FINANCIAL PREDICTIONS USING INTELLIGENT SYSTEMS

The application of advanced technologies for trading financial markets

A thesis submitted for the degree of
Doctor of Philosophy

by

Vlade Milanović

School of Engineering and Design
Department of Electronics and Computer Engineering
Brunel University

September 2007

ABSTRACT

This thesis presents a collection of practical techniques for analysing various market properties in order to design advanced self-evolving trading systems based on neural networks combined with a genetic algorithm optimisation approach. Nonlinear multivariate statistical models have gained increasing importance in financial time series analysis, as it is very hard to find statistically significant market inefficiencies using standard linear modes. Nonlinear models capture more of the underlying dynamics of these high dimensional noisy systems than traditional models, whilst at the same time making fewer restrictive assumptions about them. These adaptive trading systems can extract information about associated time varying processes that may not be readily captured by traditional models. In order to characterise the financial time series in terms of its dynamic nature, this research employs various methods such as fractal analysis, chaos theory and dynamical recurrence analysis. These techniques are used for evaluating whether markets are stochastic and deterministic or nonlinear and chaotic, and to discover regularities that are completely hidden in these time series and not detectable using conventional analysis. Particular emphasis is placed on examining the feasibility of prediction in financial time series and the analysis of extreme market events. The market's fractal structure and log-periodic oscillations, typical of periods before extreme events occur, are revealed through recurrence plots. Recurrence qualification analysis indicated a strong presence of structure, recurrence and determinism in the financial time series studied. Crucial financial time series transition periods were also detected.

This research performs several tests on a large number of US and European stocks using methodologies inspired by both fundamental analysis and technical trading rules. Results from the tests show that profitable trading models utilising advanced nonlinear trading systems can be created after accounting for realistic transaction costs. The return achieved by applying the trading model to a portfolio of real price series differs significantly from that achieved by applying it to a randomly generated price series. In some cases, these models are compared against simpler alternative approaches to ensure that there is an added value in the use of these more complex models. The superior performance of multivariate nonlinear models is also demonstrated. The long-short trading strategies performed well in both bull and bear markets, as well as in a sideways market, showing a great degree of flexibility and adjustability to changing market conditions.

Empirical evidence shows that information is not instantly incorporated into market prices and supports the claim that the financial time series studied, for the periods analysed, are not entirely random. This research clearly shows that equity markets are partially inefficient and do not behave along lines dictated by the efficient market hypothesis.

ACKNOWLEDGEMENTS

I would like to sincerely thank my supervisor, Professor John Stonham for his guidance, valuable advice and generous support throughout this research. His insight and assistance have been very valuable.

I would also like to thank the following friends, from both academia and finance, for their support, numerous discussions and help in providing data and testing: Ivana Raonić and Nataša Todorović (City University), Drago Indjić (London Business School), Mirjana Andrić (Finsoft), Richard Saldanha (Oxford Centre for Innovation) and Dejan Jeličić (Sabre Fund Management). Special thanks to Saïd Kaba (Cheyne Capital Management) for his valuable comments and feedback.

Above all, I would like to thank my family for their loving support and innumerable sacrifices and patience over the years.

Table of contents

Definitions and abbreviations used

$\Theta(\cdot)$	Heaviside Function
AANN	Auto-Associative Neural Network
ANN	Artificial Neural Networks
APT	Arbitrage Pricing Theory
AT	Asset Turnover
BM	Book to Market
CAPM	Capital Asset Pricing Model
CPI	Consumer Price Index
Cm	Correlation integral
CR	Current Ratio
D	Fractal (Correlation) Dimension
DET	Determinism (RQA measure)
DJI	Dow Jones 30 Industrial Index
*DJI**	Inflation adjusted DJI index value
DJI-F*	Filtered (smooth) version of DJI index
DJI-FD*	De-trended version of *DJI*-F* time series
DTE	Debt To Equity
DY	Dividend Yield
ε	Minimum neighbourhood distance
E	Neural network error
EBITDA	Earnings Before Interest, Taxes, Depreciation and Amortisation
EMH	Efficient Market Hypothesis
ENTR	Entropy (RQA measure)
EPS	Earnings Per Share
FNN	False Nearest Neighbour
GA	Genetic Algorithm
GPM	Gross Profit Margin
GRNN	General Regression Neural Network
H	Hurst exponent
I	Mutual information
IT	Inventory Turnover
I-R	Rank derived from fundamental financial ratios

K	Number of neural network output units
α	Largest Lyapunov exponent
λ	Logistic function growth parameter
L	The average diagonal line length in the RP
LAM	Laminarity (RQA measure)
$L_{\max}$	Longest diagonal line length in the RP
$l_{\min}$	Minimum specified line length in the RP
LOI	Line of Identity
LVQ	Learning Vector Quantisation
m	Embedding dimension
MA	Moving Average
MLP	Multi-Layer Perceptron
MPT	Modern Portfolio Theory
η	Learning rate parameter
NMSE	Normalised Mean Square Error
NN	Neural Networks
NPM	Net Profit Margin
OCFPS	Operating Cash Flow Per Share
OM	Operating Margin
P	Number of training examples
RBF	Radial Basis Function
REC	Recurrence Rate (RQA measure)
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error
ROA	Return On Assets
ROE	Return on Equity
RP	Recurrence Plots
R/S	Rescaled Range
RSI	Relative Strength Index
PB	Price to Book Ratio
P, p	Probability
PCA	Principal Component Analysis
PCF	Price to Cash Flow
PNN	Probabilistic Neural Networks

RATIO	Ratio of *DET* and *REC* (RQA measure)
RQA	Recurrence Qualification Analysis
σ	Standard deviation
S&P	Standard and Poor's Index
SPS	Sales Per Share
SOM	Self-Organising Maps
τ	Time delay
T_{pk}	Neural network desired target value for pattern p and output unit k
TREND	Trend (RQA measure)
TT	Trapping Time (RQA measure)
$V_{\max}$	Longest vertical line length in the RP
VWAP	Volume Weighted Average Price
$\bar{x}$	Average value of x
$\vec{x}$	Vector x
Y_{pk}	Neural network output value for pattern p and output unit k

Chapter 1

The Scope and Methods of Study

1.1 Introduction

The main areas of this research are the study and characterisation of financial time series in order to design advanced self-evolving trading systems based on neural networks and genetic algorithms. These powerful tools are used to discover the underlying structure and behaviour of financial time series. The trading systems are consequently used to establish whether stock returns are predictable to some degree, so that profitable trading systems can be designed after applying realistic transaction costs. Before attempting to apply prediction models to financial time series, an analysis of the underlying nonlinear systems is carried out in order to derive appropriate model parameters.

As the title of this thesis suggests, the trading models in this research are based on predictions. Most professional investment practitioners avoid this word in their prospectuses and marketing material. Claiming that an investment strategy is based on the prediction of what is believed to be random and unpredictable may seem foolish and is very unlikely to raise money from investors. For this reason certain trend followers[1] insist that purely riding the trend has nothing to do with prediction and they only follow and react to the current price. It could be argued that predictions are made by simply believing that the price is going to continue in the direction in which bets have already been placed. Assuming that trend following is a good strategy could be a prediction in itself, as this must be based on some historical analysis?

The word 'betting' is also rarely used in finance and is replaced by "investment". However, the market is a zero sum game and for every winning trade, one looses, for every successful investor/fund/investment company there are many unsuccessful ones. Are winners and losers decided upon purely by random events as suggested by some authors [55], or is there such a thing as competitive advantage? Can the existence of many hedge funds that have an outstanding performance for many years confirm the

[1] Trend following, Michael Covel, www.trendfollowing.com.

existence of competitive advantage? Or, are their returns a simple product of survivorship bias[1] within a random process? Many highly successful hedge funds use quantitative models but do not publish any information publicly. Studies such as this one may help to demystify and broaden the knowledge about certain types of quantitative models that can be used successfully in trading and fund management. These advanced trading models can aid in discovering patterns and inefficiencies in the market and help to create trading strategies that can generate profits after corrections for risk and transaction costs. There is more likelihood of achieving this if problems are tackled from different angles using a variety of different fields, tools, techniques and technologies as opposed to using a single approach that may narrow the perspective and loose sight of the bigger picture.

Most existing research in this area to date is based on linear or simple nonlinear models. These models are mathematically convenient and computationally inexpensive; however they cannot fully estimate the underlying parameters, especially if they are non-Gaussian. This thesis aims to improve the system characterisation by using nonparametric models that can approximate any distribution type and continually evolve along with changing market conditions. Empirical evidence suggests that many financial markets are nonlinear dynamic systems generating prices which do not follow a random walk and cannot be modelled using the normal distribution. They are also much more volatile at the time than standard financial theory suggests and as a result the distributions of price returns have fat tails. These highly volatile periods are clustered in time, and the volatility (standard deviation) is constantly changing over time. In addition, many time series show a presence of memory, i.e. that prices are not independent of each other [50]. For these reasons, alternative forecasting methods are used in this research, based on nonlinear optimisation models that may exploit some of these market properties and inefficiencies. They may discover some neglected areas of the markets and reveal persistent systematic price discrepancies.

The models are developed using a combination of Genetic Algorithms (GA) and different types of Neural Networks (NN) as they are very flexible tools that can model complex nonlinear problems without any assumption of the underlying distribution. These tools can utilise knowledge from various disciplines, mainly Computational

---

[1] See section 1.4.1 for more information on survivorship bias.

Intelligence, Statistics, Nonlinear Dynamics, Fractals, Chaos Theory, Recurrence Analysis, and Fuzzy Logic.

Two outcomes of this research are to determine if the markets are efficient and random, preventing the creation of a profitable trading system, or inefficient, non-random with some predictable patterns, allowing the creation of profitable trading systems. Applying these models to a few time series would not be proof enough, as the results could be explained by chance alone. Instead, they are applied to a large number of stocks and test sets data, which should enhance the credibility of the results as will be discussed.

1.1.1 Objectives of research

This thesis describes research undertaken to examine and develop methodologies for prediction of complex real world time series, based on analysis of securities from US and European markets. The objectives of the experimental studies performed in this research are to evaluate the possibility of their predictability through trading simulation tests. The main areas that this thesis addresses are:

- Evaluating whether the markets fluctuate randomly or not, and presenting some insights into financial market behaviour.
- Characterising the financial time series in terms of their dynamic nature and determining whether they are stochastic, deterministic, or chaotic.
- Calculating the dynamical invariants used for the estimation of the embedding dimension and time delay parameters.
- Analysing other relevant dynamical properties such as the degree of determinism through Recurrence Plots (RP) and Recurrence Qualification Analysis (RQA).
- Analysing market crashes using RP and the RQA.
- Examining the feasibility of prediction in financial time series, applying the most suitable prediction models and comparing their forecasting efficiency.
- Testing of both univariate and multivariate statistical models.
- Testing of both technical trading models and models based on fundamental

variables.

- Modelling the financial time series with neural networks and genetic algorithms and assessing their ability to predict future events from past histories, and highlighting their limitations.

Some models were tested using standard statistical error forecasting measures, whilst others were tested through trading performance and the evaluation of portfolio risk and returns. By analysing the root mean square error (RMSE) and similar statistical measures used to evaluate the performance of forecasting models, it is not possible to tell whether profitable trading strategies can be based upon the results. Instead, the emphasis has been placed upon the performance of trading strategies, which convert model predictions into recommendations for buy/sell actions. In order to closely simulate real trading systems, most models are evaluated using realistic transaction costs and are tested on 5 to 10 years of 'unseen' (out-of-sample) data. These models use various advanced tools and techniques adapted and combined with new ideas for financial analysis of quantitative time series data, as well as fundamental information found in companies' financial reports. It is this successful adaptation and combination of techniques applied to different markets using novel trading models that add a contribution to the knowledge on this subject. Additionally, this research characterises some financial markets over time by evaluating their statistical properties and variations in efficiency and predictability.

It is assumed that the US market is the most "efficient"[1] world market, and if a trading model performs well in this market, then it should perform equally well, if not better in other markets. In this research different methods and models are compared with their individual and combined effects on the final results across different markets

1.1.2 Research hypotheses

The aim of this research is to determine whether financial equity markets are predictable to some degree so that profitable trading strategies can be created which take into account reasonable trading costs.

---

[1] The efficient market is difficult to 'beat' as there are very few opportunities to make profitable trading systems, because the market participants act very fast and iron-out any inefficiencies that can be exploited for profitable trading

The null hypothesis can be expressed as follows:

- H0 - Profitable trading models utilising advanced nonlinear trading systems and applied to equity markets, cannot be created after accounting for reasonable transaction costs.

The corresponding alternative hypothesis is as follows:

- H1 - Profitable trading models utilising advanced nonlinear trading systems and applied to equity markets, can be created after accounting for reasonable transaction costs.

The rejection of a null hypothesis would indicate that the equity markets are not efficient or random, and predictable to a degree allowing for the creation of profitable trading systems. Whilst, the acceptance of the null hypothesis would conclude the opposite, i.e. equity markets are efficient, random and unpredictable, thus not allowing for the creation of profitable trading systems.

In addition to the main hypothesis above we can formulate an additional hypothesis:

- H2 – Neural networks represent superior forecasting models compared to other non-parametric approaches.

1.1.3 Organisation of the thesis

This thesis is presented in nine chapters:

Chapter 1 - "The Scope and the Methods of Study", defines the research hypothesis and provides the necessary background into the research area, including an overview of traditional market models and insights into financial market behaviour.

Chapter 2 - "Neural Networks and Genetic Algorithms", introduces key technologies and tools used in this research. It describes the general design features of neural networks and genetic algorithms and their practical implementation issues.

Chapter 3 - "Fractal Market Analysis", provides an introduction into the main properties of fractals and their relevance in finance. It also describes the technique for measuring the Hurst exponent and its use in deriving trading strategies. The trading simulation test applied to 30 components of the Dow Jones Industrial index is presented utilising the

Hurst exponent as an indicator for switching between different trading styles.

Chapter 4 - "Chaos Theory and Finance", describes the essential properties that lie behind chaotic nonlinear systems and their application to finance. It also portrays calculation techniques for the estimation of parameters required for reconstruction and analysis of time series. The embedding dimension is calculated using the correlation integral and false nearest neighbour methods. The average mutual entropy approach was used to estimate time delay. These dynamical invariants and parameters were estimated for a well known chaotic time series, a logistic equation, as well as for a financial time series. The resulting estimates were consequently used to determine whether the time series were of stochastic, deterministic, or chaotic nature.

Chapter 5 - "Nonlinear Systems Recurrence Analysis", introduces graphical, statistical and analytical methods for studying nonlinear dynamic systems, including state space plots, recurrent plots and recurrence qualification analysis. This chapter describes the use of these methods in analysing various dynamical properties such as the degree of determinism in financial time series, with particular emphasis on extreme market events.

Chapter 6 - "Nonparametric Time Series Forecasting", focuses on "univariate" time series modelling and forecasting using the knowledge and findings from previous chapters. Several nonparametric predictive models such as locally weighted linear, locally linear, radial basis, kernel regression, nearest neighbour, locally constant and neural networks were applied to financial time series and their forecasting performance is compared against "unconditional mean" and the "random walk" reference predictors. Four different trading systems, all utilising Probabilistic Neural Networks (PNN) and optimised using Genetic Algorithm (GA) were applied to thirty constitutes of the Dow Jones Industrial index.

Chapter 7 - "Neural Networks Models Based on Fundamental Analysis", reviews the concept of company's fundamental analysis and tests two different investing styles, the 'value' and 'growth' on a portfolio of European stocks. This chapter also applies dozens of financial ratios derived from their financial statements to trading models and portfolio selection utilising linear regression and a neural networks approach.

Chapter 8 - "Neural Networks Models Based on Technical Analysis", discusses the use of technical analysis in trading and highlights its advantages and disadvantages

compared to models based on fundamental analysis. It also presents the results of several neural networks based trading models applied to several hundred US stocks. The effect of trading costs was also evaluated. The final test in this research determines whether the return achieved by applying the neural network trading model to a portfolio of real price time series differs significantly from that achieved by applying it to a randomly generated price time series, from where conclusions that will support the thesis hypothesis are drawn.

Chapter 9 - "Conclusions and Future Research Recommendations", concludes the thesis, discusses the findings and envisages a roadmap for future research.

All test results are presented at the end of the chapters or sections rather than at the end of the thesis for continuity and ease of interpretation.

1.2 Background

In the literature on this subject, a number of different methods have been applied in order to predict financial time series. These methods can be grouped in four major categories: 1) fundamental analysis, 2) technical analysis, 3) traditional time series forecasting and 4) machine learning methods.

Fundamental analysis is the study of a company's financial condition, operation, its industry and competitors and the general economic environment in order to determine its 'real' or intrinsic value [26], [30], [34], [35], [36], [37], [38]. Typical fundamental analysts will buy/sell shares in a company if the intrinsic value is greater/less than the market price.

Technical analysis is one of the oldest methods and is based on the interpretation of patterns, trends, cycles, and formations that develop on charts, with a primary aim of identifying major turning points in the market [1], [3], [6], [7], [15].

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 into two categories: the univariate and the multivariate regression models, depending on whether they use one of (or) more variables to approximate the financial time series. Current methods however do not fully characterise the financial prediction problem.

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 them (linear or nonlinear) in order to approximate the underlying function that generated the data. They include advanced modelling tools such as neural, genetic and fuzzy systems, which exploit nonlinear relationships and spot the recurring patterns in financial markets [8], [10][10], [11], [12], [13], [19], [24]. The level of success of these methods varies from study to study and is dependent on the underlying datasets and the way that these methods are applied each time. Nonlinear chaos theory and fractal statistics have been used to explain the seemingly random behaviour of economic time series [16], [17], [18], [23], [25], [41], [42], [43], [50], [53]. Though all these approaches seem different they are all interrelated and have many common features. For example, parts of the technical analysis of the Elliott Wave Theory and Fibonacci numbers can be explained as a result of the fractal nature of the markets. Nonlinear systems are an integral part of chaos theory. Chaotic systems generate fractals. Neural networks are efficient tools that can model nonlinear systems efficiently. There is therefore a common thread between all these areas which will be covered in this thesis.

This research concentrates on machine learning Neural Networks (NNs) models, combined with fundamental analysis and technical analysis methods. In some cases the results are compared with traditional time series linear forecasting methods. There is a growing interest in the application of neural networks to financial engineering. Their ability to extract essential information buried in noisy data by nonlinear mapping of many input variables to one or many output variables and their fast response time in recall mode makes them an excellent tool for building real-time financial applications. NNs can be applied to many problems in finance where little is known about the relationship between variables and where it is not possible to derive a deterministic model. The most common analysis performed by NNs are continuous time series forecasting where they perform regression tasks, or solving classification problems where approximate probabilities of different categories are expressed as functions of input variables.

A large number of academic research papers on this subject have been published, but very few by commercial investment companies. A credibility gap exists. It could be that there is a slow acceptance of this technology in the financial world or that there is an element of secrecy involved. The following are some examples of the few successful

commercial implementations that have been published.

According to Loofbourrow, Fidelity Investment makes use of neural networks to manage a portfolio worth $3 billion [14]. Nikko Securities has implemented a hybrid neural network and fuzzy logic expert bond-rating system that has produced a better rating over long periods when compared to human expertise alone [14]. Citibank London has developed a foreign exchange trading system based on neural networks and genetic algorithm that guesses a 60 % directional change and makes a 15% yearly return [14].

One of main advantages of using such computerised models is that they introduce a stricter element of discipline in trading, removing elements of personal feeling or interpretation.

On the subject of stock market prediction, researchers and academics tend to be divided into two groups: those who believe that the market is to a certain degree predictable; and those who believe that the market is efficient, purely random, reflecting all information in current prices, making any predictions that yield positive returns impossible, as described in Efficient Market Hypothesis (EMH). Furthermore, the latter 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. However, the EMH founded on random walk theory and linear models is not well supported by empirical evidence[1] and has been questioned in recent years. Though EMH may not provide a full explanation of market behaviour, it is a good approximation of what occurs in the market most of the time. The main problem is that current market mathematical models ignore extreme situations of high volatility on the basis of a very low probable outcome. However, extreme movements in the market are relatively common. The assumption that the market is in equilibrium and that the past does not influence the present or the future is commonly held, but this could be far from reality.

Before concentrating on neural networks, nonlinear systems and forecasting models, a brief outline of current financial models and their shortcomings will be presented.

---

[1] The distribution of financial returns exhibits fat tails and is not as uniform as the shape of the bell curve. A normal distribution is generally chosen for its analytical simplicity.

1.3 Traditional financial models and their limitations

Bachelier [4] was the first to introduce the application of random theory based on normal distribution into financial market models. His work went unnoticed for over fifty years and only by the 1950s and 60s was his work established as the bedrock of the Efficient Market Hypothesis. In the EMH model, price returns are independent of the previous day's returns, retaining no element of memory, akin to tossing a coin. Thus the best forecast of a future price is the current/today's price. The EMH assumes that all public information is already discounted in the price, that investors are rational, risk averse and react in the same linear fashion[1]. However, this is far from reality as most investors do not react in the same way; some look to the short term and others follow long-term strategies. Speculators may do many trades a day, corporate treasurers may trade on weekly or monthly basis while central banks trade only occasionally. As well as operating on different time scales, the element of risk taking varies, with some investors reacting to information immediately and others waiting for it to accumulate and reach a "critical" point once a trend is confirmed. This tendency to react in a nonlinear fashion may trigger chaotic behaviour in the markets. After all human nature is not perfectly rational, an example being the number of people that play the national lottery despite the remote probability of winning.

Another assumption is that price changes are continuous, moving smoothly from one value to the next, not allowing for large jumps in either direction [53]. In a case such as this, continuous functions and differential equations could be utilised to model market behaviour. However, price jumps are very common in the financial market, from small ones caused for example by brokers skipping the intermediate values in their quotes, to large ones, often due to overnight "gaps" caused by strong positive/negative news reports.

EMH also assumes that price changes follow Brownian motion, with a normal distribution and constant variance. However, the evidence of fat tails, short term and more recently long term dependence is mounting. Real financial data does not support those EMH assumptions, but at the same time the theory that fully explains the deviation is yet to be discovered.

Traditional models, with their many assumptions, are relatively easy and convenient but

[1] Similar causes would produce similar reactions.

they do not fully capture and explain the complexity of market behaviour. They model investment behaviour by using simple linear differential equations with a single solution.

Financial engineering was founded by Markowitz [82], [83] and in his market models every stock is described with two numbers, the mean and variance, representing reward and risk respectively. His later model, termed Modern Portfolio Theory (MPT) and Sharpe's Capital Assets Pricing Model (CAPM) [52] use the same principles.

The MPT model is all about selecting a good mix of stocks to hold in a portfolio. Markowitz emphasised that investors ought to maximise expected returns on their investments for a given level of risk. He proposed the variance of returns of a portfolio as the measure of its risk, and the covariance of a stock's return with respect to the portfolio as a measure of how diversifying that stock would be for the given portfolio. His formulation led to a solution of the portfolio selection problem in terms of a quadratic optimisation problem, which is at the core of most portfolio management systems today.

Sharpe criticized the MPT model, stating that: "If everybody in the market evaluated the same selection of stocks and used Markowitz's Modern Portfolio Theory they would arrive at the same portfolio, one portfolio for all." The CAPM is a model that grew from the MPT model and is based on the T-bill rate and stocks "beta" value, which measures the strength of the linear relationship between the asset value and the return of the market overall. This model was soon modified to include as many factors as required leading to the birth of the Arbitrage Pricing Theory (APT) model. In order to accommodate changes in volatility over time, economists devised a further model called GARCH. From here on models have been evolving whenever shortfalls appear.

Black and Scholes [113] quantified the relationship between what is now a standard financial instrument called an "option" and the stock or other asset underlying it. Their option pricing formula had wide acceptance and its use in the rapidly growing derivatives segment of the financial markets is well established, and has also found wide use in other areas of finance.

Some researches have since introduced different models; Mandelbrot's model of asset returns based on fractional Brownian motion and multifractal time [50], and Ross's Arbitrage Pricing Theory that handles nonlinear relationships [51].

1.3.1 Market (in)efficiency and the distribution of financial returns

An analysis of the Dow Jones Industrial (DJI) index shown in figure 1-2 highlights some of the problems faced by traditional models' assumptions. The price shown in the figure is the actual price, not the logarithmic price commonly used in financial charts. Understandably, an index with an exponential growth may look less extreme on the logarithmic plot. It took over sixty years for the DJI to reach the 2000 mark, and less than twenty years to break the 13000 level.

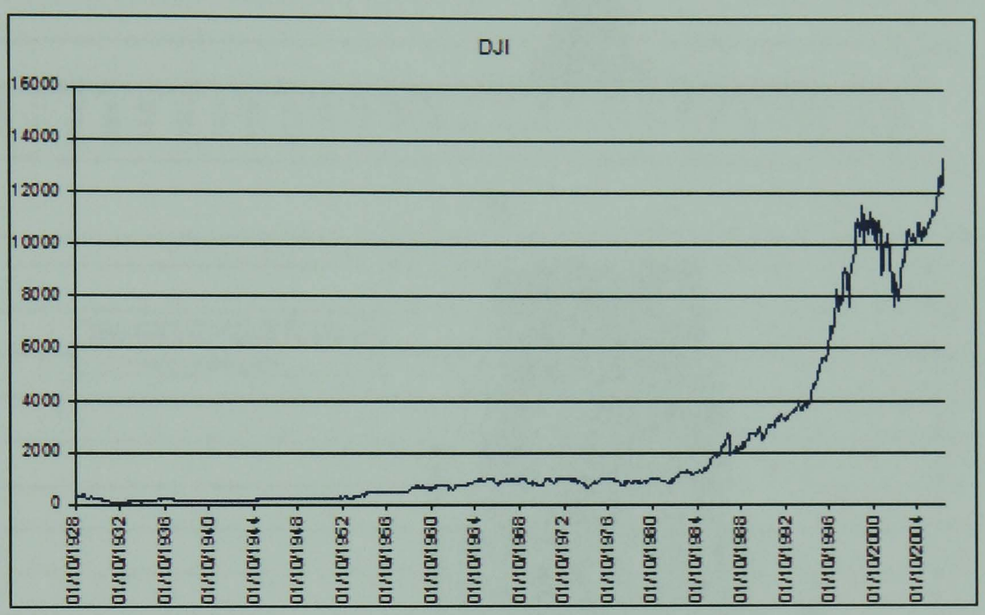


Figure 1-2. The Dow Jones Industrial index (01/10/1928 – 01/05/2007).

