the size of the space that the GA will search. For the software we constructed this is set through variables *xMax*, *yMax* and *zMax* which represent the size of the input, the first hidden and the second hidden layers respectively. For all the GAs in this study we have used *xMax* =20, *yMax*=30 and *zMax*=30. This decision was made having in mind that the larger the space the smaller the probability of neglecting network structures that might perform well in our task. On the other hand a larger space implies greater computational cost (more complicated network structures). Thus we have selected the larger space that we could search keeping in mind our computational constraints. However all the experiments proved that the most interesting part of our search space was not close to these bounds; therefore we concluded that the search space needed no expansion.

The second set of parameters includes $P_{reproduction}$, $P_{crossover}$, $P_{mutation}$, *maxGen* and *m*. $P_{reproduction}$, $P_{crossover}$ and $P_{mutation}$, are the probabilities of selecting either a reproduction, a crossover or a mutation operation, while *maxGen* is the termination criterion and *m* is the number of chromosomes (individuals) per generation. In the literature, we found that there is relationship between the values of these parameters [38]. More specifically the larger the population size the smaller the crossover and mutation probabilities. For instance, DeJong and Spears (1990, sited in [38]) suggest for large population size (m=100), $P_{crossover}$=0.6 and $P_{mutation}$=0.001, while for small population size (m=30), $P_{crossover}$=0.9 and $P_{mutation}$=0.01. In our study we applied the GA for m=40 and *maxGen*=25. Again higher values of *m* and *maxGen* imply more expensive experiments in terms of computations as well as more efficient search since the mass of the networks that will be considered is equal to *m* õ *maxGen*. We also selected $P_{crossover}$=0.6, $P_{mutation}$=0.1 and $P_{reproduction}$=0.3. Compared with the suggestions of DeJong and Spears, we used higher mutation and smaller crossover probabilities (having in mind that m=40 and *maxGen*=25). We selected this type of settings because we wanted to force the GA to make a sparser search in our space. The cost of adopting this strategy is that it is more difficult for our GA to converge to a singe network structure. We dealt with this consequence by selecting not only the best, but the three best network structures from the final generation of our GA.

Additionally the size of *Training1*, *Validation1* and *Validation2* sets had to be selected. As we have already presented in chapter 3 the size of *Training1*+ *Validation1*+ *Validation2* set (*Training set*) is predefined at 90% of all the data we have (leaving 10% of the data for testing). The software we created allows us to define the size of *Training1*, *Validation1* and *Validation2*. We refer to the size of these sets using the variables a, b and c respectively. They are defined as: a=$\frac{size\ of\ (Training1\ Set)}{size\ of\ (Training\ Set)}$, b=$\frac{size\ of\ (Validation1\ Set)}{size\ of\ (Training\ Set)}$ and c=$\frac{size\ of\ (Validation2\ Set)}{size\ of\ (Training\ Set)}$; thus they must satisfy the equation a+b+c=1. In our experiments we selected a=0.45, b=0.45 and c=0.1. We chose to split this data set like this in order to train and validate our networks in data sets with the same size and test their performance on the remaining 10% of the data (*Valiadtion2 set*)

In order to decide which was the most appropriate training algorithm of those we described in chapter 4, we experimented on specific network structures. We observed that the Resilient Backpropagation is the algorithm that converged fastest and in fewest epochs. Our observations on the performance of these algorithms agree with the observations of Demuth and Beale [10]. They also experimented on specific network structures using different data and they found that the Resilient Backpropagation converges faster than all the other algorithms we considered in our study[9]. Therefore all the networks in this study were trained using the Resilient Backpropagation algorithm with the following parameters:

Parameter	**Value**
delta_init	0.07
delta_incr	1.2
delta_decr	0.5

Table 5.6: Parameter Settings for Resilient Backpropagation.

In the next two paragraphs we present the results obtained for each one of the FTSE and S&P datasets by applying the first phase (GA search) of Experiment II using the parameters described above.

5.2.2.1 FTSE

For the FTSE data we searched the space defined above 12 times using 3 repetitions for each one of our metrics (TheilA, TheilB, TheilC and mae).

TheilA

By evaluating the networks based on TheilA the GA search for the first repetition gave us the following results:

[9] Gradient descent (incremental mode), Gradient descent (batch mode), Gradient descent with momentum (incremental mode), Gradient descent with momentum (batch mode) and Gradient descent with variable learning rate.

TheilA: Repetition 1			
Generation 1		**Generation 25**	
20-7-18-1	1.06350539553255	1-2-3-1	1.00354334885323
10-27-23-1	2.30858017421321	4-16-3-1	1.01215179900625
10-1-25-1	1.00269691369670	1-16-3-1	0.99754014836206
9-19-24-1	1.08635203261823	1-5-3-1	1.01017235795989
19-23-5-1	1.01497802812205	16-16-3-1	1.02915976153979
9-29-28-1	1.12007657009537	4-7-3-1	1.00847125353051
9-27-1-1	1.00788656553931	4-16-5-1	1.02490905958634
8-25-0-1	1.08241362171488	1-16-3-1	1.00733686649310
3-7-6-1	1.01340403091858	1-16-3-1	0.99893836431798
13-9-6-1	1.01529661748052	1-16-3-1	0.99722773505604
1-23-13-1	1.00551557236577	1-16-3-1	1.00907178974319
19-14-12-1	1.02309501148313	8-7-3-1	1.00154842651733
17-16-6-1	1.00484848467691	1-24-3-1	1.02192266247044
14-26-0-1	1.14620810155970	1-7-3-1	1.00596858445865
14-12-25-1	1.04558150203962	1-16-3-1	1.00026579681361
11-22-13-1	1.05937744091558	1-7-3-1	1.00158144785892
7-6-5-1	1.01119829442827	4-2-3-1	1.00239669676666
14-10-16-1	1.04447280928690	1-16-3-1	1.01988982852211
4-21-11-1	1.02466543026586	8-16-3-1	0.99906947893202
18-26-18-1	1.01362277667090	1-24-3-1	1.00727513395514
10-27-25-1	1.17120092130550	4-7-3-1	0.99944244510115
13-25-20-1	1.06251295833906	16-5-3-1	0.99986474756803
7-9-10-1	1.00332369301496	11-3-3-1	1.01187873218257
11-22-9-1	1.05275238968699	1-9-5-1	1.01065141746981
17-18-11-1	1.05177689669111	1-2-5-1	1.00104824838857
15-17-13-1	1.12672178790905	4-2-20-1	1.01240011451441
14-19-24-1	1.06368010060186	2-18-15-1	1.04118128784757
20-16-27-1	1.14482653303276	1-7-3-1-1	1.00483294107884
4-30-8-1	1.01845896498870	1-5-3-1	1.00907864972665
6-27-22-1	1.12487280379806	16-7-3-1	1.02807603776250
3-1-27-1	1.03064399476638	15-21-4-1	1.01826933865850
4-9-20-1	1.00651905602084	1-9-3-1	0.99692989257736
6-15-2-1	1.00508575508379	1-9-10-1	1.01923851814870
20-18-13-1	1.08370349086986	15-18-24-1	1.05450729598077
11-11-13-1	1.11253383002727	1-16-3-1	0.99468176119214
5-18-23-1	1.04264815817443	4-16-3-1	1.00052611521396
11-20-6-1	1.01970682681941	4-7-5-1	1.00898729968813
8-24-21-1	1.09730624076398	1-16-3-1	1.00393645957885
10-18-24-1	1.14351384228981	4-16-3-1	1.00718340326268
2-19-1-1	1.01585718519519	1-7-3-1	1.01469631847196
Average	**1.08678552007508**	**Average**	**1.00989628912891**
STD	**0.20405508788295**	**STD**	**0.01249876088049**

Table 5.7: The results of the First Repetition of the GA search on the FTSE data using TheilA.

The first two columns of Table 5.7 describe the initial generation that contains 40 randomly selected networks and their performance (TheilA), while the next two columns give the individuals of the last generation with their performance.

The ten network structures that was mostly visited by the algorithm as well as the frequency with which they were met in the 25th generation are indicated by the following table:

Top 10		
Structure	**Times Considered**	**Times in Final Generation**
1-9-3-1	25	1
4-2-3-1	25	1
4-10-3-1	27	0
4-16-5-1	29	1
1-2-3-1	30	1
4-16-3-1	36	3
1-7-3-1	46	4
4-7-3-1	50	2
1-16-3-1	50	9
4-9-3-1	61	0

Table 5.8: The most frequently visited networks in Repetition 1 using TheilA for the FTSE data.

From this table we have that the network with structure ‘4-9-3-1’ was considered by the algorithm 61 times and it was not present in the final generation, while network ‘1-16-3-1’ was considered 50 times and it was met 9 times in the 25th generation.

Table 5.7 indicates that the variance of the performance of the networks in Generation 1 is small and their average is close to one. Furthermore the performance of the networks belonging to the last generation indicates that most of them performed only as well as the random walk (RW) model (based on the price of the market) did; only a few of them managed to outperform slightly the RW. This can be either due to the fact that there are no structures that give significantly better results comparing to the RW model or that the path that our algorithm followed did not manage to discover these networks. Therefore, in order to have safer conclusions we selected to repeat the search twice. A second comment is that there are network structures that seem to perform very badly, for instance ‘10-27-23-1’ gave us a TheilA of 2.3. Furthermore from Table 5.7 it is clear that the GA did manage to converge to networks with smaller Theils (in both terms of mean and standard deviation). Relatively to the type of network structures we got in the final generation the only pattern we observed was a small second hidden layer and more specifically 3 neurons for most of our structures.

On the other hand Table 5.8 indicates how fit individuals proved to be throughout the search that the algorithm performed. For instance the network ‘4-9-3-1’ was visited 60 times, which implies that for a specific time period this individual managed to survive but further search of the algorithm proved that new fittest individuals came up and ‘4-9-3-1’ did not manage to have a place at the 25th generation.

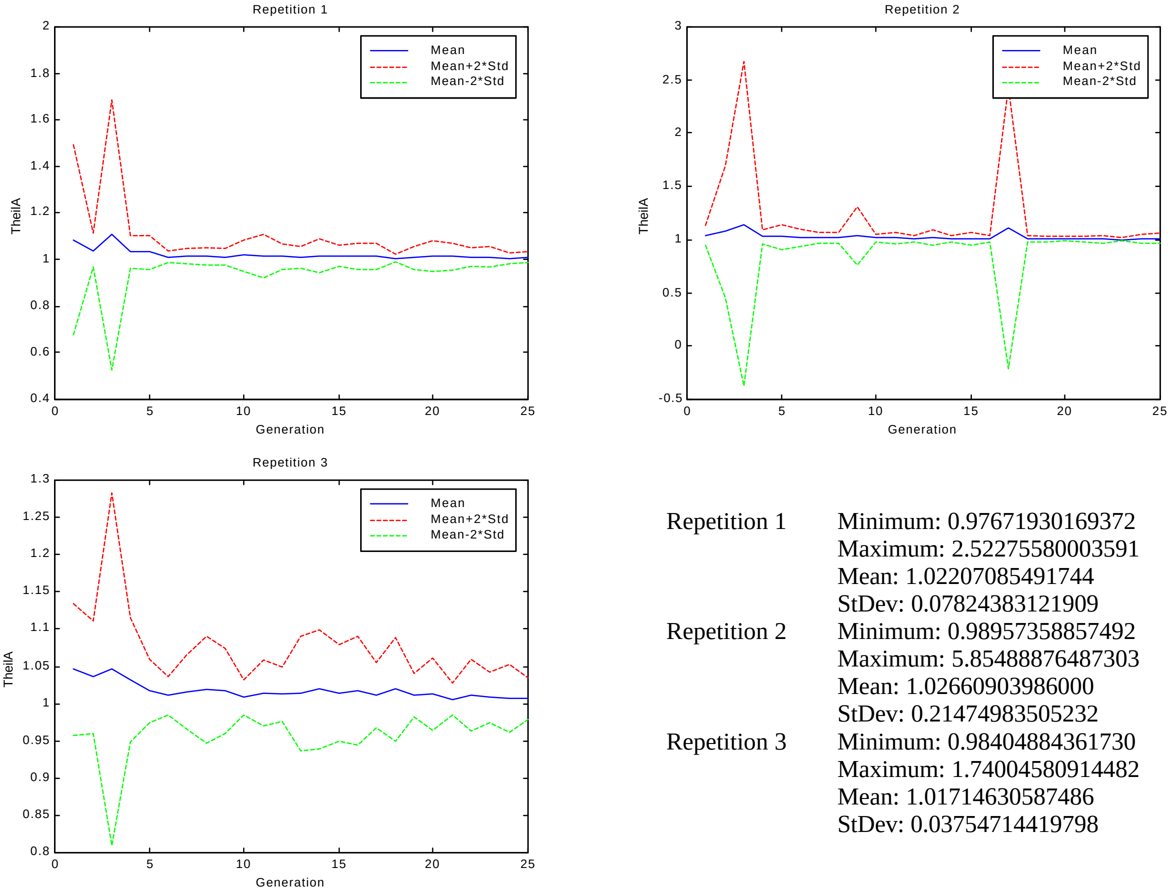
In the next step we repeated the GA search two more times. The results we obtained in terms of mean and standard deviation of the first and last generations were:

	Repetition 2		Repetition 3	
Generation	**first**	**last**	**first**	**last**
Average	1.041067446	1.011567960	1.045704902	1.006809744
Std	0.044528247	0.022214392	0.044429286	0.014169245

Table 5.9: Average and Std for Repetitions 2 and 3 for the FTSE data using Theil A.

From table 5.9 it is clear that for repetitions two and three we obtained similar results. The tables that describe the first and the last generations in details as well as the 10 mostly considered network structures are sited in Appendix I.

The following plots give the mean and the standard deviation of TheilA for each generation (from 1 to 25); their exact values are also presented in Appendix I. In addition the overall generations mean and standard deviation is reported.



Repetition 1 Minimum: 0.97671930169372
Maximum: 2.52275580003591
Mean: 1.02207085491744
StDev: 0.07824383121909

Repetition 2 Minimum: 0.98957358857492
Maximum: 5.85488876487303
Mean: 1.02660903986000
StDev: 0.21474983505232

Repetition 3 Minimum: 0.98404884361730
Maximum: 1.74004580914482
Mean: 1.01714630587486
StDev: 0.03754714419798

Figure 5.4: Mean and Std of TheilA throughout all generations for the FTSE data.

The above plots make clear that in all three repetitions the GA converged giving us networks with smaller Theils (on average). It is also clear that the standard deviation of the Theils across generations also converged to smaller values. However in none of these experiments did we obtain a network structure that clearly beats the random walk model. Furthermore the patterns we managed to observe in the topologies of the networks that belonged to the last generations are: firstly, topologies with many neurons in both hidden layers and in the input layer were not preferred and secondly, the fittest networks proved to be those with one or three neurons in the second hidden layer. The most occurrences that a specific topology managed to have in a last generation were 9, thus we discovered no network that was by far better than all the others.

The distribution that the TheilA follows for each one of the repetitions of the GA is indicated from the following plots.

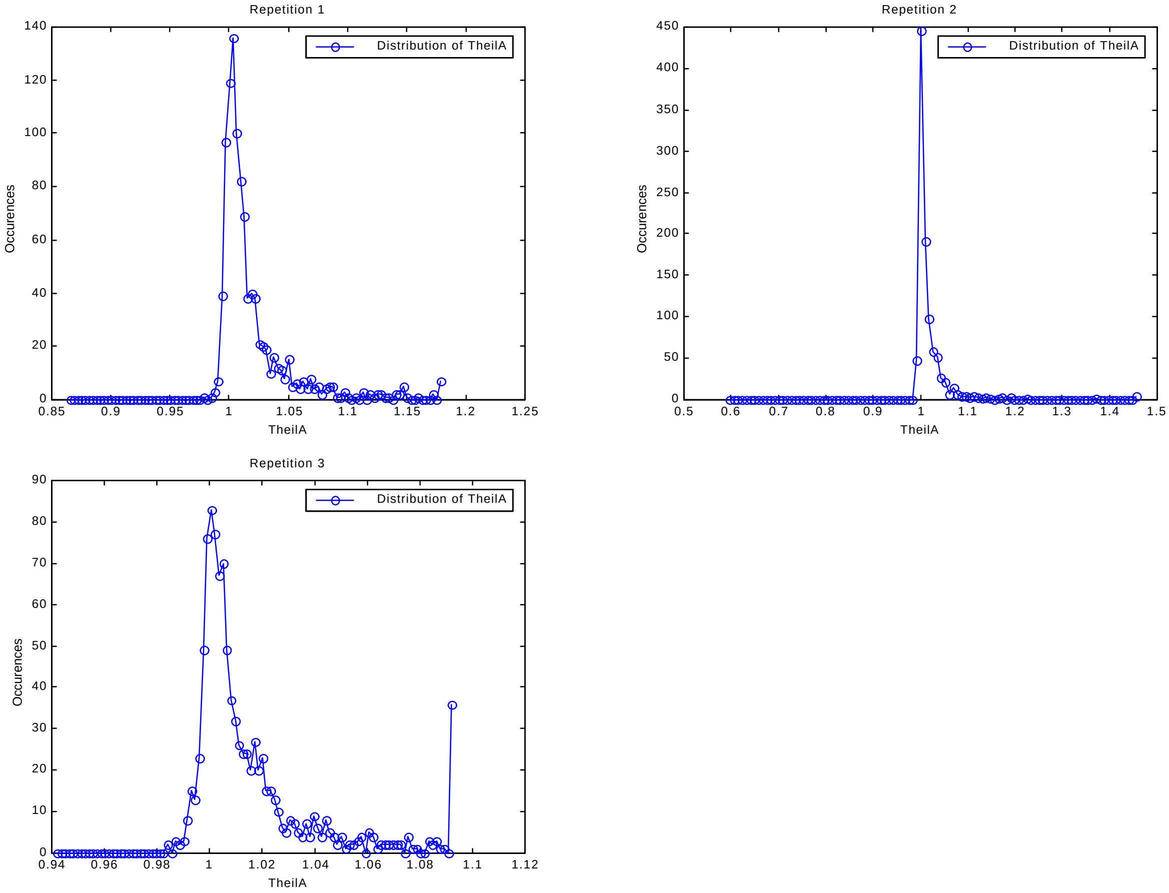


Figure 5.5: Distributions of TheilA for the FTSE data.

These plots clearly show that the best performance that a network had in terms of the TheilA statistic was close to 0.98; the best performance overall generations and repetitions was 0.976. It is also clear that all these distributions are peaked at one. Therefore the majority of network structures visited by the GA search over all repetitions performed close to one.

TheilB

Similarly, we used the GA to search the space of network topologies using TheilB as fitness function. The means and standard deviations of the first and last generation for all three repetitions are presented in the following table:

	Repetition 1		**Repetition 2**		**Repetition 3**	
Generation	**first**	**Last**	**first**	**last**	**first**	**last**
Average	0.770746593	0.731308699	0.771319021	0.734567075	0.784200620	0.734334912
Std	0.048951168	0.010639918	0.044459446	0.012125911	0.151177675	0.015118571

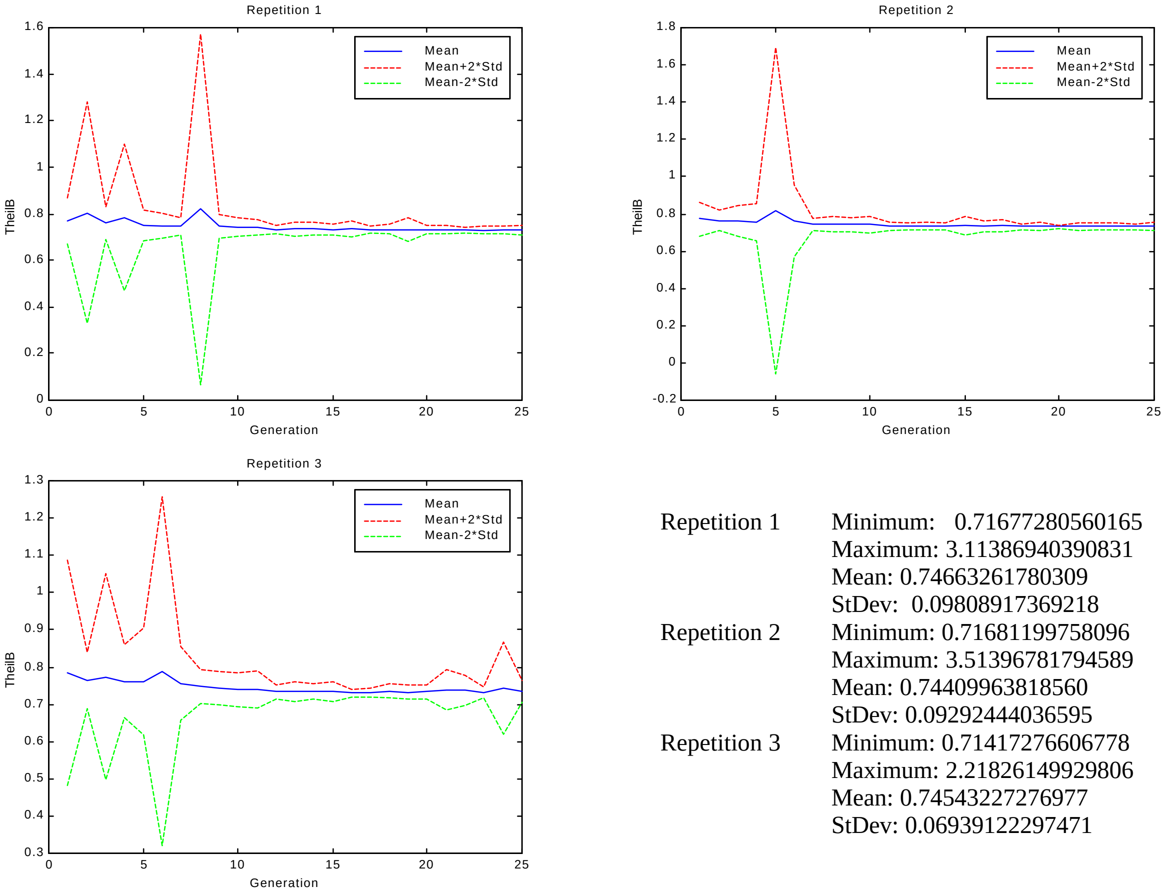
Table 5.10: Average and Std for Repetitions 1,2 and 3 using Theil B for the FTSE data.

The complete results we obtained are sited in Appendix I.

These results show that the NN models managed to beat clearly the predictions of the RW model (based on the *excess returns*) by achieving on average Theils close to 0.73. A second important comment is that the GA converged significantly both in terms of mean and standard deviation. While for TheilA the average performance in both the first and the last generations was close to one and the only thing that the algorithm managed to achieve was to reduce the variance of the performance, for TheilB we observed a significant improvement not only to the variance but to the average of the performance as well.

Further more in the first repetition of the GA search we obtained a topology that proved to be clearly the fittest; it was present 28 times in the last generation and it was visited by the GA 231 times. The topology was ‘4-4-1-1’.

The mean and standard deviation we obtained for each generation in each one of the three repetitions are depicted by the following figure:



Repetition 1 Minimum: 0.71677280560165
Maximum: 3.11386940390831
Mean: 0.74663261780309
StDev: 0.09808917369218
Repetition 2 Minimum: 0.71681199758096
Maximum: 3.51396781794589
Mean: 0.74409963818560
StDev: 0.09292444036595
Repetition 3 Minimum: 0.71417276606778
Maximum: 2.21826149929806
Mean: 0.74543227276977
StDev: 0.06939122297471

Figure 5.6: Mean and Std of TheilB throughout all generations for the FTSE data.

From these plots it is clear that the GA converged in all three repetitions both in terms of standard deviation and mean. This convergence was not always stable. For instance, in the second repetition the GA started to converge during generations 1,2 and 3; then both the standard deviation and mean of the Theils increased substantially. This is because a significant number of the characteristics of the new offspring are defined randomly and therefore there are cases in which the new offspring perform badly. We observed this phenomenon at generations 4 and 5 but then the algorithm managed to converge again excluding the individuals that were not fit.

The distributions that TheilB followed in each repetition of the GA are:

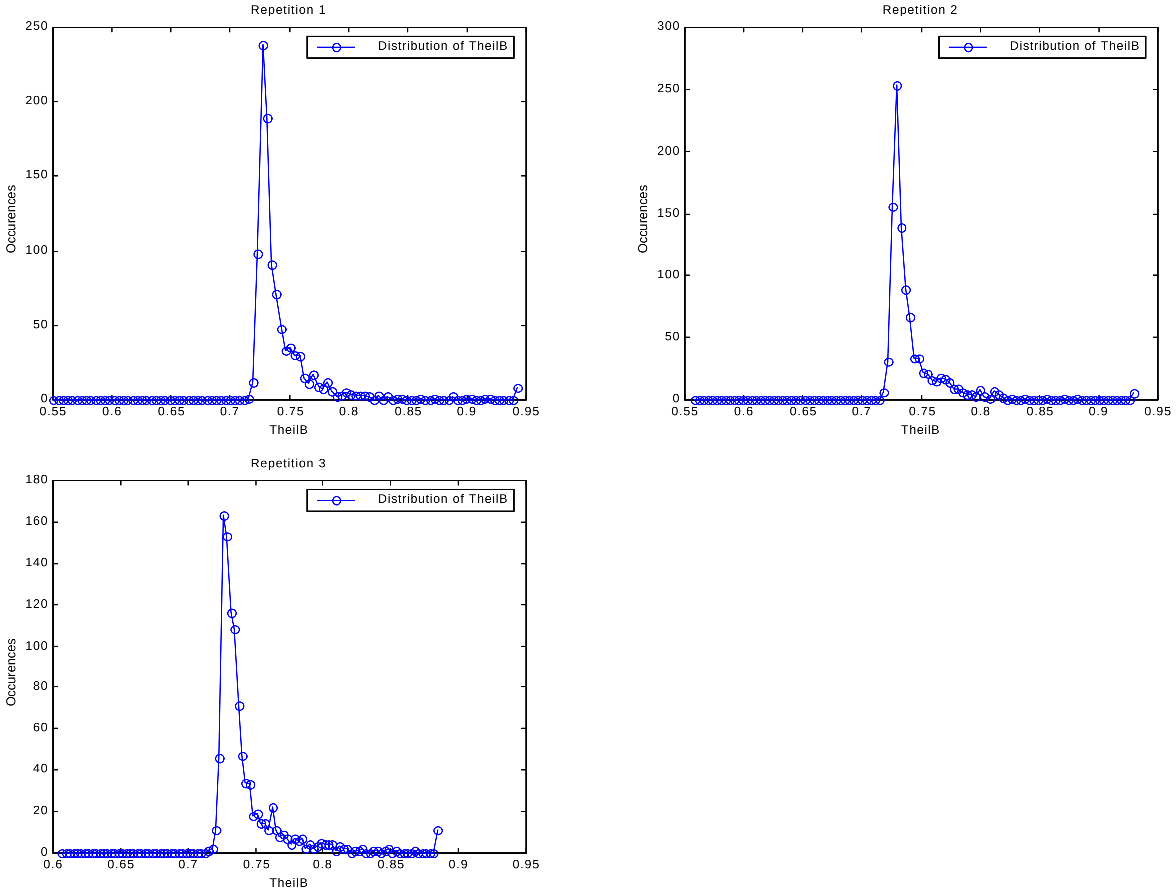


Figure 5.7: Distributions of TheilB for the FTSE data.

The distributions indicate that most of the networks visited have Theils close to 0.74 but none of them performed better than 0.714. Additionally the shape of these plots is similar to those we obtained for TheilA with the difference that they are peaked at a value close to 0.74.

TheilC

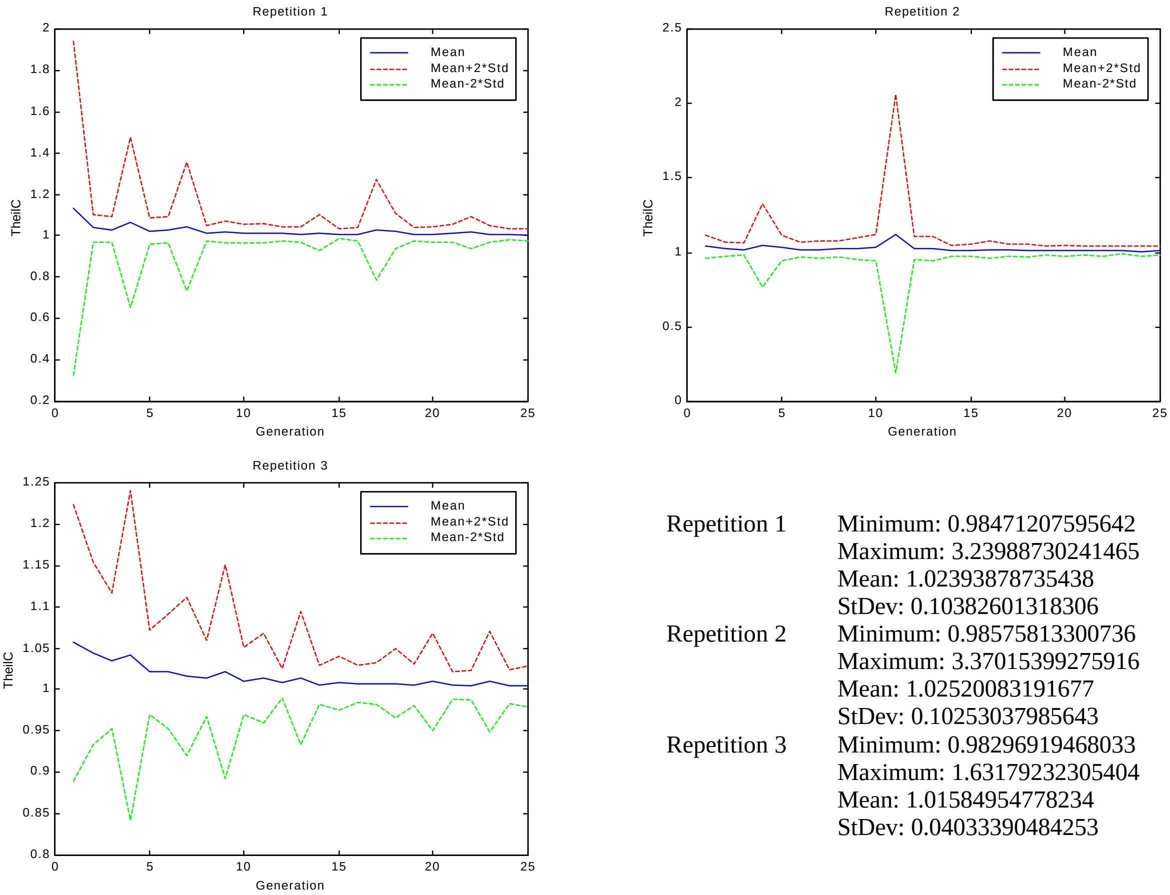
The GA search using as fitness function TheilC gave us again values close to one. Thus in this case too it is indicated that the NN models did not manage to beat clearly the model which states that the gains we will have from the stock market are exactly those that we would have from bonds. More specifically, the mean and the standard deviation of the first and the last generations for each one of the three repetitions were:

	Repetition 1		**Repetition 2**		**Repetition 3**	
Generation	**first**	**Last**	**first**	**last**	**first**	**last**
Average	1.135509075	1.004348794	1.038396719	1.011435866	1.056811551	1.003928808
Std	0.403973152	0.013590195	0.038226957	0.015577890	0.083453020	0.012347686

Table 5.11: Average and Std for Repetitions 1,2 and 3 using TheilC for the FTSE data.

Here again in repetitions one and three we had that the fittest network topologies had a second hidden layer that consisted of one or two nodes.

The mean and standard deviation throughout the 25 generations converged to smaller values but in all three repartitions they moved asymptotically close to one; similarly to the case we used as fitness function the TheilA.



Repetition 1 Minimum: 0.98471207595642
Maximum: 3.23988730241465
Mean: 1.02393878735438
StDev: 0.10382601318306

Repetition 2 Minimum: 0.98575813300736
Maximum: 3.37015399275916
Mean: 1.02520083191677
StDev: 0.10253037985643

Repetition 3 Minimum: 0.98296919468033
Maximum: 1.63179232305404
Mean: 1.01584954778234
StDev: 0.04033390484253

Figure 5.8: Mean and Std of TheilC throughout all generations for the FTSE data.

The distributions of TheilC as well are similar to those of TheilA; most of the values are close to one but none of them below 0.982.

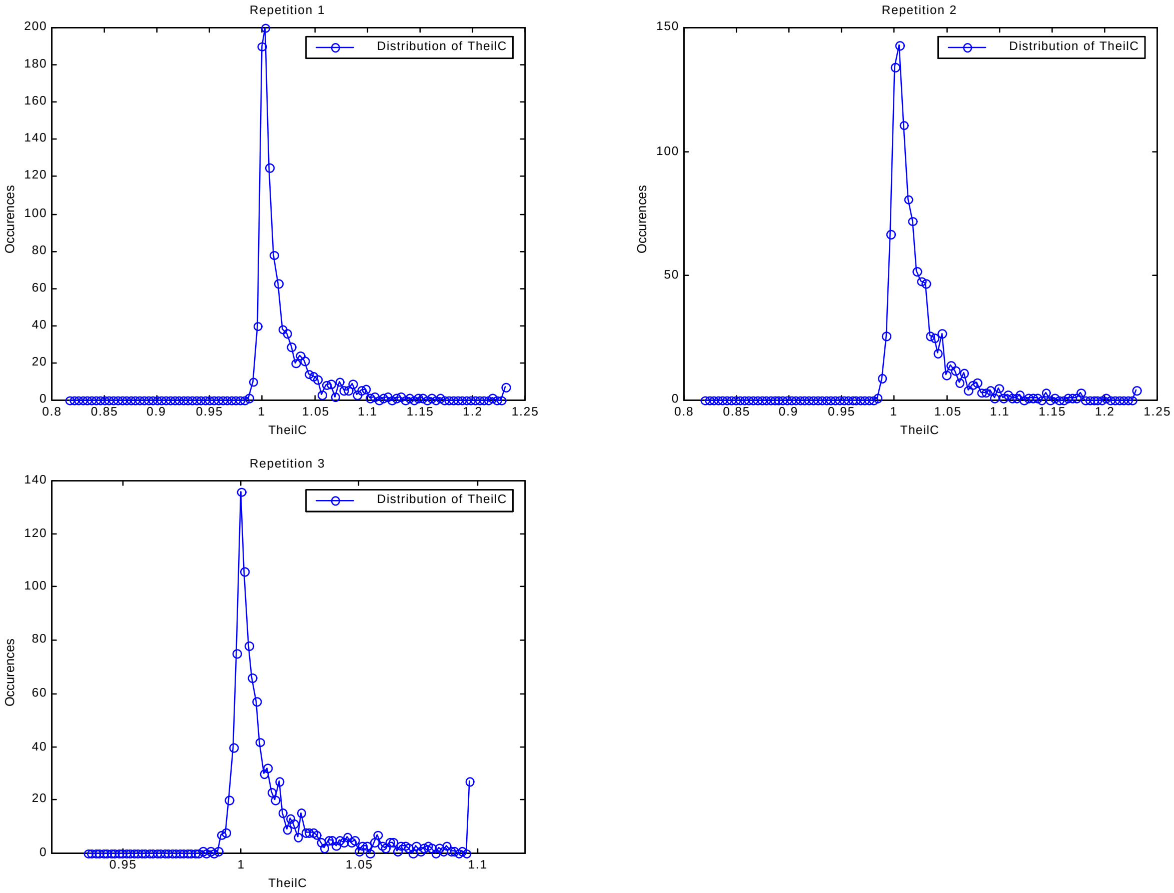


Figure 5.9: Distributions of TheilC for the FTSE data.

To sum up, from all the repetitions of the GA search using both TheilA and TheilC it became clear that the predictive power of all three types of prediction models that are involved is similar.

MAE

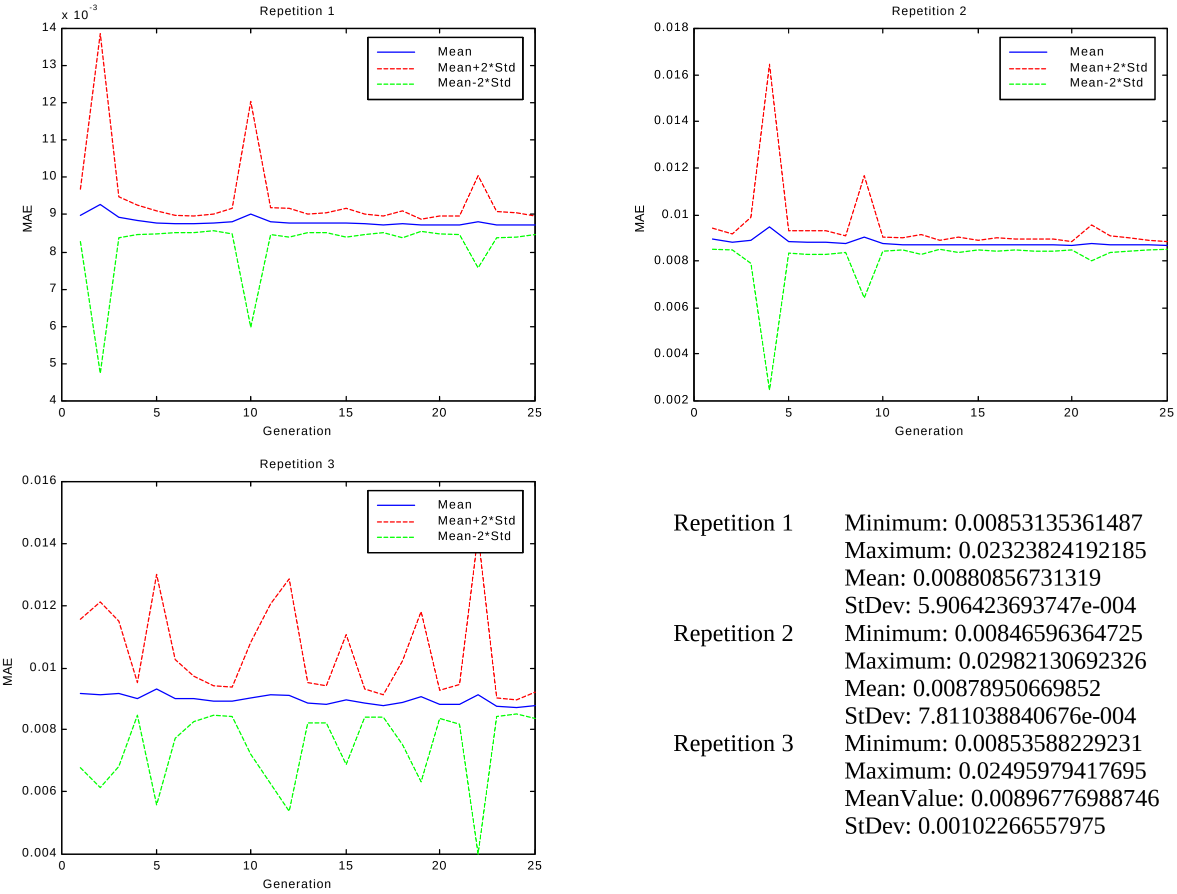
The mean absolute error (*mae*) as it has already been stated is a scale variant metric. The reason we evaluate our models based on it is that it is commonly used. Thus the comparison of the results of our study with past or future studies will be feasible. The GA search for each one of the three repetitions gave us the following results:

	Repetition 1		**Repetition 2**		**Repetition 3**	
Generation	**first**	**Last**	**first**	**last**	**first**	**last**
Average	0.008989344	0.008708330	0.008964180	0.008677859	0.009182516	0.008788057
Std	0.000353321	0.000121448	0.000214911	0.000076273	0.001194585	0.000212327

Table 5.12: Average and Std for Repetitions 1,2 and 3 using mae for the FTSE data

.

In this case also for all repetitions the GA converged to network structures that gave smaller *mae*. Consistently with all the GA searches performed so far there seem to exist a threshold beyond which our models cannot improve. In *mae* terms none of the models performed better than 0.0846.



Repetition 1 Minimum: 0.00853135361487
Maximum: 0.02323824192185
Mean: 0.00880856731319
StDev: 5.906423693747e-004
Repetition 2 Minimum: 0.00846596364725
Maximum: 0.02982130692326
Mean: 0.00878950669852
StDev: 7.811038840676e-004
Repetition 3 Minimum: 0.00853588229231
Maximum: 0.02495979417695
MeanValue: 0.00896776988746
StDev: 0.00102266557975

Figure 5.10: Mean and Std of mae throughout all generations for the FTSE data.

As the distributions of *mae* show the standard deviation is relatively small and the majority of the models are close to the mean. Therefore as for all metrics used so far there are two categories of models, having in mind their performance, those close to the mean and those that perform worse than the mean.

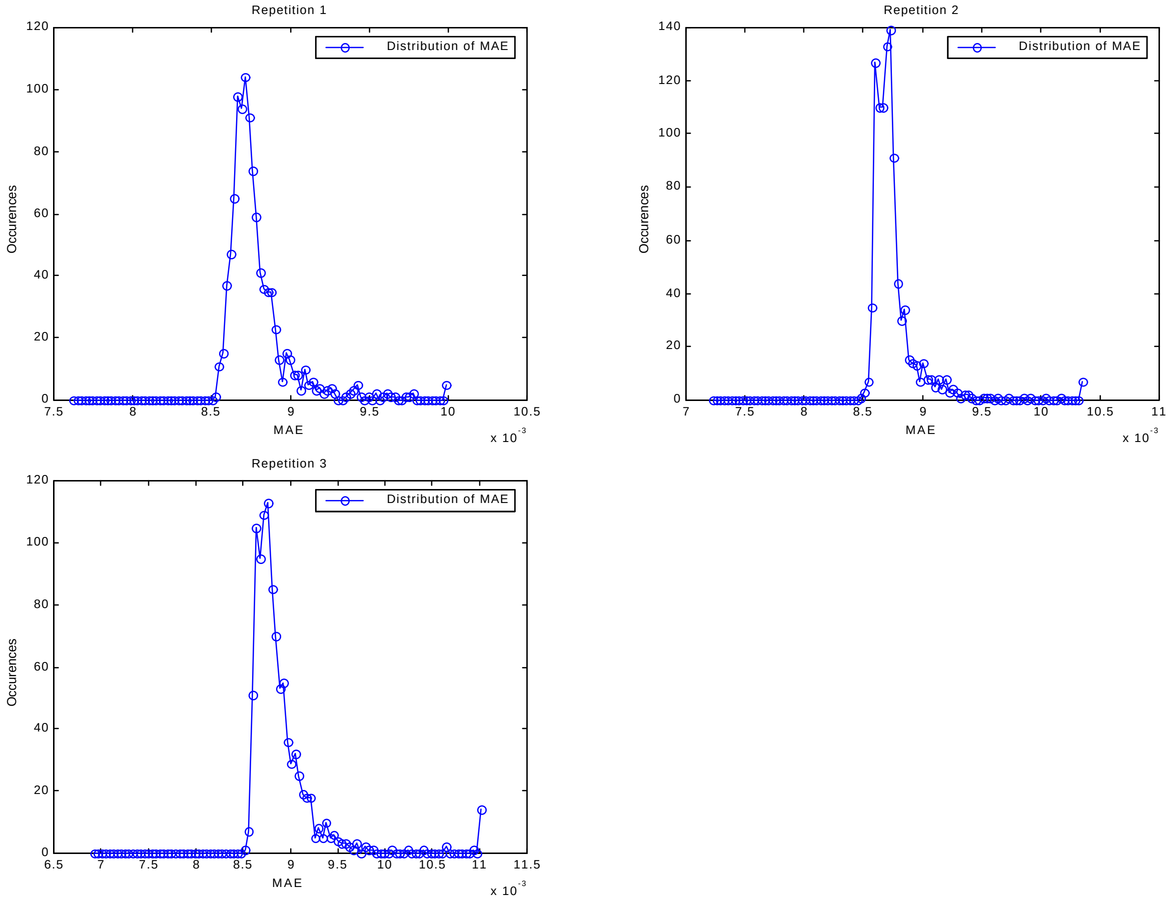


Figure 5.11: Distributions of mae for the FTSE data.

In conclusion the GA managed to indicate network structures that performed better than others on unseen data in terms of the four different metrics. From the results we obtained we can say that the NN models beat clearly only the random walk (RW) on *excess returns* (TheilB). The RW on the value of the market (TheilA) as well as the prediction model which states that the value of the market tomorrow will be such that will allows us to have exactly the same benefit as we would have if we invested in bonds (TheilC) seem to perform closely to the fittest NNs.

The repetitions of the GA showed it did converge giving us generations with smaller mean and standard deviation of the metric used each time. However this convergence was not smooth; the GA during its search went through generations with bad performance (on average) before converging to fittest generations.

Furthermore in all cases the distributions of the metrics showed that most of the network topologies considered performed close to the mean (which was different for each metric). Those networks that were not close to the mean always performed worse.

5.2.2.2 S&P

In this section we present the results we obtained from the GA search using the S&P data. Again we used the same metrics (TheilA, TheilB TheilC and mae) and we repeated the GA search three times for each metric. The complete list of results we obtained is sited in sited in Appendix II.

TheilA and TheilC

In spite the fact that we used new data (S&P data) the results we obtained from the GA search using TheilA and TheilC were similar to those we obtained for the FTSE data. For this reason we present both the results for TheilA and TheilC in the same section.

The use of TheilA as fitness function for the GA applied on S&P data and for all three repetitions gave us the following results:

	Repetition 1		Repetition 2		Repetition 3	
Generation	**first**	**Last**	**first**	**last**	**first**	**last**
Average	1.031026029	1.007563784	1.046308123	1.000894766	1.034200219	1.040926853
Std	0.036859389	0.041011029	0.090209632	0.016385514	0.050289075	0.265116720

Table 5.13: Average and Std for Repetitions 1,2 and 3 using TheilA for the S&P data.

While using TheilC we obtained:

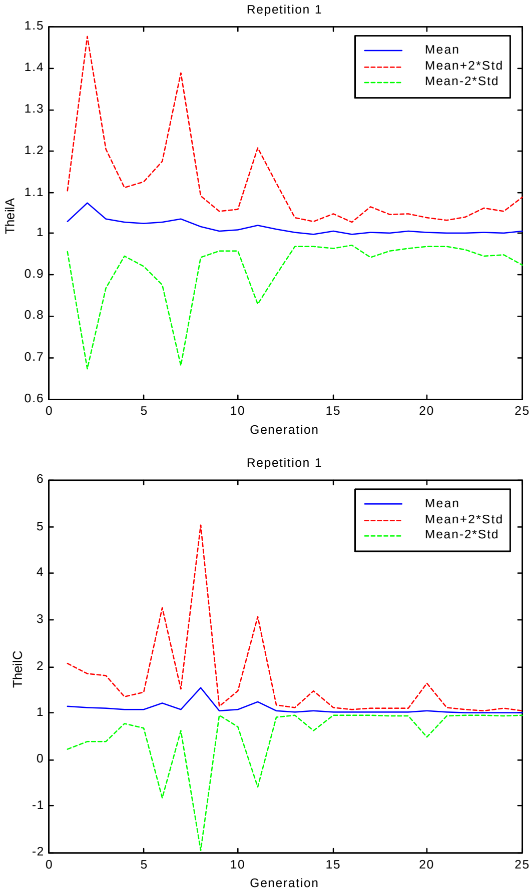
	Repetition 1		Repetition 2		Repetition 3	
Generation	**first**	**Last**	**first**	**last**	**first**	**last**
Average	1.150810389	1.002527788	1.032370207	1.004155361	1.031988377	1.001011443
Std	0.459013220	0.027139328	0.056570654	0.022147295	0.044957192	0.018254567

Table 5.14: Average and Std for Repetitions 1,2 and 3 using TheilC for the S&P data.

From all repetitions of the GA based on either the TheilA or TheilC it became clear that we none of the neural network structures did manage to perform significantly better that the random walk on the value of the market or than the predictor which states that the value of the market tomorrow will be such that will allow us to have the same profits that we would have by investing to bonds. Furthermore in the third repetition for TheilA the GA did not manage to converge to a generation with smaller Theils (in both average and standard deviation terms) comparing to the respective first generation.

Moreover the most occurrences that a neural network topology managed to have in a last generation for both TheilA and TheilC were 11 (‘3-8-3-1’). Also the topologies that were preferred by the GA were not complicated ones and more specifically most of the fittest topologies had a small number of neurons in their second hidden layer.

The next figure presents the mean and the standard deviation throughout generations 1 to 25 for the first repetition of the GA search using TheilA and TheilC. It also presents the mean, the standard deviation, the best and worse Theils overall generations in each repetition using TheilA and TheilC.



Repetition 1 Minimum: 0.95543615106433
Maximum: 2.11678033562905
Mean: 1.01441600193977
StDev: 0.06815235398677
Repetition 2 Minimum: 0.96446950743347
Maximum: 6.09555291707932
Mean: 1.03229514295346
StDev: 0.22283831470531
Repetition 3 Minimum: 0.95958074018409
Maximum: 2.67201086723304
MeanValue: 1.01506249386896
StDev: 0.07306944063927

Repetition 1 Minimum: 0.95947511864954
Maximum: 9.20182663548363
Mean: 1.08141304925925
StDev: 0.48382376757874
Repetition 2 Minimum: 0.96155747751406
Maximum: 3.20491277773598
Mean: 1.01245408293492
StDev: 0.08902262332255
Repetition 3 Minimum: 0.93472483022005
Maximum: 3.75728197821118
MeanValue: 1.01668225434724
StDev: 0.14580623775927

Figure 5.12: Mean and Std of TheilA and C for Repetition 1 throughout all generations for the S&P data.

The best TheilA that a NN gave us was 0.955; while the best TheilC we obtained was 0.934. The way that the GA converged is similar to what we had for the FTSE *excess returns*. It did not converge smoothly but for some generations the mean and the standard deviation increased rapidly while later on it managed again to select the fittest individuals and converge to generations with smaller mean and standard deviation.

The distributions for TheilA and TheilC for all repetitions of the GA search where again peaked at the value of one. Therefore we can say that in this case too we had two groups of models those that gave Theils close to one and those that gave Theils significantly larger than one. None of the NN models managed to give a Theil significantly less than one.

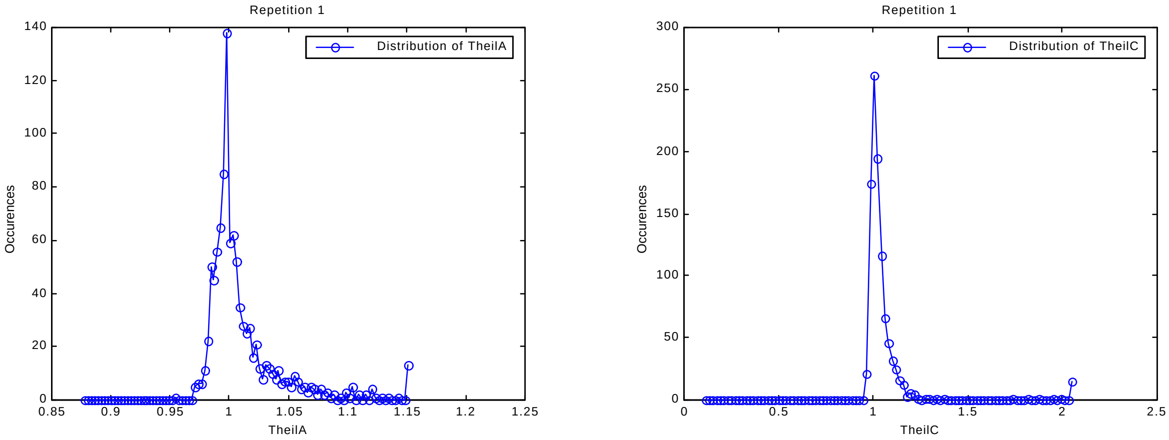


Figure 5.13: Distributions of TheilA and C for Repetition 1 and for the S&P data.

Here we present only the distribution of TheilA and TheilC for the first repetition of the GA search, the rest are similar and are sited in Appendix II.

TheilB

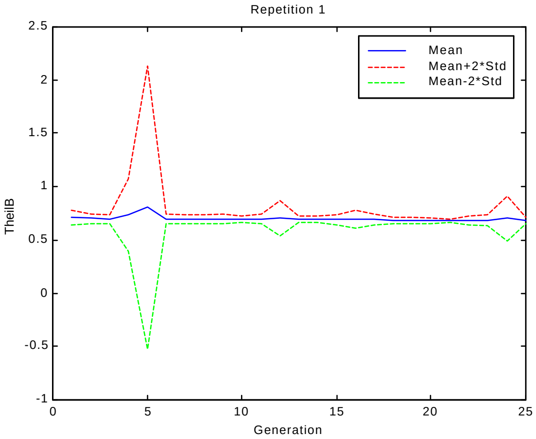
As with the FTSE dataset, the NN models did manage to beat clearly the random walk model based on *excess returns*. As the following table indicates we obtained Theils significantly less than one.

	Repetition 1		**Repetition 2**		**Repetition 3**	
Generation	**first**	**Last**	**first**	**last**	**first**	**last**
Average	0.711798721	0.683356399	0.717174110	0.694143061	0.710648728	0.687398788
Std	0.033821879	0.017246014	0.040166282	0.025064648	0.043682594	0.023336711

Table 5.15: Average and Std for Repetitions 1,2 and 3 using TheilB for the S&P data..

The Theils obtained in the last generations of all three repetitions are on average smaller than 0.7 while the respective values in the case of the FTSE data were all above 0.73. This phenomenon can be either due to the fact that the NNs managed to find more patterns in the S&P data or due to the fact that the RW model (on *excess returns*) in the case of S&P data performed worse.

For all three repetitions the GA gave final generations with smaller TheilB in both terms of means and standard deviations comparing to those of the first generations. The best TheilB obtained was 0.656, which is much better than the best TheilB we obtained for FTSE dataset.



Repetition 1	Minimum: 0.65801387240786 Maximum: 4.90344838578297 Mean: 0.69820374170794 StDev: 0.14180041538561
Repetition 2	Minimum: 0.65632890143622 Maximum: 3.06145783099977 MeanValue: 0.71351487669808 StDev: 0.14408791190612
Repetition 3	Minimum: 0.66140932153077 Maximum: 2.28875675929357 Mean: 0.69449753655453 StDev: 0.05799444472298

Figure 5.14: Mean and Std of TheilB for Repetition 1 throughout all generations for the S&P data.

The distributions of TheilB are similar to those obtained for the FTSE data with the difference that they are slightly shifted to the left. We present here only the distribution of the first repetition, the rest are sited in Appendix II.

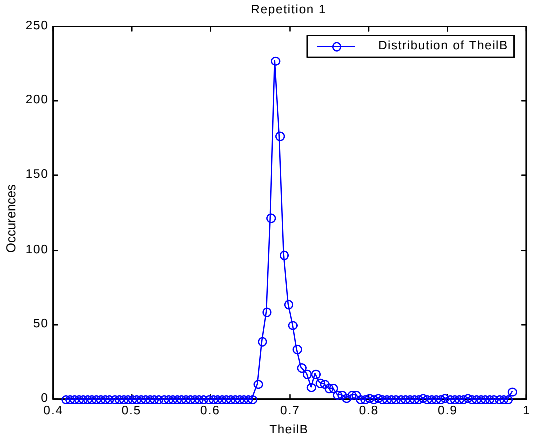


Figure 5.15: Distribution of TheilB for Repetition 1 and for the S&P data.

MAE

The performance of the NN models that were selected by the GA in *mae* terms for the repetitions one, two and three of the GA are indicated by the following table:

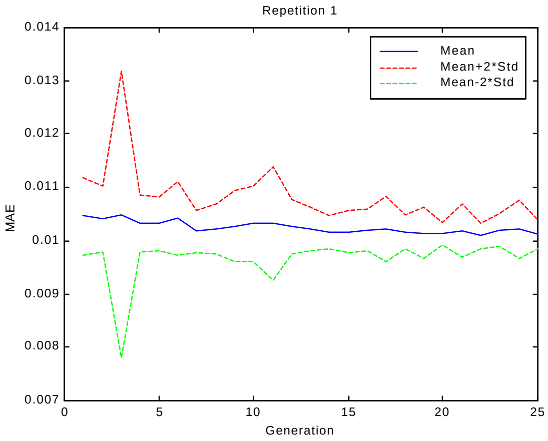
	Repetition 1		**Repetition 2**		**Repetition 3**	
Generation	**first**	**Last**	**first**	**last**	**first**	**last**
Average	0.010460053	0.010116900	0.010313421	0.010219387	0.010434636	0.010269318
Std	0.000366989	0.000130348	0.000291728	0.000307676	0.000479851	0.000584384

Table 5.16: Average and Std for Repetitions 1,2 and 3 using mae for the S&P data.

It is clear from table 5.16 that there is a very small variation in the performance of NNs but this fact due to the nature of *mae* (depended on the scale of target data) is not informative by itself.

A second comment we can make about the magnitude of the *mae* is that its larger than the one we obtained for FTSE data. Having in mind that the magnitude of the values of target data for both FTSE and S&P is almost the same we can infer that NN managed to do better in the FTSE data. Therefore the better Theils we obtained in the S&P dataset for the TheilB are due to the fact that the RW on the *excess returns* is a worse model for the S&P than it is for FTSE.

The next plot verifies our observation that the variation of the mae is very small. It also indicates that we had very small changes in the mean of the mae throughout the 25 generations. For repetitions two and three we obtained similar plots.



Repetition 1	Minimum: 0.00980390701485 Maximum: 0.01865772289184 Mean: 0.01024977636134 StDev: 3.8476067494e-004
Repetition 2	Minimum: 0.00988940190652 Maximum: 0.03591933673879 Mean: 0.01021539665984 StDev: 8.711272711571e-004
Repetition 3	Minimum: 0.00984895417788 Maximum: 0.02950493566814 MeanValue: 0.01025785873425 StDev: 7.416538535514e-004

Figure 5.16: Mean and Std of mae for Repetition 1 throughout all generations for the S&P data.

The distributions we obtained are similar to the ones obtained for the FTSE data. The difference is that for the S&P data the distributions are peaked at a smaller value, close to 0.0102 while for FTSE we had distributions peaked close to 0.0086.

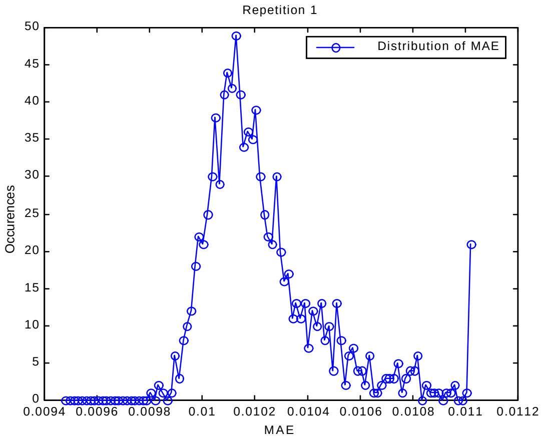


Figure 5.17: Distribution of mae for Repetition 1 and for the S&P data.