Figure 1-3 shows the DJI daily return distribution compared to the normal distribution. The index has a very high peak, which reflects the high kurtosis of 26.78 (table 1-1). The extent of its fat tails are only obvious in chart B which is a magnified part of the distribution chart A. The normal distribution graph (in red) tails off rapidly showing no changes greater than 5σ, while the DJI distribution (in green) has many changes beyond 5σ including one at 22σ.

Table 1-1. DJI descriptive statistics.

Mean	0.00008663
Standard Deviation	0.00494912
Sample Variance	0.00002449
Kurtosis	26.78288073
Skewness	-0.66802262

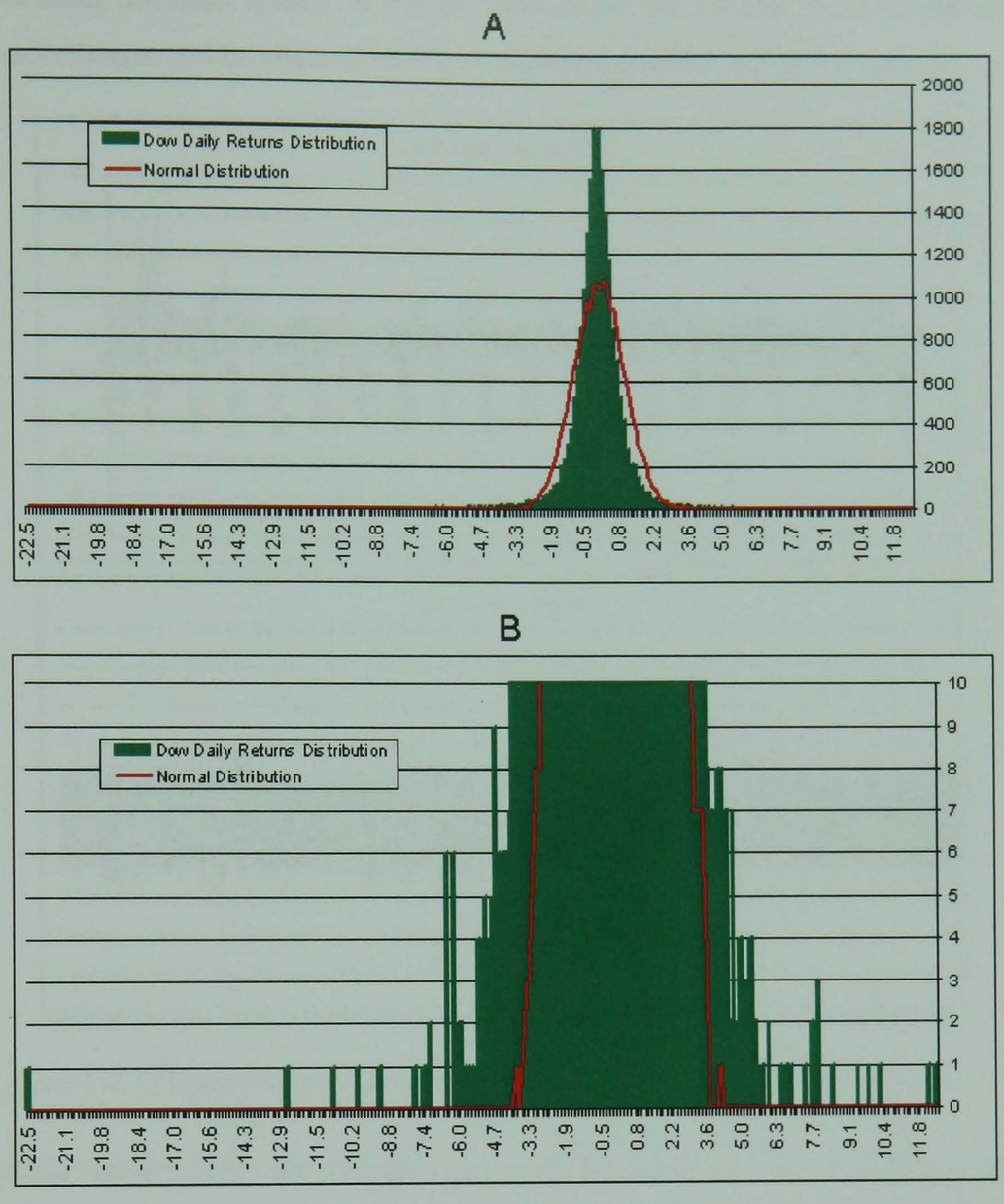


Figure 1-3. The DJI price change distribution compared to normal distribution (A) and the fat tails (B).

When comparing the following graphs in figure 1-4 the difference between the DJI changes and the normally distributed time series changes are very noticeable. The DJI changes are irregular, with alternatively narrow and widened parts and often highly volatile periods are clustered in time. There are too many very small price changes causing high kurtosis, and numerous big price changes causing fat tails, with few points in between. Whenever we have clustering, with many extreme and small variations of data as shown in figure 1-4 (DJI case), the overall mean measure does not make sense or simply does not exist. As we include more data, the sample mean either keeps increasing or decreasing. There is no one value that best describes the data as it extends over a

range of many different values. In further sections we will see that the ratio between big and small changes can be characterised by a parameter called fractal dimension.

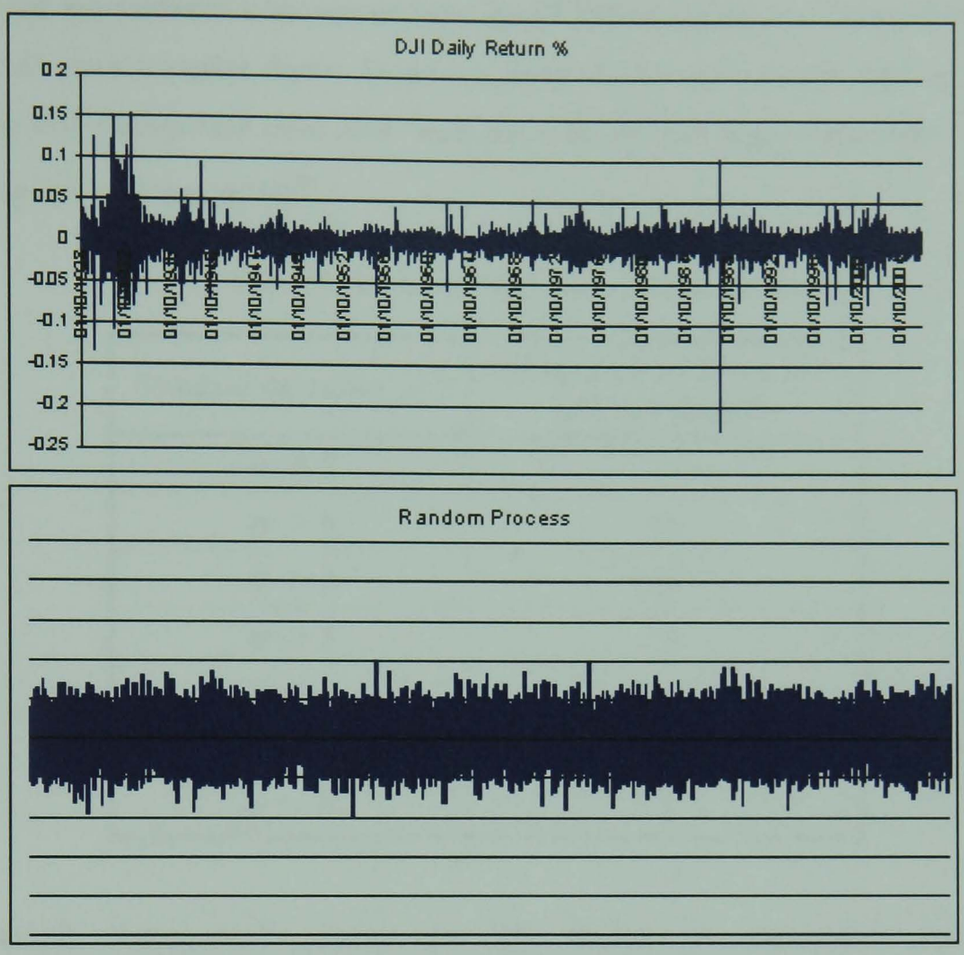


Figure 1-4. DJI price changes compared to computer simulated randomly price changes.

The media tends to give the average income in the financial City of London, but this figure is highly distorted by a few extremely large individual incomes. Similarly, the average distance between stars in the universe is a meaningless figure, as they tend to cluster in galaxies that are a huge distance apart. If the mean and the variance of the market time series are of a similar nature, then most of the current financial models are far from correct and reliable.

In order to compare the risk of these two time series, we measure the changes in standard deviation (σ) shown in figure 1-5. The DJI volatility is extreme and constantly changing over time compared to Brownian motion, which shows small and uniform changes. For normal distribution the probability of an event being larger than 8σ is 0.000000000001279. According to this, it should take around 1,555,000,000 trading years for such an event to happen, but there have been 11 occurrences in DJI

during the last 78 years (1928-2006). Similarly the 5σ event should happen every 7,000 years, and there have been 70 for the same period. Occurrences above 10σ in normal distribution are virtually nonexistent (one in 10 billion years)[1], so standard Gaussian tables don't even consider them. However, there were 6 such events, and on Oct 19th 1987, the DJI change was over 22σ with the odds of that happening under Gaussian rule being less than one in 10^{50}.

Table 1-2. Common "rare" events in DJI (1928-2006).

Standard deviation (σ)	Number of occurrences in DJI (1928-2006)
$\sigma > 5$	70
$\sigma > 6$	44
$\sigma > 7$	24
$\sigma > 8$	11
$\sigma > 9$	10
$\sigma > 10$	6
$\sigma > 20$	1

The example above clearly shows that DJI changes do not follow the normal distribution as described by the bell curve. It is interesting to note that the recent DJI price variability is similar to that which occurred 90 years ago, a time at which the markets were unregulated.

Stocks, currencies and many financial instruments are much riskier and turbulent than current models indicate. Many financial organisations are aware of this but are reluctant to put in practice a more stringent risk management system, as this would necessitate larger cash reserves in order to cover up potential losses and consequently create stricter requirements upon business. Therefore, companies introducing this would be at comparative disadvantage. However, market bubbles and crashes are inherent to all markets and it is just question of bad luck and time as to who is going to be the next big looser, as was the case for Long-Term Capital Management, Barings, Amaranth and many others.

---

[1] Older than Big Bang?

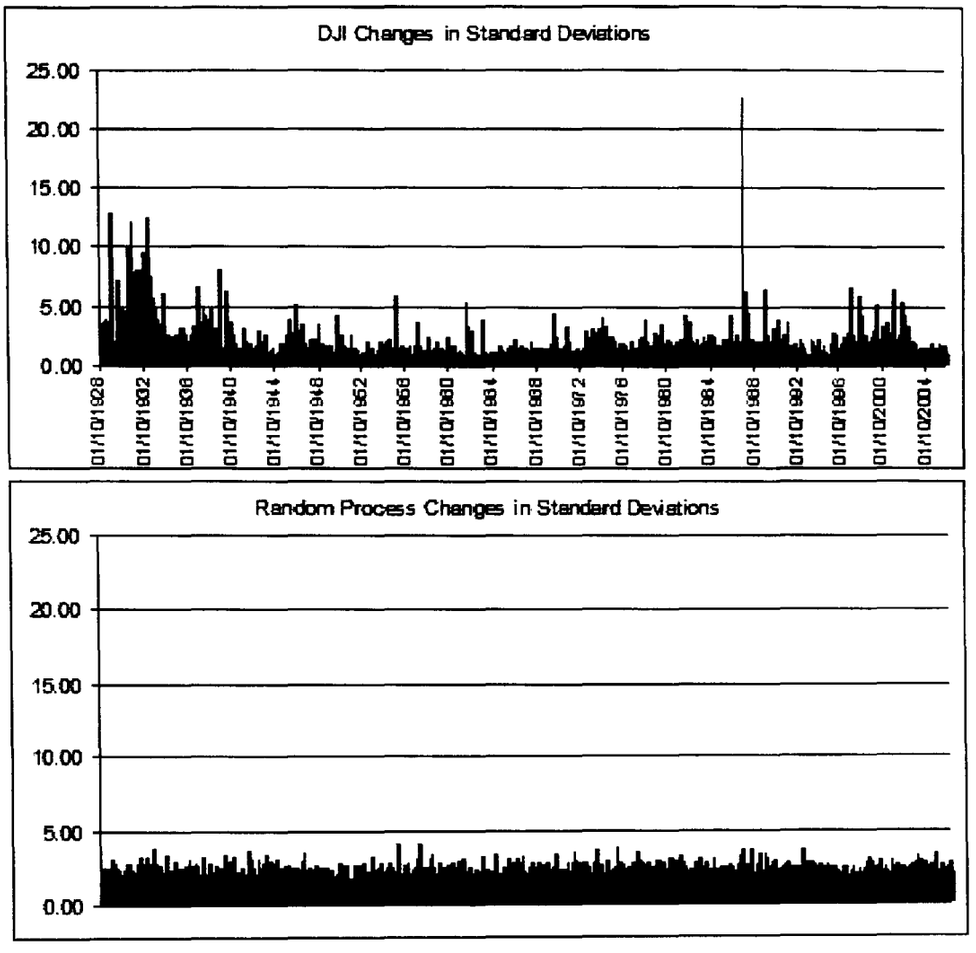


Figure 1-5. DJI changes (in standard deviations), compared to Brownian motion.

If markets are not purely random, but exhibit nonlinear behaviour expressed through market bubbles and crashes, it would be possible to create successful trading systems that could exploit market inefficiencies. The next section will describe the basic properties of a successful trading model.

1.4 The two main components of a successful trading system

Apart from risk management, the two most important ingredients of a successful trading system are the percentage of profitable trades and the average payoff ratio. The latter is derived from the average amounts of winning and losing trades. Having these principles right and armed with a good risk and money management system would put any trading strategy on a good track to perform well. Ideally, a winning probability should be more than 0.5 (50%) and the win/loss ratio should be above 1.0. Only by looking at these two measures jointly can we assess the performance of a trading system. Individually, they tell us very little, as there can be a good trading system with a 0.4

winning probability (figure 1-1, C), and a bad one with a 0.6 winning probability (figure 1-1, D). Trend following systems usually have a winning probability of below of 0.5 and a win/loss ratio of above 1, whilst reversal systems have winning probability above 0.5 and a win/loss ratio below 1.

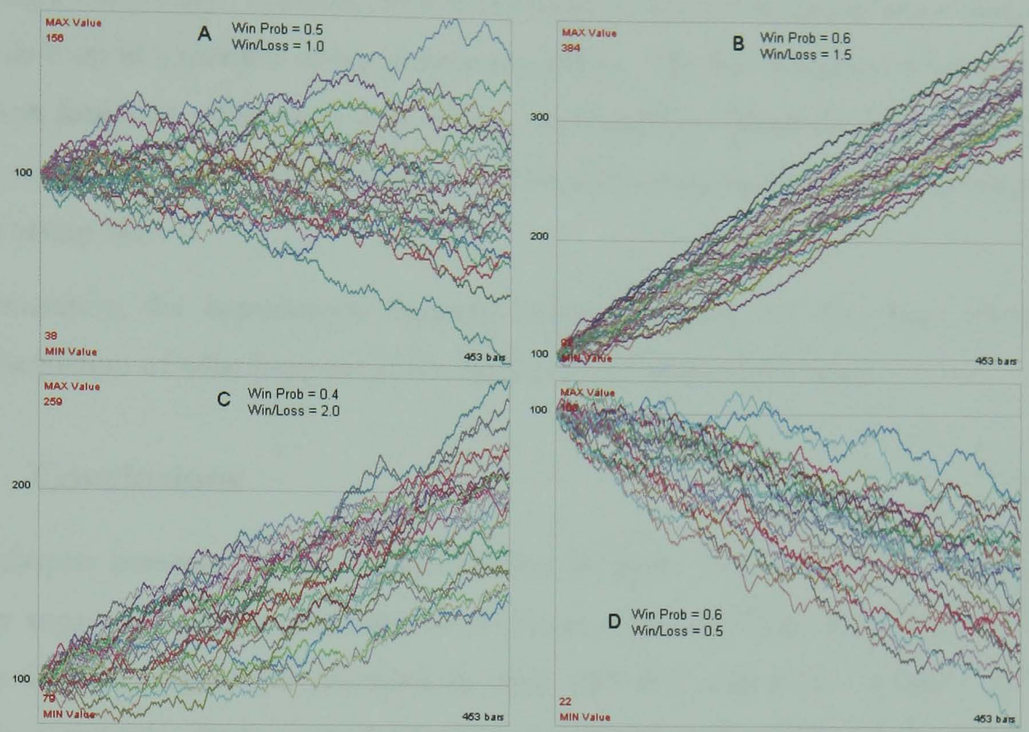


Figure 1-1. Simulated performances of 30 time series randomly generated, using different winning probabilities and payoff ratios.

Figure 1-1 shows the simulated performance of 30 time series that are generated randomly using different winning probabilities and payoff ratios. The x-axis represents time and the y-axis corresponds to a randomly generated value. All the graphs are generated for 453 bars[1] where the process of trading is absolutely random and the only difference is the one made by positive/negative mathematical expectation determined by the win/loss ratio and the winning probability parameters.

1.4.1 Survivorship bias

If we assume that the time series presented in figure 1-1 represents the performance of 30 funds, starting with an initial value of 100, that makes plot A's results very interesting. With a win probability of 0.5 and a payoff ratio of 1.0, we would expect 30

[1] Almost two trading years, assuming these are daily prices. The graphs are produced using the Java applet from http://www.hquotes.com/tradehard/simulator.html.

funds to hover randomly around the 100 value line, but instead most of them depart far from the line symmetrically on each side. On average, they don't make a profit, but some individual funds have a staggeringly good performance (the best performing one with 56% profit), and some unlucky ones have a very poor performance (the worst one showing a -62% loss). Typically, investors pull their money out of the worst performers and pile it up in a number of top performing funds. The less fortunate funds close and are soon forgotten. This cycle is continuous and analysts follow the funds that survive, often omitting the funds that disappeared from their analysis. This situation is known as survivorship bias.

Unfortunately, the hypothetical example shown in figure 1-1 (A) may represent a realistic picture of what happens in the fund industry most of the time.

1.5 Conclusions

This chapter presented evidence that markets are more volatile than standard financial theory suggests and as a result the distributions of price returns have fat tails. These highly volatile periods are clustered in time, and this makes the standard deviation change constantly. Preliminary data exploration indicates that markets exhibit nonlinear behaviour, with a probable chaotic element. If markets are not purely random and exhibit nonlinear behaviour expressed for example, through market bubbles and crashes, it would be possible to create successful trading systems that can exploit market inefficiencies. The basic requirements of such systems were presented.

The next chapter will introduce key technologies and tools, i.e. neural networks and genetic algorithms that are subsequently used in this research to build trading models.

Chapter 2

Neural Networks and Genetic Algorithms

2.1 Introduction

In the past, scientific research was mainly based on empirical and theoretical studies. In recent decades it is relying more on computational analysis and currently we see more research based on empirical data exploration, combining theory, experiments and computation with advanced tools and new algorithms. There are many time series found in finance and other disciplines that are impossible to analyse using conventional linear models, where nonlinear and non-parametric methods such as neural networks and genetic algorithms are more suitable. In this chapter these two key technologies and their practical implementations are described. Only concise and relevant description of them will be presented.

2.2 Neural networks

Artificial neural networks are very flexible tools that can model complex nonlinear problems and extract underlying functionality from data without any assumption of the underlying distribution. Their development is based on the basic understanding and functionality of biological neural networks.

2.2.1 Biological neural networks

The human cerebral cortex is made of complex network of cells called neurons. Their main role is to process and transmit neural information. Some estimates are that the brain contains between 10 billion and 1 trillion of these neurons [117]. A simplified biological neuron is shown in figure 2-1.

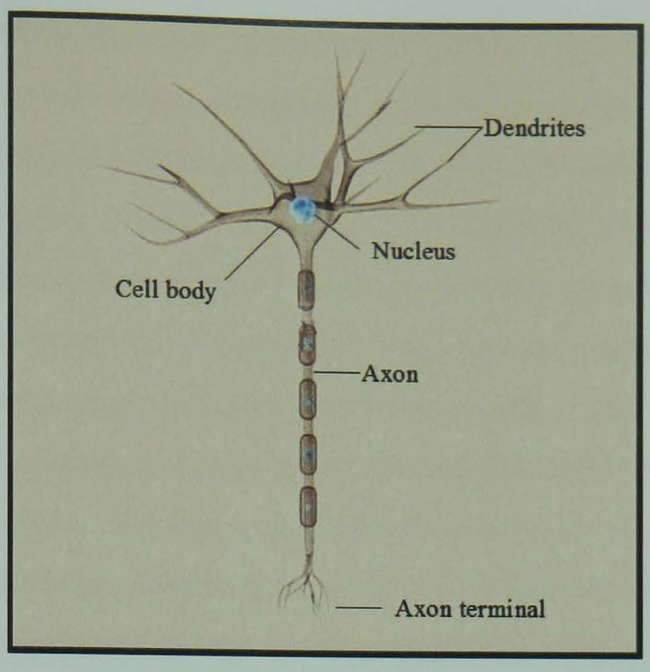


Figure 2-1. Neuron[1].

Each neuron consists of a cell body (soma) including nucleus, one or more long nerve branches called axons and a number of shorter and narrower extensions called dendrites. Among other functions dendrites act as receptors receiving impulses from other neurons and passing them on to the neuron's cell body. If stimulus from the dendrites reaches a certain threshold, the cell fires and generates an electrical impulse, which is conducted away from the cell body via the axon to a synaptic knob at its end, triggering the release of neurotransmitter molecules passing the impulse on to the neighbouring target cell.

This very simplified description of neurons and their functions is the basis for artificial neural network development. Unfortunately, we are still in the early stages of understanding the complexities of brain function and far from creating artificial models with similar abilities. However, the advances in the pattern recognition applied to areas such as language, speech and vision are remarkable.

2.2.2 Artificial Neural Networks (ANN)

There are many types of artificial neural networks; some of them mimic basic human brain functions including memory, speech and the use of language, whilst other types are developed through a scientific engineering approach of pattern recognition, classification, and prediction. They attempt to duplicate the actual parallel processing

[1] Source: Royalty-Free Stock Photos, www.shutterstock.com.

capability of the nervous system at an elementary level. The ANN are composed of numerous interconnected simple computing elements (neurons) connected by many links of variable numerical weight associated with them. Positive weights activate the neuron whereas negative weights inhibit it. Statistically speaking, these weights represent free parameters in a complex nonlinear function and their values are set by means of regression. From an AI perspective they represent memory and the learning process is achieved through the adjustment of these weights. The ANNs are in essence statistical devices, consisting of a large set of interdependencies which may incorporate any degree of nonlinearity, allowing very general functions to be modelled [96]. This chapter focuses on feed-forward neural network models because they are simpler to use than others, better understood, and closely connected with statistical classification methods.

2.2.2.1 The structure of a multilayer ANN

Probably the most successful and widely used neural network type is the multilayer back-propagation network with a graphical representation shown in figure 2-2.

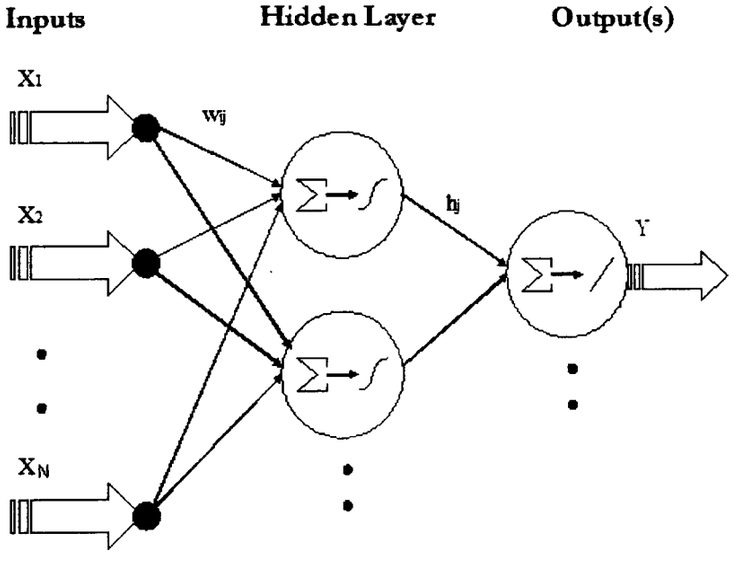
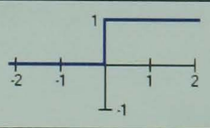
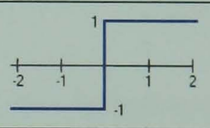
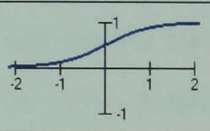
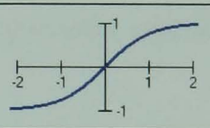
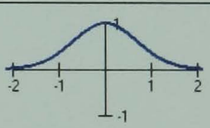
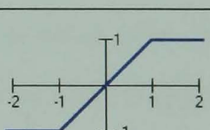
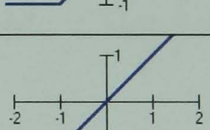


Figure 2-2. A typical two-layer neural network.