To sum up, for the S&P dataset we obtained similar results to the ones we had for the FTSE dataset. Again the NN models managed to beat clearly only the random walk based on *excess returns*. The network structures that were favored by the GA were ones with small number of neurons in the second hidden layer. Furthermore from the distributions of models based on their performance on each one of the four metrics we had that most of the networks performed close to a specific value, which was dependent on the metric we used each time.

5.2.3 Selection of the fittest Networks

In this section we study the behavior of the network topologies that were indicated to be the fittest from the Genetic Algorithm. More specifically we selected from each repetition of the GA search the 3 fittest network topologies and we trained, validated and tested them 50 times each, in order to observe the variations that they give on their performance.

Initially we defined the procedure according to which we chose the 3 fittest networks from each repetition. The selection was based on the following rules:

1. For each repetition of the GA search select those three networks that have the most representatives in the final generation.
2. If there is a tie, select those that are visited the most by the GA, based on the top 10 most visited topologies tables. If this does not break the tie, select those networks that have the simplest structures.

An example of the application of these rules follows. Applying the first repetition of the GA search for the FTSE data and for TheilA we obtained Table 5.7. In the last generation of Table 5.7 the structures with the most representatives are ‘1-16-3-1’, ‘1-7-3-1’ and ‘4-16-3-1’ with 9, 4 and 3 representatives respectively. For the second repetition of the GA in the 25th generation we obtained: ‘7-3-1-1’ 12 times, ‘7-21-1-1’ 9 times and ‘4-3-1-1’, ‘8-3-7-1’, ‘7-4-1-1’, ’20-7-2-1’ 2 times each. In this case we select ‘7-3-1-1’, ‘7-21-1-1’ and from the topologies that were met twice ‘4-3-1-1’, because comparing to the others it was the one higher ranked in the top 10 visited structures table.

The rationale behind this selection is that the networks that survived in the last generation and managed to have as many representatives as possible are the ones that proved to be the best comparing to all the networks visited by the GA. Furthermore for two networks that have the same number of representatives in the last generation the more trustworthy would be the one that passed through more comparisons with others and thus the one that was mostly visited by the algorithm.

The result of applying the above procedure for each evaluation function to both FTSE and S&P datasets is the following network structures:

	FTSE			
	TheilA	**TheilB**	**ThelC**	**MAE**
Repetition 1	1-16-3-1	4-4-1-1	14-3-2-1	6-22-2-1
	1-7-3-1	4-20-1-1	5-3-2-1	6-3-2-1
	4-16-3-1	7-4-1-1	1-3-1-1	6-9-2-1
Repetition 2	7-3-1-1	1-4-1-1	3-1-5-1	4-10-1-1
	7-21-1-1	8-4-2-1	8-7-5-1	4-5-1-1
	4-3-1-1	8-27-1-1	3-7-5-1	4-24-1-1
Repetition 3	18-3-2-1	12-4-2-1	3-2-1-1	9-2-1-1
	4-3-2-1	6-4-2-1	7-8-1-1	9-6-1-1
	18-1-2-1	3-4-2-1	1-2-1-1	17-2-5-1

Table 5.17: The fittest networks indicated from the GA for the FTSE data.

	S&P			
	TheilA	**TheilB**	**ThelC**	**MAE**
Repetition 1	2-27-3-1	5-8-1-1	7-13-7-1	6-26-4-1
	2-27-1-1	5-8-4-1	7-17-7-1	6-9-4-1
	1-27-1-1	5-14-4-1	7-26-7-1	6-22-4-1
Repetition 2	3-8-3-1	2-24-5-1	6-4-7-1	1-22-2-1
	3-28-3-1	5-6-5-1	3-4-7-1	1-7-2-1
	3-8-6-1	5-24-5-1	6-4-9-1	1-22-1-1
Repetition 3	6-7-2-1	1-8-1-1	6-3-3-1	3-19-3-1
	6-7- 4-1	1-1-1-1	6-1-3-1	3-2-3-1
	6-10-2-1	1-6-1-1	6-8-3-1	3-2-9-1

Table 5.18: The fittest networks indicated from the GA for the S&P data.

The performance of the NN models indicated from the GA search for the FTSE dataset is presented in the following table in terms of standard deviation and mean of the 50 times that each of them was trained, validated and tested:

FTSE					
	TheilA			**TheilB**	
Network	**Std**	**Mean**	**Network**	**Std**	**Mean**
1-16-3-1	0.01961388350674	1.00820477542185	4-4-1-1	0.00764454969274	0.72811491206436
1-7-3-1	0.00730024490453	1.00486862998961	4-20-1-1	0.01498936547448	0.73127557331337
4-16-3-1	0.02227707500624	1.01490731794973	**7-4-1-1**	**0.00276503486501**	**0.72957651884349**
7-3-1-1	0.00601270754165	1.00232906250210	1-4-1-1	0.01305596328701	0.73171096444132
7-21-1-1	0.01773028475136	1.00756216483921	8-4-2-1	0.05431578620745	0.73460451486896
4-3-1-1	**0.00404845348991**	**1.00169508751301**	8-27-1-1	0.00794325186130	0.72868313431556
18-3-2-1	0.02015297311220	1.01052429503033	12-4-2-1	0.00753825474067	0.73113202218734
4-3-2-1	0.00929797288659	1.00486080826453	6-4-2-1	0.00587907837145	0.73070643233284
18-1-2-1	0.00704993212946	1.00485862617409	3-4-2-1	0.01095648721657	0.73413207001749
	TheilC			**MAE**	
Network	**Std**	**Mean**	**Network**	**Std**	**Mean**
14-3-2-1	0.01198078182824	1.00649878339865	6-22-2-1	0.00016037861097	0.00874153571250
5-3-2-1	0.00856002245574	1.00217202008705	6-3-2-1	0.00011690678857	0.00868423516488
1-3-1-1	0.00503070769684	1.00178010066804	6-9-2-1	0.00008668171981	0.00873069227919
3-1-5-1	0.01504994905024	1.00719625256695	4-10-1-1	0.00010903989598	0.00869744812727
8-7-5-1	0.02070863638320	1.01600112999146	4-5-1-1	0.00008475362041	0.00867094497344
3-7-5-1	0.01389195219445	1.01342547307135	4-24-1-1	0.00010230611792	0.00867438401744
3-2-1-1	0.00407566006554	1.00137507210356	**9-2-1-1**	**0.00008050708332**	**0.00870733038017**
7-8-1-1	0.00508396558151	1.00355605401049	9-6-1-1	0.00015444671828	0.00874366288006
1-2-1-1	**0.00265946720559**	**1.00020411240575**	17-2-5-1	0.00012086822896	0.00872765526934

Table 5.19: The fittest networks indicated from the GA based on TheilA, TheilB, TheilC and MAE for the FTSE data, trained 50 times each.

The network structures that for each metric gave the smallest standard deviation are indicated in bold font.

On the other hand, the network topologies indicated as the fittest by the GA search on the S&P data are presented by the next table:

S&P					
	TheilA			**TheilB**	
Network	**Std**	**Mean**	**Network**	**Std**	**Mean**
2-27-3-1	0.01626082090294	0.99725453419637	5-8-1-1	0.01043867315151	0.68603732344611
2-27-1-1	0.01336374201237	0.99972555422971	5-8-4-1	0.01770519355545	0.67986871258394
1-27-1-1	**0.01023786430509**	**0.99961096871444**	5-14-4-1	0.01044678974676	0.67758424031563
3-8-3-1	0.01633486431797	0.99685412816632	2-24-5-1	0.02073133096139	0.69405258038825
3-28-3-1	0.03061797189849	1.00324864853158	5-6-5-1	0.01059525043127	0.67750409049392
3-8-6-1	0.05626706151184	1.01568823203417	5-24-5-1	0.01764378089270	0.68135829536472
6-7-2-1	0.01588399010866	0.99391544441447	1-8-1-1	0.00706379523034	0.68305327390241
6-7- 4-1	0.01505263024837	0.99213093392555	**1-1-1-1**	**0.00579286757204**	**0.68479563672748**
6-10-2-1	0.01332876088743	0.99413023768472	1-6-1-1	0.01025710778822	0.68304738930744
	TheilC			**MAE**	
Network	**Std**	**Mean**	**Network**	**Std**	**Mean**
7-13-7-1	0.12091917711418	1.01801983060678	6-26-4-1	0.00019506396082	0.01014302717174
7-17-7-1	0.02805762767076	1.00169434789846	6-9-4-1	0.00012645439429	0.01013162000197
7-26-7-1	0.02312257352437	0.99911646039530	6-22-4-1	0.00018126501997	0.01015238208799
6-4-7-1	0.02581666176588	0.99985924397347	1-22-2-1	0.00012603965727	0.01012772287673
3-4-7-1	0.02036573379806	1.00278153996161	1-7-2-1	0.00027350628720	0.01017535974777

6-4-9-1	0.01750129730863	1.00152217344163	1-22-1-1	0.00008240215742	0.01011905829710
6-3-3-1	0.02205633408582	0.99968154545860	3-19-3-1	0.00027204478516	0.01015573976091
6-1-3-1	**0.00888277464947**	**0.99717433585876**	**3-2-3-1**	**0.00006889143559**	**0.01011199750553**
6-8-3-1	0.02274086699402	0.99922362442379	3-2-9-1	0.00084008608651	0.01033375024713

Table 5.20: The fittest networks indicated from the GA based on TheilA, TheilB, TheilC and MAE for the S&P data, trained 50 times each.

From Tables 5.19 and 5.20 we conclude that the most stable network structures or else the network structures that gave results with the smaller variation are the simplest ones in terms of mass of neurons across their layers.

Additionally the comparison of the standard deviation and the mean even for the networks indicated as the most stable does not prove that they are able to perform clearly better than the set of network structures present in the final generations. Therefore, all that the GA helped us to do was to avoid the network structures that performed badly rather than indicating those networks that are clearly the best. Having in mind the distributions of the networks based on the magnitude of the metric used each time we conclude that there is no specific network structure that performs clearly better than others. Instead there are areas in the search space, which contain network structures that give the best performance we can have. The common characteristic that the members of these areas seem to have is that they have small number of neurons in the second hidden layer, usually one, two or three neurons; a first hidden layer that can have any number of neurons and input layer that in most cases has size smaller than ten.

The evaluation of the NN models that we have used so far was based on unseen data and more specifically on the *Validation2 Set*. We repeatedly tested our models on the *Validation2 Set* and we selected those that proved to perform optimally on it. Therefore we have adjusted our selection on the specific dataset. Since this is the case, we want to see how well the models we have selected to be among those that perform optimally will perform on totally unseen data. For this reason in the next paragraph we present the way that the four most stable topologies (one for each metric) performed on the *Test Set*.

5.2.4 Evaluation of the fittest Networks

The network structures that we consider for the FTSE and S&P data are the following:

Metric used by the GA	FTSE	S&P
TheilA	4-3-1-1	1-27-1-1
TheilB	7-4-1-1	1-1-1-1
TheilC	1-2-1-1	6-1-3-1
MAE	9-2-1-1	3-2-3-1

Table 5.21: The most stable networks for FTSE and S&P.

These are the topologies that proved to give predictions with the smallest variation among the ones indicated as fittest by the GA. Each topology was indicated by a GA search that used as fitness function TheilA, TheilB, TheilC and mae respectively.

We trained these networks on the first half of the data in the *Training Set* (Training1+ Validation1) and we validated them on the rest. Then we obtained the number of epochs beyond which the networks were over-trained and we retrained them on the full *Training Set*. Finally we tested them on the *Test Set* measuring their performance in all four metrics we used so far. We repeated this procedure 50 times for each topology. The next tables present the results we obtained for FTSE and S&P respectively:

	FTSE			
	4-3-1-1		7-4-1-1	
	STD	Mean	STD	Mean
TheilA	0.00490524354536	1.00084379707917	0.01217835101753	1.00173239514676
TheilB	0.00355189314151	0.72471227691297	0.00876287957261	0.72079219346014
TheilC	0.00490511152001	1.00081685922117	0.01217983627077	1.00185456491230
MAE	0.00005769674067	0.00861328995057	0.00011569949816	0.00860158798788
	1-2-1-1		9-2-1-1	
	STD	Mean	STD	Mean
TheilA	0.01704100294014	1.00312032669417	0.00523010070187	1.00125915576285
TheilB	0.01231229153117	0.72476426102874	0.00376342342959	0.72047602535458
TheilC	0.01704065429850	1.00309980386964	0.00523053618550	1.00134252547137
MAE	0.00016784218965	0.00861483152375	0.00004514328093	0.00856787013755

Table 5.22: The performance of the most stable networks on the FTSE data measured in all four metrics.

	S&P			
	1-27-1-1		1-1-1-1	
	STD	Mean	STD	Mean
TheilA	0.00594101933794	0.99911360283886	0.00920940387806	1.00322121138992
TheilB	0.00418916371076	0.70450039123972	0.00649378470751	0.70739677041328
TheilC	0.00594042809710	0.99901417263607	0.00920848737278	1.00312137240443
MAE	0.00006221326788	0.01036938479925	0.00009056576407	0.01039824144737
	6-1-3-1		3-2-3-1	
	STD	Mean	STD	Mean
TheilA	0.02892992971359	1.00867440565435	0.01991734563816	1.00128205208748
TheilB	0.02034385470023	0.70931128252377	0.01404976099285	0.70630764630154
TheilC	0.02892865489642	1.00862995772570	0.01991463695066	1.00114588132332
MAE	0.00030655403019	0.01046133149094	0.00021918549958	0.01040772292645

Table 5.23: The performance of the most stable networks on the S&P data measured in all four metrics.

These results are similar to those obtained by testing our models on the *Validation2 Set*. Therefore, it seems that the predictive ability of the NN models indicated by the GA is not dependent on a specific dataset. TheilA and TheilC obtained are close to one; TheilB is close to 0.72 for FTSE and 0.70 for S&P, while the MAE for FTSE is close to 0.008 and for the S&P close to 0.010.

5.2.5 Discussion of the outcomes of Experiment II

The outcomes of experiment two lead us to the following conclusions:

The benchmarks that the NN models did not manage to beat were: a) the ones that compared their predictive ability with the random walk model on the value of the market (TheilA) and b) the ones that compared their predictive ability with the model which states that the value of the market tomorrow will be such that will allow us to have the same benefit from the stock market as we would have from the bond market (TheilC). The benchmark that compared the NN models with the random walk model on the *excess returns* turned out to be easy to beat in all cases. Furthermore according to all the benchmarks that involved TheilA or TheilC there are naive prediction models (e.g the RW based on the value of the market) that can perform equally well with the best NNs, thus the NNs did not manage to outperform the predictive power of these models. The comment we must make here about TheilA, B and C is that the naï ve predictors related with TheilA and TheilC compare the prediction abilities of our models with naï ve models that indicate no change in the value of the market. Indeed, apart from TheilA which was defined to have this characteristic this statement is true for TheilC because the daily interest rate based on the Treasury Bill rates is so small that the prediction we obtain from the naï ve model used in TheilC is always very close to today’s value. However we cannot say the same for the naï ve predictor we use in TheilB (random walk on the *excess returns*); this predictor attempts to give us a prediction other that no change in the value of the market. Therefore the naive predictors that are based (or are close) to the statement that there will be no change (or no significant change) to the value of the market seems to be the most difficult to beat.

Due to the exhaustive search we performed in the experiment we have no doubt as to whether there might be a network topology for which the NNs will be able to give better predictions. The search we have done indicated that there is no specific network structure that performs significantly better than the others, rather there is a group of structures that gave us optimal performance, even though this performance is no better than that provided by a random walk.

Lastly, as our models did not manage to beat the naï ve models described they failed to find the patterns that we showed existed in the *excess returns* time series of both FTSE and S&P.

5.3 Conclusions

In chapter 3, we applied two randomness tests to both *excess returns* times series, which gave us strong indications that these series are not random, thus they contain patterns.

However, the experiments we conducted in the current chapter showed that neither the autoregressive nor the neural network models managed to find these patterns. They even failed to outperform the prediction abilities of naï ve predictors such as the Random Walk model based on the actual value of the market or the model which states that the value of the market tomorrow will be such that will allow us to have the same gains as we would have by investing in bonds.

Therefore our research was lead to the following findings:

- The excess returns times series of both FTSE and S&P are not random
- The autoregressive models did not manage to beat certain naï ve prediction models.
- The Neural Networks did not manage to beat the same naï ve prediction models.

Before drawing any conclusions based on these findings we investigate whether other studies that have been conducted so far and applied similar methodologies gave results that agree with the results we obtained from our study.

Relative to our first finding the work of Hsieh (1991) also proved that the S&P 500 time series (consisted of daily basis samples) is not random for the period from 1983 to 1998 [3]. Therefore we do have another study that agrees with the part of our work which is related to the randomness of the datasets we used.

Furthermore the work of Steiner and Wittkemper (1996, case study 3) which used data from the Frankfurt stock market proved that in terms of *mae*, the multivariate linear regression models that were applied performed closely to the best NN models. More specifically the multivariate linear regression model gave *mae* of 0.0096902 while the best NN topology 0.0094196 [12]. The difference in the performance of these models is insignificant and therefore you cannot clearly rank these models in terms of mae performance. This finding agrees with the results of our experiments; furthermore our experiments showed that such small differences in the *mae* cannot give significantly different results in the performance of models in terms of TheilA, B, or C. Having always in mind that the work of Steiner and Wittkemper uses data which is similarly scaled to the data we use, we can have some indications comparing the *mae* of their models with the ones that ours gave. The comparison states that in *mae* terms the results that their models gave are better than the ones we obtained for the S&P data but worse than the ones we obtained for the FTSE data. The conclusion we draw out of this comparison is not related to the rank we have just described; it is related to the fact that the *mae* that the models gave in both studies are close. Therefore, although in case study 3 the models were not tested against naï ve predictors, judging from the *mae* that are presented, their models performed about as well as ours. Unfortunately none of the case studies we presented in chapter 2 compared their models with naï ve predictors and thus we are obliged to make comparisons only in *mae* terms.

To sum up, we do have strong indications that the findings of our study do not contradict with the findings of other studies in the literature. Therefore keeping in mind these findings of experiments I and II we are lead to the following conclusions:

Conclusion 1:

Using the type of data we have used in conjunction with AR or feed forward NN models to capture ‘global patterns’ that exist (or not) in the stock market data you will not be able to overcome the predictive ability of naï ve predictors no matter which parameters you will select for your models.

The term ‘global patterns’ is used here to describe patterns that exist (or not) in the stock market data constantly for long time periods. The above conclusion excludes the case that other types of input data will be used and might offer extra information to the models. Another comment we have to make here is that randomness of the data does not necessarily imply that there are global patterns in it. An assumption one can make is that the time series are not random because there are patterns that are only stable over smaller chronological periods (local patterns) than the one we selected to apply to our models. If this is the case and these patterns are contradictory then it is reasonable that the AR and the NN models are not able to trace them since they are fed with lengthy (chronologically) data.

Conclusion 2:

Metrics, such as *mae*, in most cases do not reveal all the truth about the predictive power of a model. What we suggest is to undertake benchmarks using naï ve predictors. Although naï ve, as our experiments indicated, they are difficult to beat. The benchmarks we suggest is the comparison of the prediction models with: a) the RW model based on the value of the stock market and b) the model which states that the value of the stock market tomorrow will be such that will allow us to have the same profit as we would have by investing in the bond market. A third benchmark, which compared our models with the RW model based on returns proved to be lenient and thus it is not recommended. The characteristic which made the first two prediction models difficult to beat is that they predicted no change (or almost no change) in the value of the market, while the third one made predictions using more information (information for the last two days) and gave predictions different than the 'no change' in the value of the market.

Conclusion 3:

Finally, the Neural Network models are superior compared to the AR models because they are able to capture not only linear but also non linear patterns in the underlying data; but their performance is influenced by the way that their weights are initialized. Therefore the evaluation of NN models should be done not in terms of any one specific initialization of their weights, but in terms of mean and standard deviation of a number of randomly selected initializations.

Chapter 6

Conclusion

6.1 Summary of Results

In the current study prediction of the Stock Market *excess returns* on daily basis was attempted. More specifically we attempted to predict the excess returns of FTSE 500 and S&P 500 indices of the London and New York Stock Market, over the respective Treasury-Bill rates. The time series data of stock prices was transformed into the returns the investor would have if he selected the Stock Market instead of placing his capital in the bond market (*excess returns time series*). In our prediction task we used lagged values of the *excess returns time series* to predict the excess returns of the market on daily basis. We applied two different randomness tests on the *excess returns time series,* the Run and the BDS test, and we rejected randomness. Thus, we proved that the prediction task is feasible.

Our review of the literature showed that two different types of prediction models were potentially the most suitable for our purposes: a) the autoregressive AR and b) the feed forward Neural Network (NN) models. Furthermore we used the Akaike and the Bayesian Information Criteria to define the optimum lag structure for the AR models. For the FTSE data both the AIC and the BIC indicated lag of one, while for the S&P data they indicated lag structures of one and seven respectively. For the NN models we applied a Genetic Algorithm to define the optimum topology. The Genetic Algorithm did not indicate a single network topology as optimum. Instead, a number of different topologies were indicated to perform optimally. The common pattern in all of the topologies was the small number of neurons in their second hidden layer.

The parameters of these models were calculated on datasets that included data of a period of approximately eleven years and their predictive ability was measured on datasets that concerned daily data of approximately one year. We measured the performance of our models in terms of mean absolute error (*mae*) and we compared them with three naï ve prediction models:

a. The random walk (RW) model based on the value of the market.
b. The random walk model based on the *excess returns*.
c. And, a model which states that the value of the market tomorrow will be such that will allow us to have the exact same profit that we will have if we invest in bonds.

In terms of *mae* the performance of our models was close to the performance reported by other studies in literature. On the other hand, the comparison of our prediction models with the naï ve predictors described above proved that they managed to beat clearly only model b, while models a and c performed as good as the AR and the NNs. The comparison between the AR and the NNs favored insignificantly the NNs.

6.2 Conclusions

Having in mind the findings of our study we resulted in the following conclusions:

Using the type of data we have used, in conjunction with AR or feed forward NN models to capture patterns in the stock market data over long time periods, it will not be possible to improve on the predictive ability of naï ve predictors, no matter which parameters are selected for the models. Having in mind that the randomness tests rejected randomness for all our series, we believe that their inability to beat naï ve predictors is due to the following facts: a) The use of daily data makes it difficult for the models to recognize clearly trends and patterns that exist in the data, in other words daily data include high level of noise and b) We tried to find patterns in our datasets throughout long time periods, perhaps such global patterns do not exist. This does not imply randomness; there might be patterns in smaller sets (local patterns) that in total are not recognizable because those of one period refute those of another. Case study 4 indicated that such patterns do exist.

Furthermore the use of naï ve predictors as benchmarks for the models we constructed proved that, although naï ve, some of them are difficult to beat. The most difficult to beat were proved to be those that predicted no change or almost no change of the value of the Market (a and c). Out of all naï ve predictors we applied we suggest predictors a and c, or else we suggest the use of predictors that are based on the 'no change of the value of the market' concept. Evaluation of the performance using metrics such as the mean absolute error (mae) cannot depict clearly the predictive power of a model and their interpretation can easily result to misleading conclusions, as it did in case study 3 (paragraph 2.2.3.4).

6.3 Future Work

In this paragraph we indicate the directions towards which we believe that we should move in order to improve the predictive ability of our models. More specifically we discuss here three suggestions that are related with the input data, the pattern detection and the noise reduction.

6.3.1 Input Data

In our study we made the assumption that all the information we need in order to predict the *excess return time series* is included in the series. But is this assumption valid or there are other forms of input data that can offer extra information to our models? The method we would apply if it was to start our study now would be to gather those data we suspect that can influence the market and undertake Principle Component Analysis (PCA) in order to reduce the dimension of the input space by keeping simultaneously the information that the input data can offer to the models. This way we would have the certainty that we offered to our models all the information that we can gather from historic data.

6.3.2 Pattern Detection

As it has already been stated our experiments proved that neither the autoregressive (AR) nor the neural network (NN) models managed to trace patterns in the datasets they were applied to (or at least better than naï ve predictors). In order to be certain that they cannot find patterns in the *excess returns time series* we have to examine the case of tracing patterns in smaller time periods than the ones we used. Intuitively we can say that the 'rules' that the Stock Markets followed in 1988 are not the same with the ones they follow today. The financial, social and political status worldwide changes constantly and with it the way that markets function. Therefore our suggestion is that we should evaluate our models by allowing them to shift through our data by adjusting their parameters in recent historic data and make predictions on close future data.

6.3.3 Noise Reduction

Finally, we consider that the daily Stock Market data are highly noisy data. Therefore we believe that in future studies we have to find a way of reducing the noise of the data we feed our models with. Two ways in which we can achieve this target are: a) by moving from daily data to weekly or monthly average data, this way the trends that exist in the data would be clearer and thus easier to be traced by the prediction models and b) by classifying the excess returns into n-bins based on their magnitude, and then attempt to predict to which bin tomorrow's return will fall into. The simplest case of this scenario is to attempt to predict whether the market tomorrow will go up or down significantly.

Appendix I

In this appendix we present the complete list of results we obtained from the Genetic Algorithm search based on the FTSE datasets for all metrics (TheilA, TheilB, TheilC and mae) in all repetitions. More specifically we present for each repetition three tables that include: a) the first and the last generation of the GA search, b) the ten most visited network structures by the GA as well as the frequency in which they were met in the last generation and c) the mean and the standard deviation of the metric used each time throughout all generations (from 1 to 25).

- **Metric Used: TheilA**

Repetition 1

TheilA: Repetition 1			
Generation 1		**Generation 25**	
20-7-18-1	1.06350539553255	1-2-3-1	1.00354334885323
10-27-23-1	2.30858017421321	4-16-3-1	1.01215179900625
10-1-25-1	1.00269691369670	1-16-3-1	0.99754014836206
9-19-24-1	1.08635203261823	1-5-3-1	1.01017235795989
19-23-5-1	1.01497802812205	16-16-3-1	1.02915976153979
9-29-28-1	1.12007657009537	4-7-3-1	1.00847125353051
9-27-1-1	1.00788656553931	4-16-5-1	1.02490905958634
8-25-0-1	1.08241362171488	1-16-3-1	1.00733686649310
3-7-6-1	1.01340403091858	1-16-3-1	0.99893836431798
13-9-6-1	1.01529661748052	1-16-3-1	0.99722773505604
1-23-13-1	1.00551557236577	1-16-3-1	1.00907178974319
19-14-12-1	1.02309501148313	8-7-3-1	1.00154842651733
17-16-6-1	1.00484848467691	1-24-3-1	1.02192266247044
14-26-0-1	1.14620810155970	1-7-3-1	1.00596858445865
14-12-25-1	1.04558150203962	1-16-3-1	1.00026579681361
11-22-13-1	1.05937744091558	1-7-3-1	1.00158144785892
7-6-5-1	1.01119829442827	4-2-3-1	1.00239669676666
14-10-16-1	1.04447280928690	1-16-3-1	1.01988982852211
4-21-11-1	1.02466543026586	8-16-3-1	0.99906947893202
18-26-18-1	1.01362277667090	1-24-3-1	1.00727513395514
10-27-25-1	1.17120092130550	4-7-3-1	0.99944244510115
13-25-20-1	1.06251295833906	16-5-3-1	0.99986474756803
7-9-10-1	1.00332369301496	11-3-3-1	1.01187873218257
11-22-9-1	1.05275238968699	1-9-5-1	1.01065141746981

17-18-11-1	1.05177689669111	1-2-5-1	1.00104824838857
15-17-13-1	1.12672178790905	4-2-20-1	1.01240011451441
14-19-24-1	1.06368010060186	2-18-15-1	1.04118128784757
20-16-27-1	1.14482653303276	1-7-3-1-1	1.00483294107884
4-30-8-1	1.01845896498870	1-5-3-1	1.00907864972665
6-27-22-1	1.12487280379806	16-7-3-1	1.02807603776250
3-1-27-1	1.03064399476638	15-21-4-1	1.01826933865850
4-9-20-1	1.00651905602084	1-9-3-1	0.99692989257736
6-15-2-1	1.00508575508379	1-9-10-1	1.01923851814870
20-18-13-1	1.08370349086986	15-18-24-1	1.05450729598077
11-11-13-1	1.11253383002727	1-16-3-1	0.99468176119214
5-18-23-1	1.04264815817443	4-16-3-1	1.00052611521396
11-20-6-1	1.01970682681941	4-7-5-1	1.00898729968813
8-24-21-1	1.09730624076398	1-16-3-1	1.00393645957885
10-18-24-1	1.14351384228981	4-16-3-1	1.00718340326268
2-19-1-1	1.01585718519519	1-7-3-1	1.01469631847196
Average	**1.08678552007508**	**Average**	**1.00989628912891**

Table I.1: The results of the first Repetition of the GA search on the FTSE data using TheilA.

TheilA: Repetition 1		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
1-9-3-1	25	1
4-2-3-1	25	1
4-10-3-1	27	0
4-16-5-1	29	1
1-2-3-1	30	1
4-16-3-1	36	3
1-7-3-1	46	4
4-7-3-1	50	2
1-16-3-1	50	9
4-9-3-1	61	0

Table I.2: The most frequently visited networks in repetition 1 using TheilA for the FTSE data.