In this type of network the computational units are arranged into a sequence of layers; an input layer, one or more middle (hidden) layers and an output layer. The hidden layer gives the network the ability to generalise and find relevant answer from previously

unseen data. The inputs $x_1, x_2, ..., x_N$ are connected to neuron j with weights $w_{1j}, x_{2j}, ..., x_{Nj}$. Each network unit receives a number of signals from either the network inputs or the activations of incoming neurons. These signals are multiplied by the connecting weights, summed and passed through some form of a nonlinear threshold function which determines the neuron's activation level, and passed onto a next layer of neurons. These activation functions, also called squashing or transfer functions, usually map any real number into a bounded domain 0 to 1 or -1 to 1. The nonlinear transfer functions offer the possibility of building nonlinear models. The most commonly used activation functions are presented in table 2-1.

Table 2-1. Neuron activation functions.

Step (Heaviside)	$f(x) = \begin{cases} 1 & if\ (x \geq 0) \\ 0 & if\ (x < 0) \end{cases}$	$f \in (0,1)$	
Sign	$f(x) = \begin{cases} 1 & if\ (x \geq 0) \\ -1 & if\ (x < 0) \end{cases}$	$f \in (-1,1)$	
Sigmoid (logistic)	$f(x) = \frac{1}{1+e^{-x}}$	$f \in (0,1)$	
Sigmoid (hyperbolic tangent - tanh)	$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	$f \in (-1,1)$	
Gaussian	$f(x) = e^{-x^2}$	$f \in (0,1)$	
Semi-linear	$f(x) = \begin{cases} -1 & if\ (x < -1) \\ x & if\ (-1 < x < 1) \\ 1 & if\ (x > 1) \end{cases}$	$f \in (-1,1)$	
Linear	$f(x) = x$	$f \in (-\infty, \infty)$	

Early ANN models, such as perceptron, used a simple step activation function which is appropriate for discrete neural networks. The sigmoid functions are more appropriate for analogous networks and are most commonly used due to the ease of computing their derivatives which are used in the learning phase. The Gaussian functions are mainly used in Radial Basis Function (RBF) networks. The linear function is mainly

used in output units and if used in other neurons, the network will only have a linear capability to fit functions.

The output of the neuron j is specified by the following equation:

$$O_j = f_j\left(\sum_i^N w_{ij}x_i - \theta_j\right) \tag{2.1}$$

where f_j is the activation function, w_{ij} is the set of weights connecting all N inputs to the neuron j, x_i are input values, and the term θ_j is subtracted from the weighted sum representing the threshold value of the unit j that the sum must exceed to make the neuron fire.

In the case of the sigmoid activation function the equation can be written as follows:

$$O_j = \frac{1}{1+e^{-\sum_i^N w_{ij}x_i - \theta_j}} \tag{2.2}$$

In the so called feed-forward NNs the output signals are passed in one direction only, whilst Recurrent Neural Networks (RNN) may selectively route the unit output back to the earlier nodes, with the characteristics of a highly nonlinear system with feedback. This type of network is designed specifically to model time-varying patterns and it is a good choice in financial time series forecasting.

2.2.2.2 Learning algorithms

As the network's "knowledge" is stored in the weights, the learning process takes place during their updates. The *learning* or *training* algorithm finds a certain set of weights for the network to perform a desired task. Generally, there are two main approaches to learning in neural networks:

- In s*upervised learning* or *associative learning,* the network is presented with a 'teacher' representing a set of examples of corresponding inputs and desired outputs ("targets"). The training algorithm works by minimising the error between the output and the target (actual) values and propagating the error back by adjusting the network weights. In the first instance, they are initialised into small random values and as the learning progresses they are increased to a larger positive or negative value. There are many variants of supervised training methods with

back-propagation often being used in multilayer networks [97]. In the forward pass, the outputs for a specific input pattern are calculated, and the error at the output unit is determined. The learning algorithm has to select the weights in order to optimise the match between output and target values, typically by minimising the total mean squared error E over all training examples P and the output units K:

$$E = \frac{1}{P}\sum_{p=1}^{P}\left(\frac{1}{2}\sum_{k=1}^{K}\left\|T_{pk} - Y_{pk}\right\|^2\right) \tag{2.3}$$

where T_{pk} is a desired target value and Y_{pk} is the network output value. In the backward pass, weights are modified with the amount proportional to the first partial derivative of the error with respect to the weight.

$$\Delta w_{j,i} = -\eta \frac{\partial E}{\partial w_{j,i}} \tag{2.4}$$

This algorithm is called the gradient descent as it tries to move down the lowest value of an error surface. The η parameter, called *learning rate*, controls the size of the correction term that is applied to adjust the neuron weights during training. In other words, it determines the size of the descent step down the error surface. The small value of the learning rate increases the possibility of the model getting stuck at a local minima of the error surface. On the other hand, the large value may speed up the training time, but it may also cause the model to fail in finding the optimal solution.

The update rule can also be influenced by a *momentum term*, which is proportional to the size of the previous update. It acts as a smoothing parameter and helps the model to find the global error minima.

For each input-output training example one forward and one backward pass is performed. One cycle or epoch is completed when all training samples have been presented to the network. The process is reiterated until a sufficiently low error is obtained or a maximum number of training cycles is reached. Networks can be trained *incrementally* (on-line learning), where the weights are adjusted after each learning pattern presentation or in a so-called *batch* mode, in which the

weights are adjusted only after each epoch.

When the error is within acceptable limits and the network has learned or memorised all training patterns to a certain degree, it can then be (and is) considered "trained". Any new data presented to the network will produce an output value that is simply a weighted average of the target values of training cases close to the given input case.

- In *unsupervised learning* or *self-organisation* there is no priory set of categories ('teacher'); the network discovers clusters or patterns from the input data directly [98], [99]. Each of the sample inputs to the network is assumed to belong to a distinct class and is classified accordingly. Such models include Gaussian mixture models, that represent data in terms of a probability density function, and Kohonen networks, where data is modelled in terms of cluster centres and widths.

2.2.2.3 ANN Generalisation and over-fitting

When forecasting real financial data, the most important issue is the quality of future predictions (i.e. generalisation) outside the training set; how well a model fits the training data is of less importance. One way to improve the generalisation is to use a second data set, the *validation set,* in addition to the training set. The validation set is used to test the model for its predictive ability and signal when to stop the training of the ANN. Each time the network weights are adjusted using the training set, their performance is measured against the *validation* set. With an increase in the model size and the training time, the fitting error of the model decreases. However, the validation test error of the forecasts beyond the training set will usually start to increase at some point because the model will be fitting extraneous noise in the system. This is called *over-fitting.*

The right network complexity and the training time must be found for the network to generalise well. Figure 2.3 illustrates the principle of generalisation and memorisation (over-fitting), which is also called bias and the variance problem.

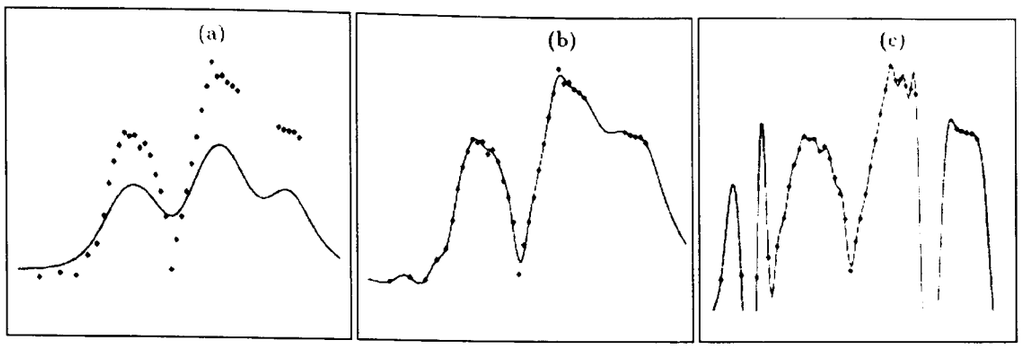


Figure 2.3. The generalisation vs. overfitting: case (a) is a typical example of poor generalisation due to an "under-trained" model or not complex enough model, (b) is a case of good generalisation, whilst case (c) is an example of an over-trained and complex model having poor generalisation. Source [100].

The over-fitting is usually controlled by *cross-validation* (or early stopping) [101], through the use of *regularisation terms* [102], and *model complexity* selection methods [103].

The early stopping or cross-validation method is an ad-hoc solution that monitors the training and cross-validation error at the same time and prematurely stops the learning algorithm that minimises the training error at the point where the cross-validation error starts to rise (figure 2.4).

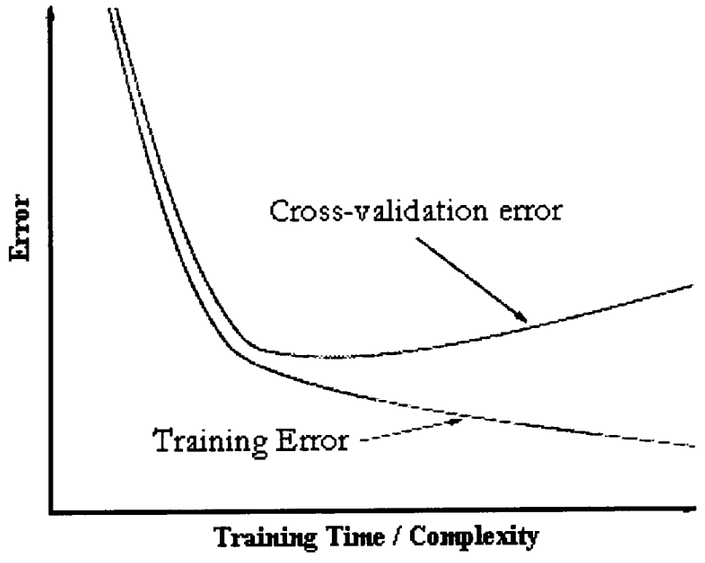


Figure 2.4. An example of a training and the cross-validation error function. Source [100].

The regularisation methods introduce an element of bias in the training algorithm, that penalise some unwanted properties, for example, excessively large weights. This *weight decay* method constrains the network by prohibiting weights from growing too large and adds an extra term to the cost function, which penalises large weights [108]. Another

simple way to avoid over-fitting is the use of large data sets.

2.2.2.4 Neural network topology

The structure of the Multi-Layer Perceptron (MLP) network is defined by the number of hidden layers and the number of neurons in each hidden layer. The model selection process should follow *Occam's razor* principle that gives preference to the simplest structures that have a minimum number of free parameters and yet model the data adequately [100]. There are a number of different ways to attempt to find an optimal network structure.

One approach is to start with a large network and use a *pruning* algorithm, which reduces the size of the network by removing the unnecessary links or units while still preserving the model's accuracy. The algorithms that remove the links are *Magnitude Based*, *Optimal Brain Damage (OBD)* [105], and *Optimal Brain Surgeon (OBS)* [104], whilst the *Skeletonisation* [106] algorithm removes the units.

The second approach starts with a minimal network size, consisting only of an input and an output layer, and the algorithm adds the hidden layer units until the test (and/or cross-validation error) falls below a given value. This type of algorithm is known as the *Cascade-Correlation (CC)* method [107].

Apart from possible improvements in network generalisation ability, both approaches above may lead to a better and smaller fitting architecture, reducing the run-time/memory requirements and finding relevant input variables.

There are no fixed rules in determining the network structure and its parameter values. The number of weights and hidden layers/nodes depends largely on the complexity of the function to be modelled, the number of training cases, the amount of noise present, the desired accuracy of generalisation and the type of activation functions used. A large number of ANNs may have to be constructed with different parameters and structures before determining an adequate model. This trial and error process can be tedious and time consuming. All this indicates that finding a good network size is an optimization problem and the best probable way to do it is by using the Genetic Algorithm (GA). A more detailed description of GA is presented in the following sections.

2.2.2.5 ANN applications in finance

ANNs can be applied to many problems in finance where little is known about the relationship between variables and where it is not possible to derive a deterministic model. The neural network prediction approach is to identify the present state of the system which is producing the time series, and search past history for similar states/patterns from which future information can be inferred. Given enough data and enough hidden neurons, neural networks are capable of approximating any continuous functional mapping. Their ability to extract essential information buried in noisy data by nonlinear mapping of many input variables to one or many output variables and their fast response time in recall mode makes them very good for building real-time financial applications. ANNs are very adaptive and handle noisy and incomplete data sets well, which makes them particularly useful in finance, where the environment is potentially volatile and dynamic.

Neural network outputs can be expressed as conditional probabilities if appropriate cost and activation functions are used [5]. In this way networks are used as classifiers, showing both the presence and the probability of class membership. They are usually trained by minimising the least square error function, but other cost functions can be used such as those that maximise risk adjusted return, the number of winning trades or any similar performance measure.

ANNs can be applied to a wide range of tasks including pattern recognition, classification, interpolation, filtering, system estimation and forecasting. There are many areas in finance where ANN's are or could be used and some of them being:

- Financial and economic forecasting, trading, investing and portfolio management.
- Risk management, credit authorisation screening, risk rating of fixed income investments, fraud detection and bankruptcy prediction.
- Securities and derivatives pricing.

The three main ANN types are Multi-Layer Perceptron (MLP), Radial Basis Function networks (RBF) and Self-Organising Maps (SOM). The other networks are variants of these three main types. For example, Recurrence Neural Networks (RNN) is a variant

of MLP, General Regression Neural Network (GRNN) and Probabilistic Neural Networks (PNN) are variants of the RBF network, and the Learning Vector Quantisation (LVQ) is a variant of SOM. In order to solve complex tasks, a combination of two or more of these types can be applied. Multiple neural network architecture can provide better results in the case of complex problems but requires a more detailed problem analysis. Complex problems can be divided into several sub-tasks and modelled individually by different networks of the same or different types respectively. An example could be the use of different networks to model different time frames in a time series. Another is the use of different networks to model different probability distributions.

The main practical applications of these neural network types and their strengths and weaknesses are presented in table 2-2.

Table 2-2. Main applicability and properties of MLP, RBF and SOM networks.

Type	Applications	Strengths	Weaknesses
MLP	Discrimination analysis Classification Interpolation Forecasting Filtering Process modelling	Suited to a wide range of problems. Interpolates and generalises well. Can accept both continuous and categorical inputs.	Training algorithm can get stuck in local minima. Will not indicate when inputs are outside the scope of the training data.
RBF	Discrimination analysis Classification Interpolation Forecasting Filtering Process modelling	Can model local data more accurately than MLP. Can indicate novel inputs. Easier and faster to train than MLP. Greater nonlinear capability than MLP. Can be trained using a mixture of labelled and unlabelled data. Models are formulated entirely in probabilistic terms, which facilitates the interpretation of confidence boundaries.	Not suited to applications with a large number of inputs (curse of dimensionality) as number of hidden units grows exponentially with number of inputs. Poor at representing the global properties of data. Limited interpolation capabilities. PNN cannot ignore irrelevant inputs without modifying basic algorithm.

SOM	Discrimination analysis Classification Data compression	Can cope with individual classes when they have a rich variety of forms. Generally better at discrimination than MLP where there is a severe imbalance in the number of examples from each class. Can be used to distinguish a signal from a noisy background where the noise comes from different sources. Does not need labelled training data, can be used when little is known about the data.	Poor at representing the global properties of data. Provide discrete rather than continuous outputs, which limit their use to classification problems. Difficult to determine when training is complete. Need to make assumptions about the dimensionality of the output map.

The *RNN networks* are particularly suitable for univariate (single) time series forecasting as they store information about time, and they are a preferable choice of tool than MLP for this type of problem. However, MLPs are better for problems where prediction depends on both the time series in question and other related time series.

The *SOM network* has a single layer of units (figure 2-5) and during training the clusters of units become associated with different classes that have statistically similar properties which are present in the data.

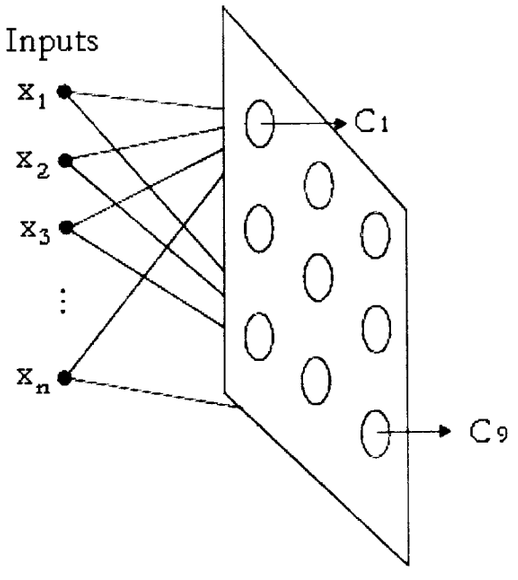


Figure 2-5. Example of a SOM neural network with nine neurons (classes).

This type of network is useful in applications where it is important to analyse a large number of examples and identify groups with similar features. They are particularly effective in applications where there is a severe imbalance in the number of examples from different groups that need to be identified. Typical examples are fault detection and risk analysis. The SOM network can also be used as a pre-processor for supervised learning to detect whether there is sufficient discriminatory information in a training set, i.e. selecting features as inputs, as well as for setting the centres of radial basis functions in an RBF network.

Another ANN type not yet mentioned is the *Auto-Associative Neural Network (AANN)* which is a particular type of MLP. Its structure is made of two MLP networks connected "back to back" (figure 2-6). Their main use is in data validation, fault detection and data compression.

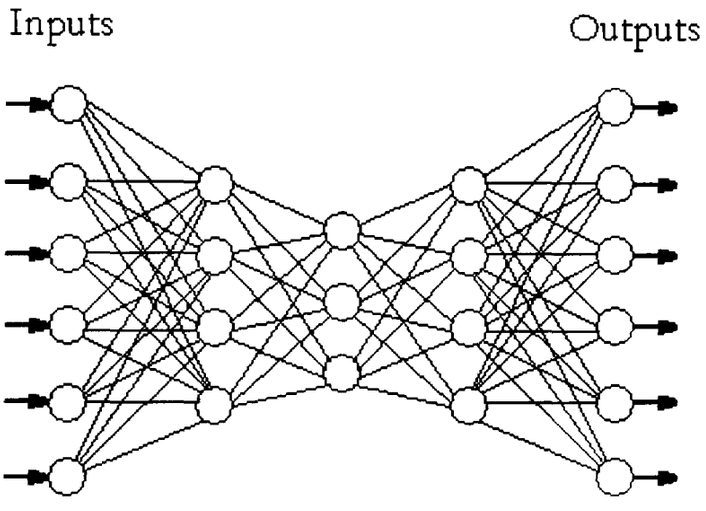


Figure 2-6. An example of an auto-associative neural network architecture.

These networks are trained with a target data set that is identical to the input data set. The key part in this network is the middle layer, which has a smaller number of units compared to inputs and outputs. The network learns the relationships within the input data and compresses them to a number of parameters equal to the size of the middle layer. Once the network has been trained, i.e. all output values match the input values, it can be used for data validation. When valid data is presented to the network, the output values will be close to the input values. If one of the input parameters is invalid

the outputs will be significantly different from the inputs and the difference can be easily detected. For data compression, the two last layers can be removed from the trained network, leaving the middle layer to produce the compressed version of the input data. This is equivalent to the nonlinear version of Principal Component Analysis (PCA).

2.2.3 Data selection and pre-processing

Regardless of the methods used to obtain forecasts, approaches that put more emphasis on meaningful data selection and pre-processing produce better results. This is particularly important for financial data, as the inner workings of the financial markets are hidden and the inputs can be/are obscure and difficult to select. The success of neural network applications in time series forecasting, as with any statistical modelling tool, depends heavily on the information within the data itself. This data needs to be pre-processed to a suitable form for use with neural networks. Pre-processing is needed in order to scale the data into a desired range, to reduce the number of inputs, remove noise, and for encoding textual (categorical) data. Examples of typical pre-processing can be simple noise filtering using moving averages or more complex tasks such as Fourier or Wavelets transforms. Care needs to be taken not to remove chaotic or nonlinear components from the time series by filtering, as these elements can improve model performance. Neural networks can also be used for the successful filtering of time series. Unlike polynomial and smoothing splines, they are easy to extend to multiple inputs and outputs without an exponential increase in the number of parameters. Another pre-processing concern is the handling of data outliers. Most financial models ignore them on the basis that volatility before and after a large shift in price is similar, but in practice a single outlier is often followed by several others in the same or opposite direction. By including these large values in the model without disturbing its properties, big losses can be prevented in volatile periods. Neural network models are highly nonlinear tools and they are capable of modelling extreme values. If these outliers disturb the property of the model, they could be scaled into a range, and then an extra variable added representing their presence (i.e., 0 for no outlier, -1 for the negative, and +1 for the presence of positive outlier). Another variable can also be added, indicating its size.

2.2.4 Limitations of neural networks

One of the main criticisms levelled at neural networks is that they cannot easily justify their answers. This "black box" problem has been addressed by extracting the rules from the trained neural networks. Commercial tools have already been developed to solve this problem. More trust may be gained by a better understanding of the representation of neural network outputs.

2.3 Genetic Algorithm (GA)

A genetic algorithm is a computational method modelled on the Darwinian natural selection mechanism, specifically the survival of the fittest. The basic principles of GAs were proposed by Holland [109], and further developed by Goldberg [110] and Koza [111]. GAs solve problems by removing less fit members from the population and selectively breeding the fittest ones that in the previous generation had found the best solutions[1]. The *breeding* procedure includes three basic genetic operations: *reproduction*, *crossover* and *mutation*.

2.3.1 The basic genetic algorithm procedure

The algorithm starts with the creation of an initial population (first *generation*) which is made of a number of randomly selected individuals. The fitness of each member is tested using an objective function known as fitness function. The population is also given an overall fitness rating based on the ratings of its members. The fitness value indicates how close an individual or population is to the solution.

A new generation is formed based on the fittest individuals of the previous generation. The *reproduction* process selects a fit member from the previous generation and passes it to the new generation without applying any change to it. The *crossover* process randomly selects two fit parents from the survivors' pool and recombines some of their characteristics (*genes*) in a random way to produce two new offsprings. Both reproduction and crossover operations pass only the existing information from generation to generation. To introduce novelty an additional operation, *mutation*, is used. Mutation generates an offspring by randomly changing the values of genes at one

---

[1] The individuals (candidate solutions) are usually represented as binary coded strings of fixed length. A position, or set of positions in an individual is called a *gene*.

or more gene positions of a selected individual. Though the mutations can cause some abnormalities to an offspring, they occasionally improve the fitness of the population.

Each iteration in a genetic algorithm creates a new generation. With a pure replacement strategy, the whole population is replaced by a new one. With an elitist strategy, a proportion of the population survives to the next generation. After dozens or even hundreds of 'generations', a highly fit population eventually emerges representing near-optimal solutions to the problem under consideration. The search is terminated after a number of evolution cycles (generations) are reached, when the amount of variation of individuals between two successive generations virtually disappears, or when a pre-defined value of fitness function is reached.

A flowchart of the GA operations described is shown in figure 2-7.

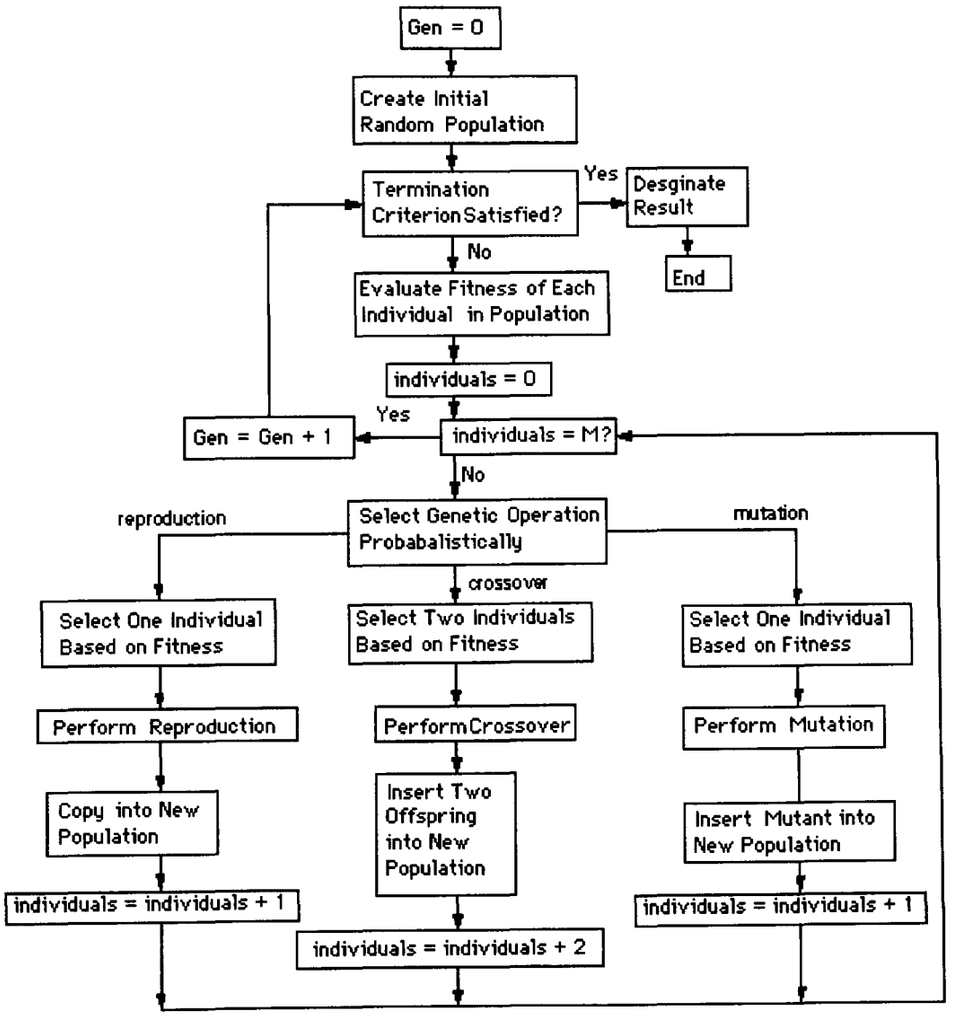


Figure 2-7. Genetic algorithm flowchart. Source [111].