TheilA: Repetition 1		
Generation	**Mean**	**STD**
1	1.08678552007508	0.20405508788295
2	1.03935078221888	0.03602938840117
3	1.10714011077766	0.29021390916461
4	1.03299940158742	0.03427207205730
5	1.03185315512521	0.03611051076312
6	1.01213054130883	0.01325122529524
7	1.01465125813734	0.01718716677158
8	1.01291125746099	0.01891781262948
9	1.01206527576169	0.01686073134542
10	1.01800895566742	0.03424491355265
11	1.01621771456402	0.04686524668659
12	1.01285307428506	0.02724353547456
13	1.01019678515465	0.02360160963930
14	1.01617158007458	0.03697318528337
15	1.01605007723046	0.02247191709692
16	1.01556420422961	0.02759774903873
17	1.01515654809810	0.02735970200787
18	1.00661233527168	0.00848711748322
19	1.00912721263499	0.02502545116507
20	1.01403566680733	0.03275040959333
21	1.01260517766040	0.02886033959374
22	1.01209680795179	0.01980344746897
24	1.01134892925481	0.02171552074739
24	1.00594271246919	0.01179454551189
25	1.00989628912891	0.01249876088049

Figure I.3: Mean and Std of TheilA throughout all generations in repetition 1 for the FTSE data.

Repetition 2

TheilA: Repetition 2			
Generation 1		Generation 25	
2-24-26-1	1.11813766718096	20-7-2-1	1.00184661293818
4-8-19-1	1.17547334620910	8-3-7-1	1.01982076483303
3-11-7-1	1.02290497005809	7-3-1-1	0.99906109307869
17-20-3-1	1.04370455350328	7-21-1-1	1.01492191599385
11-8-1-1	1.01618179638309	4-21-1-1	0.99355051782963
14-19-14-1	1.06544645423544	7-3-1-1	1.00406950269047
7-3-2-1	1.00306435368849	3-21-1-1	1.00451122525557
12-10-10-1	1.02774092907466	7-3-1-1	1.00071844558870
13-26-0-1	1.06849786152071	7-4-1-1	0.99846880533535
14-29-12-1	1.06570885228095	12-21-20-1	1.05364777222097
3-28-8-1	1.01503466812073	7-3-1-1	1.00162607211633
20-29-13-1	1.10093332444841	7-21-1-1	1.00099231062299
2-21-16-1	1.03497249080513	7-4-1-1	1.00838621581456
1-10-8-1	1.02292798211608	12-3-20-1	1.06941416411512
8-3-1-1	1.00696517211468	8-3-7-1	1.03367141438730
3-26-4-1	0.99041696462295	20-7-2-1	1.01929737935196
9-25-11-1	1.04154889257945	7-4-20-1	1.01707982819114
6-23-10-1	1.01175916610557	12-21-1-1	1.01397585383556
14-16-17-1	1.13148582523777	15-7-30-1	1.11219281900278
3-15-20-1	1.16615487946461	7-3-1-1	1.00393592062611
17-24-6-1	1.03067901498808	7-21-1-1	0.99744081489836
8-21-27-1	1.02313159538776	7-21-1-1	1.00021557930861
18-5-2-1	1.01082904625174	4-3-1-1	1.00062657570077
17-6-19-1	1.01704232318767	7-8-1-1	1.00077838160404
15-16-3-1	0.99898079746182	4-3-1-1	0.99647920075373
2-10-1-1	1.00580552885653	20-7-22-1	1.02148015510226
8-8-19-1	1.03099355302045	7-3-1-1	1.01479569929501
12-24-4-1	0.99808296787421	7-3-1-1	1.00537717836183
7-26-12-1	1.04283095885825	7-3-1-1	1.00235838587736
10-28-20-1	1.02868081928977	7-21-7-1	1.02311176788348
20-8-5-1	1.02389450695595	7-3-1-1	0.99960937160581
7-19-6-1	1.00808883654680	7-21-1-1	1.00295503621555
11-11-12-1	1.02528904060568	7-21-1-1	1.00355797862279
16-16-27-1	1.09480676948671	7-21-1-1	1.00432958891701
14-3-3-1	0.98957358857492	2-8-1-1	1.01128687384650
15-6-14-1	1.03634943283935	7-21-1-1	1.00621614313891
12-22-12-1	1.01704910192153	7-21-1-1	1.00000534794055
8-8-12-1	1.02584117445122	7-3-1-1	0.99477243986251
19-24-11-1	1.03814028682443	7-3-1-1	1.00088293250760
2-16-10-1	1.06754836075064	7-3-1-1	1.00525034292916
Average	**1.04106744634709**	**Average**	**1.01156796070500**

Table I.4: The results of the second Repetition of the GA search on the FTSE data using TheilA.

TheilA: Repetition 2		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
9-3-7-1	16	0
12-21-2-1	16	0
2-3-1-1	16	0
12-3-1-1	18	0
4-3-1-1	20	2
7-1-1-1	23	0
7-28-1-1	25	0
7-3-7-1	41	0
7-21-1-1	48	9
7-3-1-1	172	12

Table I.5: The most frequently visited networks in repetition 2 using TheilA for the FTSE data.

TheilA: Repetition 2		
Generation	**Mean**	**STD**
1	1.04106744634709	0.04452824737631

2	1.07666112823947	0.31101831986027
3	1.14731797667752	0.76385704141927
4	1.02714164901533	0.03396242058969
5	1.02615290449167	0.05818261348035
6	1.02276162812629	0.04115356995870
7	1.02096276305827	0.02416273516764
8	1.01721341176217	0.02434382410621
9	1.03476139479343	0.13677545850175
10	1.01413546878791	0.01939270874050
11	1.01411110942541	0.02942813022045
12	1.01076692489960	0.01620021736039
13	1.01917932504770	0.03689994429588
14	1.00922504654240	0.01433801040983
15	1.01008914497963	0.03135775463240
16	1.00787911533916	0.01633475868070
17	1.11092307633909	0.66079694013901
18	1.00562319752609	0.01606409492627
19	1.00757091693364	0.01336817190826
20	1.00628795659829	0.01096133188920
21	1.00493068979808	0.01113060772240
22	1.00791199066484	0.01806210093525
24	1.00332611650870	0.00620794256018
24	1.00765765389310	0.02100123935119
25	1.01156796070500	0.02221439207968

Figure I.6: Mean and Std of TheilA throughout all generations in repetition 1 for the FTSE data.

Repetition 3

TheilA: Repetition 3			
Generation 1		**Generation 25**	
5-29-9-1	0.99816748884017	4-3-23-1	1.01006614077786
2-29-19-1	1.09854728841160	4-3-2-1	1.01515098653632
3-8-25-1	1.02106240158269	18-3-2-1	1.00740640796372
17-28-28-1	1.19559897349382	2-2-21-1	0.99715018267172
13-13-6-1	1.01578585304786	4-30-2-1	1.01211821671603
15-12-0-1	0.99301574507889	13-1-5-1	0.99824039352116
14-23-3-1	1.07551267584720	4-1-2-1	1.00179127439447
20-30-2-1	0.99633481903537	18-3-2-1	0.99810573548878
14-18-14-1	1.08288786858866	18-1-4-1	1.01670631310201
13-29-16-1	1.12156627154233	18-1-5-1	1.03112721490660
19-3-4-1	1.01869802871080	18-3-2-1	1.01195404438461
13-22-5-1	1.00986672387720	7-1-2-1	1.00335253299096
9-10-29-1	1.04552330640827	7-3-4-1	1.00298894043260
15-12-20-1	1.08340137503222	18-1-2-1	1.00428343897891
17-21-4-1	1.11057320042797	4-3-4-1	0.99879935785336
20-28-1-1	0.99247692322188	6-1-4-1	1.00056493476465
18-19-16-1	1.03413139443074	4-3-2-1	0.99902127396198
17-10-4-1	1.00502613761163	4-3-4-1	1.00217008648749
19-5-27-1	1.09675975922842	14-3-17-1	1.07589279997397
9-13-21-1	1.03511485755200	18-3-2-1	1.00112679360615
14-8-15-1	1.06146983619133	18-1-4-1	1.00030086043270
11-16-20-1	1.07638544136352	18-1-5-1	1.00121367450953
12-19-17-1	1.02038574768492	4-3-2-1	1.00541045074627
15-27-19-1	1.01537002830371	4-1-4-1	0.99956198380551
2-20-9-1	1.07324166385468	18-3-2-1	1.00738837334297
5-7-28-1	1.00815496450509	13-3-2-1	1.00882617176923
14-9-16-1	1.03983171675952	18-1-2-1	0.99982786362810
20-8-3-1	1.01233880839697	18-3-2-1	1.00351555721259
20-5-26-1	1.00440687448385	18-3-2-1	1.00147389261749
3-21-22-1	1.10705072537003	11-6-10-1	1.02854596539663
3-29-4-1	1.04223667103795	4-3-2-1	0.99984366724032
17-17-3-1	0.99854716192115	18-1-2-1	1.00241106485686
17-19-29-1	1.08915751504048	7-1-2-1	1.00061579922015
4-13-7-1	1.01826717422046	6-3-4-1	1.00433115068912

18-9-9-1	1.01792679186201	18-16-4-1	1.01672120329510
3-4-9-1	1.01758234329760	6-3-4-1	0.98901578570703
1-3-18-1	1.04889446555361	4-3-2-1	1.00225292665755
8-3-28-1	1.01569791260743	18-3-2-1	0.98981524151261
19-19-23-1	1.07113447894981	18-3-2-1	1.00334554868613
17-16-5-1	1.06006467149002	7-3-2-1	1.01995552296237
Average	**1.04570490212160**	**Average**	**1.00680974434504**

Table I.7: The results of the third Repetition of the GA search on the FTSE data using TheilA.

TheilA: Repetition 3		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
7-3-2-1	16	1
20-30-2-1	18	0
20-30-4-1	19	0
20-17-2-1	22	0
18-1-2-1	22	3
4-30-4-1	30	0
4-30-2-1	33	1
4-3-4-1	45	2
18-3-2-1	48	9
4-3-2-1	49	5

Table I.8: The most frequently visited networks in repetition 3 using TheilA for the FTSE data.

TheilA: Repetition 3		
Generation	**Mean**	**STD**
1	1.04570490212160	0.04442928664854
2	1.03592523678858	0.03782682803399
3	1.04561346461301	0.11821214549655
4	1.03168815333548	0.04194332789306
5	1.01669796518995	0.02106891344061
6	1.01079576244850	0.01269078989694
7	1.01546998892848	0.02465587821756
8	1.01862141977780	0.03568230941754
9	1.01696161114399	0.02854026600792
10	1.00801276807151	0.01175593102620
11	1.01406906806150	0.02230958725612
12	1.01199572013683	0.01833673833645
13	1.01376159078949	0.03853727942206
14	1.01961680335092	0.03995135536375
15	1.01387049461365	0.03245406573664
16	1.01720892970320	0.03667665729174
17	1.01080768608946	0.02185646735145
18	1.01968820665845	0.03491308409599
19	1.01089223061804	0.01469030491868
20	1.01277741300207	0.02421245241713
21	1.00576676983489	0.01087026955868
22	1.01093147745980	0.02402188369827
24	1.00778003771309	0.01683386501529
24	1.00719020207630	0.02275792050249
25	1.00680974434504	0.01416924596758

Figure I.9: Mean and Std of TheilA throughout all generations in repetition 3 for the FTSE data.

❑ Metric Used: TheilB

Repetition 1

TheilB: Repetition1			
Generation 1		**Generation 25**	
17-20-26-1	0.79782489901060	4-4-1-1	0.72624612002890
1-23-12-1	0.91464782836897	4-4-1-1	0.72548064571460
5-1-20-1	0.76736227737376	4-4-1-1	0.72652977737784

2-12-15-1	0.75859197147524	4-4-1-1	0.73405956725773
11-4-20-1	0.73331165221857	4-4-1-1	0.72464984015399
14-12-15-1	0.75796024647650	4-4-4-1	0.76174453112638
14-28-29-1	0.82628871315510	4-4-1-1	0.76005304367667
11-16-17-1	0.87357011987555	4-30-1-1	0.73962413774381
3-3-20-1	0.73643075294891	4-4-1-1	0.72501636126733
20-25-17-1	0.74299880298685	4-4-1-1	0.72392018612007
20-20-18-1	0.75934378970419	4-4-1-1	0.72881049362646
7-22-24-1	0.74748140203163	4-4-1-1	0.72190082909092
1-25-14-1	0.73426885526110	4-4-1-1	0.72587841879972
7-18-19-1	0.88641351836711	4-4-1-1	0.73397489389653
14-25-4-1	0.72303704729465	13-13-13-1	0.75243344253102
11-21-19-1	0.74447460436379	4-20-1-1	0.72551921199357
10-1-26-1	0.90376200763706	4-4-1-1	0.72564947085065
5-9-0-1	0.74954222649254	4-4-1-1	0.72868713115569
11-1-25-1	0.76654062438866	4-20-1-1	0.72659318915690
2-19-4-1	0.76137234018741	4-4-1-1	0.73052783841434
3-11-0-1	0.75216067119434	4-20-1-1	0.72446516684313
3-30-22-1	0.81221652266947	4-4-1-1	0.72451852217900
20-15-1-1	0.72335873987966	4-20-1-1	0.73646662343769
3-5-1-1	0.72831507349469	4-4-1-1	0.73203752942552
7-20-7-1	0.75580848112956	4-4-1-1	0.72664261786896
20-23-21-1	0.78534702072948	4-4-1-1	0.72710091643839
7-2-0-1	0.72711531735740	4-4-1-1	0.72528051638007
8-7-13-1	0.74474973638331	4-4-1-1	0.72821783350419
4-23-25-1	0.73008494790145	4-4-1-1	0.73167255060008
17-13-27-1	0.77524256613413	4-4-1-1	0.72556654753416
3-14-12-1	0.75681840093618	6-20-1-1	0.72479444898119
19-16-11-1	0.75544362001270	4-20-1-1	0.72763019291423
16-25-16-1	0.74736720109773	6-4-1-1	0.72822099896469
10-17-28-1	0.77067698285754	11-24-14-1	0.76778426344268
19-12-14-1	0.73160606408486	4-4-1-1	0.72928007427502
10-30-12-1	0.74467316296382	4-24-1-1	0.73242308470911
17-13-21-1	0.77192557147770	4-4-1-1	0.72870646033906
1-3-18-1	0.72876219983049	4-4-1-1	0.73067328227481
11-4-8-1	0.78654851366164	4-4-1-1	0.72819602812919
2-14-30-1	0.81641925843869	4-20-1-1	0.72537119340421
Average	**0.77074659329633**	**Average**	**0.73130869954071**

Table I.10: The results of the first Repetition of the GA search on the FTSE data using TheilB.

TheilB: Repetition 1		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
7-4-4-1	10	0
3-20-1-1	11	0
3-19-1-1	13	0
7-20-4-1	18	0
7-19-1-1	19	0
7-19-4-1	22	0
7-20-1-1	55	0
7-4-1-1	79	0
4-20-1-1	97	6
4-4-1-1	231	28

Table I.11: The most frequently visited networks in repetition 1 using TheilB for the FTSE data.

TheilB: Repetition 1		
Generation	**Mean**	**STD**
1	0.77074659329633	0.04895116803940
2	0.80467739321147	0.23756742909623
3	0.76045367399991	0.03459716794868
4	0.78281940951393	0.15766828731724
5	0.75108841699404	0.03223155848477
6	0.74831874049448	0.02614997642620
7	0.74664313157668	0.01935973263036
8	0.82018033721078	0.37640961448349
9	0.74591594072900	0.02497184849329

10	0.74358097825910	0.01987993826175
11	0.73982733138751	0.01672987541425
12	0.73365259111065	0.00909445043497
13	0.73587272042600	0.01504699792679
14	0.73635836199576	0.01337626748970
15	0.73347424395583	0.01148152080390
16	0.73455510558827	0.01731003870323
17	0.73034853187984	0.00727871706537
18	0.73252303930752	0.01065345603196
19	0.73278387811338	0.02613906018687
20	0.73144540201148	0.00972029392602
21	0.73026983594524	0.00966730383648
22	0.73058783386472	0.00632433787093
24	0.72852518134133	0.00781936719191
24	0.72985807332325	0.00740410215431
25	0.73130869954071	0.01063991802019

Figure I.12: Mean and Std of TheilB throughout all generations in repetition 1 for the FTSE data.

Repetition 2

TheilB: Repetition 2			
Generation 1		**Generation 25**	
10-17-16-1	0.76255605064605	19-4-1-1	0.74078352853616
11-24-6-1	0.79995204529445	8-19-3-1	0.72337560308454
15-15-19-1	0.81497699756415	1-4-1-1	0.72758169447964
15-10-22-1	0.85731762161783	1-4-1-1	0.72832878976486
19-25-1-1	0.72670769704184	8-4-2-1	0.73472438343445
15-13-7-1	0.79722671749949	1-4-1-1	0.72963808228680
10-24-14-1	0.75724522565000	8-27-2-1	0.73413610078024
10-24-24-1	0.78984825511080	19-15-1-1	0.73122666193466
13-23-6-1	0.73716475253647	8-27-1-1	0.73446833051836
10-24-11-1	0.75608268461777	12-27-1-1	0.73121559201995
18-15-4-1	0.73143476457716	11-15-6-1	0.73638550375113
14-5-1-1	0.73993035757330	8-15-2-1	0.72215990417771
19-17-24-1	0.79015658877448	1-4-1-1	0.72901157502917
7-9-9-1	0.78273922142957	1-4-2-1	0.73399399669247
13-17-19-1	0.76242172604959	1-4-1-1	0.76110526186065
6-11-17-1	0.73415207053374	8-27-1-1	0.73960245057910
10-2-22-1	0.73142398946553	8-27-3-1	0.73057376376052
11-9-4-1	0.73820759673540	8-4-2-1	0.72662565701605
19-3-17-1	0.75905542349020	8-4-2-1	0.72519902896838
11-20-15-1	0.80915317759243	4-28-3-1	0.73428185046846
17-28-27-1	0.77392365312294	2-27-20-1	0.77604891964398
13-22-27-1	0.77404377826343	8-4-1-1	0.72764973555345
3-25-5-1	0.73087740578314	8-4-2-1	0.74950012216169
20-17-24-1	0.79266344519399	19-4-2-1	0.72628551346770
9-21-2-1	0.74565645988304	8-4-2-1	0.72671767380432
10-20-21-1	0.88034882421206	1-27-2-1	0.72664049748227
7-8-0-1	0.76994442977303	1-15-1-1	0.73293676242410
2-24-16-1	0.75240319809561	1-15-2-1	0.72602615817803
20-19-30-1	0.76985226884836	8-27-1-1	0.72648983222761
20-22-23-1	0.77541720329423	1-4-1-1	0.72748270897881
12-23-13-1	0.77290214153685	19-4-1-1	0.73840660656907
13-5-21-1	0.72228274016821	1-27-1-1	0.75813837715832
20-16-6-1	0.75833389816282	1-4-2-1	0.72983628826863
4-8-23-1	0.72774666929478	12-27-2-1	0.72323756221698
9-19-26-1	0.77668271360736	1-27-1-1	0.72685534831228
20-21-24-1	0.94979587410256	8-27-2-1	0.72373717107182
20-27-4-1	0.74084134200264	8-27-2-1	0.75994167960964
7-26-4-1	0.75693014648560	8-27-1-1	0.75009938167650
20-19-20-1	0.76318305005566	19-27-1-1	0.72643210000112
13-9-12-1	0.74117866553761	8-27-1-1	0.74580283919670
Average	**0.77131902178060**	**Average**	**0.73456707592866**

Table I.13: The results of the second Repetition of the GA search on the FTSE data using TheilB.

TheilB: Repetition 2		
Top 10		
Structure	Times Considered	Times in Final Generation
8-27-2-1	17	3
1-27-1-1	22	2
8-4-2-1	23	5
8-27-1-1	23	5
8-4-1-1	26	1
1-4-2-1	26	2
1-4-1-1	28	6
19-27-2-1	38	0
1-15-1-1	38	1
1-27-2-1	46	1

Table I.14: The most frequently visited networks in repetition 2 using TheilB for the FTSE data.

TheilB: Repetition 2		
Generation	Mean	STD
1	0.77131902178060	0.04445944686711
2	0.76442157749507	0.02856377327300
3	0.76129349745041	0.04111999844688
4	0.75535715868057	0.05068644155445
5	0.81693560427959	0.43772442822191
6	0.76221362671730	0.09760803893110
7	0.74346274163658	0.01580153177733
8	0.74599086591583	0.02093897414905
9	0.74164990571886	0.01939217891660
10	0.74347510847027	0.02159351599604
11	0.73390951145325	0.01177502920533
12	0.73346321237139	0.00874922125478
13	0.73375037042562	0.01025618334001
14	0.73320490042625	0.00856283213162
15	0.73726156173633	0.02439442204190
16	0.73491758789636	0.01446666449812
17	0.73593488188912	0.01544095823912
18	0.73094067421410	0.00783955610980
19	0.73311226640041	0.01201815470220
20	0.72979997926893	0.00567693217366
21	0.73193488334635	0.01023653031223
22	0.73110120070384	0.00841515113653
24	0.73268356890142	0.00906320401021
24	0.72979017153292	0.00776409317098
25	0.73456707592866	0.01212591192593

Figure I.15: Mean and Std of TheilB throughout all generations in repetition 2 for the FTSE data.

Repetition 3

TheilC: Repetition 3			
Generation 1		Generation 25	
6-6-23-1	0.74293593850553	3-4-2-1	0.73952218870213
11-19-18-1	0.80663630134728	3-4-2-1	0.72470169708509
13-6-1-1	0.73023084993958	12-4-2-1	0.72921636359840
10-17-8-1	0.72459022847379	12-4-2-1	0.72545271427825
12-6-30-1	0.75730406932868	5-4-2-1	0.73412703076486
18-12-9-1	0.77460459492606	12-10-24-1	0.75124840457393
15-1-17-1	0.80463496312655	12-4-2-1	0.72777199206735
12-17-24-1	0.78461372281948	12-4-2-1	0.72478677085075
8-18-11-1	0.76225098250142	12-4-2-1	0.73646485031424
9-15-9-1	0.75536320819926	12-4-2-1	0.73719350193292
6-15-10-1	0.77669302015239	6-4-2-1	0.73687837296668
3-22-25-1	0.80170088491421	12-2-5-1	0.72172802173590
1-18-3-1	0.73175128217320	3-3-2-1	0.72899376205883
8-3-21-1	0.78174725589263	12-4-2-1	0.75079038769573

12-1-3-1	0.74383098656436	12-3-16-1	0.72179549143087
5-18-16-1	0.75473668788821	5-4-2-1	0.73208390020234
5-26-14-1	0.74226868484395	12-9-11-1	0.72658980095446
13-15-6-1	0.76132787248539	14-4-8-1	0.74622538224083
6-23-24-1	0.74986988370283	5-3-2-1	0.72998609759347
13-7-11-1	0.73326520531415	12-4-16-1	0.74062676853138
3-4-29-1	0.74726004082433	12-4-2-1	0.73064543178641
17-29-1-1	0.72616319277977	2-11-23-1	0.81380359847045
13-18-6-1	0.74267043549530	12-4-2-1	0.72747936993212
5-4-10-1	0.73042152290924	12-4-5-1	0.74634642272964
13-11-30-1	0.80693970027569	18-4-2-1	0.73364857026966
17-29-25-1	0.80149747302230	15-4-2-1	0.73282527986571
1-7-0-1	0.73896805012793	12-4-2-1	0.72491529473417
12-14-28-1	1.69183076649090	12-4-2-1	0.72712925641952
3-7-27-1	0.75362126614326	12-4-2-1	0.73676794502639
13-28-0-1	0.83767534606713	12-4-2-1	0.72782020445862
3-21-19-1	0.76087409282584	6-4-2-1	0.73084875004671
11-23-8-1	0.73178143464029	12-4-2-1	0.72345528713565
16-26-14-1	0.74618410727018	12-4-2-1	0.72698769494556
15-15-3-1	0.73083156366432	12-4-2-1	0.73054532211167
9-25-9-1	0.89051024335805	6-4-2-1	0.74040228376699
17-19-11-1	0.75188516757149	12-4-2-1	0.72295977817666
2-8-16-1	0.76338625503702	12-4-2-1	0.74588435960812
19-5-8-1	0.73461236363999	6-4-2-1	0.72493574989443
20-4-13-1	0.73165942302732	12-4-2-1	0.72593237228551
16-4-14-1	0.72889573235544	5-10-2-1	0.73388003290609
Average	**0.78420062001562**	**Average**	**0.73433491260371**

Table I.16: The results of the third repetition of the GA search on the FTSE data using TheilB.

TheilB: Repetition 3		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
3-4-6-1	11	0
12-28-2-1	11	0
12-27-2-1	12	0
12-28-1-1	13	0
12-1-1-1	19	0
3-4-1-1	21	0
3-4-16-1	23	0
12-4-1-1	53	0
3-4-2-1	131	2
12-4-2-1	247	19

Table I.17: The most frequently visited networks in repetition 3 using TheilB for the FTSE data.

TheilB: Repetition 3		
Generation	**Mean**	**STD**
1	0.78420062001562	0.15117767558147
2	0.76298502842435	0.03857488804010
3	0.77348370147352	0.13823222913569
4	0.76148698172130	0.04867290111032
5	0.76162962695504	0.07193547862366
6	0.78747710690305	0.23360690788595
7	0.75595743469110	0.04858207629080
8	0.74724781344474	0.02314303823169
9	0.74278773089516	0.02207925782823
10	0.73955757165335	0.02282004731523
11	0.73935179796536	0.02537101034329
12	0.73294340706796	0.00952915618925
13	0.73346929377066	0.01329082246750
14	0.73363709371557	0.01002702346253
15	0.73409868840459	0.01317178693899
16	0.73047621697051	0.00505292088679
17	0.73094387730163	0.00608332851271
18	0.73407638633097	0.00953079495459
19	0.73146904096522	0.00951466087344
20	0.73350210683748	0.00952783177244

21	0.73819756476797	0.02730341590094
22	0.73714505657302	0.02071786523091
24	0.73220892036801	0.00727128205486
24	0.74313883942427	0.06121214459898
25	0.73433491260371	0.01511857138976

Figure I.18: Mean and Std of TheilB throughout all generations in repetition 3 for the FTSE data.

❑ Metric Used: TheilC

Repetition 1

TheilC: Repetition 1			
Generation 1		**Generation 25**	
15-3-23-1	1.02593630699472	1-3-2-1	1.00337135708724
9-22-14-1	1.03598308619011	5-3-2-1	0.99724835686376
11-25-5-1	1.03189353787474	18-3-12-1	1.00029409637481
19-30-16-1	1.02678392758229	5-3-1-1	1.00295843100880
1-22-27-1	1.04878415138520	5-10-2-1	0.99681261036375
14-8-16-1	1.05101170915501	5-3-18-1	0.99860174054073
7-7-24-1	0.99266010831460	14-3-1-1	0.99951949648666
15-6-15-1	1.24523180295957	14-3-2-1	1.01555108114665
2-9-21-1	1.03470837129502	11-3-1-1	1.00096317466229
10-14-17-1	1.06733633562815	5-3-2-1	0.99935221231490
2-24-24-1	1.08114261594356	5-10-1-1	0.99971051037159
3-10-8-1	1.02517700027565	1-3-2-1	0.99689959843596
18-18-26-1	1.10048590348531	14-3-2-1	1.00983872288599
8-13-30-1	1.02296341135953	1-10-1-1	0.99786033322468
3-8-1-1	0.99723827774618	14-3-2-1	1.01134272904002
18-4-14-1	1.03481494598171	14-3-12-1	1.00311910264319
14-22-13-1	1.03992766782176	5-3-1-1	1.00947499234082
8-1-8-1	1.08122858463245	18-3-2-1	1.00200606686272
13-16-8-1	1.02047545037958	14-3-2-1	1.00284778485376
13-19-17-1	1.04757605275091	1-3-1-1	0.99760648199359
14-9-0-1	1.06366848666875	11-10-1-1	1.00003237662241
15-11-13-1	1.07306781669173	3-3-2-1	1.00134714246391
18-25-20-1	3.23988730241465	3-3-1-1	0.99882409193274
7-15-26-1	1.04669607202909	3-3-1-1	0.99389705667613
2-7-24-1	1.05952079488423	1-3-1-1	0.99656729742038
13-30-6-1	1.00912577445924	18-10-12-1	1.02615262246525
7-10-15-1	1.03538191146802	11-3-1-1	1.00099942664480
8-14-11-1	1.02219447149530	10-17-26-1	1.07534924081868
9-21-20-1	1.05365270394263	5-3-2-1	1.00067235689485
5-10-2-1	1.00682301192624	3-10-1-1	1.00537583023018
8-8-19-1	1.02591173310196	1-10-1-1	0.99716602400200
14-20-26-1	1.13381817624170	11-3-1-1	1.02780621111697
4-6-18-1	1.01901901472843	5-3-2-1	0.99237207146608
2-3-27-1	1.00119154652541	1-3-1-1	1.00378299827320
12-20-20-1	1.01432362494014	1-3-1-1	0.99988140323891
19-9-30-1	1.06313719517930	1-3-2-1	1.00187782922889
17-20-9-1	1.01521306953014	3-10-1-1	1.00138224253317
17-15-29-1	2.38438553175818	14-3-2-1	1.00269621867499
7-8-19-1	1.00934249399307	5-3-2-1	0.99949500448009
4-15-16-1	1.13264304110830	14-3-2-1	1.00289546698067
Average	**1.13550907552106**	**Average**	**1.00434879479166**

Table I.19: The results of the first Repetition of the GA search on the FTSE data using TheilC.

TheilC: Repetition 1		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
3-10-2-1	15	0
5-3-2-1	16	5
4-3-1-1	17	0
3-3-1-1	24	2

1-3-2-1	24	3
11-3-1-1	31	3
3-3-2-1	38	1
14-3-1-1	40	1
11-3-2-1	47	0
14-3-2-1	72	6

Table I.20: The most frequently visited networks in repetition 1 using TheilC for the FTSE data.

TheilC: Repetition 1		
Generation	**Mean**	**STD**
1	1.13550907552106	0.40397315241053
2	1.03798037616445	0.03258630114314
3	1.03109000066956	0.03011804991534
4	1.06559369913039	0.20490738454098
5	1.02460266408795	0.03177198806815
6	1.02918137566744	0.03213562874884
7	1.04349184177495	0.15524470194918
8	1.01487792558385	0.01843770955181
9	1.02014682226235	0.02624025037345
10	1.01137144157985	0.02214717599637
11	1.01383217620482	0.02397693705126
12	1.01103787770476	0.01676462668561
13	1.00921166673809	0.01849660269077
14	1.01478002936466	0.04417103643028
15	1.00968323941814	0.01271905404460
16	1.00934123118825	0.01571423210930
17	1.02768388955141	0.12117398285663
18	1.02318040347753	0.04135691828891
19	1.00760477588269	0.01489073879709
20	1.00890832936560	0.01803108317812
21	1.01196902174108	0.02179472491668
22	1.01648993933705	0.03874000923503
24	1.00863069818654	0.01977130671574
24	1.00792238846531	0.01234806934442
25	1.00434879479166	0.01359019547158

Figure I.21: Mean and Std of TheilC throughout all generations in repetition 1 for the FTSE data.

Repetition 2

TheilC: Repetition 2			
Generation 1		**Generation 25**	
10-2-26-1	1.03655337078927	8-4-14-1	1.01719990760559
14-19-27-1	1.05538527031850	3-7-5-1	1.01714738333103
13-17-15-1	1.04342383822502	4-7-14-1	1.02903464527979
5-15-4-1	1.01067965660619	4-7-5-1	1.00750393257849
7-11-5-1	1.00103469044560	8-7-5-1	0.99803729509583
19-7-16-1	1.03472554411983	3-1-5-1	1.00099602099411
9-6-14-1	1.01069526078735	3-2-5-1	1.00987598160475
6-13-12-1	1.03911899480850	3-1-14-1	1.04647696605205
5-15-14-1	1.06361476917331	4-1-14-1	0.99931339628120
9-5-3-1	1.00310583341832	4-7-5-1	1.06405163953827
8-21-23-1	1.12062082132039	3-4-5-1	1.01904575843927
6-29-19-1	1.05938120835122	8-1-5-1	1.00816521922633
16-3-9-1	0.99817869436170	3-1-5-1	1.00067117367900
4-21-18-1	1.20094391851956	4-1-5-1	0.99760706387498
7-14-30-1	1.05194611073932	8-7-5-1	1.00351252803548
4-14-16-1	1.04178283947176	8-7-5-1	1.00576746701579
20-22-19-1	1.05053612287807	3-7-5-1	0.99536869693794
17-13-9-1	1.03625659427663	3-1-5-1	1.00122694414256
20-5-20-1	1.04908972321100	3-7-5-1	1.00764796349476
17-28-5-1	1.02350344088493	4-2-5-1	0.99599623816473
5-11-27-1	1.06007805730425	3-1-5-1	1.00240485943143
18-8-5-1	1.06925092745343	3-4-5-1	1.00373767313624
16-14-18-1	1.02873757154273	3-1-5-1	1.00177312026809

7-12-30-1	1.08710281233147	8-7-5-1	1.00371980827939
16-27-4-1	1.03009418724811	4-1-5-1	1.00077491171257
7-14-4-1	1.00961471132723	4-7-14-1	1.02412559654018
17-7-7-1	1.01458868467658	4-1-5-1	1.00254205164671
10-3-0-1	1.00604553837792	8-1-5-1	1.00889960991655
5-5-0-1	1.02078706170967	3-2-5-1	1.01847624908139
3-3-14-1	1.00975497319317	3-7-14-1	1.04365434267609
16-2-1-1	1.01424188871495	3-7-5-1	1.03597708888867
20-25-11-1	1.02840986192786	3-1-5-1	1.00199821877471
11-18-13-1	1.02435151634814	4-2-14-1	1.03387331977261
2-4-6-1	1.00351432048884	3-7-5-1	1.00948223820116
15-22-20-1	1.08018310212650	8-7-5-1	1.02148921642927
11-9-4-1	1.00429944212166	3-2-14-1	1.00010886782646
8-17-11-1	1.00555094290246	3-4-14-1	1.01693393074859
14-30-5-1	0.99726816083567	3-2-5-1	1.00478164725574
11-27-9-1	1.05874742069763	3-1-5-1	0.99839936562070
16-13-7-1	1.05267090072190	4-1-5-1	0.99963631623437
Average	**1.03839671961892**	**Average**	**1.01143586634532**

Table I.22: The results of the second Repetition of the GA search on the FTSE data using TheilC.

TheilC: Repetition 2		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
3-2-5-1	17	3
4-1-5-1	19	4
14-4-5-1	23	0
8-4-5-1	26	0
4-7-5-1	31	2
3-7-5-1	32	5
8-1-5-1	36	2
14-7-5-1	46	0
3-1-5-1	63	7
8-7-5-1	105	5

Table I.23: The most frequently visited networks in repetition 2 using TheilC for the FTSE data.

TheilC: Repetition 2		
Generation	**Mean**	**STD**
1	1.03839671961892	0.03822695795035
2	1.02511982206707	0.02397340248970
3	1.02021942796966	0.02048337689386
4	1.05043277748730	0.13871543781286
5	1.03147311434886	0.04105035980513
6	1.01948291362973	0.02627953655254
7	1.02196301083359	0.02903958929662
8	1.02430266345901	0.02771158111552
9	1.02612730920877	0.03621783782763
10	1.03262812598979	0.04371434365855
11	1.12536365988086	0.46909451927490
12	1.02902398168248	0.03724927116347
13	1.02566041011639	0.03959187164820
14	1.01447010307616	0.01842506159039
15	1.01292273316017	0.02045183997605
16	1.01963634403375	0.02896695980843
17	1.01619936468210	0.02064670500389
18	1.01242682436156	0.02206486921513
19	1.01384727911084	0.01514713211335
20	1.01466839103730	0.01811668712588
21	1.01226138194281	0.01467094251392
22	1.01096434504512	0.01650245159862
24	1.01300812191500	0.01258750969559
24	1.00798610691671	0.01663381672781
25	1.01143586634532	0.01557789029345

Figure I.24: Mean and Std of TheilC throughout all generations in repetition 2 for the FTSE data.

Repetition 3

TheilC: Repetition 3			
Generation 1		Generation 25	
1-13-10-1	1.01196508296996	12-23-9-1	1.02084349748395
12-22-12-1	1.51701115607429	10-14-1-1	0.99994510751456
6-26-11-1	1.03126667385435	3-2-1-1	1.00588156237620
15-16-2-1	0.99932693575644	7-2-1-1	1.00186443156866
11-11-11-1	1.08649928881197	7-8-1-1	1.00361598589367
7-19-12-1	1.03068724363147	7-8-1-1	0.99325675408299
3-22-14-1	1.06849581824158	1-8-1-1	0.99783843853638
8-9-5-1	1.01808478655022	7-2-1-1	0.99788583302726
19-2-10-1	1.01305625098337	1-1-1-1	1.00312361688543
6-20-21-1	1.06752790805617	3-8-1-1	0.99803688595290
9-11-5-1	1.00178893705345	11-2-1-1	1.07088710265842
19-17-27-1	1.12812578120427	8-8-1-1	0.99913591929483
12-14-14-1	1.06275407783698	7-11-1-1	1.02159302312244
15-8-9-1	1.05827879825117	3-14-1-1	0.99998581250209
14-22-20-1	1.04294361758566	6-8-1-1	1.00005407226925
19-25-12-1	1.07445738452332	1-2-1-1	1.00192851793992
18-1-2-1	0.99960363982928	3-2-1-1	1.00596772084431
14-3-17-1	1.02473967356294	8-14-1-1	0.99582218081463
12-1-0-1	1.01440200564075	1-2-1-1	0.99791634354506
7-9-5-1	1.02238932230959	3-2-1-1	1.00900460901345
19-12-12-1	1.01955328537027	3-25-1-1	1.00394191832084
11-30-27-1	1.13445513328596	3-2-1-1	0.99795866981909
12-25-30-1	1.02746942224864	3-2-1-1	1.00978991960966
6-29-10-1	1.00900675226308	7-8-1-1	1.00138759043449
15-27-26-1	1.05803278121184	3-2-1-1	0.99753703020452
8-26-26-1	1.05267107903009	1-2-1-1	0.99980296613168
4-26-21-1	1.05843043437764	8-2-1-1	0.99938605574754
11-22-13-1	1.09864395671269	3-14-1-1	1.00397076315221
18-14-22-1	1.09044336817123	1-2-4-1	0.99946404914577
16-3-10-1	1.02616675050456	3-2-1-1	0.99513068297738
3-6-0-1	1.02373174156495	3-2-1-1	1.00655997711277
13-21-22-1	1.09689385923325	3-14-1-1	1.00261142990486
3-17-9-1	1.02351376256036	11-8-1-1	1.00118059689926
18-11-13-1	1.02502554943982	6-8-1-1	1.00052521245670
14-19-10-1	1.06066641038923	3-2-1-1	0.99987973557755
15-1-0-1	1.00908833652573	3-2-1-1	1.01050848597008
10-2-1-1	0.99641869830255	1-2-4-1	0.99996233666810
2-18-26-1	1.03173819906985	3-2-1-1	1.00634377296038
8-20-5-1	1.02399611452763	7-2-1-1	0.99691242506699
8-27-27-1	1.13311205121694	3-8-1-1	0.99971130452011
Average	**1.05681155171834**	**Average**	**1.00392880845016**

Table I.25: The results of the third Repetition of the GA search on the FTSE data using TheilC.

TheilC: Repetition 3		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
3-6-1-1	24	0
8-3-1-1	24	0
6-2-1-1	24	0
6-6-1-1	25	0
7-8-1-1	31	3
8-8-1-1	31	1
3-14-1-1	31	2
3-8-1-1	40	2
6-8-1-1	42	2
3-2-1-1	58	11

Table I.26: The most frequently visited networks in repetition 3 using TheilC for the FTSE data.

TheilC: Repetition 3		
Generation	**Mean**	**STD**
1	1.05681155171834	0.08345302055968

2	1.04357616836592	0.05494210684191
3	1.03521179471531	0.04101974858396
4	1.04175972761468	0.09962968783019
5	1.02108923496715	0.02568697215322
6	1.02225751413128	0.03453057774673
7	1.01588931202835	0.04786512182141
8	1.01336187854518	0.02316709203519
9	1.02172539382458	0.06494489909299
10	1.01013125193652	0.02024340192529
11	1.01385745356517	0.02679167928423
12	1.00806950650078	0.00880076844817
13	1.01379033965025	0.04010478612978
14	1.00558934285116	0.01185278648792
15	1.00795597710469	0.01622553431482
16	1.00703398841169	0.01124911475968
17	1.00716858900608	0.01249239339100
18	1.00738397997532	0.02074599809597
19	1.00554016776548	0.01265055932840
20	1.00925934556580	0.02940863999516
21	1.00541407181875	0.00819104038935
22	1.00502861374714	0.00879756487632
24	1.01017251746689	0.03034653774740
24	1.00423216483188	0.01028352798556
25	1.00392880845016	0.01234768690916

Figure I.27: Mean and Std of TheilC throughout all generations in repetition 3 for the FTSE data.

❑ Metric Used: MAE

Repetition 1

MAE: Repetition 1			
Generation 1		Generation 25	
18-15-29-1	0.00972124986667	6-3-2-1	0.00868427980048
16-15-10-1	0.00877673761978	6-22-2-1	0.00865447608136
5-6-23-1	0.00891051947370	6-1-2-1	0.00858813374620
20-12-21-1	0.00872705430365	6-20-2-1	0.00862569101658
6-9-3-1	0.00869710184667	11-26-9-1	0.00877028308233
14-6-13-1	0.00902607687438	6-3-2-1	0.00853135361487
5-17-27-1	0.00955041442937	6-1-2-1	0.00867213493446
20-4-27-1	0.00858536429038	6-3-2-1	0.00871208777344
17-26-2-1	0.00864180214005	6-3-2-1	0.00875471331486
19-20-19-1	0.00884666186967	6-3-2-1	0.00863561231051
8-10-12-1	0.00887081524655	6-22-2-1	0.00884817751580
20-23-3-1	0.00878141268714	6-3-2-1	0.00866013791024
3-9-18-1	0.00899374697932	6-1-2-1	0.00863422050258
10-10-27-1	0.00897309124154	6-3-2-1	0.00864928605432
12-3-1-1	0.00873206436583	6-22-2-1	0.00867839461032
10-13-9-1	0.00889687378157	6-20-2-1	0.00861246494076
8-13-24-1	0.00894324640393	6-22-2-1	0.00871951911985
8-23-25-1	0.00915426659893	6-9-2-1	0.00869223699977
3-12-7-1	0.00896620130055	6-22-2-1	0.00855575101310
19-2-27-1	0.00867976167075	6-3-2-1	0.00921396976875
17-14-30-1	0.01039621774471	6-3-2-1	0.00879469940856
9-7-3-1	0.00884404011489	6-22-2-1	0.00871712990846
13-16-13-1	0.00944668899992	6-3-2-1	0.00870162816656
5-20-23-1	0.00901243244162	6-22-2-1	0.00883944638464
20-4-18-1	0.00865417035085	6-9-2-1	0.00876704440055
13-24-14-1	0.00941793468744	6-22-2-1	0.00895133642145
4-15-16-1	0.00877659907427	6-22-2-1	0.00874513518887
2-18-20-1	0.00937184410246	6-22-2-1	0.00863636134353
11-2-27-1	0.00897068431244	6-3-2-1	0.00869019039138
18-8-10-1	0.00888245423601	6-16-18-1	0.00885193139134
12-8-28-1	0.00939793781624	6-9-2-1	0.00866161747229
6-22-2-1	0.00873621745520	6-20-2-1	0.00867230781519

6-3-21-1	0.00876358258486	6-22-2-1	0.00885278996875
8-10-17-1	0.00874098476160	6-3-2-1	0.00855832481974
1-15-13-1	0.00912819561004	6-22-2-1	0.00871339401391
5-9-10-1	0.00884405723294	6-3-2-1	0.00859279636663
6-11-4-1	0.00866647833221	13-1-2-1	0.00867129786273
15-14-13-1	0.00918707452686	6-22-2-1	0.00873971462032
3-16-0-1	0.00898128026728	6-22-2-1	0.00864540087871
10-5-17-1	0.00888045236944	6-22-2-1	0.00863775451904
Average	**0.00898934475029**	**Average**	**0.00870833063633**

Table I.28: The results of the first Repetition of the GA search on the FTSE data using MAE.

MAE: Repetition 1		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
3-3-2-1	16	0
3-9-2-1	17	0
6-1-2-1	18	3
3-7-2-1	25	0
6-20-2-1	28	3
6-16-2-1	33	0
6-9-2-1	40	3
3-22-2-1	68	0
6-3-2-1	102	13
6-22-2-1	165	15

Table I.29: The most frequently visited networks in repetition 1 using MAE for the FTSE data.

MAE: Repetition 1		
Generation	**Mean**	**STD**
1	0.00898934475029	0.00035332146784
2	0.00928440396751	0.00227860107989
3	0.00892718825394	0.00028564757771
4	0.00884261515381	0.00019855821514
5	0.00878938652122	0.00015523049383
6	0.00875488224217	0.00011489559859
7	0.00874814482435	0.00010964657459
8	0.00878618515200	0.00011065342783
9	0.00881219136233	0.00016829048108
10	0.00901058663670	0.00151927544552
11	0.00882249238259	0.00018045077483
12	0.00877864289294	0.00018950823660
13	0.00876655189764	0.00013064768318
14	0.00878792194224	0.00013327277366
15	0.00878609609235	0.00019456205381
16	0.00873833694645	0.00014497241828
17	0.00873386287771	0.00010730833115
18	0.00873864947919	0.00018220936539
19	0.00871195280365	0.00008219948753
20	0.00872174002837	0.00011499489960
21	0.00872086371588	0.00012454658379
22	0.00880915111966	0.00061816366915
24	0.00871897499454	0.00017954218192
24	0.00872568615577	0.00016551860545
25	0.00870833063633	0.00012144813268

Figure I.30: Mean and Std of MAE throughout all generations in repetition 1 for the FTSE data.

Repetition 2

MAE: Repetition 2			
Generation 1		**Generation 25**	
4-20-29-1	0.00910685202639	4-5-1-1	0.00863541421825
10-22-21-1	0.00929378150623	4-10-1-1	0.00856599620030
7-7-16-1	0.00911903376524	4-5-1-1	0.00871488523699
12-5-4-1	0.00883720813502	4-2-1-1	0.00864911250171

11-4-18-1	0.00890534972873	4-5-1-1	0.00870528589543
7-2-24-1	0.00874509975229	4-5-1-1	0.00866316739761
11-18-1-1	0.00886358959881	4-26-15-1	0.00900999257388
12-15-29-1	0.00913278190338	4-7-1-1	0.00869817309000
13-3-23-1	0.00877404086728	4-10-1-1	0.00863747417365
2-5-21-1	0.00896217602044	4-2-1-1	0.00867195231522
7-6-3-1	0.00871357559745	4-10-1-1	0.00863591121872
4-7-0-1	0.00871278082116	4-24-1-1	0.00865562986823
19-24-23-1	0.00912673768459	4-3-1-1	0.00868033584098
4-1-4-1	0.00860908894043	4-5-1-1	0.00870402248270
15-4-28-1	0.00891434423342	4-10-1-1	0.00866125695271
12-12-18-1	0.00919767994232	4-10-1-1	0.00874789025836
2-20-1-1	0.00884885953568	4-3-1-1	0.00862830301856
20-20-13-1	0.00922639219903	4-24-1-1	0.00858689676531
16-25-21-1	0.00929437137030	4-10-1-1	0.00879278863021
5-7-28-1	0.00874110418965	4-5-1-1	0.00868254120351
15-11-0-1	0.00896936052959	4-7-1-1	0.00865822465685
1-1-30-1	0.00869667184604	4-5-1-1	0.00866255762752
15-28-29-1	0.00956457641402	4-10-1-1	0.00859640777061
8-4-10-1	0.00919894570270	4-24-1-1	0.00862747863008
8-24-0-1	0.00903194444188	4-10-1-1	0.00875347771847
15-10-28-1	0.00933298743440	4-5-1-1	0.00875235677188
2-2-10-1	0.00879241347003	4-24-1-1	0.00871466978993
10-5-7-1	0.00887952334172	9-5-1-1	0.00865521062571
3-18-6-1	0.00885315386873	4-24-1-1	0.00859131835426
2-3-16-1	0.00870788089428	4-3-1-1	0.00861253794580
1-1-13-1	0.00874060170979	4-10-1-1	0.00860959378599
19-5-24-1	0.00896812557146	4-24-1-1	0.00858480492802
14-21-22-1	0.00914891241715	4-10-1-1	0.00872627909366
14-4-30-1	0.00892338872239	4-2-1-1	0.00869863768500
2-26-23-1	0.00907402022963	4-10-1-1	0.00874411431092
17-5-14-1	0.00900486353861	4-5-1-1	0.00870606338296
11-9-14-1	0.00894949130915	4-5-1-1	0.00873682984398
20-13-17-1	0.00882948082486	4-5-1-1	0.00866740863700
13-11-16-1	0.00905788637045	4-24-1-1	0.00860480162461
7-3-25-1	0.00871814152316	4-10-1-1	0.00868456406711
Average	**0.00896418044945**	**Average**	**0.00867785917732**

Table I.31: The results of the second Repetition of the GA search on the FTSE data using MAE.

MAE: Repetition 2		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
11-2-1-1	9	0
4-3-1-1	10	3
4-9-1-1	13	0
4-24-1-1	17	7
4-1-1-1	20	0
4-7-1-1	20	2
4-19-1-1	25	0
4-2-1-1	111	3
4-5-1-1	145	11
4-10-1-1	171	12

Table I.32: The most frequently visited networks in repetition 2 using MAE for the FTSE data.

MAE: Repetition 2		
Generation	**Mean**	**STD**
1	0.00896418044945	0.00021491112595
2	0.00882519532730	0.00017999658034
3	0.00890847174519	0.00049510683972
4	0.00947243811638	0.00349395011968
5	0.00883321074771	0.00025144610690
6	0.00878587445198	0.00025777889118
7	0.00880597460072	0.00026518213990
8	0.00874469851872	0.00017440373774
9	0.00903535177039	0.00132169657237
10	0.00873701044106	0.00015309453325

11	0.00872914534725	0.00012210195513
12	0.00870862424203	0.00021035430604
13	0.00870846387100	0.00008404757738
14	0.00870664798721	0.00015984556888
15	0.00869591370301	0.00010355994819
16	0.00871849054105	0.00013450060533
17	0.00869299331398	0.00011916720727
18	0.00869874200689	0.00012878599374
19	0.00869750065933	0.00012731881638
20	0.00867296992890	0.00009398474016
21	0.00877427351247	0.00038416543205
22	0.00873282495574	0.00018211561655
24	0.00871929493363	0.00013811994922
24	0.00869151711414	0.00009762570032
25	0.00867785917732	0.00007627391053

Figure I.33: Mean and Std of MAE throughout all generations in repetition 2 for the FTSE data.

Repetition 3

MAE: Repetition 3			
Generation 1		**Generation 25**	
9-18-6-1	0.00865725266898	9-30-1-1	0.00889152737719
20-6-30-1	0.00895240327346	12-14-5-1	0.00891380747521
20-4-5-1	0.00904845346457	15-30-1-1	0.00864708875187
15-24-26-1	0.00901408812660	9-14-1-1	0.00876106145566
15-16-7-1	0.00898923178902	1-2-1-1	0.00869987157346
3-3-6-1	0.00889132302369	17-2-5-1	0.00885751893075
9-25-23-1	0.00923316802102	9-2-1-1	0.00864215330168
6-7-27-1	0.00886283069982	17-2-5-1	0.00874375291972
9-27-28-1	0.00916976010885	1-2-4-1	0.00863252595120
12-12-0-1	0.00925824142155	19-2-1-1	0.00869622778946
20-17-1-1	0.00869057378029	1-2-1-1	0.00864999704989
15-16-9-1	0.00864897788727	9-6-1-1	0.00887804989550
16-1-11-1	0.00884916094533	17-2-1-1	0.00878062097174
6-21-17-1	0.00936353884797	9-6-1-1	0.00885913190899
10-12-23-1	0.00903095970356	17-6-10-1	0.00897236408932
10-4-22-1	0.00872355300211	17-12-1-1	0.00875663119065
7-24-20-1	0.00943413030509	9-6-1-1	0.00936442586176
11-6-24-1	0.00901295129659	12-30-1-1	0.00873040059843
16-5-29-1	0.00873000355225	9-30-5-1	0.00971354863764
11-28-26-1	0.01639337421547	15-2-1-1	0.00884062671625
17-27-14-1	0.00905416718624	9-30-1-1	0.00872695239465
4-3-11-1	0.00977099662963	1-2-1-1	0.00870809493213
5-8-16-1	0.00916073409238	17-2-5-1	0.00883557428706
5-4-17-1	0.00881574712371	9-2-5-1	0.00856791481129
5-25-17-1	0.00883126266076	15-30-1-1	0.00859144488516
20-6-28-1	0.00908954544771	9-30-1-1	0.00875014026028
8-6-3-1	0.00875170872971	12-2-1-1	0.00873842002998
17-21-17-1	0.00901444132651	15-2-1-1	0.00874657723062
20-10-1-1	0.00874104804208	9-2-5-1	0.00862456319611
9-12-7-1	0.00910724700802	17-2-5-1	0.00853893735409
2-4-29-1	0.00898027680706	9-2-1-1	0.00881921234457
6-24-20-1	0.00936382644490	17-2-1-1	0.00869006418613
7-5-5-1	0.00876981622838	9-2-1-1	0.00869411402953
3-14-25-1	0.00914456447640	9-6-1-1	0.00870008438791
4-23-5-1	0.00870241920776	15-2-1-1	0.00909621733973
3-8-18-1	0.00886223502629	17-30-1-1	0.00860083087461
19-15-11-1	0.00881746093788	9-2-5-1	0.00885526099485
20-7-25-1	0.00925173238778	9-2-1-1	0.00882351211862
16-26-27-1	0.00922801095741	9-2-1-1	0.00864101401446
3-11-7-1	0.00888945666675	9-6-1-1	0.00874202121812
Average	**0.00918251683802**	**Average**	**0.00878805708341**

Table I.34: The results of the third Repetition of the GA search on the FTSE data using MAE.

	MAE: Repetition 3	
	Top 10	
Structure	Times Considered	Times in Final Generation
17-2-1-1	18	2
9-6-29-1	20	0
9-2-24-1	22	0
9-2-11-1	24	0
9-6-11-1	26	0
9-2-5-1	36	3
9-6-24-1	42	0
9-6-5-1	55	0
9-6-1-1	56	5
9-2-1-1	103	5

Table I.35: The most frequently visited networks in repetition 3 using MAE for the FTSE data.

	MAE: Repetition 3	
Generation	Mean	STD
1	0.00918251683802	0.00119458516803
2	0.00914842802760	0.00149602577539
3	0.00916296044402	0.00117238799254
4	0.00898908454527	0.00026119323173
5	0.00929868332462	0.00184899283046
6	0.00899525937609	0.00064003738245
7	0.00898938481793	0.00036465574474
8	0.00894118827839	0.00023973175697
9	0.00892147660347	0.00023587064167
10	0.00902235726579	0.00090625709386
11	0.00914962197157	0.00145456005575
12	0.00911534169104	0.00187158187296
13	0.00887168137131	0.00033275617599
14	0.00883221941329	0.00029864062209
15	0.00896830611358	0.00104921095002
16	0.00884087629838	0.00022887215476
17	0.00877479170502	0.00018103503628
18	0.00887481813997	0.00067410419385
19	0.00906942591748	0.00136775819649
20	0.00883263518041	0.00023154102604
21	0.00882025781654	0.00032198030364
22	0.00914905451396	0.00256819837559
24	0.00873382809305	0.00014487145697
24	0.00872199235632	0.00011337961631
25	0.00878805708341	0.00021232700109

Figure I.36: Mean and Std of MAE throughout all generations in repetition 3 for the FTSE data.

Appendix II

In appendix II we present the complete list of results we obtained from the Genetic Algorithm search based on the S&P datasets for all metrics (TheilA, TheilB, TheilC and mae) in all repetitions. More specifically we present for each repetition three tables that include: a) the first and the last generation of the GA search, b) the ten most visited network structures by the GA as well as the frequency in which they were met in the last generation and c) the mean and the standard deviation of the metric used each time throughout all generations (from 1 to 25). We present as well the figures which depict: a) the mean and the standard deviation for each metric and each repetition throughout all 25 generations and b) The distribution for each metric in each repetition.

- **Metric Used: TheilA**

Repetition 1

TheilA: Repetition 1			
Generation 1		**Generation 25**	
10-9-3-1	1.01674032123354	2-10-3-1	0.99617102462064
11-8-19-1	1.06152126924323	2-27-3-1	0.99314061108058
8-16-12-1	0.99741854995173	2-27-1-1	0.99286125024988
11-26-3-1	1.00965490224592	2-10-3-1	1.02802014941916
16-5-17-1	1.02660616864941	2-27-3-1	1.06496157467294
19-8-17-1	1.03204549213933	2-27-3-1	1.00767330608999
12-9-26-1	1.00253200965297	2-25-13-1	1.01631550284512
2-27-9-1	1.01016294837508	1-27-1-1	0.99814602390372
12-9-22-1	1.12140366253561	2-27-3-1	0.99772182877575
5-3-14-1	0.98507744075772	15-27-1-1	1.00688275376560
5-17-1-1	1.00171701448101	20-10-3-1	1.00161738790905
12-3-29-1	0.99058305290029	2-27-3-1	0.98002218134218
19-9-21-1	1.02995035716883	2-27-3-1	0.99269079809153
9-25-26-1	1.06959026441482	1-25-1-1	0.99818805089038
14-23-14-1	1.08247398192704	2-25-13-1	0.99548025051562
14-27-1-1	1.02367460511902	15-27-3-1	1.00614286190132
16-26-11-1	0.99732068967256	2-27-1-1	1.00302482432601
13-28-15-1	0.97833415614826	2-29-1-1	0.99887165876673
10-15-1-1	0.98518295150281	2-10-13-1	0.99142977676360
20-18-16-1	1.04891889982107	2-25-1-1	0.99748074099401

1-21-17-1	1.00622900927090	2-27-3-1	0.99599663291651
1-9-10-1	1.02913465242850	2-10-3-1	0.99470069037489
15-27-7-1	1.00757509920938	1-27-1-1	0.99805520343116
20-10-7-1	1.02872845236987	2-27-3-1	0.99326228194394
4-18-18-1	1.02603701913459	2-27-1-1	0.99804610190767
5-26-24-1	1.01630113429424	2-27-13-1	1.04908872403441
1-30-6-1	0.99933564406770	2-25-3-1	0.98523490098971
20-18-14-1	1.02479058657248	2-27-1-1	1.01413611950403
19-2-28-1	1.04932259067562	2-25-1-1	0.98865826922526
4-25-14-1	1.00462661998162	2-27-3-1	0.98413214510922
1-17-23-1	1.09859855723603	2-27-1-1	0.99839252213163
18-11-6-1	1.04755684068926	1-25-1-1	1.00248052000429
20-16-1-1	1.01338529617733	3-7-1-1	0.99280273841346
17-24-19-1	1.06476058304878	15-11-20-1	1.24115210119972
19-3-3-1	1.07832171866701	2-27-1-1	0.99807048081832
10-15-16-1	1.03916685246834	3-27-1-1	1.00496325771956
4-28-30-1	1.13343465805010	1-7-3-1	0.99749706125562
7-20-10-1	0.98948140807188	15-24-6-1	1.01093309968983
13-10-15-1	1.04998410604530	3-7-1-1	0.99806395511542
10-18-29-1	1.06336160608450	1-27-1-1	0.99004201146639
Average	**1.03102602931209**	**Average**	**1.00756378435437**

Table II.1: The results of the first Repetition of the GA search on the S&P data using TheilA.