The three operations, *reproduction*, *crossover* and *mutation* that produce a new population are applied according to a probabilistic schema. Each of them has its own corresponding probability values, and the sum of all three is equal to one. Therefore, the number of offspring derived from these operations is proportional to their probabilities respectively.

2.3.2 Genetic algorithm properties

GAs use a minimum of information about a problem, they only require a quantitative estimation of the quality of a possible solution. This makes them easy to use and applicable to most optimisation problems. Furthermore, they do not use an "exhaustive search" technique, meaning that every possible combination is tested to find the best one. For reasons of efficiency, the exhaustive search optimisers tend to limit the number of variables they use. The genetic algorithm, by contrast, does not try every possible combination and is usually much faster than exhaustive search algorithms.

GAs do not require a continuous or differentiable search space. "Newtonian" or "hill climbing" type optimisers often do impose such conditions. Exhaustive search algorithms can only work in continuous spaces if they are instructed to search in increments and thus they are more likely to become stuck in *local minima.*[1] This is less likely for GAs because they are searching many points in the search space simultaneously, and are not searching in increments (i.e. "hill climbing"). Since GAs don't search in orderly increments, they never know when they have found the optimal solution. Therefore, they use arbitrary stopping criteria when some number of generations pass with no better solutions being found, or when some amount of time has passed.

GAs cannot always be guaranteed to find the optimal solution because they are not trying every possible combination. However, trading models, estimated from the training set, usually do not require the optimal solution because it will most likely over-fit and generalise poorly.

Choosing the right population size can also be difficult, as a small population provides an insufficient sample size over the space of solutions for a problem, and a large population requires a lot of evaluation which makes its processing slow.

---

[1] A local minima is a solution that the optimiser can never get out of in order to find better solutions.

GAs can fail if the parameters they are given are too restrictive. Optimisation may end early if the genetic algorithm cannot reproduce, i.e. if there is not enough diversity in the optimisation parameter search space. This happens when too many parameter combinations yield the same result.

In general GAs are a good optimisation choice in complex problems with a large number of parameters, when the objective function is not smooth (i.e., not differentiable) and where there are multiple local minima.

2.3.3 Genetic algorithm application

Genetic algorithms have been found to be very powerful in solving optimisation problems that appear to be difficult or even unsolvable by traditional methods. They have been applied to the two main topics of this research: machine learning and financial applications. GAs were used to determine the best neural network topology (architecture) as well as optimising the learning algorithm parameters. The procedure examines a large number of networks and selects the one expected to have the best possible performance.

In finance GAs can be used in many innovative ways, for instance in portfolio optimisation, i.e. by minimising financial risk while simultaneously trying to maximise return. In this research they are mainly used to solve the following problems:

- Finding optimal trading strategies by finding the rules to predict a rise/fall of the financial security.
- Finding the best inputs to a neural network.
- Optimising technical indicators (used as the neural network inputs) by finding the best parameters for the particular trading period.
- Finding the best buy/sell thresholds in trading rules for neural nets.
- Finding the best parameters for indicators used in limit orders, stop orders, and stop limit orders.

2.4 Conclusions

This chapter has provided an overview of the structure and operation of neural networks and genetic algorithms and the reasons for their use in this research. More powerful multi-layered networks, combined with genetic learning algorithms are particularly suited to complex financial problems for which conventional statistical models are not suited. Their combination can dramatically reduce the complexity of the solution and produce more robust models. To be able to create such models we have to be certain that such a task is feasible. If the market fluctuates randomly then there is no space for predictions.

The next chapters will introduce fractal and nonlinear market analysis which will be used to study market properties that can be utilised and exploited by NN and GA tools.

Chapter 3

Fractal Market Analysis

3.1 Introduction

In the previous chapter, neural networks and genetic algorithms were described as two good candidate technologies for discovering underlying patterns and the dynamics of nonlinear financial time series. This chapter will outline the principles of fractal market analysis and its relevance to this research. The central theme of this research is based around the assumption that financial markets are nonlinear dynamic systems, generating prices that do not follow the random walk and the normal distribution. Both fractals and chaos[1] are hallmarks of nonlinear dynamic systems. The physics behind this is an area of intensive research in many branches of science. The primary focus of this chapter will be on the analysis of the qualitative properties of financial time series and testing their predictability, nonlinearity and complexity. Many financial time series show a presence of "memory" (long dependencies) when scrutinised carefully utilising the correct tools. Trends, fractal scaling and occasional erratic behaviour are all common features found in financial markets and are indications and characteristics of nonlinear dynamic systems.

Before going into more detail on this subject an introduction to the concept will be made, using simple examples that are relevant to real financial data.

3.2 Fractals

A fractal is a shape or pattern whose parts, property or structure is similar to the whole [53]. It has a special kind of invariance or symmetry that is present in the whole and its parts. They are found throughout nature, examples being clouds, lightning patterns, trees, human lungs and snail shells. Fractals are an important discovery because they change the way in which we analyse and understand experimental data. The notion that nonlinear dynamic systems create fractals will be discussed further ahead.

---

[1] Nonlinear behaviour that appears erratic and random.

Peters [17] gave a good explanation saying that they give structure to complexity, and beauty to chaos.

Until recently, scientists described nature using Euclidean geometry and the continuous mathematics of smooth forms made of lines, curves and planes. We tend to think in a 'three dimensional way', which is a gross simplification of the real, rugged character of nature, where most things are not three dimensional. Through the introduction of fractal geometry founded by Mandelbrot [46], [47], which deals with the ruggedness of structures, we are able to study jagged and irregular objects that have a fractal dimension, a numerical measure of "roughness".

3.2.1 The fractal dimension

A fractal dimension is the measure of how much a multi-dimensional space is occupied by an object. For example, a jagged line like a coastline has a fractal dimension of between 1 and 2, whilst a sheet of paper crumpled roughly into the shape of a ball has a dimension of between 2 and 3. The fractal dimension characterises how the property measured depends upon the resolution at which it is measured. It could be argued that the fractal dimension could be a better choice than volatility in assessing the "roughness" of financial time series. Essentially, it describes the changes in the variability of a measurement across a range of sample sizes in terms of a power law scaling relation. This could be compared to taking measurements of a coastline's length using different sized rulers and recording how the length changes as the ruler size changes.

In order to estimate the fractal dimension of a time series, methods such as rescale range analysis and the correlation dimension are often applied. These methods will be discussed in more depth in the following sections.

3.2.2 Randomness, determinism and fractals

In order to produce a fractal object, there must be a combination of local randomness and an element of global determinism. The first introduces innovation and variety, whilst the second represents the rule or hidden order that 'shapes' this stochastic process. It is amazing to find how the combination of these two properties in their simplest form can produce astonishingly complex structures. Figure 3-1 illustrates how

trees, that appear to be complex natural objects, can be generated easily by an iterated process using simple equations, whose constants are randomly drawn from pre-specified values found in a number of different sets. There is no way to replicate the similar tree structure using Euclidean geometry.

$$x_{n+1} = ax_n + by_n + e \qquad x_{n+1} = r\cos(\theta)x_n - s\sin(\varphi)y_n + e$$

$$y_{n+1} = cx_n + dy_n + f \qquad y_{n+1} = r\sin(\theta)x_n - s\cos(\varphi)y_n + f$$

Figure 3-1. Computer generated fractal trees[1],showing the power of fractals and chance working together.

It can be noted that each tree branch is different, but shares certain global properties, so that small branches are similar to bigger ones, and to the tree as a whole. This "self-similarity" is more obvious in symmetric fractals, like the Sierpinski triangle (figure 3-2). Self-similar fractals scale the same way in all directions. The fractals that scale more in one direction than another are called self-affine.

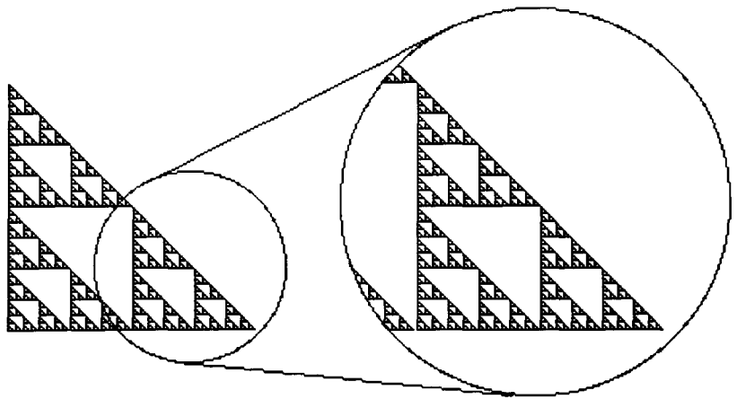


Figure 3-2. Self-similar fractal properties.

[1] Source: http://local.wasp.uwa.edu.au/~pbourke/fractals/

Barnsley [48] developed Iterated Function Systems for generating fractal shapes. Using a simple deterministic rule and randomness, a Sierpinski gasket can be generated by drawing a starting point anywhere inside the triangle, and iterating each following point half way to the randomly drawn triangle corner A, B or C. After 10,000 iterations a clear pattern emerges, as shown in figure 3-3.

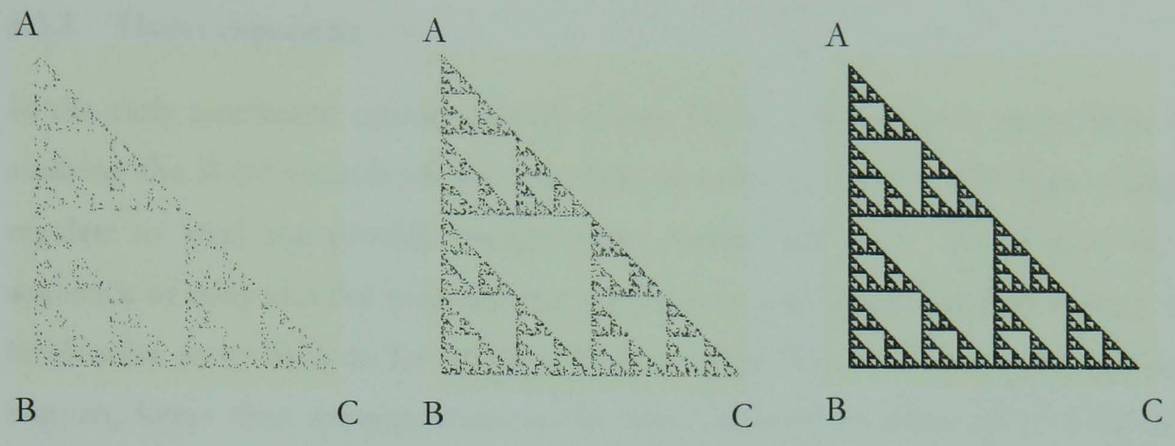


Figure 3-3. The random generation of a Sierpinski triangle.

Just by looking at these fractal objects, a repeating pattern can be recognised by the naked eye.

3.3 Fractals and financial time series

The two main elements of the system that create fractal objects, local randomness and global determinism, can both be found in financial markets. Randomness is generated by a myriad of traders and investor's actions. National and international rules and financial institutions' internal restrictions create global determinism. These restrictions can be of a legal nature or rooted in the investor's own investment strategy. As a result, financial time series are generated with a hidden order in the shape of probability distributions and fractal time. When looking at price charts without the identifying legend one cannot tell if the data covers a single day, week, month, year or decade. In order to acquire a better understanding of how real financial time series are generated, fractal shapes can be used to create time series that very closely resemble reality [16].

West and Goldberger [45] postulated that the fractal structure is nature's way of creating stable and more error-tolerant 'systems'. Power low scaling is a distinctive feature found in many of these systems. For example, the average diameter of the lung's

airways progressively scales down according to a power law. Peters [18] made an interesting analogy with this fractal structure, with power scaling, global determinism and local randomness as the cause of stability in the financial markets. This scaling was discovered by the Italian economist V. Pareto in the 19th century whilst using it to study population wealth distribution.

3.3.1 Hurst exponent

In the early nineteenth century Harold Edwin Hurst, a hydrologist, spent many years studying the flood records of the river Nile in order to build a dam high enough to regulate its level and provide enough water during dry years. He noticed that the sequence of rainy and dry years did not appear to be random; larger than average water levels were more likely to be followed by even larger levels. Then the reverse would happen, lower than average water levels were followed by even lower water levels. Following this, he devised the rescaled-range R/S statistics [40] to capture this effect. The method provides a relatively simple and robust procedure for studying anomalies in random events. It is a non-parametric test which makes no assumptions about data under investigation. The equation for calculating R/S is:

$$\frac{R_{(n)}}{S_{(n)}} = \frac{\underset{1\le k\le n}{Max}\left\{\sum_{i=1}^{k}\left(x_i-\bar{x}_n\right)\right\} - \underset{1\le k\le n}{Min}\left\{\sum_{i=1}^{k}\left(x_i-\bar{x}_n\right)\right\}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}_n\right)^2}} \tag{3.1}$$

The numerator[1] $R_{(n)}$ is the range from peak to trough in the summed deviations from the mean, and the denominator $S_{(n)}$ is a standard deviation of the data series. Hurst

---

[1] A breakdown of the steps required to calculate the $R_{(n)}$ numerator.

Calculate the mean over the entire data sample.

$\bar{x}_n = (1/n)(x_1 + x_2 + \ldots + x_n)$

Calculate the deviation from the mean for each data value.

$y_1 = x_1 - \bar{x}_n$, $y_2 = x_2 - \bar{x}_n$, ... $y_n = x_n - \bar{x}_n$

Calculate the running sums of the above differences up to each point in time series.

$Y_1 = y_1$, $Y_2 = y_1 + y_2$, ... $Y_n = y_1 + y_2 + \ldots + y_n$

The numerator range is $R_{(n)} = \underset{1\le k\le n}{Max}(Y_k) - \underset{1\le k\le n}{Min}(Y_k)$

found that R/S was governed by the power law found in many natural phenomena of the form:

$$\frac{R_{(n)}}{S_{(n)}} \propto Cn^{H} \tag{3.2}$$

where C is a constant and H is a Hurst exponent. In practice, the H is estimated by calculating the average rescale range over multiple sections of data[1].

$$E\left[\frac{R_{(n)}}{S_{(n)}}\right] = Cn^{H} \qquad \text{as } n \to \infty \tag{3.3}$$

The estimate of H is obtained from the slope of the regression line of R/S values against the window size n in log-log space.

$$\log\left(R_{(n)} / S_{(n)}\right) = \log(C) + H\log(n) \tag{3.4}$$

In his experimental study, using Monte Carlo simulation, Hurst found that for random, independent white noise processes of ordinary Brownian motion, H is equal to ½, therefore values different to ½ suggest the presence of long-term dependence. William Feller [44] mathematically proved that for Gaussian series the R/S statistic would increase in proportion to $n^{1/2}$ for large values of n. Note that the efficient market hypothesis assumes that H has a value of ½. In general H can take values in the range $0 \leq H \leq 1$, where $H > ½$ is indicative of persistent, trend reinforcing series with a positive long range correlation, while $H < ½$ suggests anti-persistent, mean reversion series with negative autocorrelation (figure 3-4). The further H is from ½, the stronger the indication of trend reinforcement or mean reversion and long-term dependence.

[1] Wavelets analysis is a novel technique which generally fares well in comparative studies [43].

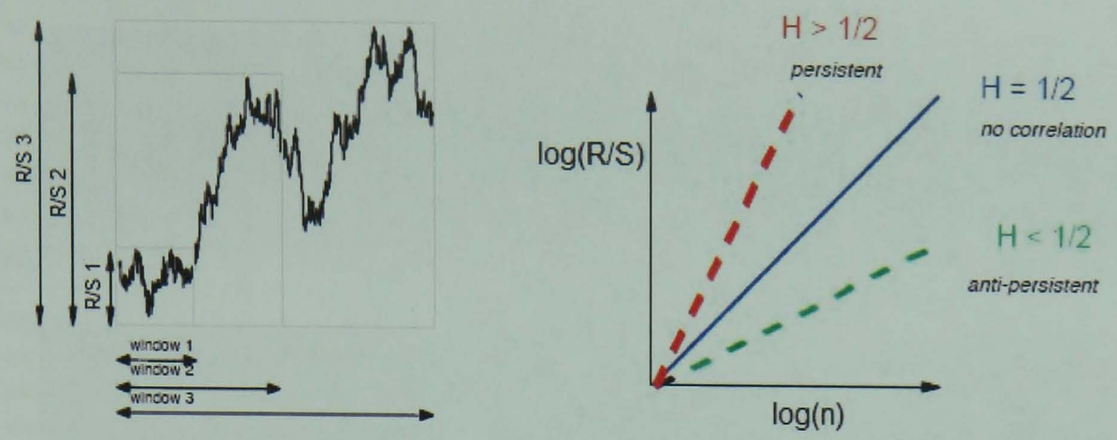


Figure 3-4. Graphical representation of the R/S analysis.

The Hurst pattern plays a crucial part in fractal analysis and has been observed in many financial markets. In the early seventies Mandelbrot [41], [42] used it in his studies of long-term dependence in economics and financial time series. The distribution of DJI returns, shown previously, mimics the distribution of data that Hurst found in water levels. There are many small variations around the mean, such as common market rallies (river floods) and market crashes (river droughts). The sequence of these events is of even greater significance than the variations themselves in causing long-term correlations that cannot be observed using standard statistical tests. Due to this long-term memory, the past continues to influence present fluctuations. It can take anything from days to decades for this effect to fade away.

Figure 3-5 shows the Hurst exponent*[1] calculated for the Dow Jones Industrial index (1930-2007) for rolling windows of different length. The sensitivity to the number of points used is clearly visible from the graphs. The more data used, the better an estimate of H^* can be achieved. For a window of short size, the H^* value is overestimated.

[1] Note that the Hurst exponent* in this case is an estimate for a single particular window length and is not constant. The real H exponent is a constant, representing the slope of the regression line between $\boldsymbol{R/S}$ values plotted against the window size $\boldsymbol{n}$ in log-log space.

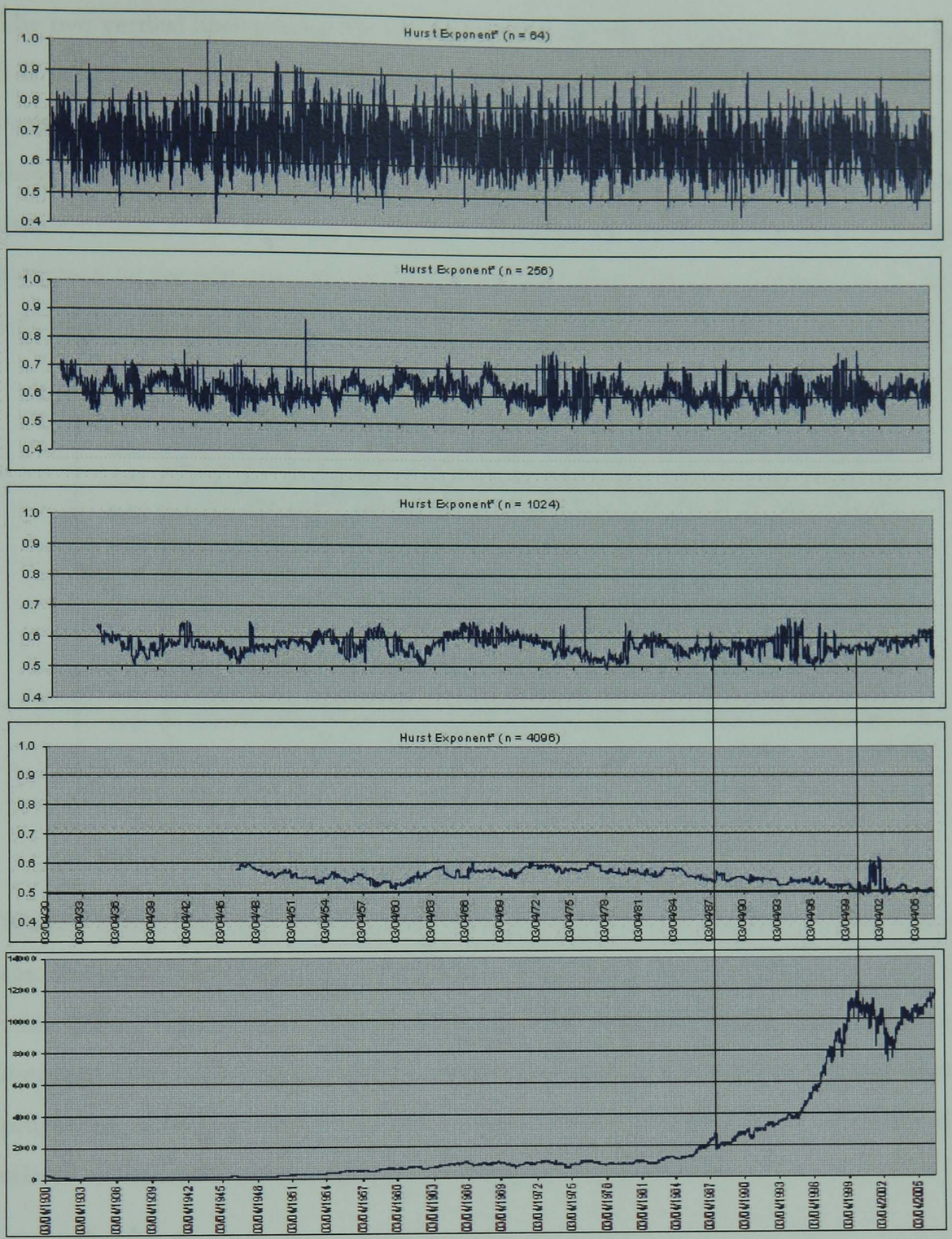


Figure 3-5. The Hurst exponent* calculated using daily data from the DJI (1930-2007) on a rolling window of different data points. The graph of the DJI index is shown at the bottom.

From the last plot (window size n=4096) we can extrapolate that the value of *H* declines from 0.6 towards the 0.5 mark, which could be an indication of the market becoming more efficient and random. The Hurst exponent is less volatile and drops off asymptotically as the size of the window data is increased to reach a 'real' value (figure 3-6).

The two vertical lines joining the DJI index plot and the H exponent plots in the figure 3-5, indicate the times just before the market crashes of Oct 1987 and Jan 2000. In both cases the H exponent declined just before the crash and rose after the crash (for a short time in the case of the low resolution (n=4096) plot and for a longer duration/time in the higher resolution (n=1024) plot). These low values of H could be explained by the nervous state of the market just before dramatic changes occur. In this type of unsettled market, the correlation could be lower resulting in low values of H.

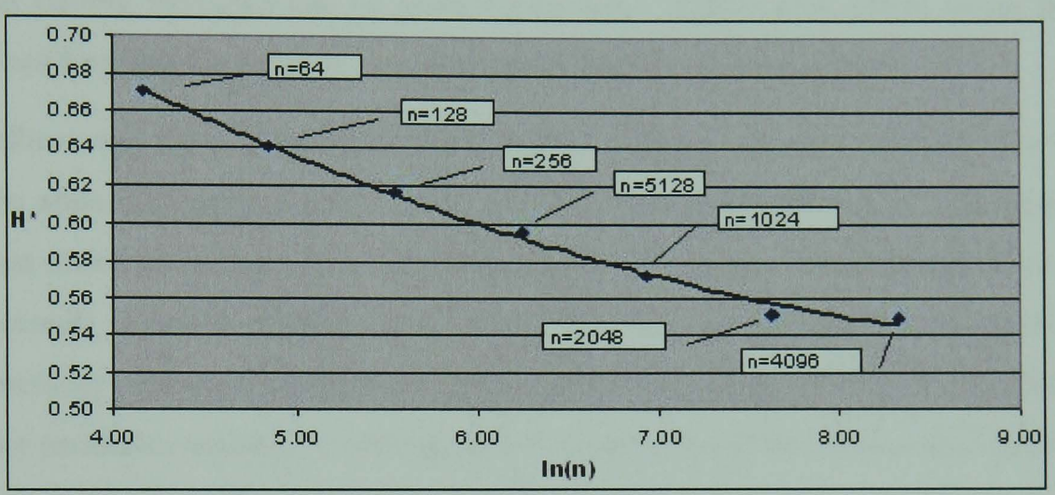


Figure 3-6. The Hurst exponent plot against the natural logarithm of data set size.

We saw earlier that fractal objects exhibit 'self-similarity'. In the case of time series, we can recognize fractal characteristics when we find that certain properties do not change with different time scales. In the case of R/S analysis, the Hurst exponent is the invariant value. It is estimated from varied sized data sets which have similar statistical properties, a fact confirmed by the constant Hurst exponent. The H exponent is related to the fractal dimension by way of the following relation:

$$D = 2 - H \tag{3.5}$$

The fractal dimension is an important concept in fractal geometry and represents the measure of roughness of an object or, in this case, of a time series. It is also widely used in nonlinear chaos theory as an estimation of the complexity of the system under investigation.

The next section will investigate whether the Hurst exponent can be used to gain a statistical edge in forecasting stock price time series.

3.3.2 The Hurst exponent as the measure of predictability

It has been widely reported that most economic and financial time series are persistent and trend reinforcing with $H > ½$. The persistent changes reinforce each other causing a trend (up or down) that once started tends to keep going. This property could be exploited by trend following trading strategies that are very popular amongst many traders. On the other hand, for time series with $H < ½$, their anti-persistent changes contradict each other and any trend started is likely to reverse itself. In that case price reversal trading strategies can be used successfully. Either way, prices series are not purely random and they may have at least some degree of predictability.

Mandelbrot sees these types of technical trading systems as fool's gold [53]. He claims that the apparent patterns observed in financial time series are typical occurrences in data that scales according to a power law and that they cannot be predicted. Events can unexpectedly appear to form patterns and cycles that are the properties of most long-term memory processes in which seeming patterns appear and disappear and cannot be used for profitable trading. However, in his seminars he admits that several individuals claim to use his research to trade successfully.