	TheilA: Repetition 1	
	Top 10	
Structure	**Times Considered**	**Times in Final Generation**
1-27-5-1	17	0
1-27-7-1	17	0
11-27-1-1	18	0
11-27-3-1	20	0
4-27-3-1	31	0
2-27-5-1	38	0
1-27-1-1	40	3
1-27-3-1	51	0
2-27-1-1	77	6
2-27-3-1	153	9

Table II.2: The most frequently visited networks in repetition 1 using TheilA for the S&P data.

	TheilA: Repetition 1	
Generation	**Mean**	**STD**
1	1.03102602931209	0.03685938937668
2	1.07515876345684	0.20113917810003
3	1.03657408510144	0.08345848935369
4	1.02877704484134	0.04147552202173
5	1.02367819524646	0.05088335539216
6	1.02672603238816	0.07464021324217
7	1.03585102028240	0.17657016017945
8	1.01680902823607	0.03673615402445
9	1.00708812163881	0.02366692213190
10	1.00886601985485	0.02469038544834
11	1.01891953395836	0.09411205805880
12	1.01222973326057	0.05581799449547
13	1.00380755343800	0.01759827519680
14	0.99885813558073	0.01516077770100
15	1.00582130586593	0.02136918286419
16	0.99979665848459	0.01432169121900
17	1.00377706767460	0.03003203275375
18	1.00164688066194	0.02163475758187
19	1.00612533534150	0.02123338722841
20	1.00346669465632	0.01706522542128
21	1.00092473180676	0.01580683002060
22	1.00185611532501	0.01988124507503
24	1.00387535417187	0.02920974492628
24	1.00117682355519	0.02667998089688
25	1.00756378435437	0.04101102941005

Figure II.3: Mean and Std of TheilA throughout all generations in repetition 1 for the S&P data.

Repetition 2

TheilA: Repetition 2			
Generation 1		Generation 25	
9-28-1-1	1.00382890795919	2-8-3-1	0.99164419230536
4-9-1-1	0.99964182072060	3-26-6-1	0.98149079822034
14-14-2-1	1.00400000221744	1-8-3-1	0.99806759373223
19-24-22-1	1.12324795520694	3-8-3-1	0.99858126457773
7-23-29-1	1.01502784920761	3-8-3-1	1.01008739926850
15-2-16-1	1.04784944232386	3-28-3-1	0.99931259628266
12-2-15-1	1.00744862911104	3-8-8-1	1.05483250891300
2-20-7-1	1.50756627541676	3-26-15-1	1.06409823203046
2-27-28-1	1.04728927064332	7-30-10-1	1.00033542683217
9-3-22-1	1.16499120841988	3-22-3-1	0.99343183708843
8-21-8-1	1.01663538373660	3-8-3-1	0.98452311650220
13-15-5-1	1.01563178435262	3-8-3-1	0.98906151142631
9-13-10-1	1.03917739597152	3-12-8-1	0.98874801698079
15-5-18-1	1.04604238576553	3-8-3-1	0.98452142481181
14-19-24-1	1.00127424332762	3-8-8-1	1.00485014160360
12-11-24-1	1.15617180506800	3-6-3-1	1.00143626700946
5-20-1-1	0.99310701711722	3-8-2-1	0.99639637661406
11-15-22-1	1.03287285052711	3-8-3-1	0.99563885940733
17-9-5-1	1.03308412492395	2-8-3-1	1.02914617603958
9-28-4-1	0.99145796362705	3-28-3-1	1.00430588051317
14-13-30-1	1.01003281078810	3-8-6-1	0.98822474063799
8-2-20-1	0.99042402366628	3-8-3-1	1.00714933389583
14-7-7-1	1.02651783301094	2-8-3-1	0.99073438910318
7-5-10-1	1.02210718727811	3-8-3-1	0.98538048971950
17-29-11-1	1.04030997367422	3-11-3-1	0.99764525388294
3-12-24-1	1.02826813517832	3-8-6-1	1.00881010375646
16-23-9-1	1.00182027003093	3-28-3-1	1.00607625094766
15-16-5-1	1.03143037850492	3-22-3-1	1.00161121404454
13-6-3-1	1.00653832071521	3-8-3-1	1.00050315354271
15-7-19-1	1.05442534560716	3-28-3-1	0.99649679695580
16-18-27-1	1.06395030932731	3-8-6-1	0.99016179042100
5-17-22-1	1.02731182595737	3-12-3-1	0.98892292398429
4-1-20-1	0.99755188563277	3-12-3-1	0.99809776934564
15-6-12-1	1.21904950775875	3-8-3-1	0.99546869356681
7-11-10-1	1.01705727915059	3-12-3-1	0.99418152809191
3-26-12-1	0.99166260903375	3-8-3-1	0.99611890187084
10-8-27-1	1.01786185027106	3-28-2-1	1.00063377364922
2-18-12-1	0.97369310146501	3-6-2-1	0.99716195996563
15-7-29-1	1.07230371477054	3-28-8-1	1.00899082777641
14-28-12-1	1.01366225432207	3-26-3-1	1.01291113416774
Average	**1.04630812329468**	**Average**	**1.00089476623713**

Table II.4: The results of the second Repetition of the GA search on the S&P data using TheilA.

TheilA: Repetition 2		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
1-8-8-1	16	0
1-26-3-1	16	0
3-8-2-1	18	1
3-22-3-1	20	2
3-28-8-1	24	1
3-8-12-1	27	0
3-26-3-1	39	1
3-26-8-1	48	0
3-8-8-1	50	2
3-8-3-1	93	11

Table II.5: The most frequently visited networks in repetition 2 using TheilA for the S&P data.

TheilA: Repetition 2		
Generation	**Mean**	**STD**
1	1.04630812329468	0.09020963254430

2	1.05804997868736	0.15746028217796
3	1.14866745286417	0.62149960277385
4	1.16817408956494	0.80103338903237
5	1.03020761290137	0.03962951767915
6	1.06022901211300	0.16483343543114
7	1.02078867378932	0.02762307913314
8	1.01472780175725	0.02552452814620
9	1.02433632686368	0.04915280791921
10	1.01261124502234	0.03994771040294
11	1.01037665034755	0.03078120241819
12	1.05582614165554	0.24040679149753
13	1.02390169823368	0.04823449023874
14	1.04469698992567	0.20900662508499
15	1.00503115234038	0.01583682083762
16	1.01592560279906	0.05820414314095
17	1.00410733590162	0.02305575139415
18	1.00727429712349	0.01987909076842
19	1.00906363153772	0.04348136859581
20	1.00029563130607	0.01407444614466
21	1.00752591263887	0.04412970661510
22	1.02053112283162	0.11737558836433
24	1.01341730951552	0.06545756092918
24	1.00441001458449	0.03898519565716
25	1.00089476623713	0.01638551410462

Figure II.6: Mean and Std of TheilA throughout all generations in repetition 2 for the S&P data.

Repetition 3

TheilA: Repetition 3			
Generation 1		**Generation 25**	
12-9-20-1	0.98892887948957	8-6-2-1	0.98924684615325
3-5-12-1	1.06749253580088	6-7-4-1	0.98676150477196
9-18-16-1	1.04145100883170	11-10-2-1	0.99499422543083
20-12-6-1	1.02448368209515	12-22-2-1	0.99791425787662
7-7-9-1	1.02089673002327	6-7-2-1	1.01372490919178
5-19-18-1	1.07608189039220	12-7-2-1	1.00670505555468
9-20-29-1	1.05682427916788	1-10-2-1	1.01275248139166
2-17-21-1	0.99145668282378	6-6-2-1	0.99794663021328
4-12-3-1	0.99725423516924	1-3-16-1	0.98572435651090
20-14-7-1	1.02756726057120	8-5-2-1	1.01448177281263
2-17-28-1	1.06084018008798	19-7-22-1	1.01784642582344
12-17-2-1	1.00736639010816	6-10-2-1	1.00360562075025
11-2-28-1	1.04449189983866	6-6-4-1	0.99734218938980
5-3-19-1	0.99271867579604	6-5-4-1	0.98005273498159
19-16-18-1	1.03649467715470	12-10-4-1	0.98982536368004
8-5-4-1	0.97482624376742	14-5-4-1	1.01799319411598
17-7-5-1	1.03358033638196	19-7-22-1	1.04090741373705
15-24-15-1	1.00431840405814	6-5-2-1	0.98742298707030
10-20-30-1	1.09071670369404	6-7-2-1	0.98220961873415
3-20-20-1	0.99494479790514	4-30-29-1	2.67201086723304
13-5-5-1	1.01649119202247	12-10-2-1	0.99164213878266
9-16-25-1	1.09411478012979	3-1-4-1	0.99663723696081
4-27-28-1	1.02096632483180	6-7-2-1	0.99856726233776
6-23-30-1	1.00745373215685	6-6-5-1	0.96253303737967
17-2-0-1	1.00141553330503	6-6-4-1	0.99992050755834
12-22-27-1	1.00243550238285	6-6-22-1	1.04344386759136
5-16-16-1	0.98933100436657	6-4-4-1	1.00157816020721
11-4-9-1	0.99681944734160	8-10-2-1	1.00140323647320
6-11-13-1	1.01212621822098	12-7-4-1	0.98467065007620
14-14-11-1	1.05734804094705	6-4-2-1	1.00306374149422
20-4-3-1	1.04916052903597	12-10-4-1	0.98217393018468
16-28-30-1	1.25918116183688	6-7-4-1	0.97188152266876
2-29-24-1	1.02356273290070	12-10-2-1	0.98498649938606
10-5-14-1	0.99892559339866	4-7-4-1	0.98956810916584

2-30-18-1	1.00674693160346	6-7-4-1	1.00438751206512
19-26-26-1	1.11144882162755	6-7-2-1	0.99050153754485
1-21-13-1	1.01132149995692	12-10-2-1	1.02874421602256
1-24-27-1	1.04077085248445	6-10-2-1	1.03839942817996
13-24-24-1	1.02214635313019	6-10-2-1	0.97460282005598
10-4-21-1	1.11350704933510	3-6-2-1	0.99890026640013
Average	**1.03420021985430**	**Average**	**1.04092685339896**

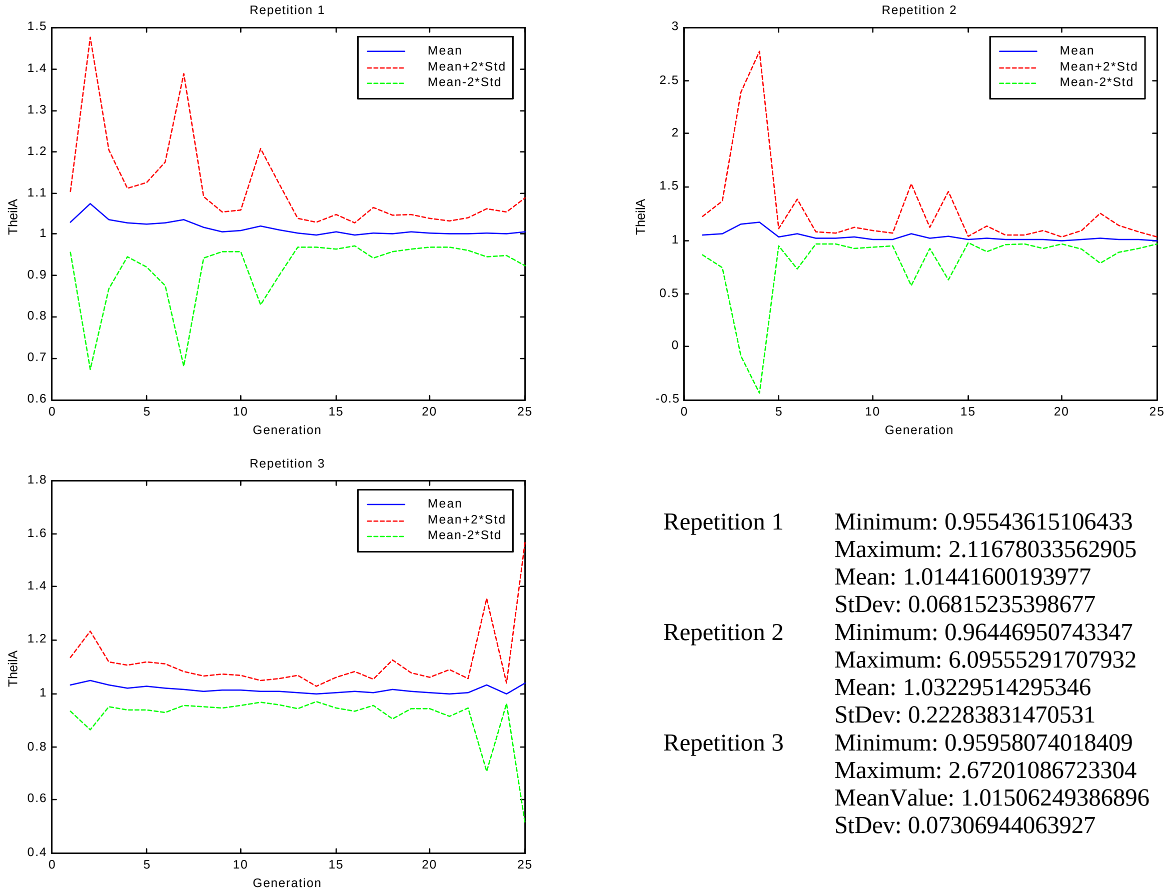
Table II.7: The results of the third Repetition of the GA search on the S&P data using TheilA.

TheilA: Repetition 3		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
11-7-2-1	14	0
3-7-4-1	15	0
3-14-2-1	17	0
12-14-2-1	18	0
3-5-2-1	27	0
6-10-2-1	36	3
3-10-2-1	43	0
3-10-4-1	44	0
6-7-2-1	44	4
3-7-2-1	59	0

Table II.8: The most frequently visited networks in repetition 3 using TheilA for the S&P data.

TheilA: Repetition 3		
Generation	**Mean**	**STD**
1	1.03420021985430	0.05028907541858
2	1.04847993440806	0.09146271967752
3	1.03336720446340	0.04211376465622
4	1.02190651963839	0.04306416461799
5	1.02807575061685	0.04605627957523
6	1.02025947109491	0.04540223032579
7	1.01774271453529	0.03148993363317
8	1.00898322149018	0.02884002896741
9	1.01063789275734	0.03214695091722
10	1.01143402039930	0.02832553334035
11	1.00860141726667	0.02102981183655
12	1.00733088359693	0.02449937399139
13	1.00519649106312	0.03182746792233
14	0.99989357528340	0.01399523789161
15	1.00430636695155	0.02920888927487
16	1.00740212635680	0.03771533050717
17	1.00433262919855	0.02454580411744
18	1.01681354490186	0.05517028338221
19	1.00865153093969	0.03387558995550
20	1.00304746573789	0.02978227440289
21	1.00128793342908	0.04396480629388
22	1.00161308962195	0.02787699881861
24	1.03135833163462	0.16165680097021
24	1.00071315808487	0.01973543167513
25	1.04092685339896	0.26511672068132

Figure II.9: Mean and Std of TheilA throughout all generations in repetition 3 for the S&P data.

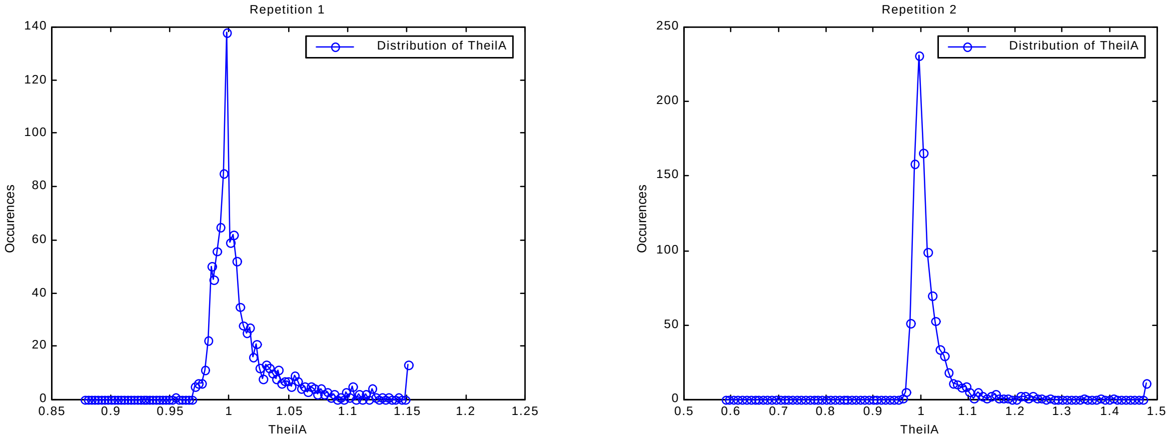


Repetition 1 Minimum: 0.95543615106433
Maximum: 2.11678033562905
Mean: 1.01441600193977
StDev: 0.06815235398677

Repetition 2 Minimum: 0.96446950743347
Maximum: 6.09555291707932
Mean: 1.03229514295346
StDev: 0.22283831470531

Repetition 3 Minimum: 0.95958074018409
Maximum: 2.67201086723304
MeanValue: 1.01506249386896
StDev: 0.07306944063927

Figure II.1: Mean and Std of TheilA throughout all generations for S&P data



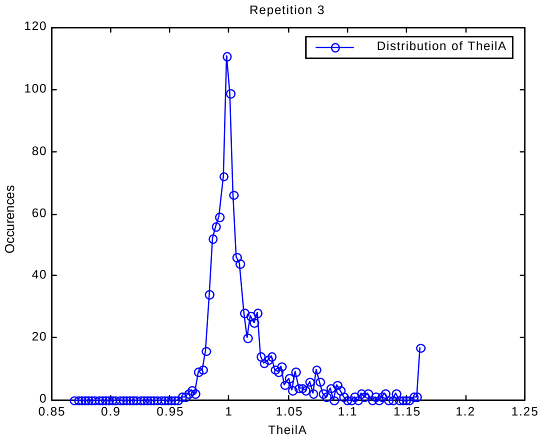


Figure II.2: Distributions of TheilA for the S&P data.

❑ Metric Used: TheilB

Repetition 1

TheilB: Repetition1			
Generation 1		**Generation 25**	
6-29-10-1	0.69402640608358	5-8-4-1	0.74245392734983
1-8-30-1	0.69221953636456	5-14-1-1	0.68201955182204
16-11-14-1	0.70457250494219	5-8-1-1	0.68292785962835
19-21-1-1	0.69344103699837	5-30-1-1	0.67063045728494
16-16-29-1	0.71664594722315	5-8-28-1	0.68932271061489
15-28-7-1	0.73782589353192	5-14-1-1	0.67614000764103
19-1-24-1	0.75639552449061	5-15-11-1	0.67040771021457
11-12-14-1	0.69224854944453	5-8-4-1	0.67862873053819
19-29-0-1	0.86694465064241	5-8-1-1	0.68128618372687
3-23-26-1	0.68530912213062	5-14-4-1	0.66229141047792
11-29-2-1	0.69662374672499	5-8-1-1	0.68198817678510
20-23-2-1	0.69539841018868	5-8-1-1	0.68299402459088
5-25-16-1	0.67810337052783	5-8-4-1	0.73001306515439
11-26-11-1	0.73403699385381	5-8-4-1	0.67827142597409
4-19-28-1	0.70360969831549	5-8-11-1	0.68094019747031
5-23-5-1	0.66263222564995	5-8-1-1	0.66431776468434
9-13-12-1	0.73103887185464	5-8-1-1	0.67785050693753
16-14-15-1	0.70646850957191	5-14-4-1	0.68095477947510
15-10-1-1	0.71447470230284	5-8-1-1	0.68337507717604
10-9-2-1	0.69269414744817	1-8-29-1	0.68783013746018
6-25-10-1	0.69098114690989	5-8-4-1	0.67841253085053
12-14-12-1	0.73102906286734	5-8-11-1	0.70810041072052
7-20-16-1	0.70179263212717	5-15-4-1	0.66380280434184
10-10-4-1	0.68494139378727	5-8-4-1	0.70455854213615
8-25-4-1	0.68188940868894	5-5-4-1	0.66684441391692
7-26-28-1	0.70695807037813	5-14-4-1	0.67547314260186
9-8-8-1	0.75674567199455	5-8-1-1	0.68268464846331
10-1-4-1	0.68826759650912	3-29-10-1	0.68286114248838
14-5-1-1	0.69158353298780	5-8-4-1	0.65979698432567
19-11-8-1	0.74337892397498	5-8-1-1	0.70203338922179
11-18-7-1	0.68873884646873	5-8-4-1	0.66917116531587
1-13-14-1	0.68750210793105	5-22-27-1	0.67761097337484
6-17-30-1	0.73386157863712	5-8-4-1	0.67409492537802
19-30-11-1	0.69323736699251	5-8-1-1	0.68302034556526
7-4-11-1	0.70584742716032	5-8-4-1	0.69702786803216
19-5-22-1	0.71121931205457	5-8-4-1	0.71561061937991
2-21-29-1	0.73470402428047	5-5-4-1	0.66586916472402
16-24-12-1	0.71696095056949	5-8-1-1	0.67906554733599
18-19-16-1	0.72228668741762	5-6-1-1	0.68039653720582

20-6-17-1	0.74531327951611	1-5-4-1	0.68317713752578
Average	**0.71179872173859**	**Average**	**0.68335639994778**

Table II.10: The results of the first Repetition of the GA search on the S&P data using TheilB.

TheilB: Repetition 1		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
11-14-1-1	14	0
15-21-1-1	15	0
6-5-1-1	16	0
6-21-1-1	17	0
5-5-1-1	25	0
5-14-4-1	28	3
5-5-4-1	34	2
5-14-1-1	42	2
5-8-1-1	64	11
5-8-4-1	104	11

Table II.11: The most frequently visited networks in repetition 1 using TheilB for the S&P data.

TheilB: Repetition 1		
Generation	**Mean**	**STD**
1	0.71179872173859	0.03382187902311
2	0.70141180367685	0.02328051021601
3	0.69515957008248	0.02050218565968
4	0.73510368814496	0.17101654993517
5	0.80291814310880	0.66524160479464
6	0.69560812801306	0.02393160832130
7	0.69470226493938	0.02054786634272
8	0.69295361172128	0.02243389936192
9	0.69493269979713	0.02260243084875
10	0.69297844147294	0.01691673407539
11	0.69610076439150	0.02384675958139
12	0.70452168576096	0.08391258127204
13	0.69059628126288	0.01556658636915
14	0.69032219055344	0.01501335052918
15	0.68934845553206	0.02308888136983
16	0.69598890348128	0.04124601881118
17	0.69012273885057	0.02507016739980
18	0.68480823107874	0.01491875799046
19	0.68473065011144	0.01591618394063
20	0.67915074392762	0.01241722585699
21	0.67917119827313	0.00929109749427
22	0.68228983689630	0.02018829457647
24	0.68588679030745	0.02474692772195
24	0.70113159962782	0.10423227445488
25	0.68335639994778	0.01724601473687

Figure II.12: Mean and Std of TheilB throughout all generations in repetition 1 for the S&P data.

Repetition 2

TheilB: Repetition 2			
Generation 1		**Generation 25**	
10-17-13-1	0.71283243582108	20-3-7-1	0.68291785198663
14-9-23-1	0.74407599616666	9-26-15-1	0.72816142823464
7-11-4-1	0.69051663510100	3-6-5-1	0.67226008117247
16-2-20-1	0.72625211166162	3-24-10-1	0.71372809959724
2-12-9-1	0.69010764344036	6-27-11-1	0.81065124017392
3-11-26-1	0.68906525504324	6-27-7-1	0.67433384581398
3-16-19-1	0.74834809318176	5-24-5-1	0.70633707605777
12-1-4-1	0.70050761568713	3-6-10-1	0.70372551773268
14-23-26-1	0.72141147477459	9-3-5-1	0.69276241771761
13-23-28-1	0.69828511491073	6-24-11-1	0.69272944026763
14-30-10-1	0.70067039968538	3-24-11-1	0.69318431697219

10-28-19-1	0.82591159259536	2-24-5-1	0.67251184848005
20-24-18-1	0.74659018199425	5-24-5-1	0.66896718665487
14-10-3-1	0.69468024609951	11-27-5-1	0.69124669094238
6-6-9-1	0.67738854073737	6-6-5-1	0.69356816222432
2-23-22-1	0.69532854159151	3-9-5-1	0.67784940744659
9-18-16-1	0.70057330089406	5-24-10-1	0.67797063462224
12-24-7-1	0.73182855289443	9-24-5-1	0.69807755975257
16-12-21-1	0.75821845191271	9-27-10-1	0.70981876042601
8-5-16-1	0.68694773089132	9-24-10-1	0.74380033086422
2-24-19-1	0.69203996579844	2-6-5-1	0.68191454752207
19-2-18-1	0.70245994729280	5-5-5-1	0.69540756002251
9-30-22-1	0.70884530073371	5-26-15-1	0.70645316154481
14-28-11-1	0.85825374255046	5-6-5-1	0.66781309175073
20-12-16-1	0.69166410996147	3-11-5-1	0.67216181179770
10-12-26-1	0.76340043320400	2-24-5-1	0.70987348526188
14-7-14-1	0.74471850305307	11-24-5-1	0.69808759854852
5-26-27-1	0.69985233504398	3-24-5-1	0.69023295326908
9-4-2-1	0.73484066323215	2-9-10-1	0.67876318800019
10-11-5-1	0.69181549589757	3-9-5-1	0.67816251383342
19-8-27-1	0.70189552284871	15-13-18-1	0.68712156929140
11-30-22-1	0.79777464357140	2-24-5-1	0.70073397168819
6-3-27-1	0.70322100647675	9-27-10-1	0.68834427224697
9-3-15-1	0.73788007654446	2-24-5-1	0.68669736631903
6-19-21-1	0.70847492999187	11-6-5-1	0.67777008512657
5-14-27-1	0.67665960368114	5-27-5-1	0.68319286606697
8-24-1-1	0.68156768110030	3-9-10-1	0.68668208484530
13-2-12-1	0.68849561240277	5-6-5-1	0.68449749018671
1-10-28-1	0.69467827848856	9-24-10-1	0.71373874980683
7-5-16-1	0.66888666624490	5-26-5-1	0.67347217637990
Average	**0.71717411083006**	**Average**	**0.69414306101622**

Table II.13: The results of the second Repetition of the GA search on the S&P data using TheilB.

TheilB: Repetition 2		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
6-5-11-1	13	0
5-5-5-1	14	1
5-24-5-1	14	2
6-24-11-1	15	1
3-5-11-1	15	0
6-5-5-1	22	0
3-24-11-1	29	1
3-5-5-1	30	0
6-24-5-1	54	0
3-24-5-1	70	1

Table II.14: The most frequently visited networks in repetition 2 using TheilB for the S&P data.

TheilB: Repetition 2		
Generation	**Mean**	**STD**
1	0.71717411083006	0.04016628224090
2	0.74080094767905	0.21418127706747
3	0.70032737488139	0.02956473921564
4	0.74651661568058	0.15975860517528
5	0.77474255509688	0.23876414867291
6	0.71071139639619	0.03100438598837
7	0.78261468401863	0.37294811949089
8	0.70691588978741	0.03169752384880
9	0.70362564763802	0.03056063713399
10	0.70003204801989	0.04177420744423
11	0.72784390569737	0.15666732941127
12	0.68737154379940	0.01622069312188
13	0.69257524035989	0.04395240421040
14	0.69803926331108	0.06696781479911
15	0.69337435223419	0.02271992980077
16	0.74304037503525	0.32268991018835
17	0.69340383733837	0.04516544169418

18	0.69093384728701	0.02486720407850
19	0.69746830167550	0.05458505109784
20	0.72931201013539	0.26306919086252
21	0.68728681390040	0.01243947589559
22	0.69362499762337	0.04400261935057
24	0.71419570552291	0.10647267775306
24	0.71179739248754	0.12403524250175
25	0.69414306101622	0.02506464852767

Figure II.15: Mean and Std of TheilB throughout all generations in repetition 2 for the S&P data.

Repetition 3

TheilC: Repetition 3			
Generation 1		**Generation 25**	
8-13-0-1	0.71916442001415	3-8-1-1	0.68266969632829
7-30-25-1	0.87963098934679	3-1-1-1	0.73842352043891
17-28-6-1	0.71660758761507	1-1-1-1	0.69810312773029
16-3-13-1	0.69659099538981	1-1-1-1	0.68565899556320
18-1-30-1	0.70850688696663	1-8-1-1	0.68360612697136
8-14-4-1	0.73120262494365	1-8-4-1	0.67454777413676
20-1-2-1	0.68883963811900	1-8-1-1	0.67657973566247
3-6-15-1	0.70400038486082	3-6-1-1	0.68022211434828
11-3-7-1	0.68524757152200	1-6-1-1	0.67502220755333
2-1-5-1	0.70635467350505	1-8-1-1	0.68332312387201
7-10-5-1	0.66760745550999	1-6-1-1	0.67139771909341
7-28-5-1	0.68435168857503	1-6-1-1	0.68131886267254
3-2-19-1	0.70302942353287	1-1-4-1	0.68249737207816
8-25-10-1	0.68393490518279	1-6-1-1	0.68451845852368
15-16-24-1	0.73023438600910	1-1-1-1	0.68324166734494
8-17-3-1	0.71999936576128	1-1-1-1	0.68350036212110
15-16-10-1	0.70103419456432	1-1-1-1	0.68425247043304
11-23-4-1	0.70807709401127	1-1-1-1	0.68376416260145
18-29-27-1	0.74625088580754	1-6-1-1	0.67713450803963
1-8-24-1	0.72582144383957	3-1-1-1	0.68473829178320
12-13-4-1	0.68653911012982	1-1-1-1	0.68440086890815
8-21-12-1	0.67606801964813	1-6-1-1	0.68276288930984
14-26-7-1	0.70463914154845	1-8-1-1	0.68236209000969
1-6-29-1	0.67453895498009	14-22-1-1	0.69350951180859
1-20-4-1	0.68411672983186	1-8-4-1	0.69350204096795
15-23-2-1	0.67210010716423	4-1-1-1	0.69265187385419
2-3-5-1	0.68496736360049	1-8-1-1	0.68326440919969
5-11-8-1	0.67402749228686	11-6-1-1	0.68739282509479
1-19-14-1	0.68061629351643	1-12-1-1	0.68546340464013
14-26-22-1	0.86148052618263	1-8-1-1	0.68198659406048
19-9-25-1	0.76599155868841	3-2-4-1	0.68277603258168
11-15-29-1	0.70163376586139	8-25-28-1	0.81617246080167
15-11-9-1	0.67899509022506	1-8-1-1	0.68200121371241
16-30-24-1	0.73736264297528	1-8-1-1	0.68335246212669
2-12-29-1	0.68781111478460	3-1-1-1	0.67588511006702
6-15-15-1	0.70402177848009	1-8-1-1	0.67772004218626
11-20-1-1	0.68789431474524	3-8-1-1	0.67441594863378
13-8-8-1	0.73949906704385	1-8-1-1	0.68338998448511
3-24-25-1	0.72882940982318	1-6-1-1	0.67280342182900
11-2-15-1	0.68833004161005	1-8-4-1	0.68561807037001
Average	**0.71064872845507**	**Average**	**0.68739878879858**

Table II.16: The results of the third Repetition of the GA search on the S&P data using TheilB.

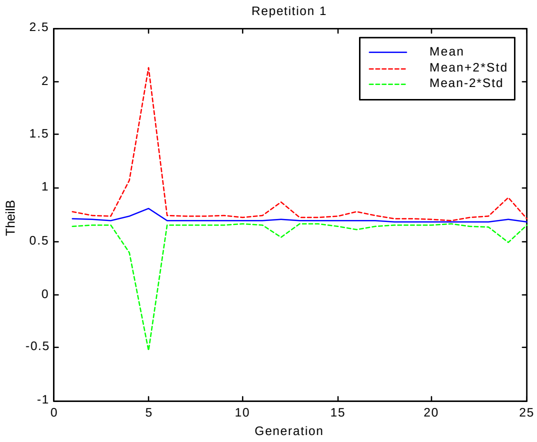
TheilB: Repetition 3		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
1-1-4-1	14	1
1-2-1-1	19	0
1-26-1-1	19	0
1-8-4-1	19	3

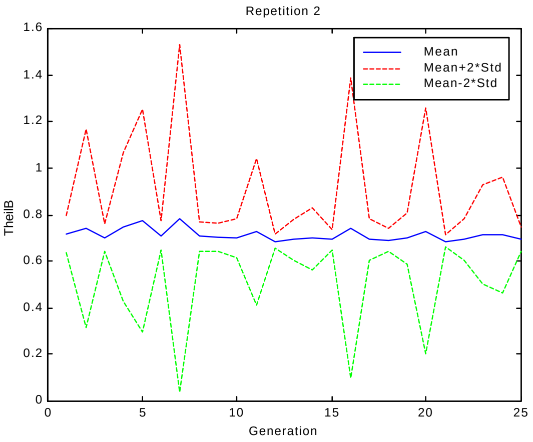
3-8-1-1	23	2
1-8-1-1	37	10
1-6-1-1	51	7
3-1-1-1	53	3
1-1-1-1	56	7
3-6-1-1	62	1

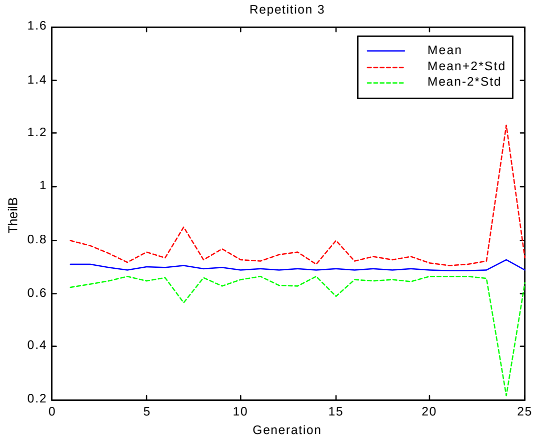
Table II.17: The most frequently visited networks in repetition 3 using TheilB for the S&P data.

TheilB: Repetition 3		
Generation	**Mean**	**STD**
1	0.71064872845507	0.04368259478079
2	0.70737541991906	0.03522081801646
3	0.69865278686966	0.02480095044211
4	0.69011796931026	0.01372494302950
5	0.70082099035223	0.02651996058387
6	0.69718063514431	0.01842019563136
7	0.70634564839249	0.07035365118345
8	0.69291554106168	0.01658292617884
9	0.69596404823704	0.03437592342208
10	0.68962956039022	0.01887190797087
11	0.69133664636112	0.01453505117466
12	0.68949116661525	0.02834634108135
13	0.69074785727804	0.03172875559544
14	0.68692633292179	0.01153190781914
15	0.69373933215179	0.05276327306662
16	0.68795516206104	0.01719646355370
17	0.69282944888119	0.02299106482548
18	0.68898525130176	0.01838667595506
19	0.69102926619919	0.02323978779823
20	0.68796271711173	0.01196846161570
21	0.68547996904250	0.01047762379208
22	0.68630319815234	0.01143863317073
24	0.68790537157307	0.01611159069729
24	0.72469657728196	0.25385680701469
25	0.68739878879858	0.02333671167448

Figure II.18: Mean and Std of TheilB throughout all generations in repetition 3 for the S&P data.







Repetition 1	Minimum: 0.65801387240786 Maximum: 4.90344838578297 Mean: 0.69820374170794 StDev: 0.14180041538561
Repetition 2	Minimum: 0.65632890143622 Maximum: 3.06145783099977 MeanValue: 0.71351487669808 StDev: 0.14408791190612
Repetition 3	Minimum: 0.66140932153077 Maximum: 2.28875675929357 Mean: 0.69449753655453 StDev: 0.05799444472298

Figure II.3: Mean and Std of TheilB throughout all generations for S&P data

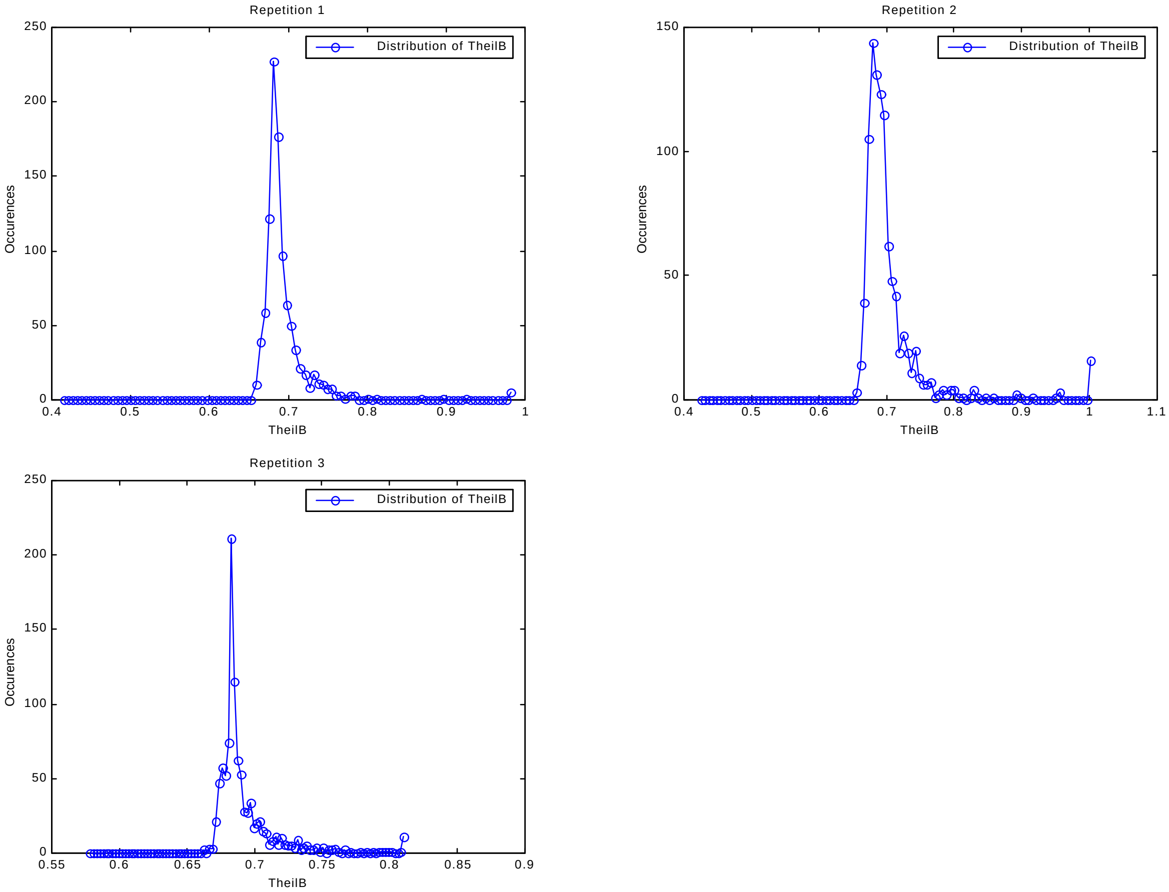


Figure II.4: Distributions of TheilB for the S&P data.

❑ Metric Used: TheilC

Repetition 1

TheilC: Repetition 1			
Generation 1		**Generation 25**	
19-28-8-1	1.01731908253829	7-13-7-1	0.99814272979818
17-18-23-1	1.01776685688234	4-23-7-1	0.99795284802898

11-10-14-1	1.01502385726268	7-17-7-1	0.96951169816745
4-8-20-1	0.98074855226521	1-13-7-1	1.01932865335758
19-11-19-1	1.08280675720682	1-17-7-1	1.00524309930048
2-22-7-1	1.02626878260514	7-6-13-1	1.12505360330236
14-14-28-1	1.09488804140101	7-17-7-1	1.01827353998814
15-17-7-1	1.03597351998646	5-26-12-1	0.96671365987616
11-15-26-1	1.00384213878940	4-13-7-1	0.98577246051998
3-27-21-1	1.16812815644992	1-6-7-1	0.98867882077986
13-23-12-1	1.08195230423459	7-13-2-1	0.97131690636599
3-4-25-1	1.00416872000730	1-23-7-1	1.00349000467989
18-18-29-1	1.09149726435464	7-17-16-1	1.02591512097822
14-20-13-1	3.13276676848730	1-13-7-1	1.00138535559976
14-8-2-1	0.98316310294573	7-13-7-1	0.98945534278777
8-25-1-1	1.05286720293340	4-6-6-1	0.98139032187209
20-15-13-1	1.03434164479650	2-17-20-1	1.01018487848453
11-7-29-1	1.00749175768566	7-13-16-1	0.98032682768603
8-7-7-1	1.02213733635671	4-6-7-1	1.00307421168352
13-16-29-1	1.19205442900254	7-17-12-1	1.03174197995604
12-12-10-1	0.98092529488107	7-13-7-1	1.01829267947295
2-10-25-1	1.03595285766531	5-17-23-1	1.01681861094554
11-29-26-1	3.10042217056751	16-20-30-1	1.05925351899758
4-27-28-1	1.07994669963528	4-6-12-1	0.98633021651383
9-29-11-1	1.07732852408503	7-26-7-1	1.00797969816978
3-17-18-1	1.10015189394248	1-26-7-1	1.00248638659167
4-13-7-1	1.01942974711795	8-17-12-1	1.02013374156683
19-20-13-1	1.06545620186317	7-6-7-1	0.98904183184807
15-19-24-1	1.07761073677863	7-17-7-1	1.00762280906556
10-14-0-1	1.06152326233530	7-26-7-1	0.99966569858334
2-2-14-1	0.99941420819261	7-13-16-1	1.01944832952608
20-16-13-1	1.03309929009005	7-12-12-1	0.98776502091965
7-6-0-1	1.01588814336733	4-6-7-1	0.98748569376481
14-26-20-1	1.01613994122301	7-17-16-1	0.97828066782478
17-6-27-1	1.05727268023110	2-13-7-1	0.98625636228499
13-19-5-1	1.04752365524730	6-6-16-1	0.98800776822600
14-16-25-1	1.11108617247612	4-12-7-1	0.99108154541474
10-23-24-1	1.04127807693848	7-13-7-1	1.00276582399729
17-28-30-1	1.04420917412263	7-6-12-1	0.99061846102576
17-24-30-1	1.02255055556200	7-26-6-1	0.98882463010328
Average	**1.15081038906285**	**Average**	**1.00252778895139**

Table II.19: The results of the first Repetition of the GA search on the S&P data using TheilC.

TheilC: Repetition 1		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
8-5-12-1	11	0
7-6-7-1	12	1
7-13-7-1	12	4
2-26-12-1	14	0
7-26-12-1	14	0
7-6-12-1	14	1
7-26-7-1	15	2
8-6-7-1	17	0
2-13-12-1	22	0
8-26-12-1	28	0

Table II.20: The most frequently visited networks in repetition 1 using TheilC for the S&P data.

TheilC: Repetition 1		
Generation	**Mean**	**STD**
1	1.15081038906285	0.45901322062614
2	1.12002831853205	0.35929179730852
3	1.10158146313436	0.35458298145988
4	1.06415867440215	0.14943776724159
5	1.06689651499668	0.19655074576051
6	1.21304516710555	1.01568105591982
7	1.07719461122750	0.22418445854493

8	1.54080703699033	1.74355247404193
9	1.04093063596092	0.04805864666446
10	1.08313823721226	0.19616635337521
11	1.23924457579548	0.91843058937567
12	1.03847006685350	0.06520006343272
13	1.03241137011405	0.04231942099160
14	1.05392843070739	0.20836896576336
15	1.03016724728643	0.03937635807287
16	1.01905793492849	0.03180702623512
17	1.01909623587441	0.03380121922683
18	1.01611985934268	0.03635511711087
19	1.01567023848110	0.03701060400241
20	1.05784862537154	0.28573401389322
21	1.02544956321815	0.04875543080990
22	1.01004153870473	0.02921630635168
24	1.00830818696947	0.02608147579613
24	1.00839352025788	0.03994011477012
25	1.00252778895139	0.02713932856248

Figure II.21: Mean and Std of TheilC throughout all generations in repetition 1 for the S&P data.

Repetition 2

TheilC: Repetition 2			
Generation 1		**Generation 25**	
20-28-24-1	1.06217767825929	6-4-7-1	0.99287916843563
11-8-30-1	1.07840273362353	1-3-9-1	0.99972448106942
15-3-1-1	1.00116859408204	3-3-7-1	0.99044245849590
19-11-27-1	1.04084219858633	6-4-9-1	1.00935358471270
11-26-5-1	0.97707536324280	7-13-7-1	1.03675487245426
18-12-12-1	1.01488240106260	3-4-1-1	0.99891987615435
15-18-8-1	1.02188789690440	3-3-7-1	1.00120846521469
17-26-18-1	1.04559907543637	3-3-1-1	1.00151655667184
6-15-9-1	1.32508491633297	6-4-7-1	1.00687084376682
7-13-12-1	0.99051557857617	4-6-7-1	0.98802841332583
3-16-19-1	1.02099508041393	7-23-24-1	1.06436554071977
10-18-17-1	1.05545644057526	6-4-9-1	0.98941734088501
8-3-15-1	1.02673075679420	6-25-7-1	0.99595614615815
17-28-9-1	1.03594575784108	3-4-7-1	0.98914422210900
4-15-14-1	0.99440155673454	3-4-7-1	0.99179939427543
10-17-25-1	1.01200127053257	4-3-7-1	0.99021532625432
20-10-9-1	1.04442592500618	4-6-7-1	1.01062395446783
4-13-2-1	0.99749520906810	3-4-7-1	1.00314417686150
11-16-27-1	1.08559011749108	6-3-7-1	0.99256248634962
8-17-20-1	1.03944330932485	6-4-7-1	0.99249342281028
8-22-27-1	1.09205474035813	6-3-9-1	1.00172365012060
11-14-20-1	1.05269078411314	3-4-7-1	1.01171458398735
4-2-20-1	1.00741221618469	7-22-0-1	1.04919494726432
4-18-4-1	0.96319387754033	6-4-7-1	0.98506711192513
17-1-2-1	0.99488813021926	6-4-9-1	0.99890290243690
3-17-5-1	1.02693763933513	4-3-7-1	0.98987830658354
19-22-16-1	1.07558358402188	3-30-7-1	1.00720329192511
6-28-7-1	0.98838218570635	19-3-9-1	1.00581621261333
20-2-13-1	1.01987014678329	6-4-7-1	1.00087621737651
11-4-26-1	1.04531415090429	3-13-7-1	1.02116837497123
3-12-23-1	1.00665249780319	3-20-7-1	0.99362097177874
4-24-17-1	1.00898451294384	6-30-7-1	0.98362768338278
6-11-7-1	0.98867156608099	3-20-2-1	0.98364706210897
13-4-8-1	1.01606200487068	6-25-7-1	0.99984533775869
3-23-21-1	1.03915686612047	6-25-7-1	1.01018154902036
3-3-3-1	0.99231114613215	1-4-7-1	0.99288584333619
12-15-22-1	1.03731672596817	6-3-7-1	1.03338059484848
10-3-1-1	1.00051387427026	6-23-7-1	0.98660915380679
9-13-6-1	1.00400157663589	6-3-7-1	1.08597112360685
16-23-7-1	1.06468823197557	4-4-7-1	0.97947879120799

Average	**1.03237020794640**	**Average**	**1.00415536103131**

Table II.22: The results of the second Repetition of the GA search on the S&P data using TheilC.

TheilC: Repetition 2		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
6-4-9-1	18	3
3-6-7-1	18	0
3-4-2-1	21	0
3-3-9-1	26	0
3-13-7-1	27	1
4-4-7-1	30	1
4-3-7-1	31	2
6-4-7-1	40	5
3-3-7-1	58	2
3-4-7-1	58	4

Table II.23: The most frequently visited networks in repetition 2 using TheilC for the S&P data.

TheilC: Repetition 2		
Generation	**Mean**	**STD**
1	1.03237020794640	0.05657065455373
2	1.01614393921500	0.05187475295012
3	1.01016059635000	0.02110125180680
4	1.00234996615996	0.02053948109474
5	1.00576176779611	0.02600444267451
6	1.01013091400801	0.04207053163655
7	1.01586261065026	0.06169990369676
8	1.07658039065531	0.36596671384823
9	1.00655751865237	0.02029176469790
10	1.02917967836887	0.16803095434958
11	1.00811622684539	0.03043250622790
12	1.00636352590795	0.03062219384388
13	1.00948269229709	0.03578304986517
14	1.01820689400517	0.09078202248591
15	0.99911235319766	0.01289625696924
16	1.00479797986096	0.02979133084417
17	1.01302286278111	0.04790720001658
18	1.00929317918300	0.03697120892311
19	1.00128777454089	0.02500862287800
20	0.99896067622350	0.01504147830072
21	1.01729156612463	0.04033411067374
22	1.00782920354181	0.03137631722520
24	1.00454588783890	0.02560848884762
24	1.00378830019127	0.02534215590043
25	1.00415536103131	0.02214729569558

Figure II.24: Mean and Std of TheilC throughout all generations in repetition 2 for the S&P data.

Repetition 3

TheilC: Repetition 3			
Generation 1		**Generation 25**	
5-25-2-1	1.00364292643304	7-4-5-1	0.98114496499651
14-17-8-1	1.06096301088784	3-1-3-1	0.99894515849469
7-9-22-1	1.04450569868948	6-3-30-1	0.99971351688454
15-16-30-1	1.02618259488852	6-21-14-1	1.08688157352096
17-1-5-1	1.02460030708669	6-3-3-1	1.01061142531444
1-7-11-1	0.99653559496828	3-3-30-1	0.99659195017947
3-4-13-1	1.02672449459175	6-8-3-1	1.01841676854069
16-6-5-1	0.98181194691247	6-3-14-1	1.01103894411632
4-22-28-1	1.01550858528331	6-1-3-1	0.99916452123975
20-7-12-1	1.01439927241823	6-21-3-1	0.97838187415264
7-5-27-1	1.02671556549764	6-21-5-1	1.00076067277786
8-16-4-1	0.98129219363814	6-4-5-1	0.97439610927693

9-8-24-1	1.07222901641924	6-4-26-1	1.02415676595732
8-6-2-1	1.00548169139534	1-16-3-1	0.99901056892274
3-1-26-1	1.01566513712496	6-3-3-1	1.00001860585609
2-21-21-1	1.00784445017699	6-1-3-1	0.99762200677821
6-20-18-1	0.99085745561966	7-20-5-1	1.01315330370910
4-15-14-1	1.01188007782690	6-21-3-1	0.99807085140847
9-17-10-1	1.11684914807256	6-16-3-1	1.01523918552402
18-20-25-1	1.09653392002525	6-3-3-1	0.98612093987832
5-3-24-1	1.00083572070873	3-4-3-1	0.99150263526705
13-4-9-1	1.01642343643323	6-3-3-1	1.00085238422724
18-23-6-1	1.02958421207990	3-21-3-1	1.00407264172778
16-21-4-1	0.98747684503884	6-4-3-1	0.98998756167666
3-14-21-1	1.06315507751588	3-1-5-1	1.00212268740161
6-3-21-1	1.05075757403626	3-3-3-1	0.98659139066831
20-12-18-1	1.11970584109305	6-3-3-1	1.01256111677542
3-30-19-1	1.02666713275115	6-4-5-1	0.98460463618307
10-29-17-1	1.01952342285834	3-16-3-1	1.00338100462916
5-16-0-1	0.99774111336865	6-1-3-1	0.98627954186294
14-6-13-1	1.01228527380682	6-4-3-1	0.99664628309507
4-13-15-1	1.00369903162600	3-3-5-1	0.98253993465572
16-1-26-1	1.16371359376290	3-4-3-1	1.00640097103321
2-20-12-1	0.99484591705222	6-3-3-1	0.99843473001173
10-3-18-1	1.00658268289922	6-21-14-1	1.01700309776023
6-30-18-1	0.98327191992003	3-3-30-1	1.01569283653616
20-6-30-1	1.06407669401275	3-1-5-1	0.99618001389025
14-3-13-1	1.01225474508108	6-1-3-1	0.99913481179003
5-3-25-1	1.05235391039384	6-8-3-1	0.98188380509502
9-26-27-1	1.15435786290132	6-8-3-1	0.99514595528172
Average	**1.03198837738241**	**Average**	**1.00101144367744**

Table II.25: The results of the third Repetition of the GA search on the S&P data using TheilC.

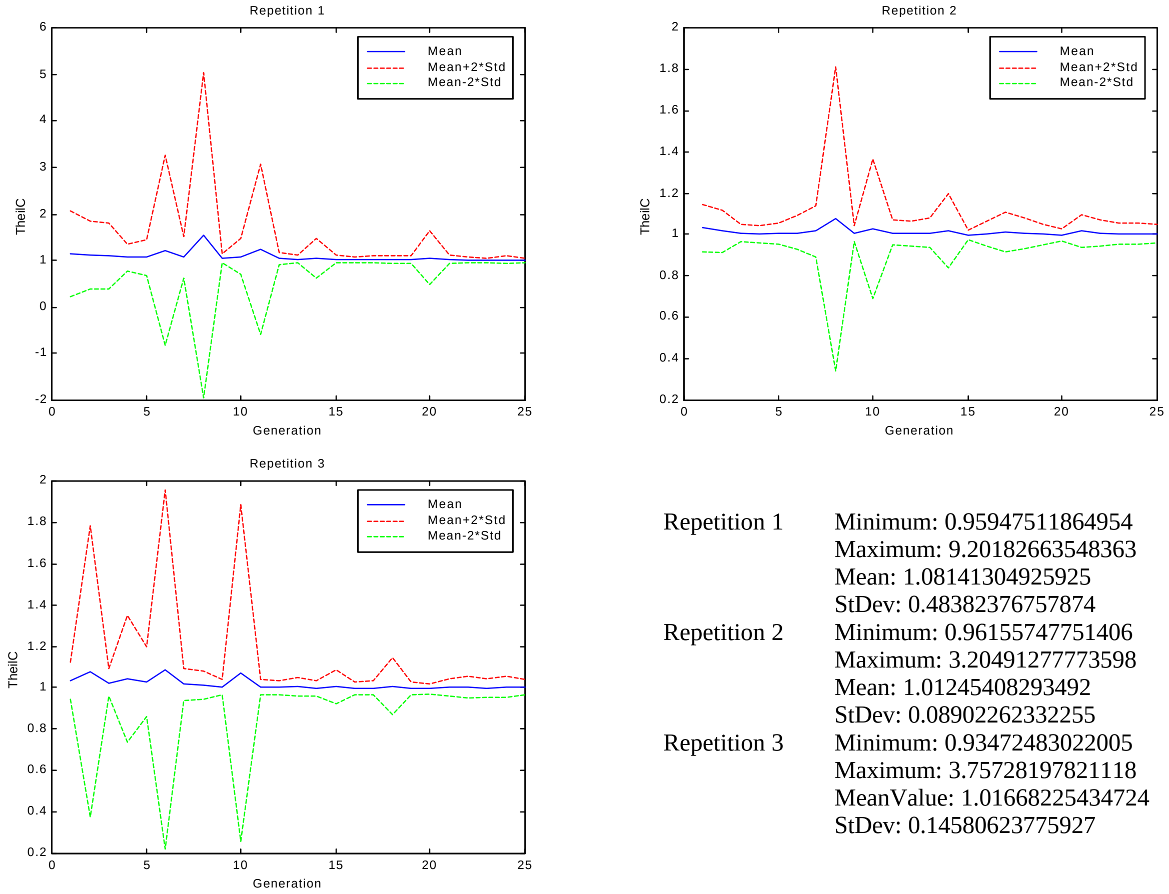
TheilC: Repetition 3		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
3-21-3-1	19	1
3-3-3-1	20	1
6-4-3-1	21	2
4-3-5-1	22	0
6-21-5-1	27	1
6-21-3-1	30	2
6-4-5-1	33	2
3-3-5-1	36	1
6-3-3-1	47	6
6-3-5-1	52	0

Table II.26: The most frequently visited networks in repetition 3 using TheilC for the S&P data.

TheilC: Repetition 3		
Generation	**Mean**	**STD**
1	1.03198837738241	0.04495719251051
2	1.07751992993298	0.35268246169815
3	1.02437165612007	0.03287815572548
4	1.04425968798508	0.15380363433146
5	1.03111307646401	0.08435654218854
6	1.08879344877237	0.43462512737003
7	1.01591339865194	0.03746149697033
8	1.01306184445016	0.03450195182923
9	1.00193863905894	0.01824666591080
10	1.07296894247444	0.40649476561851
11	1.00203258202310	0.01804776630134
12	1.00060072605838	0.01761156307757
13	1.00565978157107	0.02302560410018
14	0.99716326134888	0.01792317213655
15	1.00565582967919	0.04138664009639
16	0.99631603456002	0.01587538998262
17	0.99941007619267	0.01675582951651
18	1.00717317480486	0.06796700817528

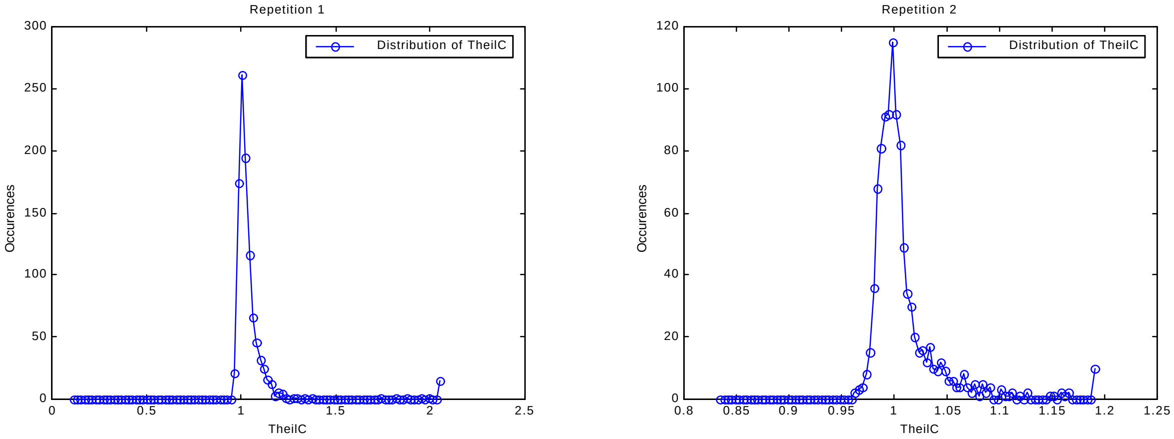
19	0.99831309347034	0.01630745123277
20	0.99459017452436	0.01266585121335
21	1.00153010270498	0.02134512615499
22	1.00427504249839	0.02651149665148
24	0.99870675193108	0.02169844925055
24	1.00268928234377	0.02492875587567
25	1.00101144367744	0.01825456736131

Figure II.27: Mean and Std of TheilC throughout all generations in repetition 3 for the S&P data.