The two views put forward in the last couple of paragraphs appear contradictory. If time series exhibit long lasting trends, a simple trend following strategy such as a moving average crossover one would be profitable. Trending strategies enter positions after a trend has developed and tend to go with the momentum. A similar approach can be made in the case of oscillatory time series where price reversal trading strategies can be applied. Reversal strategies enter positions when a price move is about to occur, or when a new trend is about to start. They try to anticipate a change in momentum toward the long term trend. The main challenge is knowing when it is most advantageous to apply the trend following or the trend reversal strategy.

One way to test this is to use the Hurst exponent to classify financial time series, and accordingly specify the most suitable trading strategy. The idea is to apply two basic trading strategies, the trend following and the reversal trading strategy to a number of stocks and use the Hurst exponent as a trending/reversal indicator to switch between these two strategies accordingly. If the combined strategy performs better than either individual one it would indicate that the use of the Hurst exponent gives an extra edge

and predictive power. In the case of the trend following strategy, a simple Moving Average (MA) crossover trading strategy was used, and for the reversal strategy, the Relative Strength Index (RSI) was used. Both of these strategies can be utilised in either trend following or reversal systems depending on the choice of parameters used. Using shorter parameter periods would track short term trend changes and act as a reversal system while the use of longer periods would mimic the trend following strategy. The five following systems were tested: *MA system*, *RSI system*, *Expert System 1*, *Expert System 2A* and *Expert System 2B*.

The *MA system* is a basic crossover trading system where the buy/sell signals are generated when the fast[1] moving average crosses above/below the slow moving average as shown in figure 3-7.

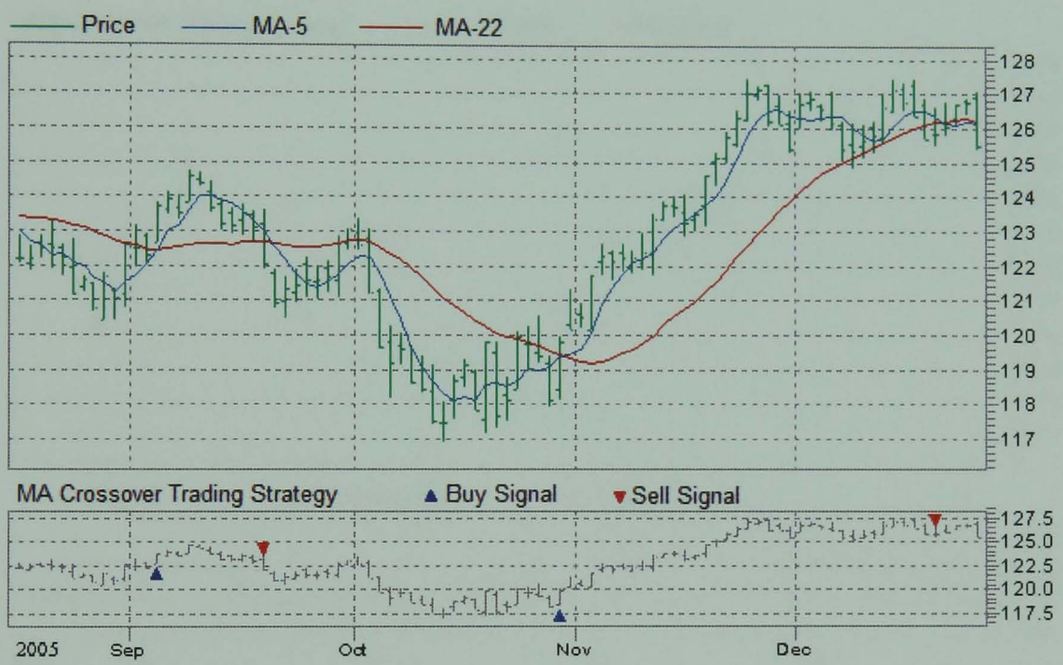


Figure 3-7. An example of the MA crossover trading strategy. The buy/sell signals are generated when the fast moving average (blue line) crosses above/bellow the slow moving average (red line).

The Moving Average (MA) crossover system is used to determine the start of an uptrend/downtrend. Comparing a short period MA to long period MA provides a means of measuring how short term prices fluctuate compared to longer term prices. A short term MA crossing above a longer term MA indicates a possible up trending market and the opposite applies for a downtrend market. In this case, the MA crossover trading strategy enters a long position when the 5 day MA crosses above the

[1] The fast moving average uses less periods (days) in calculations, whilst the slow one uses more periods.

22 day MA and the sell signal is generated when the 5 day MA crosses below the 22 day MA. As long as the time series has long trends, this simple trend following trading strategy will be profitable. Many frequent reversals will cause this strategy to fail, due to late buy/sell signals introduced by the time lag in moving averages.

The *RSI system* is based on an indicator representing a measure of investor sentiment, and as such is used to forecast turns in the market. The RSI indicator oscillates in the range of 0 to 100 based on changes in price over time, but it typically takes up a value of between 20 and 80. It is a normalised ratio of the sum of the up-moves to down-moves. The closer the index is to 100, the stronger the indication of an overbought market. The closer the index is to 0, the stronger the indication of an oversold market. Figure 3-8 shows an example of the RSI crossover trading strategy.

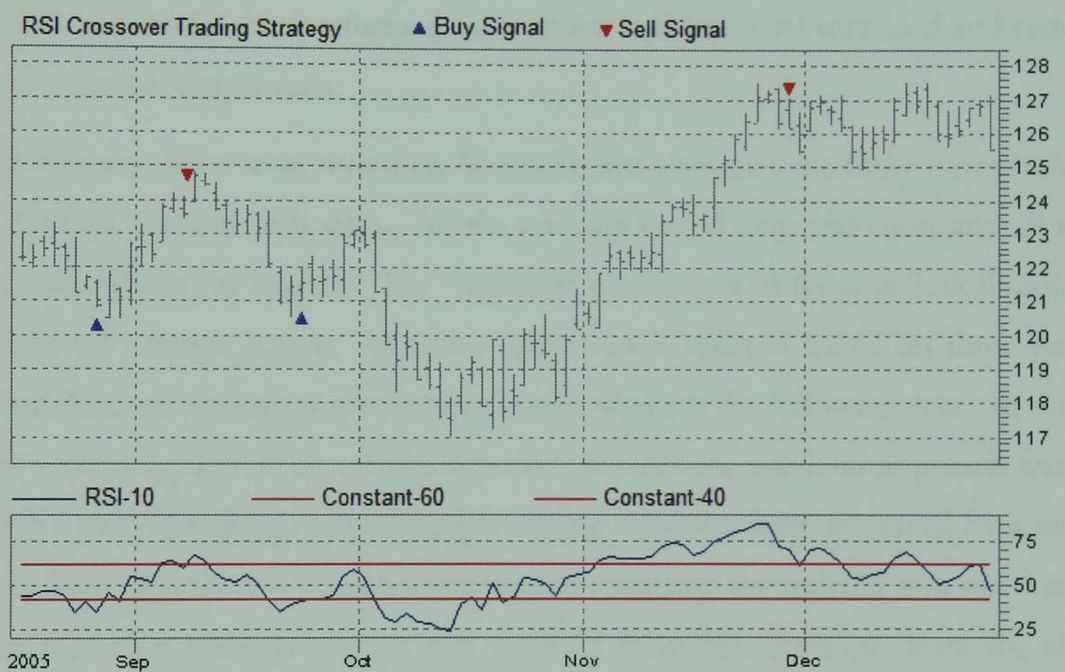


Figure 3-8. Example of the RSI crossover trading strategy. The buy signal is generated when the 10 day RSI crosses above 40 and the sell signal is generated when the RSI crosses bellow 60.

The traditional concept behind the RSI as a trading strategy is to signal the purchase of a security when the RSI crosses above 30 and to sell when the RSI crosses below 70. However, the interpretation of the RSI indicator depends on the particular time series characteristics. It can be noted that the signals generated in this example are closer to the turning points. There is no lag, as in the case of the MA system, indicating that this system is more suitable for a time series that has many reversals. The specified parameters shown in the MA and RSI systems are only there for illustration purposes;

the actual parameters utilised are found using a genetic algorithm for each individual stock and trading period.

The *Expert System 1* is a combination of the two systems above. It uses the Hurst exponent of 0.7 as a fixed threshold value to switch between the *MA system* signals and the *RSI system* signals. If the average H^* value[1] of the time series is above 0.7 then MA *system* trading signals were used, otherwise *RSI system* trading signals were used.

The *Expert System 2A* is essentially the same as the *Expert System 1,* except that the H^* threshold value is not fixed. A genetic algorithm was used to find the optimal value for H^* threshold value as well as for the other indicators' parameters.

The *Expert System 2B* is the same system as the *Expert System 2A* except that the genetic algorithm used the validation set in addition to the training set to find the best model. The optimised model that performed best on the validation set was used and tested on one year's out-of-sample tests.

All five systems above were tested on 30 stocks that comprise the Dow Jones Industrial index over 10 years of daily data. All the systems traded long and short and all stocks were continually in the market[2]. The "optimal" parameters of these technical indicators were selected using a genetic algorithm optimisation method based on three years of historical data and out-of-sample tests were recorded for the following year. In order to test the performance over an extended period, the moving windows approach was used in which a pair of training/out-of-sample testing windows were advanced by a number of years. This process was repeated ten times by shifting the training and test windows by one year at a time, producing ten years of long out-of-sample tests for all five systems. All trading tests presented in this research use a similar moving window training/test approach which is shown schematically in figure 3-9.

---

[1] The H* was calculated as the average of three values, estimated using rolling window sizes of 30, 60 and 90 days.

[2] All stocks were always either in long or short position.

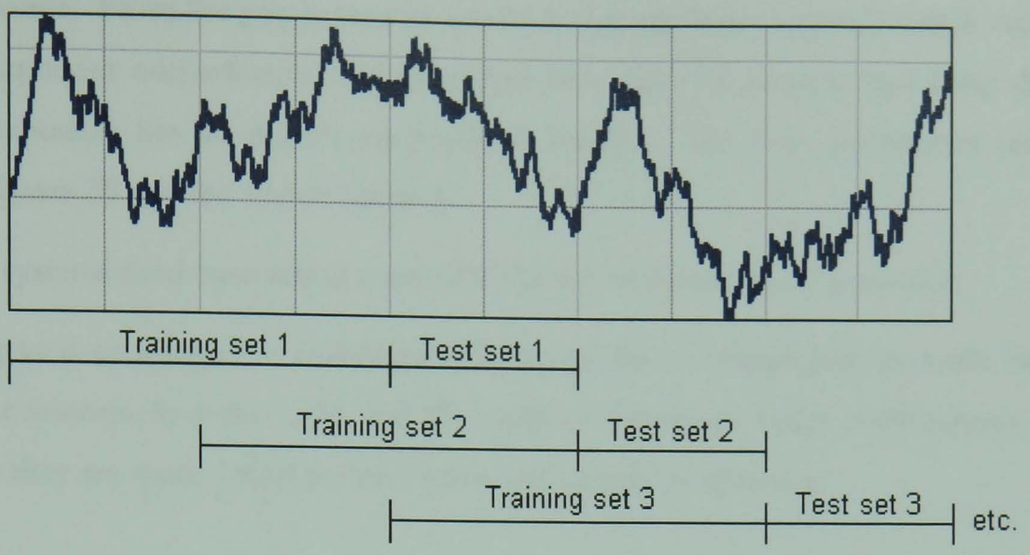


Figure 3-9. Training and out-of-sample tests illustration.

The advantage of using a moving window approach over using a single training/test cycle one is that it allows for the fact that the prediction model may change over time. Training strategies that were optimal in the past may not be optimal when projected too far into the future.

The performance of all five systems, compared to the buy and hold strategy, is presented in figure 3-10.

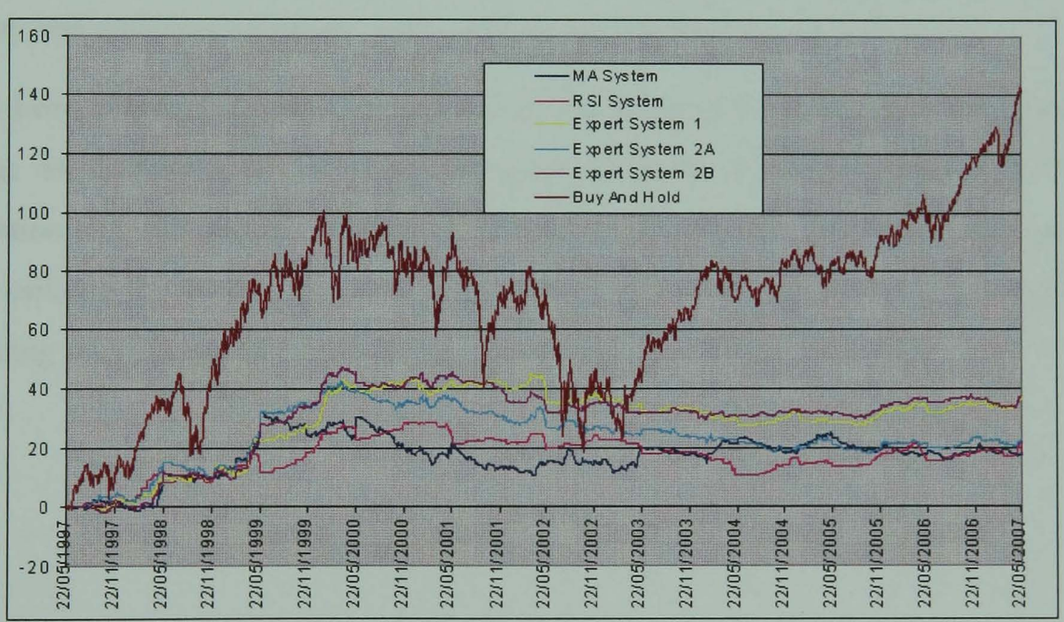


Figure 3-10. Systems performances utilising Hurst exponent.

It should be noted that this test was not performed with the aim of creating a good trading system; it was only used to compare the relative performance of these different trading strategies and to see if the use of the Hurst exponent can improve their

performance. From the graph, we can conclude that all three 'Expert' models, using the Hurst exponent outperformed the individual MA and RSI models, indicating that the Hurst exponent has an impact on portfolio returns. The best performers were the *Expert System 2B* and the *Expert System 1*.

All five systems used transaction costs of 0.1% per trade and $0.01 per share.

Coming to a conclusion is somewhat difficult in that a comparison is made between different systems. Systems 1, 2A and 2B could be showing a better performance simply because they are more complex/have more parameters to optimise.

3.4 Conclusions

This chapter presented the main properties of fractals and their relevance in finance. Characteristics of nonlinear systems such as fractal scaling, trending and erratic behaviour are common features found in financial markets. The two main elements of the system that create fractal objects, local randomness and global determinism, can both be found in financial markets. Randomness is generated by the actions of many investors, whilst national and international rules and financial institutions' internal restrictions create global determinism. Fractal structures and power scaling cause stability and resilience in nature, and could be said to do the same in finance.

It has been shown that the Hurst exponent can be used as a measure of predictability and as an indicator in switching between different trading strategies. A trading simulation test was applied to 30 constitutes of the Dow Jones Industrial index, over ten years, utilising the Hurst exponent. It has been found that this strategy performs better, indicating that the use of the Hurst exponent improves predictive power.

Chapter 4

Chaos Theory and Finance

4.1 Introduction

This chapter describes the essential properties that lie behind chaotic nonlinear systems and their application to finance. Both fractal analysis and chaotic systems are used in this research to gain a better understanding of the financial markets, and the findings are consequently used in developing AI based trading models. Various calculation techniques for the estimation of parameters required for the reconstruction and analysis of time series are presented and consequently used to determine whether the time series are of stochastic, deterministic, or chaotic nature.

Historically, nonlinear dynamic systems known as deterministic chaos have been studied in the physical sciences but they have recently been receiving a great deal of attention in the field of finance and economics. Chaotic processes are typically produced by nonlinear feedback systems and analysed in so-called state-space[1], which is a dynamical plot of variables that defines the system. A low dimensional[2] chaotic system can mimic stochastic behaviour and can produce data that may look random and unpredictable, though there is a clear deterministic formula behind it. This chapter will investigate the presence of the chaotic deterministic components that may be buried deep in noisy real-world financial time series. Before an attempt is made to model and predict such series, tests will be performed on a logistic equation; a simple and well-known chaotic nonlinear system that is used to model population growth. This simple nonlinear equation can generate complicated time series that cannot be modelled by linear approximation and a more general framework is required. It was impossible to model and predict the behaviour of similar systems until the emergence of nonlinear models such as neural networks that can adaptively explore a large area of potential models.

The characterisation methods that can extract some of the essential properties which lie

---

[1] Also known as phase-space modelling.

[2] A system with a few degrees of freedom, i.e. the system can be described with small number (less than ten) of differentiable equations.

behind chaotic nonlinear systems will be demonstrated and used as a guide for further analysis and modelling. Most tests in the following sections will be conducted on four time series; the artificially generated logistic map chaotic time series and the three pre-processed variants of the Dow Jones Industrial (DJI) index.

4.2 The logistic equation

The logistic equation was discovered in 1845 by the Belgian mathematician Pierre-Francois Verhulst and was used to model the growth of populations. It is probably the simplest representation of a nonlinear dynamical system capable of complex chaotic behaviour, and is described by the following formula:

$$x_{(k+1)} = \lambda x_k (1 - x_k) \tag{4.1}$$

The formula is very basic, but it can exhibit unusually diverse regimes and provide some clear guidelines and resemblance to the behaviour of financial markets.

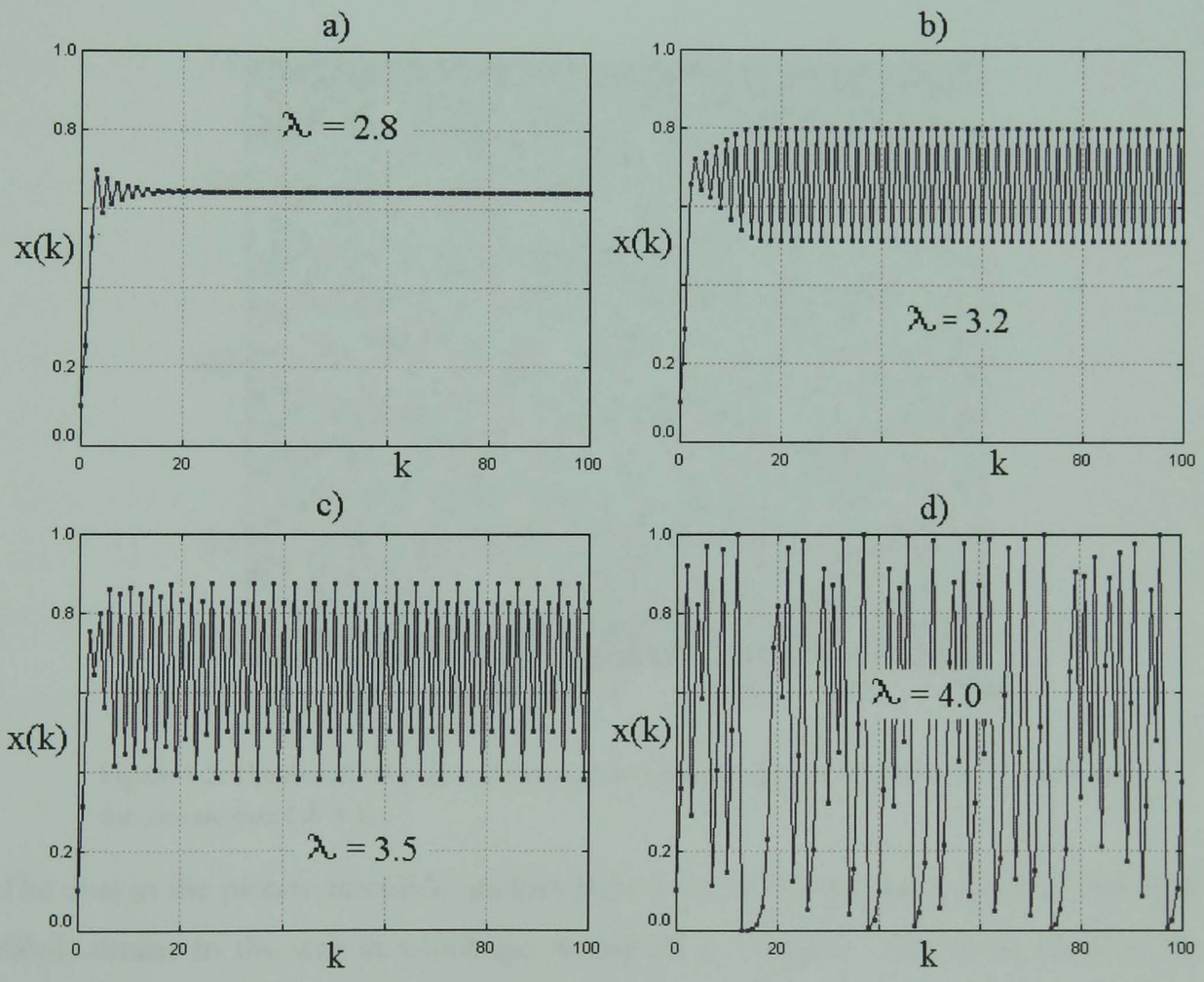


Figure 4-1. The logistic map, first 100 points using initial value x(0)= 0.1and different λ values.

These regimes show stable, bifurcating, intermittent and completely chaotic behaviour that will be described in later sections. The transitions between them are controlled by a single growth parameter λ. Figure 4-1 presents the dynamic of the logistic equation created using different values for the λ parameter.

For the value of $\lambda = 2.8$ (graph a.) the process settles to a unique stable equilibrium[1]; for $\lambda = 3.2$ (graph b.) the system oscillates between two states; for $\lambda = 3.5$ (graph c.) the system oscillates between four states; and for $\lambda = 4$ (graph d.) the system has an infinite number of solutions, it is aperiodic, appearing random and unpredictable, but in fact it is completely chaotic, unstable and deterministic at the same time. In the case of a), b), and c) the system is globally stable and will converge to one or more steady state solutions irrespective of the initial starting value. However, in case d.), a small perturbation of the initial condition will cause it to move widely at an unpredictable future time. In this chaotic state, the processes may appear as pure noise, as shown in figure 4-2.

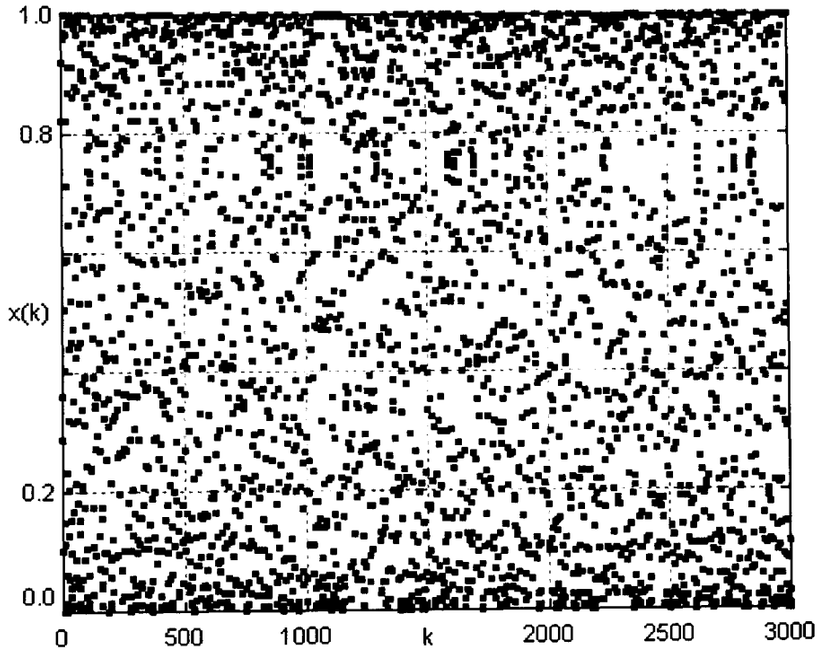


Figure 4-2. The first 3000 points of the logistic equation appear to be randomly distributed for the chaotic case ($\lambda = 4$).

The dots in the picture resemble random behaviour; the whole space appears uniformly filled, similar to the way in which gas would fill a container. But if we examine the distribution of dots closely, there are noticeably more dots closer to the lower and upper

[1] A single solution.

bounds. This is clearly visible in the frequency distribution diagram (figure 4-3).

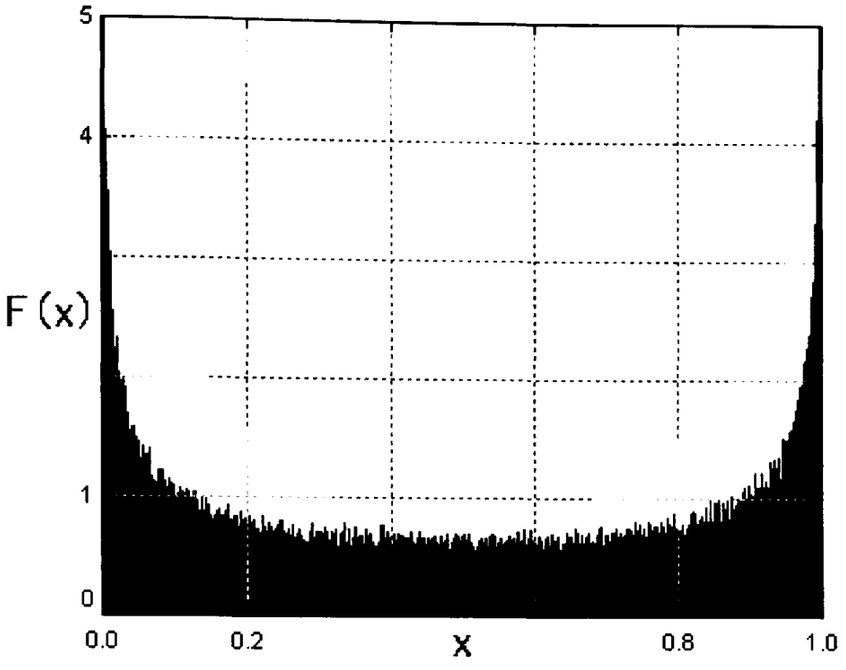


Figure 4-3. Frequency distribution diagram of the logistic map, $\lambda = 4$, x(0)=0.1, 100000 iterations.