Repetition 1 Minimum: 0.95947511864954
Maximum: 9.20182663548363
Mean: 1.08141304925925
StDev: 0.48382376757874
Repetition 2 Minimum: 0.96155747751406
Maximum: 3.20491277773598
Mean: 1.01245408293492
StDev: 0.08902262332255
Repetition 3 Minimum: 0.93472483022005
Maximum: 3.75728197821118
MeanValue: 1.01668225434724
StDev: 0.14580623775927

Figure II.5: Mean and Std of TheilC throughout all generations for S&P data



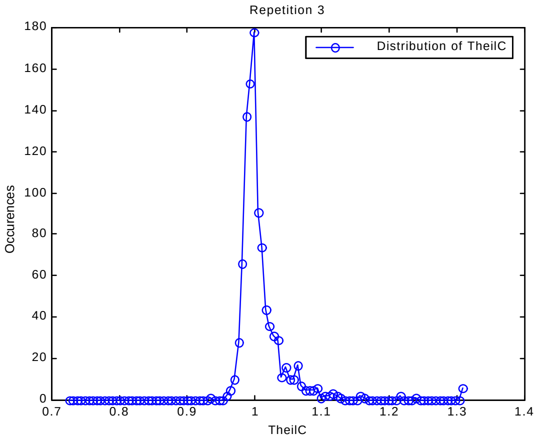


Figure II.6: Distributions of TheilC for the S&P data.

❑ Metric Used: MAE

Repetition 1

MAE: Repetition 1			
Generation 1		Generation 25	
8-26-17-1	0.01082148677787	13-2-4-1	0.01018086944186
1-11-19-1	0.01027539359744	5-29-4-1	0.01025466382724
18-23-9-1	0.01033594635379	6-9-4-1	0.01003349123162
9-29-29-1	0.01165545754764	6-22-4-1	0.01019623187658
15-28-7-1	0.01028731024147	6-9-4-1	0.01024907871274
13-20-3-1	0.01012573956822	13-2-4-1	0.01022911795834
3-6-29-1	0.01018434110585	4-3-24-1	0.01029937435314
10-13-29-1	0.01078501379046	6-26-4-1	0.01008630989484
12-7-24-1	0.01055129654693	6-9-4-1	0.01005359315849
3-25-15-1	0.01059555267155	6-2-4-1	0.01007997324157
1-24-11-1	0.01007180588011	13-2-4-1	0.01001765771189
20-29-21-1	0.01153937589747	6-26-4-1	0.01007285661144
6-15-22-1	0.01078922786428	6-22-4-1	0.01009894728859
8-23-15-1	0.01055309878220	6-26-4-1	0.01012038931752
6-15-12-1	0.01006163191842	6-26-4-1	0.01014469377460
11-23-19-1	0.01021781356864	6-24-4-1	0.01014618721788
17-25-20-1	0.01011060161736	6-26-8-1	0.01052742594827
10-16-27-1	0.01032377200493	6-26-4-1	0.00997822770917
7-24-18-1	0.01032789211906	13-2-4-1	0.01036412317804
11-23-4-1	0.01031558993825	6-9-4-1	0.01012919239167
4-4-9-1	0.01010062997034	6-26-4-1	0.01027004895053
12-8-25-1	0.01056444295301	6-24-4-1	0.01016168620318
3-29-30-1	0.01056723481496	6-24-4-1	0.01009454980765
4-15-17-1	0.01042068414873	6-22-4-1	0.01022978828066
9-24-22-1	0.01006241360333	6-26-4-1	0.01015008808923
19-9-4-1	0.01053170707434	20-16-4-1	0.01009073411875
20-30-24-1	0.01082652296372	6-26-4-1	0.00983047478051
18-16-19-1	0.01043071759173	6-26-4-1	0.01012645241927
13-16-26-1	0.01080469249030	6-9-4-1	0.00998556703755
11-3-24-1	0.01014775738173	6-9-4-1	0.00989313802305
16-30-16-1	0.01068841726874	6-26-4-1	0.01000915070926
11-7-7-1	0.01009846945984	6-24-4-1	0.00993013250097
12-15-16-1	0.01068760616444	6-9-4-1	0.01014556451736
8-9-4-1	0.01028721659842	6-2-4-1	0.01012801070643
14-8-19-1	0.01032018565074	6-26-4-1	0.01005305496014
1-9-9-1	0.01024687463883	6-9-4-1	0.01003803795492
17-24-28-1	0.01063022567389	6-26-4-1	0.00993344934517
18-24-17-1	0.01086925606620	6-22-4-1	0.01002516348673
1-7-21-1	0.01005463584433	6-22-4-1	0.01014284329963

5-23-9-1	0.01013410054665	6-9-4-1	0.01017566114932
Average	**0.01046005346741**	**Average**	**0.01011690002965**

Table II.28: The results of the first Repetition of the GA search on the S&P data using MAE.

MAE: Repetition 1		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
6-20-4-1	14	0
6-19-4-1	16	0
6-9-8-1	17	0
6-9-11-1	20	0
12-9-4-1	21	0
6-9-12-1	23	0
6-26-4-1	42	12
6-22-4-1	42	5
6-24-4-1	53	4
6-9-4-1	208	9

Table II.29: The most frequently visited networks in repetition 1 using MAE for the S&P data.

MAE: Repetition 1		
Generation	**Mean**	**STD**
1	0.01046005346741	0.00036698955031
2	0.01041190346023	0.00030505419144
3	0.01049539122420	0.00134648160959
4	0.01031635703317	0.00026671326211
5	0.01031677703834	0.00024966264000
6	0.01042261441872	0.00034434546370
7	0.01017621732596	0.00020223572476
8	0.01022179688198	0.00023692735144
9	0.01026817729007	0.00033169403989
10	0.01032202531960	0.00035393722123
11	0.01031907408842	0.00053265437380
12	0.01026289618738	0.00025520156019
13	0.01022619247343	0.00020450279448
14	0.01016382852648	0.00015119078171
15	0.01016530153170	0.00019833516641
16	0.01020409839946	0.00019801293950
17	0.01021584902694	0.00030912126295
18	0.01017021305833	0.00015550118608
19	0.01015047614309	0.00024435996339
20	0.01013550789649	0.00010608272551
21	0.01019151826061	0.00025377555163
22	0.01009294169913	0.00012021678220
24	0.01020137473906	0.00015279738737
24	0.01021692351367	0.00027348351891
25	0.01011690002965	0.00013034879532

Figure II.30: Mean and Std of MAE throughout all generations in repetition 1 for the S&P data.

Repetition 2

MAE: Repetition 2			
Generation 1		**Generation 25**	
12-8-14-1	0.01061952071906	1-7-2-1	0.01001007873398
15-23-27-1	0.01070182089781	1-19-4-1	0.01016153427295
3-30-9-1	0.01018003542960	1-22-2-1	0.01007575427999
19-7-18-1	0.01091125845474	1-7-2-1	0.00997335395368
14-8-4-1	0.01016895350561	1-7-1-1	0.01032138908545
19-4-14-1	0.01007609845537	6-22-2-1	0.01041825555326
17-29-13-1	0.01018153960122	1-7-2-1	0.01012233266405
4-10-2-1	0.01005558740288	1-22-2-1	0.01006281494254
17-29-19-1	0.01103265595856	1-7-2-1	0.01008702396355
3-16-4-1	0.01016699281026	1-12-1-1	0.01009687975505
17-15-23-1	0.01058824909676	1-7-2-1	0.01011527398696

15-11-15-1	0.01041236513722	1-22-1-1	0.01036188784720
8-24-9-1	0.01000909661910	1-22-2-1	0.01015508574221
18-17-0-1	0.01043836969710	1-22-2-1	0.01031810615012
5-7-13-1	0.01034750019893	1-22-2-1	0.01014997656796
6-27-13-1	0.01063510801161	1-7-1-1	0.01006925605898
4-15-6-1	0.01000767592185	1-22-1-1	0.01049884609528
12-25-1-1	0.01004597534891	1-22-2-1	0.01004195101056
2-15-5-1	0.01018518725356	1-22-2-1	0.01058703761546
16-11-29-1	0.01080663667206	6-22-2-1	0.01106494312374
5-1-8-1	0.01008865070814	1-7-1-1	0.01021334291007
6-4-21-1	0.01010419628811	1-12-1-1	0.01009051642187
20-17-6-1	0.01031458460684	1-7-2-1	0.01000892312319
8-3-13-1	0.01030025052740	1-22-2-1	0.01011749595975
20-14-24-1	0.01039665229254	20-23-23-1	0.01169731957641
7-18-19-1	0.01053120070625	1-7-2-1	0.01014012565679
19-22-1-1	0.01008613558996	1-22-2-1	0.01026993188154
5-24-14-1	0.01038903448452	1-7-2-1	0.01019903030010
6-2-4-1	0.00988940190652	1-22-2-1	0.01009495415453
3-30-5-1	0.00997760291548	1-22-2-1	0.01010237876559
7-20-10-1	0.01018997308305	1-22-2-1	0.01014192779225
11-7-7-1	0.01010683875638	1-22-1-1	0.01008312179469
20-2-25-1	0.01020866746597	1-22-1-1	0.01009249414311
6-8-9-1	0.00996034458774	1-22-2-1	0.01009430857259
3-2-21-1	0.01023964515497	1-22-5-1	0.01013520692974
19-27-1-1	0.01032801408326	1-12-1-1	0.01009884697809
15-11-30-1	0.01100302364945	1-22-2-1	0.01009603943686
1-2-2-1	0.01009507039063	1-7-1-1	0.01007983654398
2-12-12-1	0.01022521406019	1-22-2-1	0.01019792435930
5-23-22-1	0.01053174335775	1-22-2-1	0.01012997654455
Average	**0.01031342179518**	**Average**	**0.01021938708120**

Table II.31: The results of the second Repetition of the GA search on the S&P data using MAE.

MAE: Repetition 2		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
6-2-2-1	17	0
6-2-1-1	17	0
6-22-1-1	18	0
1-2-1-1	24	0
1-12-2-1	25	0
1-2-2-1	27	0
6-22-2-1	54	2
1-7-2-1	72	8
1-22-1-1	73	4
1-22-2-1	160	16

Table II.32: The most frequently visited networks in repetition 2 using MAE for the S&P data.

MAE: Repetition 2		
Generation	**Mean**	**STD**
1	0.01031342179518	0.00029172827038
2	0.01024527968772	0.00036620515007
3	0.01020781589501	0.00016871155123
4	0.01020573101983	0.00016073062138
5	0.01025570233463	0.00056421710020
6	0.01015273221242	0.00014582434786
7	0.01079352026520	0.00407706246931
8	0.01018119032119	0.00022216388591
9	0.01028804893218	0.00061772106307
10	0.01016684773465	0.00029762463784
11	0.01018803712457	0.00018124086054
12	0.01018765534536	0.00032988769229
13	0.01035210480876	0.00078162886624
14	0.01016199086112	0.00014074525602
15	0.01015144118232	0.00015363159368
16	0.01013365722448	0.00008554035076
17	0.01012364548914	0.00010122513119

18	0.01012509235873	0.00012100932051
19	0.01018424939633	0.00027135499847
20	0.01013650911176	0.00019928435170
21	0.01014494657141	0.00025238223659
22	0.01016864734467	0.00027638722176
24	0.01012568176948	0.00007574897826
24	0.01017158062873	0.00023732166063
25	0.01021938708120	0.00030767688317

Figure II.33: Mean and Std of MAE throughout all generations in repetition 2 for the S&P data.

Repetition 3

MAE: Repetition 3			
Generation 1		**Generation 25**	
3-19-10-1	0.01033437364966	3-2-9-1	0.00999551029143
17-24-21-1	0.01097296265831	7-11-3-1	0.01002251215414
4-30-13-1	0.01057143472116	3-2-3-1	0.01010151158032
4-20-12-1	0.01014783165871	3-2-3-1	0.01003620417210
13-9-3-1	0.01046426210220	3-19-3-1	0.00998681296993
16-25-9-1	0.01053713480109	3-2-9-1	0.01016262772498
14-7-2-1	0.01041129695636	3-19-3-1	0.01041665406289
7-2-16-1	0.01009411471769	3-11-9-1	0.01028798253078
16-22-15-1	0.01105293191777	3-2-3-1	0.01005607533735
16-2-5-1	0.00993659092232	3-19-3-1	0.01014618181002
3-12-18-1	0.01120065755211	3-11-9-1	0.00999033563591
9-4-14-1	0.01026868829576	3-19-3-1	0.01022339117734
20-21-1-1	0.01033003592977	7-30-29-1	0.01365646745190
19-26-3-1	0.01014067841733	3-12-1-1	0.01002470106983
18-9-1-1	0.01004172597260	3-2-3-1	0.01015132545909
16-28-9-1	0.01078608192637	3-19-3-1	0.01010732212111
4-23-8-1	0.01034585428016	3-2-3-1	0.01000608412603
15-20-17-1	0.01048222851366	3-2-9-1	0.01010126382177
14-9-30-1	0.01000035981498	3-2-3-1	0.01007543952001
7-29-26-1	0.01007022402815	3-19-3-1	0.01001121665420
11-27-20-1	0.01240441643928	3-19-1-1	0.01013510260381
14-4-25-1	0.01044589314371	3-12-9-1	0.01003080381138
9-21-24-1	0.01103799752853	3-19-1-1	0.01011542123362
16-6-20-1	0.01011813814072	14-25-0-1	0.01076179890738
9-28-17-1	0.01041877219201	7-2-9-1	0.01018721013366
13-9-4-1	0.01017024796633	7-2-1-1	0.00990344340126
13-25-7-1	0.01021292897667	20-15-10-1	0.01052204584358
17-23-28-1	0.01093644621452	3-13-6-1	0.01035854889887
3-8-0-1	0.01005296344796	7-2-9-1	0.01038938110287
18-29-6-1	0.01012913180668	3-11-9-1	0.01017535459143
20-25-16-1	0.01049766670799	19-21-23-1	0.01075983642058
17-3-1-1	0.01010518397569	3-12-28-1	0.01048288833839
9-29-9-1	0.01008641357279	3-11-1-1	0.01005875206094
6-17-11-1	0.01044070559333	3-19-3-1	0.01022942228014
17-14-9-1	0.01044075247411	3-19-3-1	0.01035745124680
2-13-26-1	0.01024966640167	7-19-3-1	0.01020242733745
7-4-14-1	0.00987221179891	3-12-1-1	0.01030189934430
8-29-13-1	0.01018161273335	3-19-3-1	0.01021818212223
13-20-14-1	0.01131340579108	3-2-9-1	0.01009597204686
11-26-6-1	0.01008144096557	3-2-3-1	0.00992719517675
Average	**0.01043463661768**	**Average**	**0.01026931891434**

Table II.34: The results of the third Repetition of the GA search on the S&P data using MAE.

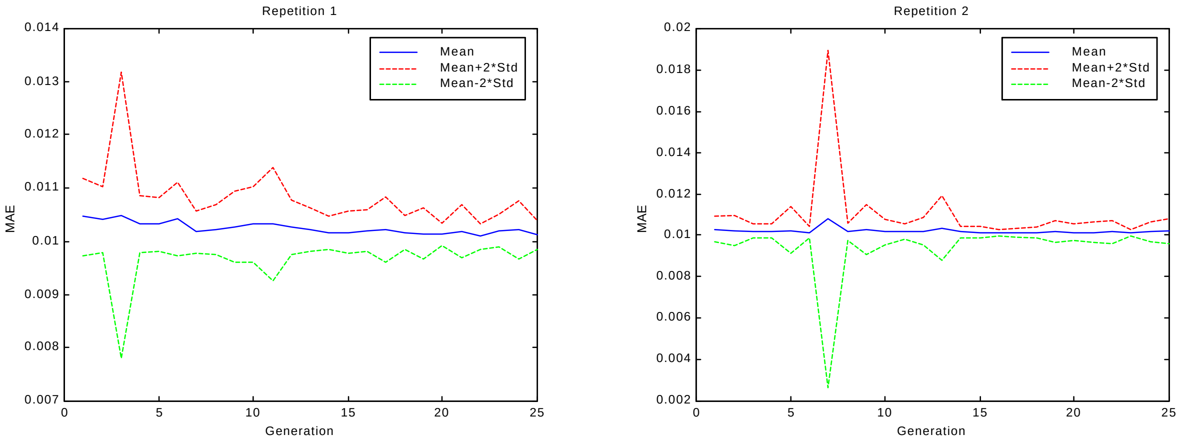
MAE: Repetition 3		
Top 10		
Structure	**Times Considered**	**Times in Final Generation**
3-19-1-1	19	2
7-2-9-1	20	2
3-12-1-1	21	2
3-12-9-1	22	1

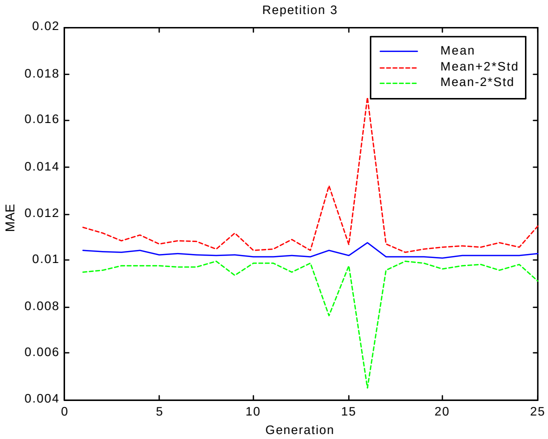
3-19-3-1	26	9
7-2-1-1	28	1
3-2-3-1	28	7
3-14-1-1	32	0
3-2-1-1	41	0
3-2-9-1	44	4

Table II.35: The most frequently visited networks in repetition 3 using MAE for the S&P data.

MAE: Repetition 3		
Generation	**Mean**	**STD**
1	0.01043463661768	0.00047985147492
2	0.01038563099491	0.00039042746740
3	0.01032175710452	0.00027100571948
4	0.01042954307744	0.00032540767903
5	0.01024617133289	0.00023771320203
6	0.01029355863461	0.00028277333313
7	0.01024641588921	0.00027126191413
8	0.01021487279662	0.00013085158123
9	0.01026138243293	0.00045085330090
10	0.01015267275046	0.00013716872743
11	0.01016650215329	0.00016068620961
12	0.01018522395239	0.00034844036877
13	0.01013844588551	0.00014068857570
14	0.01041987769172	0.00139422038167
15	0.01020193203535	0.00022507837164
16	0.01073813192864	0.00311941223590
17	0.01015601045978	0.00028542162492
18	0.01013659100774	0.00009296598643
19	0.01016204433271	0.00015622259278
20	0.01011637045638	0.00023768049959
21	0.01020663859573	0.00020876138275
22	0.01019087833275	0.00019067470163
24	0.01017288307835	0.00030220310993
24	0.01019897790019	0.00018711566351
25	0.01026931891434	0.00058438451446

Figure II.36: Mean and Std of MAE throughout all generations in repetition 3 for the S&P data.





Repetition 1	Minimum: 0.00980390701485 Maximum: 0.01865772289184 Mean: 0.01024977636134 StDev: 3.8476067494e-004
Repetition 2	Minimum: 0.00988940190652 Maximum: 0.03591933673879 Mean: 0.01021539665984 StDev: 8.711272711571e-004
Repetition 3	Minimum: 0.00984895417788 Maximum: 0.02950493566814 MeanValue: 0.01025785873425 StDev: 7.416538535514e-004

Figure II.7: Mean and Std of mae throughout all generations for S&P data

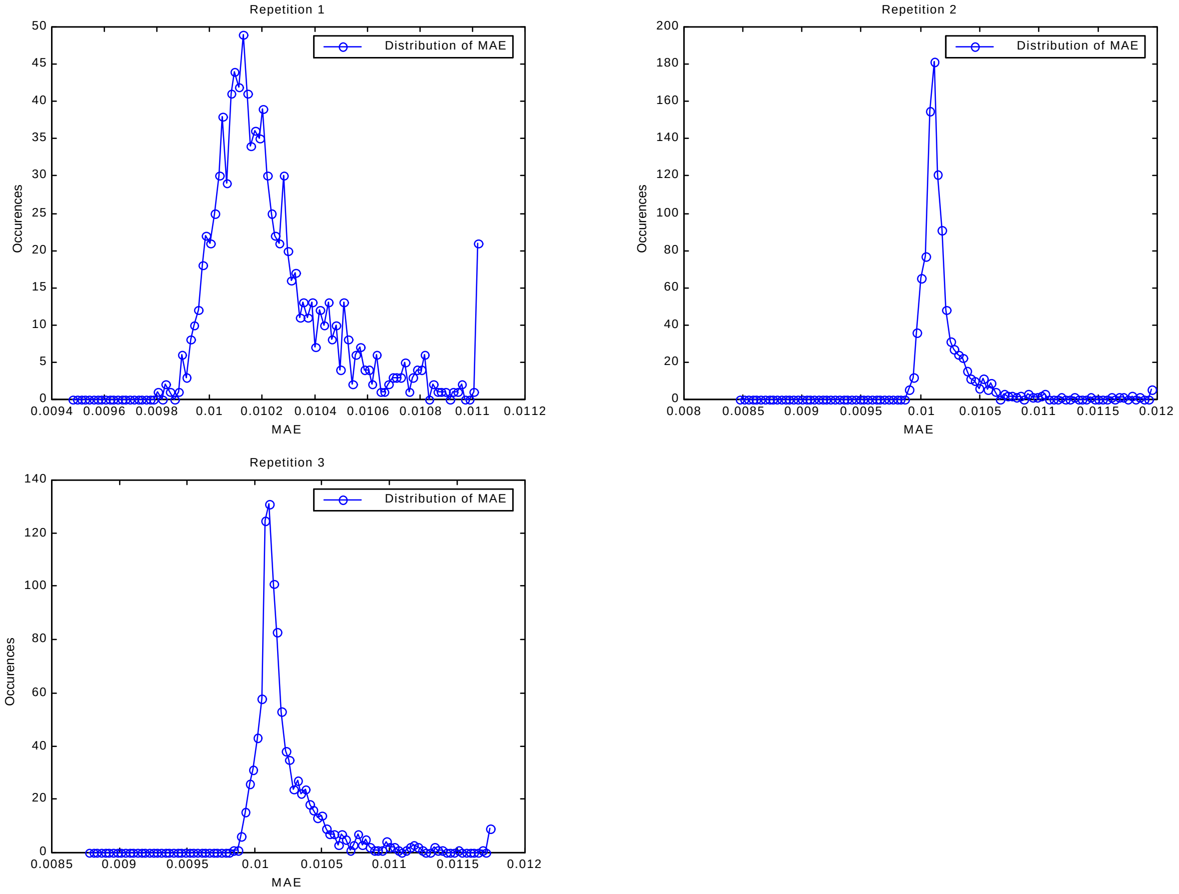


Figure II.8: Distributions of mae for the S&P data.

References

[1] Malkei B. G. (1999, 7th ed.). *A random walk down wall street.* New York, London: W. W. Norton & Company.

[2] Helstrom T. & Holmstrom K. (1998). Predicting the stock market. Published as Opuscula ISRN HEV-BIB-OP-26-SE.

[3] Hsieh A. D. (1991). Chaos and non-linear dynamics: Application to financial Markerts. *Journal of Finance*, Vol.46, pp 1833-1877.

[4] Tsibouris G. & Zeidenberg M. (1996). Testing the efficient market hypothesis with gradient descent algorithms. In Refenes, A. P. *Neural networks in the capital markets*. England: John Wiley & Sons Ltd, pp 127-136.

[5] White H. (1993). Economic prediction using neural networks: The case of IBM daily stock returns. In Trippi R. R. & Turban E. *Neural networks in finance and investing*. Chicago, Illinois, Cambridge, England: Probus Publishing Company, pp 315-329.

[6] Maddala G.S. (1992). *Introduction to econometrics.* New York, Toronto: Macmillan Publishing Company.

[7] Pesaran H. M. & Timmermann A. (1994). Forecasting stock returns: An examination of stock market trading in the presence of transaction costs. *Journal of Forecasting*, Vol. 13, pp 335-367.

[8] Azoff E. M. (1994). *Neural network time series forecasting of financial markets.* Chichester: John Wiley and Sons.

[9] Mitchell M.T. (1997). *Machine learning*. New York: The McGraw-Hill Companies.

[10] Demuth H. & Beale M. (1997). *Neural network toolbox: for use with matlab*, 4th edition, 3rd version. U.S.: The MathWorks Inc. (Online: http://www.mathworks .com/access/helpdesk/help/toolbox/nnet/nnet.shtml)

[11] Medsker L., Turban E. & Trippi R.R. (1993). Neural network fundamentals for financial analysts. In Trippi R. R. & Turban E. *Neural networks in finance and investing*. Chicago: Probus Publishing Company, pp 3-27.

[12] Steiner M. & Wittkemper H. G. (1996). Neural networks as an alternative stock market model. In Refenes, A. P. *Neural networks in the capital markets*. England: John Wiley & Sons, pp 137-149.

[13] Chenoweth T. & Obradovic (1996). A multi-component nonlinear prediction system for the S&P 500 index. *Neurocomputing*. Vol. 10, Issue 3, pp 275-290.

[14] DataStream web site http://www.primark.com/pfid/index.shtml?/content/ datastream.shtml

[15] Han J. & M. Kamber (2001). *Data mining: concepts and techniques*. San Francisco: Academic Press.

[16] Lindgren B. W. (1976). *Statistical theory*. 3rd edition. N.Y., London: Macmillan

[17] Bennet J. D. (1998). *Randomness* 2nd edition. U.S.A.: President and Fellows of Harvard College.

[18] Knuth E. D. (1981). *The art of computer programming*. Vol. 2. 2nd edition. U.S.A. Addison-Wesley.

[19] Brock W.A., Dechert W.D. & Scheinkman, J.A. (1987), “A test for independence based on the correlation dimension,” *University of Wisconsin, Department of Economics*. (Revised version in Brock W. A., Dechert W. D., Scheinkman J. A.,

and LeBaron B. D. (1996), *Econometric Reviews*, 15, 197-235.)

[20] Kosfeld R., Rode S. (1999). Testing for nonlinearities in German bank stock returns. Paper presented at the *Whitsun conference of the German statistical society, section empirical economic research and applied econometrics*, May 26-28, Heidelberg.

[21] Barnes L. M. & De Lima J.F.P. (1999). Modeling financial volatility: extreme observations, nonlinearities and nonstationarities. *Working Paper: University of Adelaide Australia.*

[22] Barkoulas J.T., Baum C. F. & and Onochie, J. (1997) A nonparametric investigation review of the 90-Day T-Bill rate. *Review of Financial Economics*, Vol. 6, Issue 2, pp 187-198.

[23] Afonso A. & Teixeira, J. (1999) Non-linear tests for weekly efficient markets: evidence from Portugal. *Estudos de Economia,* Vol. 2, pp 169-187.

[24] Johnson D. & McClelland R. (1998). A general dependence test and applications. *Journal of Applied Econometrics*, Vol. 13, Issue 6, pp 627-644.

[25] Koèenda E. (1998). *Exchange rate in transition.* Praha: CERGE UK, Charles University.

[26] Robinson D. M. (1998). Non-Linear dependence asymmetry and thresholds in the volatility of Australian futures markets, Research Paper Series, Department of Accounting and Finance, University of Melbourne.

[27] Kyrtsou C., Labys, W. and Terraza, M. (2001). Heterogeneity and chaotic dynamics in commodity markets. Research Paper, West Virginia University, No. 7.

[28] Kanzler L. (1998). BDS Matlab code, Revision 2.41. Department of Economics, University of Oxford. (Online: http://users.ox.ac.uk/~econlrk)

[29] Brockwell J. P. & Davis A. R. (1996) *Introduction to time series forecasting.* New York, Berlin, Heidelberg: Springer-Verlang.

[30] Holden K., Peel A.D. & Thomson L.J. (1990). *Economic forecasting: An introduction.* Cambridge: Cambridge University Press.

[31] Theil H. (1966). *Applied economic forecasting*. Amsterdam: North-Holland Publishing Company.

[32] Microfit 4.0 Website. http://www.intecc.co.uk/camfit/

[33] Gurney K. (1997). *An introduction to neural networks*, London: UCL Press.

[34] Bishop M. C. (1996). *Neural networks for pattern recognition*. New York: Oxford University Press.

[35] Goldberg E. D. (1989). *Genetic algorithm in search, optimization, and machine learning*. New York: Addison-Wesley.

[36] Mitchell M. (1997). *An introduction to genetic algorithms.* Cambridge Massachusetts: MIT Press.

[37] Koza R. J. (1992). *Genetic programming, on the programming of computers by means of natural selection.* Cambridge Massachusetts: MIT Press.

[38] Man F. K., Tang S. K. & Kwong S. (1999). *Genetic algorithms: Concepts and designs*. Heidelberg: Springer-Verlang.

[39] The Mathworks web site. http://www.MathWorks.com