The frequency distribution contains fat tails, and is not flat, as would be the case with uniformly distributed white noise. It is an example of stable distribution. The process is entirely non-stationary; the mean and variance are undefined, and they keep changing constantly as we add more observations. The chaotic system does not have a steady equilibrium state; in fact there are indefinitely many states. This is also visible from figure 4-4, showing steady equilibrium solutions of all systems defined by the $2.8 \leq \lambda \leq 4$ parameter range.

Three different regions are distinguishable: the region of convergence for parameter values less than 3, the periodic region for parameter values in the interval [3, 3.56], and the chaotic region for values in the interval [3.57, 4].

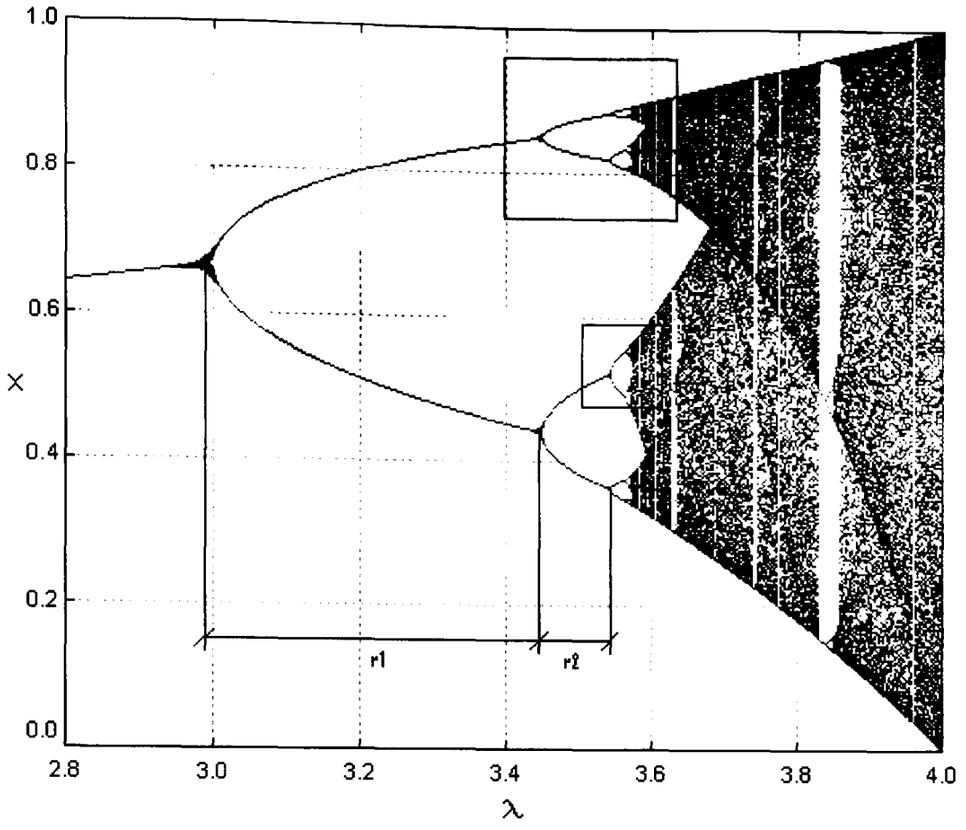


Figure 4-4. Logistic map bifurcation diagram.

We can see that the map branches into two, then four, then eight and so on. The sequence follows a geometric progression, but soon looks a mess. These transitions, through a period doubling mechanism, which often lead from order (determinism) into chaos, are called bifurcations. Bifurcations follow a pattern, occurring closer and closer together to infinitum. The right hand side of the picture shows the chaotic regions of many solutions that are cyclically interspersed with clear "windows of stability" with a small number of solutions. This alternation between stability and chaos is called intermittency.

We can also observe from the logistic map bifurcation diagram that chaotic systems show self-similarity and fractal patterns. The big picture and the two squares within the picture are self-similar, being the same map at different scales. The relative separation between the bifurcations is a constant value known as a Feigenbaum constant.

$$F = \frac{r1}{r2} = \frac{r2}{r3} = \frac{r_n}{r_{n+1}} = 4.669201... \tag{4.2}$$

The time series produced by chaotic systems are highly sensitive to initial conditions. Even slight changes in the initial values can produce very different outcomes and make

long term predictions impossible. In the case of the logistic equation this is visible after only twenty iterations. Figure 4-5 shows two paths, starting from two initial values that differ only in the sixth decimal place.

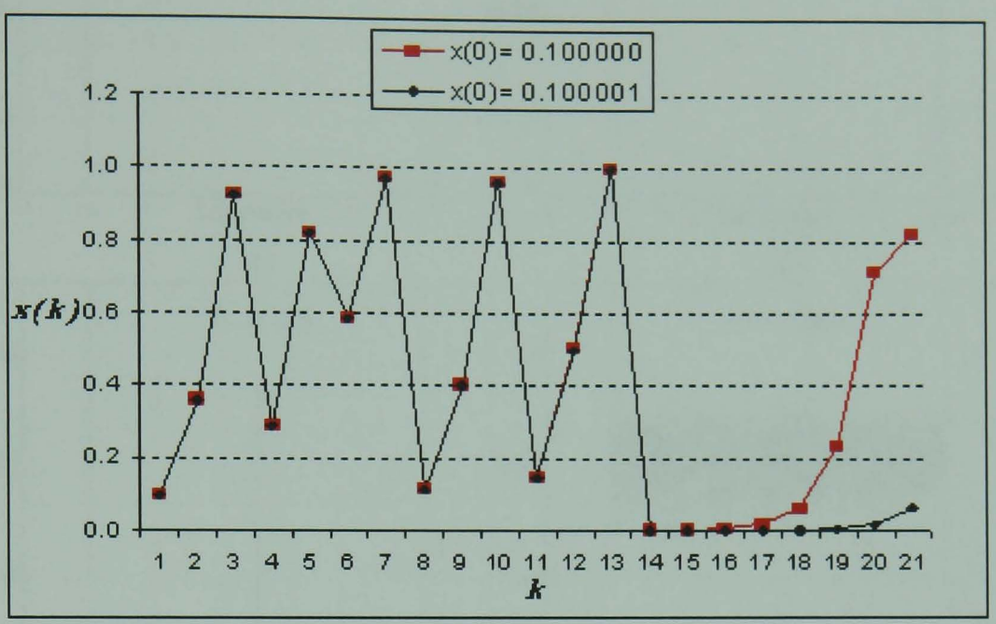


Figure 4-5. Logistic map sensitivity to initial values, $\lambda = 4$.

Often, time series characterisation is carried out in a frequency domain which is applicable to simple and linear problems. For such systems spectral (Fourier) analysis is useful, and it is possible to derive the number of degrees of freedom of the system from the number of peaks (modes), see figure 4-6 a), b) and c). However, these conventional methods are ineffective for nonlinear chaotic processes as they have a continuous broadband Fourier spectra instead. As the parameter λ is increased, more and more sub-harmonics appear until deterministic chaos is reached. This implies that filtering by frequency cannot be applied to discrete chaotic time series since a signal and noise have similar spectral properties.

This is illustrated in the figure 4-6 d).

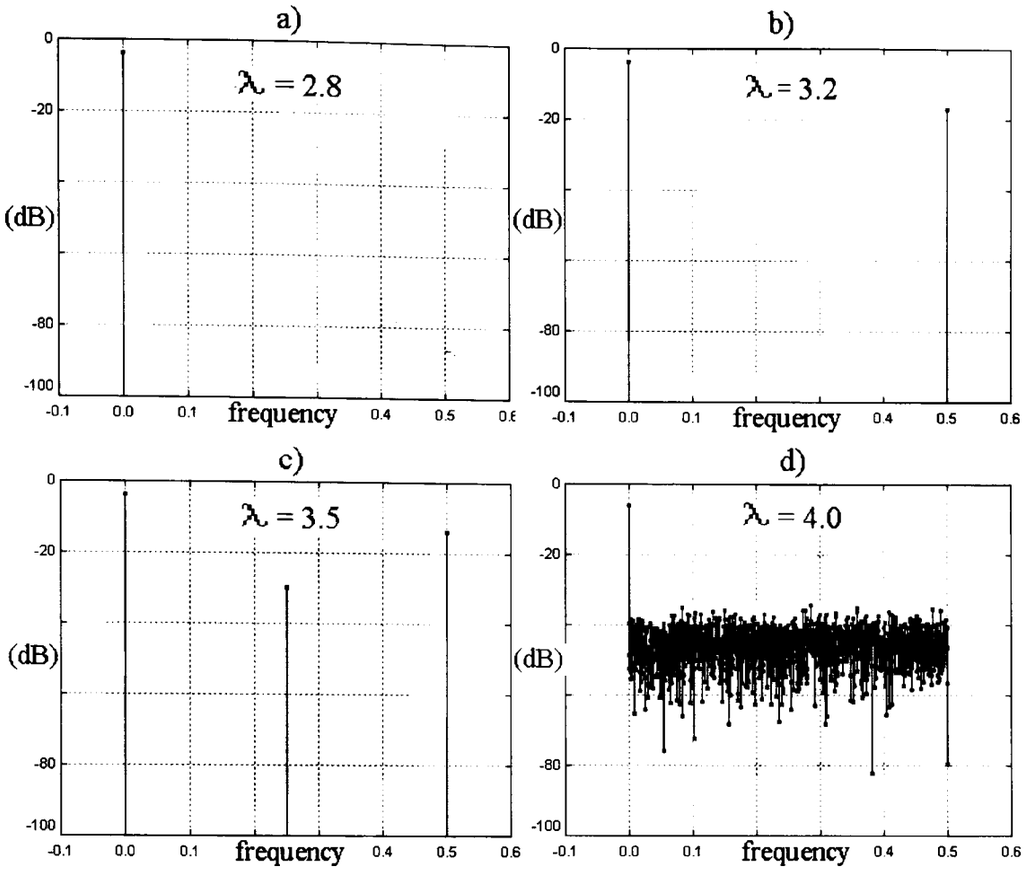


Figure 4-6. Logistic Map Fourier Analysis; initial value x(0)=0.1; discard first 1000, display next 3000 iterations.

In addition to linear tests, such as spectral analysis and autocorrelation functions, various other methods have been developed for nonlinear time series characterisation. These include state-space[1] embedding, calculation of correlation dimension D, Lyapunov exponents, entropies and mutual information. These techniques can discover regularities that are completely hidden in the time series and are not detectable to conventional analysis, as in the case of Fourier transform. These properties characterise useful concepts of the underlying system that have practical implications and will be discussed in more detail in following sections.

4.3 State-space embedding

The basic idea behind state-space embedding is that we can use the current and the lagged values of a single time series, which is produced by a multidimensional chaotic system, to reconstruct the dynamics of that underlying process without knowing the equation of motion. The idea may seem farfetched, but it has been proven mathematically, by Takens [56], that it is possible to reconstruct the entire dynamics of a

[1] Also called phase-space embedding.

chaotic system from a relatively small number of observables[1]. This is because the combined effect of many degrees of freedom of a nonlinear dynamical system are buried in the observable time series through the main state vector components. So we need to embed a univariate time series into a sufficiently high-dimensional space of delay coordinates in order to recover the full geometrical structure of a nonlinear system. The number of minimum lags required to model the underlying process is called *embedding dimension*, and the time lag is termed embedding separation or *time delay*.

Given a time delay τ and embedding dimension m, the reconstructed state space is obtained from a delay coordinate vector:

$$\vec{x}_t = \left(x_{t-(m-1)\tau}, \ldots, x_{t-\tau}, x_t\right) \tag{4.3}$$

The model is defined as a function of the delayed vector:

$$y_t = f(\vec{x}_t) \tag{4.4}$$

It has been shown earlier that a chaotic time series looks random when plotted in time domain. However the same series shows a clear pattern when viewed in state-space plot, which removes the time element. For two and three dimensional systems we can visually inspect the data through a state-space portrait of the system. The higher dimensional cases can be also explored using applications specifically developed for multivariate data visualizations[2]. The two dimensional case is illustrated for the Logistic Map by plotting the pairs of points $x_{(t-1)}$ and $x_{(t)}$ shown in figure 4-7.

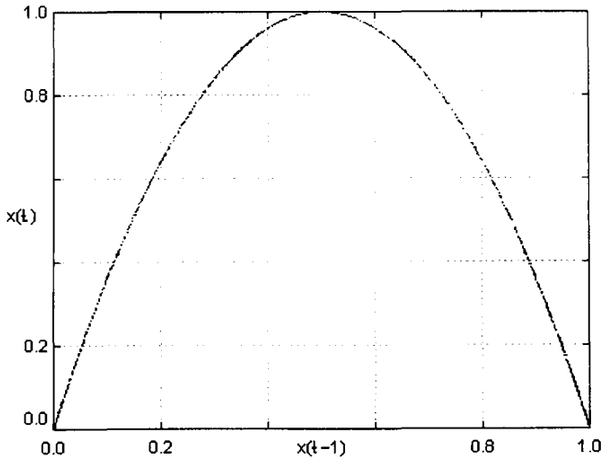


Figure 4-7. Logistic equation state-space plot.

[1] Note that Takens' theorem assumes a purely deterministic noiseless system. Before the Takens proof Ruelle [58], [65] and Packard et al. [59] used embedding for analysing chaotic time series.

[2] A good example is XGobi, freely available from http://www.research.att.com/areas/stat/xgobi/

The well defined structure of a reconstructed state space becomes visible in the geometric shape of a parabolic curve. However, the position of each point is entirely irregular, jumping all over the curve in a random manner, with no point visited twice. The geometric shape, or state space trajectory that dynamics evolve toward, is common to chaotic systems and is denominated the *strange attractor*[1]. This shape is the equilibrium of the system, which is dynamic and constantly changing, but is bound to a region defined by the attractor. The cycles of the attractor are non-periodic and their trajectories in state space never intersect. Often chaotic attractors are self-similar and have a fractal Hausdorff dimension[2].

A state space trajectory that is properly embedded will have the same dimension as the attractor of the underlying system. As we increase the embedding dimension, the dimensionality of the attractor should not change, as long as the embedding dimension is higher than the attractor's. This is because the attractor's points are correlated and clamped together.

The time series embedding approach can be useful in the modelling of complex systems such as financial markets that are influenced by a large number of factors. In such applications, it is fundamental that the embedding dimension and time delay are chosen appropriately. The system of heuristics developed for detecting nonlinearity and chaos in time series, as well as for calculating the suitable embedding dimension and an adequate time delay will be explained in following sections.

4.4 Estimating the embedding dimension

The embedding dimension should be chosen sufficiently high to capture the complexity of the system and influence the main participating variables. Using a dimension larger than the minimum required by the data will lead to excessive requirements in terms of the number of data points and computation time. Furthermore, the noise by definition has an infinite embedding dimension so an excessively large value would inflate the noise to the level of masking the true signal of the process. If the dimension is set too small, the dynamic system in question will be under-determined.

---

[1] They are also called fractal or chaotic attractors.

[2] Two characteristics mentioned previously, local randomness and global determinism that jointly produce fractal shapes are demonstrated here. The local uncertainty of where the point at any time is bound by the global shape of the parabolic curve.

Several statistical techniques have been developed to estimate the dimensionality of a system that generates an experimental time series. One way to estimate it is to use the Akaike Information Criterion (AIC) [57], but this measure relies heavily on linearity assumptions. For nonlinear systems, methods based on box counting, correlation sum techniques [62] or k-nearest-neighbour distances [61] are preferable.

4.4.1 Correlation dimension analysis

The method that is often used to estimate the optimal embedding dimension is through the calculation of the correlation integral, which counts the number of points enclosed in a hyper-sphere that is centred in a state-space, whilst letting the radius of the hyper-sphere grow until all points are enclosed. The correlation integral $C_m(\varepsilon)$ is calculated for the range of dimensions m using the following equation:

$$C_m(\varepsilon) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{i,j=1, i\neq j}^{N} \Theta\left(\varepsilon - \| \vec{x}_i - \vec{x}_j \|\right) \tag{4.5}$$

Where: $\Theta(\varepsilon, \vec{x}_i, \vec{x}_j) = 1$ if $(\varepsilon - \| \vec{x}_i - \vec{x}_j \|) > 0$; 0 otherwise[1]

N is the number of observations.

ε is the distance (scale length)

The $C_m(\varepsilon)$ value represents the probability that two randomly selected points in state space are less than ε units apart. As the value of ε is increased, the $C_m(\varepsilon)$ will grow at the rate of ε^D.

$$C_m \propto \varepsilon^D \quad \text{or} \quad \log(C_m) \approx D\log(\varepsilon) \tag{4.6}$$

The slope of the linear regression line produced by plotting $\log(C_m)$ against $\log(\varepsilon)$ will estimate the correlation dimension[2] D of the reconstructed state space for the embedding dimension m. If we calculate the correlation dimension D for the number of successive embedding dimensions m we can distinguish three important cases:

a. If with the increase of the embedding dimension m, the estimate of the correlation dimension D increases to infinity or to an extremely large value, then

---

[1] $\Theta(\cdot)$ is known as the Heaviside Function.

[2] A non-integer value for the correlation dimension is called a fractal dimension.

the time series is Gaussian and random. In this case, there are no predictable variations around a trend and any forecasting attempt is bound to fail. A reconstruction of the Brownian motion in phase space would lead to an attractor with an infinite correlation dimension. This is illustrated in figure 4-8 where the correlation dimension is calculated for the embedding dimension range from 1 to 20.

b. If with the increase of the embedding dimension, the correlation dimension also increases, but remains much below the embedding dimension, then the time series observations are non-random and correlated. For such time series, the Hurst exponent, described previously, is not equal to 0.5. In this case, the possibility of long-term predictions are possible.

c. If by increasing the embedding dimension, the correlation dimension starts to saturate, then the time series is most probably chaotic [72]. As mentioned before, the reason for this saturation is that the attractor's fractal dimension does not change as long as the embedding dimension is higher than the attractor's. In this case accurate short-term predictions are attainable within a future period specified by the largest Lyapunov exponent. The converging value of D is the estimated value for the "true" fractal dimension of the underlying attractor.

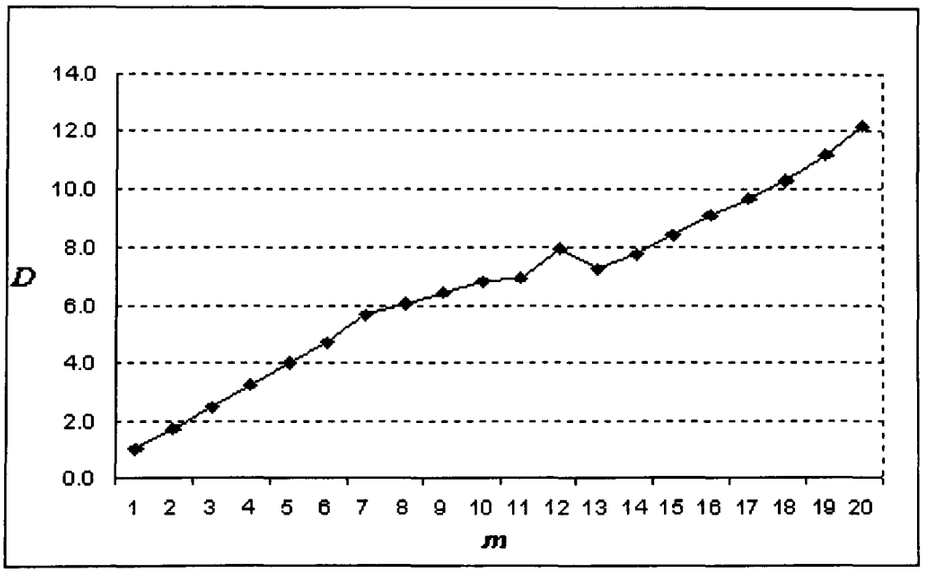


Figure 4-8. Case a.), the correlation dimension for Brownian motion grows to infinity as the embedding dimension increases.

From these cases, we can see that in addition to estimating the optimal embedding

dimension (the number of variables required to model the underlying system), the correlation integral can be used to distinguish between random and nonlinear chaotic systems. In case c.), we can estimate the optimal embedding dimension m according to Takens theorem. The time series needs to be embedded in a m -dimensional space such as $m \geq 2D+1$ in order to reconstruct the dynamics of the underlying attractor. However, given enough data, Sauer et al. [73] have shown that the estimated embedded dimension m is equal to the smallest integer greater than or equal to the fractal dimension D of the attractor. The logistic map's calculated correlation dimension for the increasing embedding dimension is shown in figure 4-9. The correlation dimension analysis shown in the graph confirms the fact that the logistic equation is one-dimensional. The estimated correlation dimension remains close to one, despite the increase in the embedding dimension m.

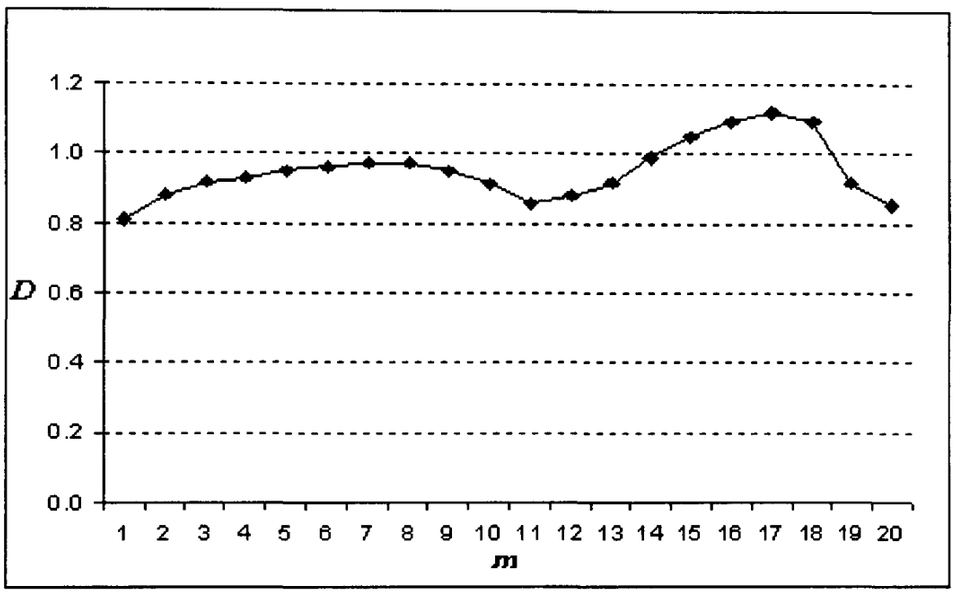


Figure 4-9. The logistic map's correlation dimension calculated for the increasing embedding dimension.

In addition to the logistic map, the correlation dimension and other tests are applied to the Dow Jones Industrial (DJI) financial index series. Seventy-five years of monthly data (1932-2007) were used in this study. The reason for using the monthly data is because the index was adjusted for inflation in order to detrent time series and to show realistic growth. The shortest period for which inflation data is available is on a monthly basis. This may not be much of a problem, as will be observed later, in nonlinear analysis it is better to decrease the number of observations by reducing the sampling rate rather than the observation time. Consumer Price Index monthly data

were gathered and the inflation effect was removed from the DJI using the following formula:

$$DJI^* = \ln(DJI_T) - (a \cdot \ln(CPI_T) + C) \tag{4.7}$$

where DJI^* is adjusted DJI in month T

DJI_T is the price of DJI in month T

CPI_T is the Consumer Price Index in month T

a and C are constants derived by regressing the log of DJI against the log of the CPI.

Figure 4-10 shows the inflation adjusted DJI* plot and its smooth version (DJI*-F). The filtered version is created by removing noise, using a General Regression Neural Network, although any adaptive moving average filter can produce similar results.

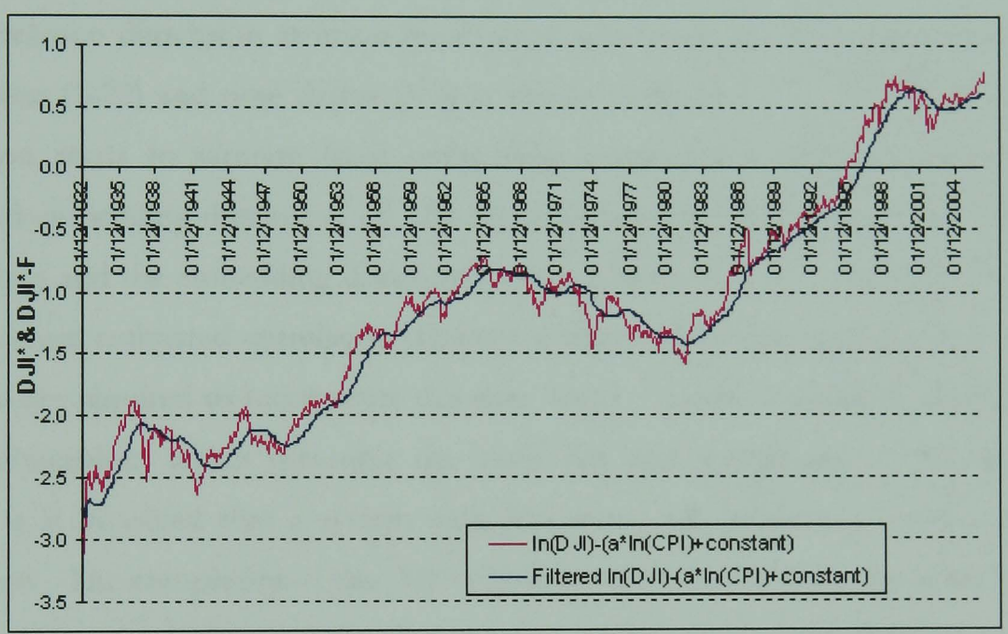


Figure 4-10. An inflation adjusted logarithm of the DJI and its smooth version.

Both time series exhibit long trends, even after removing the inflationary influence. As the analysis of nonlinear systems, especially chaotic ones, is mainly based on trajectories and cycles of the underlying attractor, we remove the trend from the FJI*-F time series by eliminating the 20 month moving average from it. The resulting de-trended time series, which we will call DJI*-FD, has visible cycles of different lengths and is shown in figure 4-11.

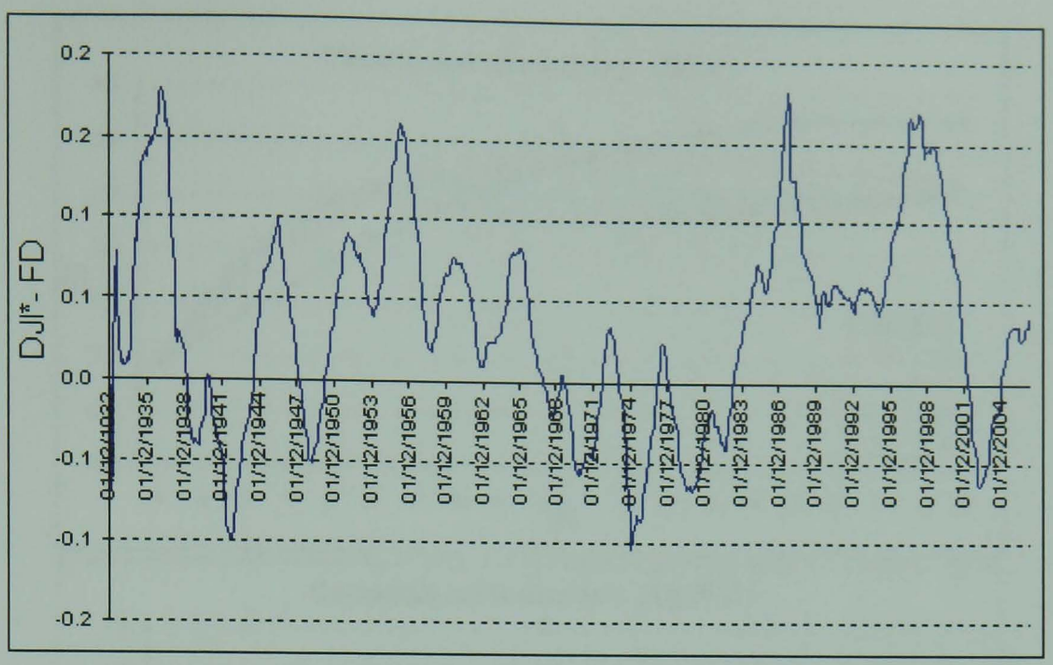


Figure 4-11. The DJI*-FD time series, a de-trended version of the DJI*-F time series produced by removing the 20 month moving average.

The correlation dimension estimate for all three time series for the range of embedding dimensions (1-20) and time delays (1-5) is shown in figure 4-12. The DJI* correlated dimension starts to saturate from embedding dimension 5 onwards, therefore, the estimated embedding dimension of 2.64 was calculated as the average value for all five time delays and the embedding dimension varying from 5 to 20. For DJI*-F the time series average estimated correlation dimension was 1.43, which suggests that only two variables are required to reconstruct this time series. As the correlation dimension is a fractal dimension, which measures the variability and complexity of the underlying system, it is expected that a system with less noise will produce a lower correlation dimension. The complexity of the DJI*-FD time series lies between the other two time series and it's estimated correlation dimension value is 2.40.

The presence of a low fractal correlation dimension indicates that all these time series have a low dimensional attractor, and as a result could be classified as chaotic.

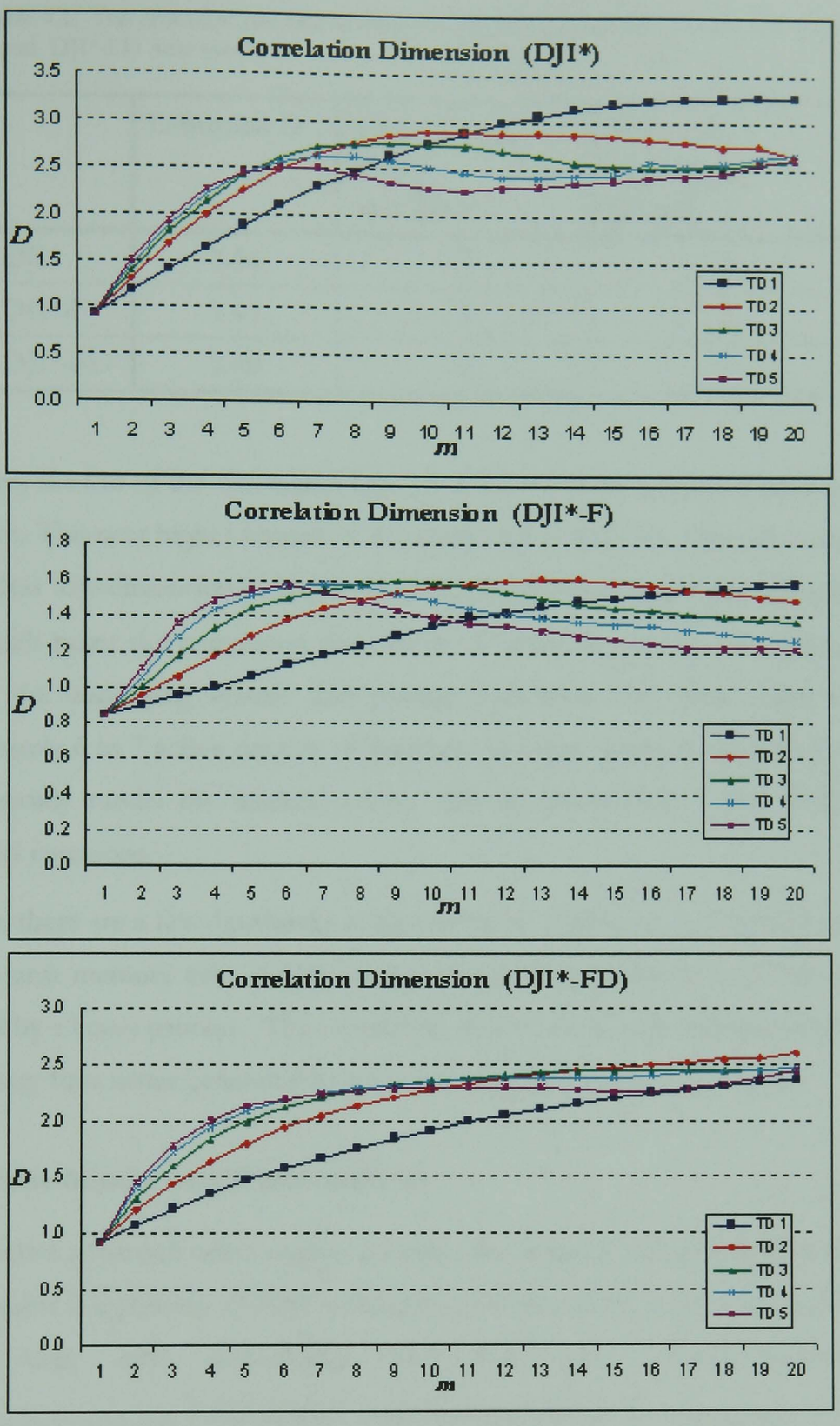


Figure 4-12. The correlation dimension estimates for DJI*, DJI*-F and DJI*-FD time series for varying embedding dimensions (1-20) and time delays (1-5).

The estimated optimal embedding dimension for all three time series is shown in table 4.1.

Table 4.1. The estimated embedding dimension from the correlation dimension for DJI* , DJI*-F and DJI*-FD time series.

	Estimated D	Takens's upper bound $m \geq 2D+1$	Sauer et al. (next higher integer)
DJI*	2.64	7	3
DJI*-F	1.43	4	2
DJI*-FD	2.40	6	3

In practice, the use of the theoretical limit $m \geq 2D+1$ is the preferred option for noisy time series. The next higher integer of the estimated correlation dimension can be used for noiseless low-dimensional chaotic systems. Some research use the embedding value m at which point the correlation dimension D starts to saturate. According to these findings, the underlying system that produced all three DJI time series above has approximately 6 to 7 active degrees of freedom. In other words, it would be justified to mathematically model the market activity with no more than 7 first-order ordinary differential equations.

However, there are a few drawbacks to this method. Theiler et al. [75] have shown that the long-term memory time series[1] can be classified as nonlinear and chaotic, even if produced by a linear process. The correlation dimension is well defined only in the case of stationary time series generated by a low-dimensional dynamical system.

4.4.2 False nearest neighbours method

An alternative approach often used to estimate the optimal embedding dimension is the False Nearest Neighbours (FNN) technique, introduced by Kennel et al. [64]. This method uses trial embedding and for each m-dimensional vector $\vec{x}_t = \left(x_{t-(m-1)\tau}, \ldots, x_{t-\tau}, x_t\right)$ the nearest neighbouring vector $\vec{y}_t = \left(y_{t-(m-1)\tau}, \ldots, y_{t-\tau}, y_t\right)$ is found, based on the Euclidean distance[2]. From here, the embedding dimension is expanded by one for both vectors and the new distance is calculated. If the expanded dimensional vector is large enough for the attractor to unfold, then the new distance

[1] Also called long coherence time.

[2] Euclidean distance is defined as $\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\ldots+(x_n-y_n)^2}$

should be small and similar to the one measured previously. If not, the neighbour is classified as being a "false neighbour". A simple example of this is illustrated in figure 4-13 showing how two nearby points in a one-dimension view (line) are wide apart when seen on a true attractor (circle) in a two-dimensional space.

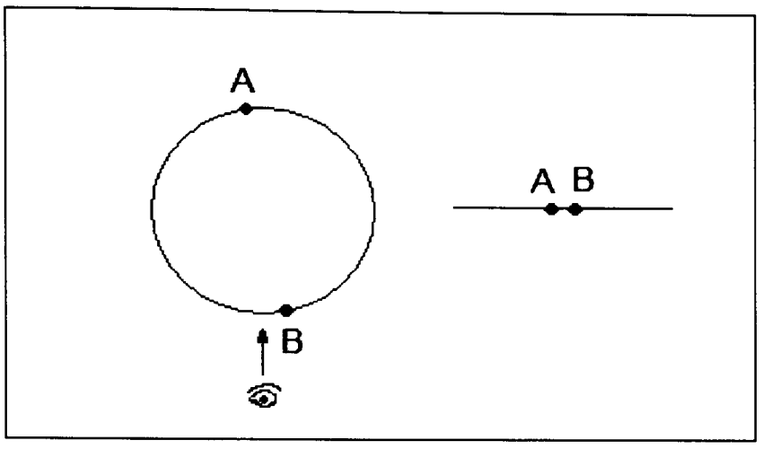


Figure 4-13. False nearest neighbour graphical interpretation.

When the embedding dimension is sufficiently large, the fraction of vectors that have a false nearest neighbour converges to zero. Results obtained with the false nearest neighbour method for the logistic function is shown in figure 4-14 and for all three financial time series are shown in figure 4-15. As expected, for the logistic function the number of false nearest neighbour drops to zero for the embedding dimension $m = 1$.

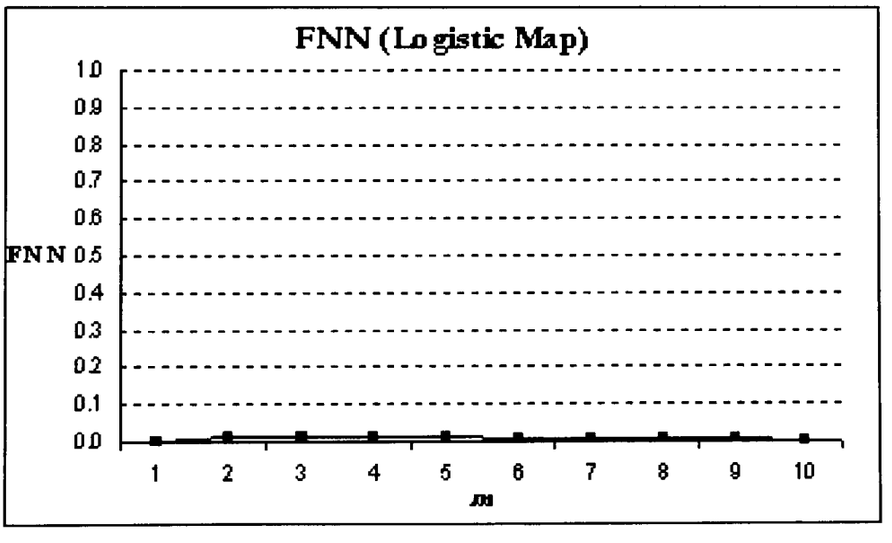


Figure 4-14. The false nearest neighbour method as an estimator of the optimal embedding dimension for the logistic function time series.

For financial time series DJI*, DJI*-F and DJI*-FD the number of false neighbours

approaches zero around embedding dimension of 3, 2 and 3 respectively.

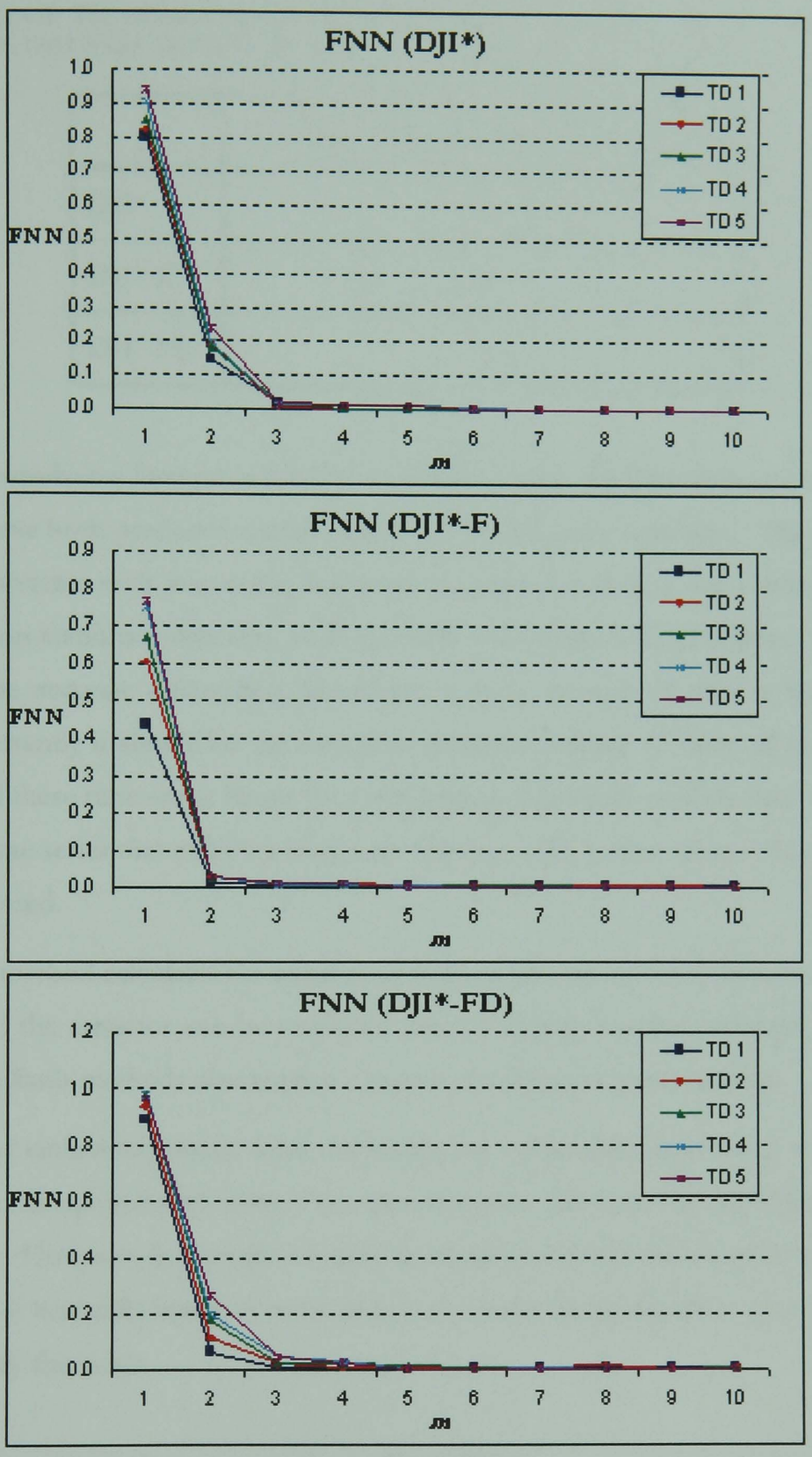


Figure 4-15. The false nearest neighbour method as an estimator of the optimal embedding dimension for DJI*, DJI*-F and DJI*-FD time series for varying embedding dimensions (1-10) and time delays (1-5).

These results are equal to the estimates obtained using the next higher integer of the

correlation dimension as shown on page 68 (table 4.1).

Table 4-2. The estimated embedding dimension using the false nearest neighbour method for DJI* , DJI*-F and DJI*-FD time series.

	Estimate of D using the FNN method
DJI*	3
DJI*-F	2
DJI*-FD	3

Both the correlation integral and FNN methods require the time delay to be specified and they have both produced similar results for the different time lags. They work well on stable systems with low noise, but when presented with real data, including short, noisy, or non-stationary data sets, both methods are fraught with problems. In order to estimate the accurate embedding dimension, a large amount of data is required, and most importantly it should be for long time periods. Having 75 years of monthly data may model these time series better than using many thousands of daily data values. For financial time series that exhibit a long term memory effect, more time rather than more data is required.

The FNN method calculates the number of false neighbours globally, however, the local dynamic of the attractor can be modelled using a slightly smaller or larger embedding dimension. Both methods also require a degree of subjective interpretation.

For optimal embedding, apart from estimating the embedding dimension, we also need to estimate the optimal time delay. For noise-free data, the choice of time delay is of less importance. However, in the case of noisy financial time series the choice is of increased relevance. It is usually better to overestimate the embedding dimension, but not so high as to amplify the noise.

4.5 Estimating the appropriate time delay

Estimating the correct time delay is essential for a correct reconstruction of the underlying time series dynamic. Choosing a value which is too small results in slow changes of a set of very similar delayed vectors, whilst a large value creates a set of

uncorrelated vectors making the reconstructed state space look random. A suitable time delay has to be large enough that the information we get from the new delayed vector is significantly different but still correlated to the vector measured previously. Unfortunately, there is no rigorous way to determine the optimal value of time delay. Two common ways of selecting an appropriate delay include finding the first minimum, in either the (linear) autocorrelation function or the (nonlinear) mutual information function. The use of a classical autocorrelation function is not appropriate for nonlinear systems as it only detects linear correlations.

Wolf et al. [66] have shown that the appropriate time delay for a given embedding dimension can be estimated from the following relation:

$$\tau = \frac{Q}{m} \tag{4.8}$$

where m is the embedding dimension

Q is the average cycle period.

Q can be estimated using the rescale range (R/S) analysis presented in section 3.3.1. The average cycle period is equal to a time at which the R/S plot changes its slope.

A preferable way to compute a time delay is by using the average mutual information function[1], which detects both linear and nonlinear correlations. Fraser and Swinney [76] first used this information-theoretical measure to find the optimal time delay, which corresponds to the first minimum of the mutual information between the time series $x(t)$ and the delayed time series $x(t-\tau)$. The mutual information is estimated from entropies of these two time series.

The entropy of the distribution of the discrete random variable x_i with probability distribution $p_x(x_i), i = 1,...N$ is defined as[2]:

$$H(x,N) = -\sum_{i=1}^{N} p_x(x_i)\log_2(p_x(x_i)) \tag{4.9}$$

The entropy value represents the average number of bits required to describe an isolated

---

[1] This concept originates from the application of information theory developed by Shannon [67].

[2] Assuming that the discrete variable x_i has been digitised to integer values lying between 1 and N, then the probability of the particular value x_i that has been observed n times is $p_x(x_i) = n/N$.

observation x_i and can range from 0 to $\log_2 N$. The mutual information between the two time series, $x(t-\tau)$ and $x(t)$, is the difference between the sum of their individual entropies and the joint entropy:

$$I(x_{t-\tau}, x_t, N) = H(x_{t-\tau}, N) + H(x_t, N) - H(x_{t-\tau}, x_t, N) \tag{4.10}$$

where the individual entropies are:

$$H(x_{t-\tau}, N) = -\sum_{x_{t-\tau}=1}^{N} p\left(x_{(t-\tau)}\right)\log_2\left(p\left(x_{(t-\tau)}\right)\right) \tag{4.11}$$

$$H(x_t, N) = -\sum_{x_t=1}^{N} p(x_t)\log_2(p(x_t)) \tag{4.12}$$

and the joint entropy is:

$$H(x_{t-\tau}, x_t, N) = -\sum_{x_{t-\tau}=1}^{N}\sum_{x_t=1}^{N} p\left(x_{(t-\tau)}, x_t\right)\log_2\left(p\left(x_{(t-\tau)}, x_t\right)\right) \tag{4.13}$$

The mutual information measures in bits the degree to which the knowledge of one variable specifies the other. If there is no dependence among these variables then the mutual information will approach zero. In other words no knowledge can be gained for the second sample by simply knowing the first.

Figure 4-16 shows the average mutual information values calculated for the DJI*, DJI*-F and DJI*-FD time series.

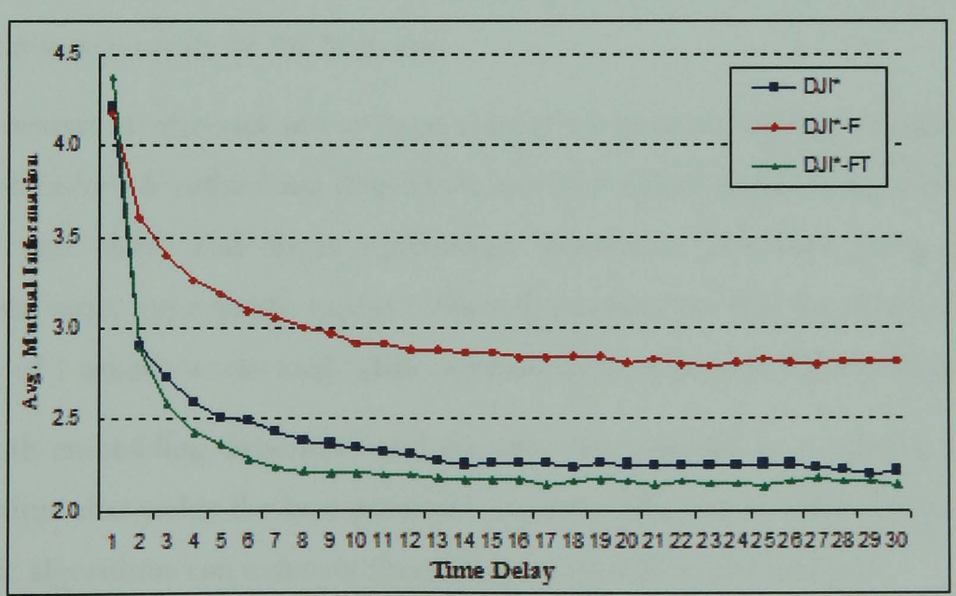


Figure 4-16. Average mutual information method as an estimator for the optimal time delay of

DJI*, DJI*-F and DJI*-FD time series.

As can be seen from the graphs, the mutual information reaches its first minimum at the time delay of 13 months for the DJI* time series. For the DJI*-F time series the first local minimum is at 11 months and for DJI*-FD it is at 8 months. These estimates seem larger than they should be. Even for the logistic equation that has an optimal time delay of 1, the estimate using this approach was 10 (figure 4-17).

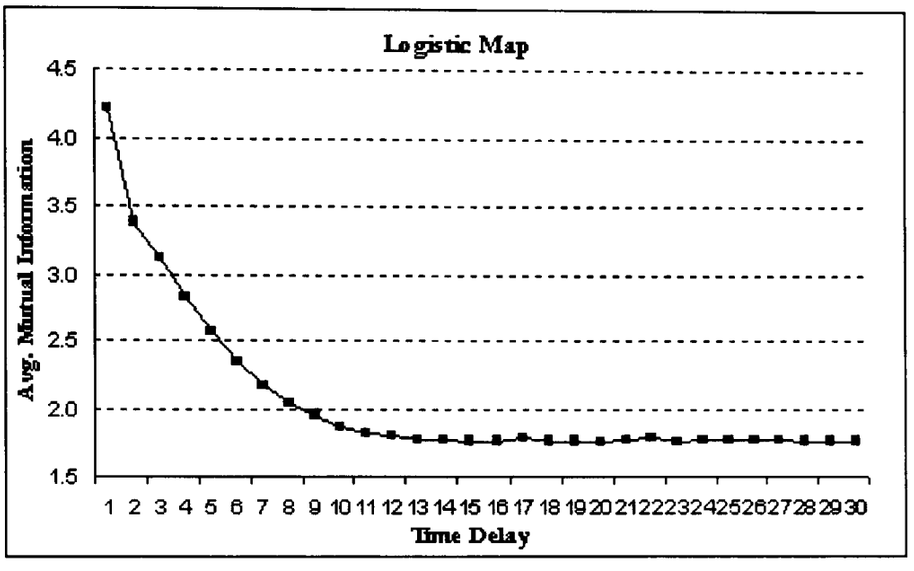


Figure 4-17. The average mutual information method as an estimator for the optimal time delay of the logistic map time series.

Typically the time delay should be selected in order to minimise the interaction between the points of the time series considered, but in practical forecasting applications this value may not necessarily be the best one.

The false nearest neighbours and average mutual information techniques can be used in the first indication of embedding dimension and time delay, respectively. Often, smaller values of time delay and larger embedding dimension estimated using the above methods are more appropriate values. We will see later on that for discrete systems a time delay of 1 usually works well, while continuous systems should have larger values.

Ideally, both embedding dimension and the time delay should be estimated together in one procedure that yields the best predictive model. The way in which neural networks and genetic algorithms can estimate these parameters will be shown later.

4.6 Lyapunov exponents

As the dynamic of the attractor evolves in time, the nearby geometry of the cloud of points diverge away and converge back again. The Lyapunov exponents measure the average local exponential rates of divergence or convergence of neighbouring trajectories' points. The exponents can be calculated for each dimension of a dynamic system or a reconstructed model. The presence of at least one positive Lyapunov exponent confirms sensitivity to initial conditions, this being a hallmark of nonlinear systems that are a necessary prerequisite for deterministic chaos[1]. This implies that the trajectories diverge from each other rapidly, and make long term predictions impossible[2]. The magnitude of the positive exponent reflects the time scale at which system dynamics become unpredictable. Whilst the positive exponent measures how rapidly nearby points diverge from one another, the negative exponent measures the contraction of the state space's time scale.

There are now a variety of algorithms for estimating Lyapunov exponents from time series. An early algorithm developed by Wolf et al. [66] is in wide use for calculating the largest Lyapunov exponent using experimental data. Other techniques fit various forms of functions to the embedded data and calculate the local divergence of these functions. The exponents can also be estimated using a neural network [69], [70].

In the particular case of the logistic map, the Lyapunov exponent is derived from the equation:

$$\alpha = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \log_2 | \lambda(1-2x_i) | \qquad (4\text{-}14)$$

For convenience the exponents are measured in bits per time interval - a legacy of the Shannon [67] information theory.

The Lyapunov exponents graph of the logistic map for the range of λ values is shown in the figure 4-18.

---

[1] Not all systems that are sensitive to the initial condition are chaotic.

[2] A good example is the weather forecast.

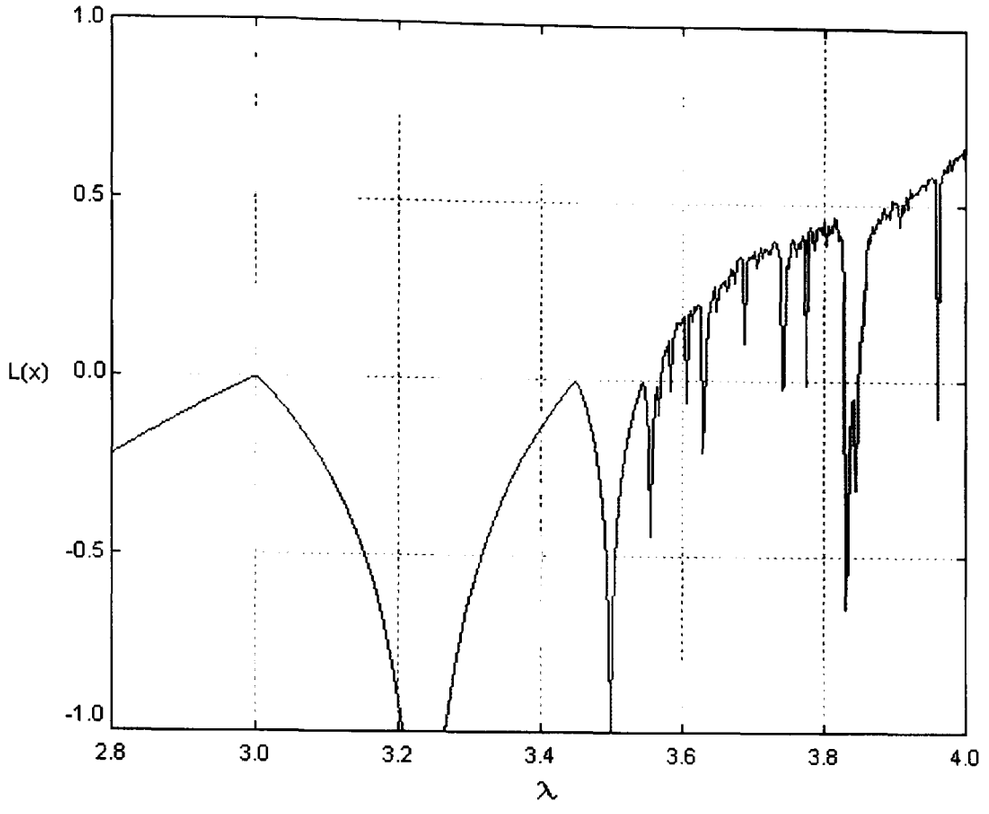


Figure 4-18. Logistic Map Lyapunov Exponents for the range of λ values; initial value x(0)=0.1; discard first 100, display next 400 iterations.

If we compare this picture to the logistic map figure 4-4 we can see how the exponent's values dip into negative territory for the non-chaotic regions ('clear windows'), while they are positive in the chaotic regions.

The final values of the estimated largest Lyapunov exponent for the three financial time series is presented in table 4-3 and how it evolves over time is shown in figure 4-19.

Table 4-3. The estimated largest Lyapunov exponents for DJI*, DJI*-F and DJI*-FD time series using Wolf at al. method.

	DJI*	**DJI*-F**	**DJI*-FD**
Largest Lyapunov exponent	0.01046	0.00684	-0.00109

For the DJI* and DJI*-F time series the exponent is positive over the whole period while the DJI*-FD exponent takes both positive and negative values. These results indicate that the DJI* and DJI*-F possess properties typical of deterministic chaotic signals, while the DJI*-FD oscillates between chaotic and non-chaotic periods.

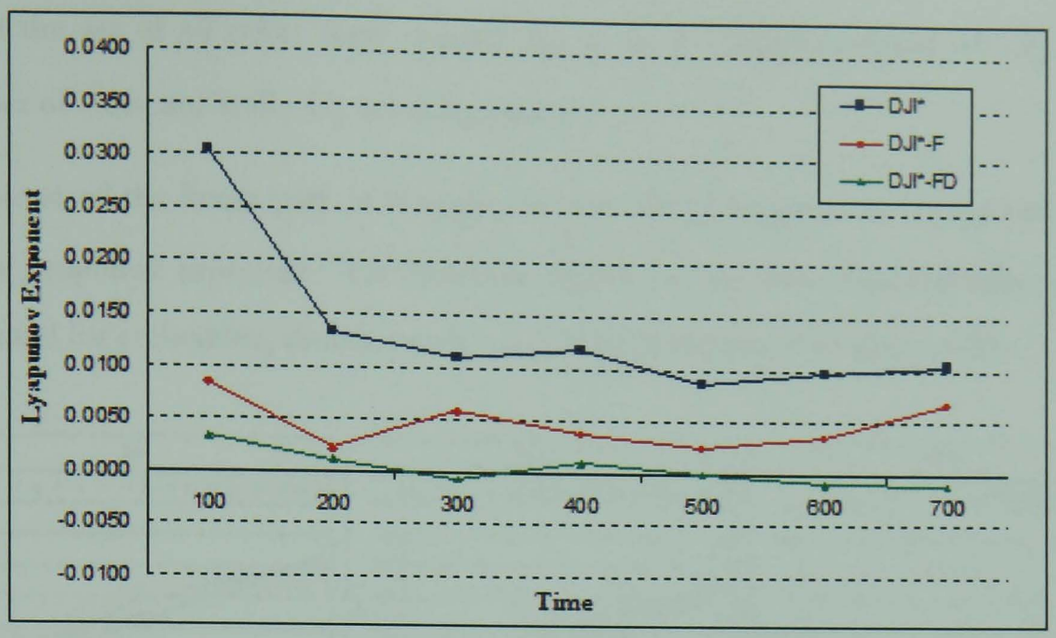


Figure 4-19. The largest Lyapunov exponent of DJI*, DJI*-F and DJI*-FD time series.

Knowing the largest Lyapunov exponent value is of practical importance as it can be used to estimate the ability to forecast future time periods reliably. However, it is very difficult to correctly estimate the Lyapunov exponents from noisy and short time series. The correct estimate of the largest Lyapunov exponent can be undetectable due to the autocorrelation present in the series under study, and therefore, the time series can be wrongly labelled as chaotic [95].

Also, the Wolf et al. algorithm requires a large amount of data when applied to complex systems. The minimum number of necessary data is estimated at 10^D, where D is the dimensionality of the attractor. This would require a million data points for a six dimensional system, which could present a problem for the analysis of what are often short financial time series.

To resolve the shortfalls of the Wolf et al. model, Kantz [84] and Rosenstein et al. [85] independently developed an algorithm that allows a robust estimation of the maximal Lyapunov exponent of noisy and short time series. The algorithm tests the exponential divergence of nearby trajectories directly by calculating:

$$S(\Delta n) = \frac{1}{N-m}\sum_{n=m+1}^{N}\log\left(\frac{1}{u_n}\sum_{x_{n'}\in U_n}\left|\vec{x}_{n'+\Delta n}-\vec{x}_{n+\Delta n}\right|\right) \qquad (4\text{-}15)$$

for different embedding dimensions m, neighbourhoods of different sizes ε and a few hundred different reference trajectories. The $\vec{x}_n$ is a delay vector of dimension m, and

U_n is the set of all other delay vectors $\vec{x}_{n'}$ in an ε - neighbourhood of $\vec{x}_n$. The number of elements in the U_n set is denoted u_n.

The slope of the linear part (if it exists) of the $S(\Delta n)$ function is interpreted as the largest Lyapunov exponent. The function $S(\Delta n)$ for the three financial time series is calculated for embedding dimension of 1 to 10 and is presented in figure 4-20.

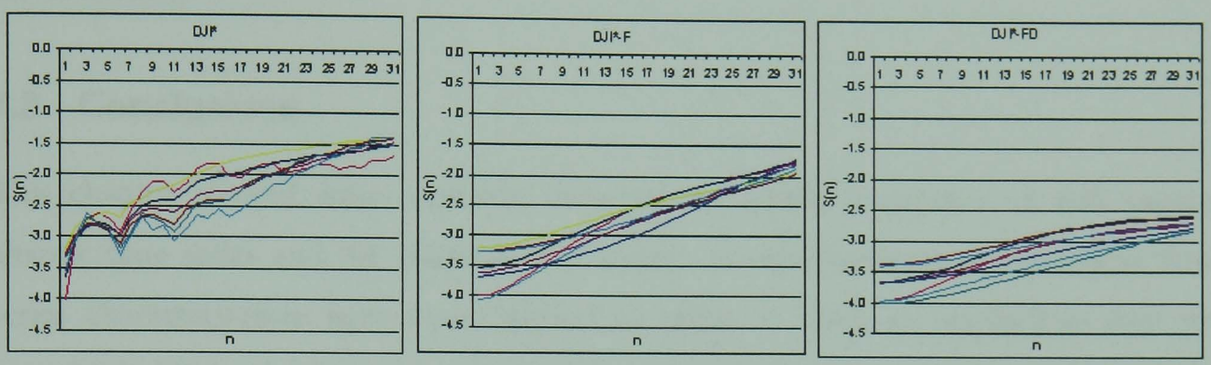


Figure 4-20. The function $S(\Delta n)$ calculated for the DJI*, DJI*-F and DJI*-FD time series and embedding dimension 1 to 10.

The function shows a rather robust positive linear slope for all three time series, confirming the existence of the positive Lyapunov exponent for each time series, indicating that they could be chaotic (table 4-4).

Table 4-4. The estimated largest Lyapunov exponents for DJI*, DJI*-F and DJI*-FD time series using S(n) function.

	DJI*	DJI*-F	DJI*-FD
Largest Lyapunov exponent	0.05816	0.05717	0.03222

4.7 Alternative methods

The reliable estimate of the dynamical invariants of the financial time series can be a difficult task due to the presence of serial correlations, noise, and the relatively short length of data. Those parameters required for successful modelling of nonlinear systems could be found experimentally through the use of neural networks. The degrees of freedom of the system can be estimated by training many neural networks using a different number of neurons in the hidden layer. The number of hidden units when the minimum error is reached can be seen as the number of degrees of freedom. An equivalent procedure can estimate the embedding dimension by varying the number

of input units. The Lyapunov coefficient can be estimated by looking at out-of-sample errors as a function of future prediction time. The interpretation of these figures can be a problem as NNs can produce different results depending on the NN types being used.

From the perspective of applied analysis, the best parameters are those that give the best prediction results when used in the forecast model. The complexity of those models used for predictions may not be related to the complexity of the system that produced the time series.

4.8 Conclusions

This chapter showed some similarity between the typical properties of well known chaotic time series and the financial time series. Several methods for nonlinear time series characterisation have been applied in order to discover regularities that are completely hidden in these time series and are not detectable to conventional analysis. The correlation integral was used to estimate the appropriate embedding dimension of three pre-processed variants of the DJI time series. This method found that these time series are possibly generated by a low-dimensional chaotic system. The false nearest neighbour method confirmed these results. This technique was used to estimate the appropriate embedding dimension, while the average mutual information method was used to estimate time delay. The Lyapunov exponent estimate confirmed sensitivity to initial conditions, indicating that financial time series are produced by a nonlinear system. However, the correct estimate of these parameters can be difficult for short, noisy, or non-stationary financial time series.

Ideally, both the embedding dimension and the time delay should be estimated together in one procedure to yield the best predictive model. These properties characterise useful concepts of the underlying system that have practical implications which will be used in the following chapters.

Chapter 5

Recurrence Analysis of Nonlinear Systems

5.1 Introduction

The previous two chapters presented methods for analysing financial time series and estimated the embedding parameters required for the reconstruction of dynamical systems. Those methods gave us an insight into the characteristics of financial systems that generate the time series under study. Further knowledge can be gained through the use of recurrence analysis, an elegant and very intuitive method for analysing the behaviour of dynamical systems. Using such tools such as *state space* and *recurrence plots,* it is possible to visualy assess the properties of the underlying system at a glance. More detailed information of the system's behaviour can be gained using *recurrence qualification analysis.* This chapter describes the use of these methods in analysing various dynamical properties such as the degree of determinism, with particular emphasis on extreme market events.

5.2 State space plots

As was shown earlier, state space plots can be used to visualise the spatial structure of the dynamical system. As the correlation dimension calculated for the smoothened version of DJI (DJI*-F) was 1.43, we can assume that only two variables (i.e. the next larger integer, Sauer et al. [73]) are required to model the underlying dynamics. If this estimate is correct then the two-dimensional state space plot may reveal more insights. Figure 5-1 shows the state space plots for the DJI*-F time series reconstructed using the increasing time delay. The small circles/ellipses and dents visible on the charts represent small cycles and the large ones represent big cycles. In between these big cycles we have a long period of a trending market represented by an almost straight line. Are these cyclical parts chaotic regions that are separated by periods of calm and stable trending markets, or as explained earlier is intermittency at play?

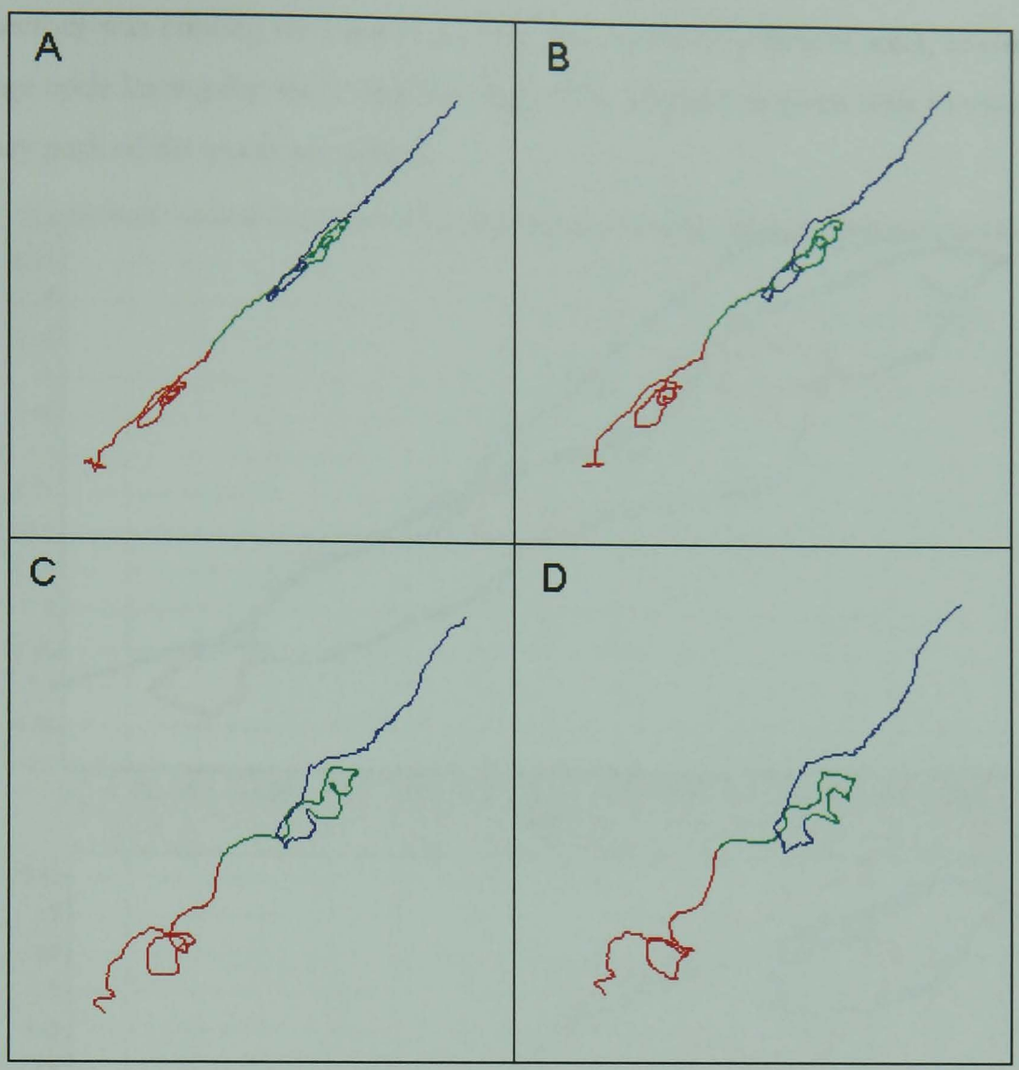


Figure 5-1. State space plots of filtered DJI*-F time series using time delays of 5 (A), 10 (B), 20 (C) and 30 (D) months respectively.

It would be interesting to compare the length of these long periods of cyclical and trending markets in order to find some scaling invariant, but unfortunately, we do not have enough of them to make any viable conclusion. Using the seventy five years of monthly data, we only have two examples of each, with a possible third cyclical period in the starting phase. Will this current third cycle starts to spiral backwards as the previous ones, indicating a market downturn that may last decades? It would be of interest to see the development of this plot in the long term. The shortage of a long span of data highlights a common problem with financial data analysis.

The way in which this plot evolves in real time is represented by the different colours. Just to illustrate the dynamics of the state space trajectory in the middle part of figure 5-1 B, the middle and the top parts are enlarged and shown in figure 5-2 for the time delay of 10 months. The thin red line is a hypothetical continuation of the future that mimics what happened in the graph above it, representing the 1960's to 1980's period, where

the trajectory was circling backwards in four/five cycles of different sizes, encompassed by a large cycle lasting for about twenty years. The example is given only to visualise the trajectory path of the previous period.

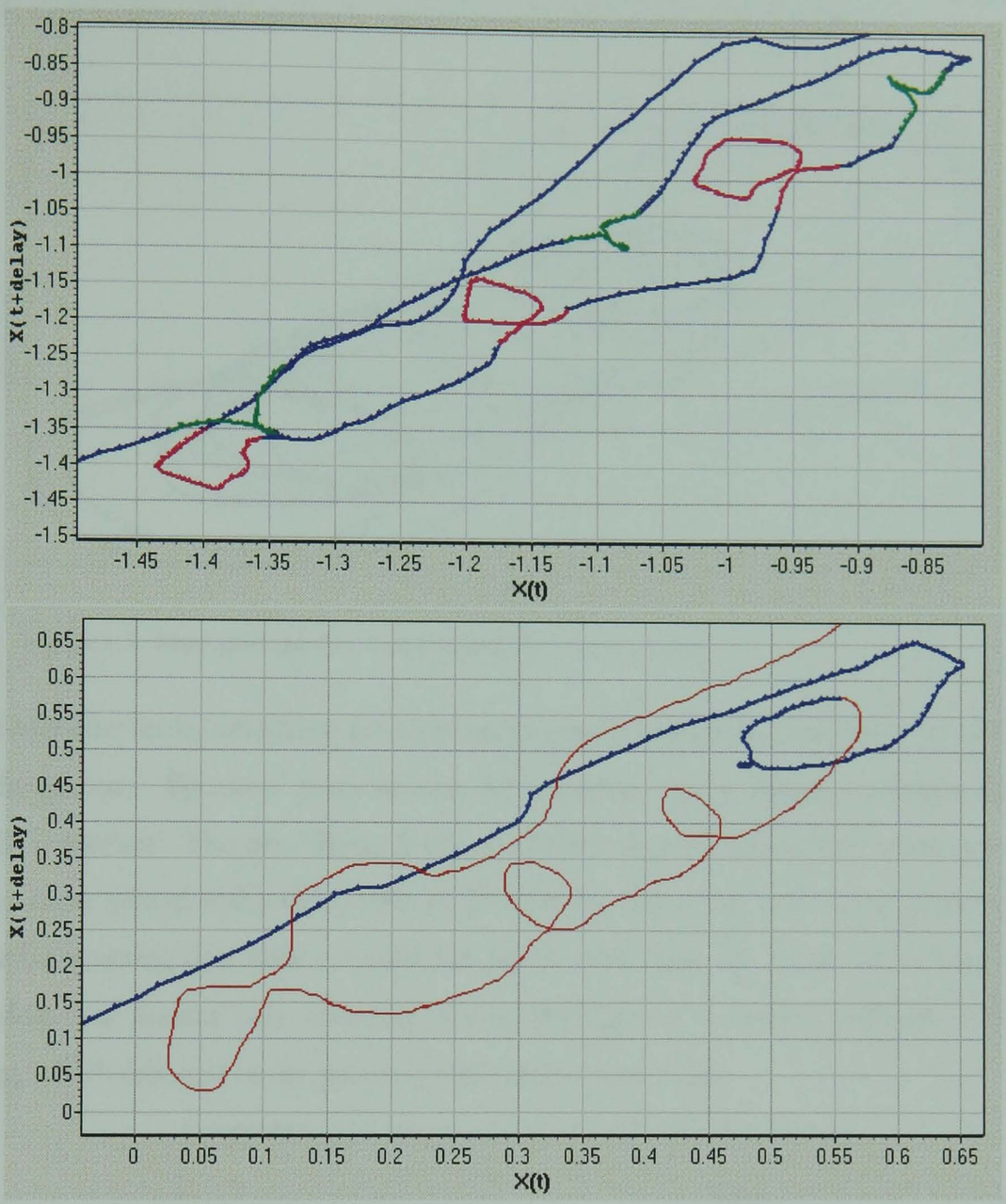


Figure 5-2. DJI*-F state space dynamics showing irregular cycles. The time delay used is 10 months. The top graph represents the 1960-1980 period. The bottom graph is the current state (blue line), and the hypothetical future development of the plot (red line), shown only to clarify the dynamics of the plot above.

As has been shown before, the attractor is a defined shape, a spatial pattern that the trajectory is evolving around. By examining the top plot in figure 5-2, we can notice three small dents (cycles) highlighted in green, and three bigger cycles highlighted in pink, almost equally spaced apart. The green dents represent the DJI cycles formed in an upward trend, whilst the pink circles represent bigger cycles from a downward slope.

The trend presented in this time series causes these cycles to be apart. By removing the trend from the time series, the dents would get close in space and so would the circles, resembling a more defined attractor. Figure 5-3 shows a state space plot of a de-trended DJI*-F time series. The aperiodic cycles are clearly visible in the graph.

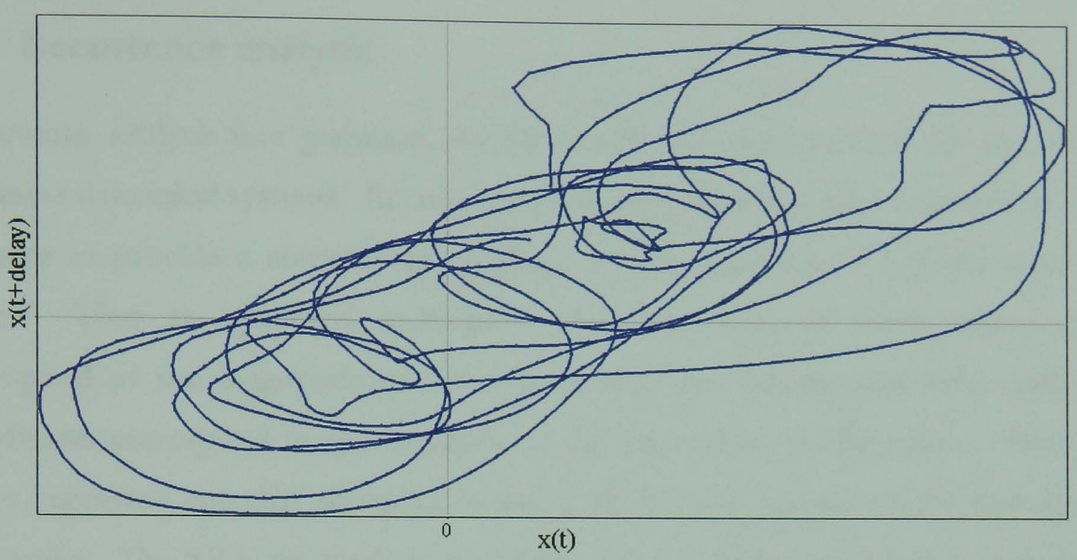


Figure 5-3. State space plot of a de-trended DJI*-F time series.

One cautionary note, searching for patterns in graphs and plots could be very subjective and dangerous. Spurious patterns can be found in many different shapes that are actually irrelevant. The next figure 5-4 is an unfiltered (with noise) DJI* state space plot for the same period and a delay time of 15 months. The first half of the picture in red resembles a sitting dog with a visible tail on the left, legs, fur, head, and a long snout. This does not sound very scientific but it highlights a common problem with data mining, which can find many patterns where there aren't any.

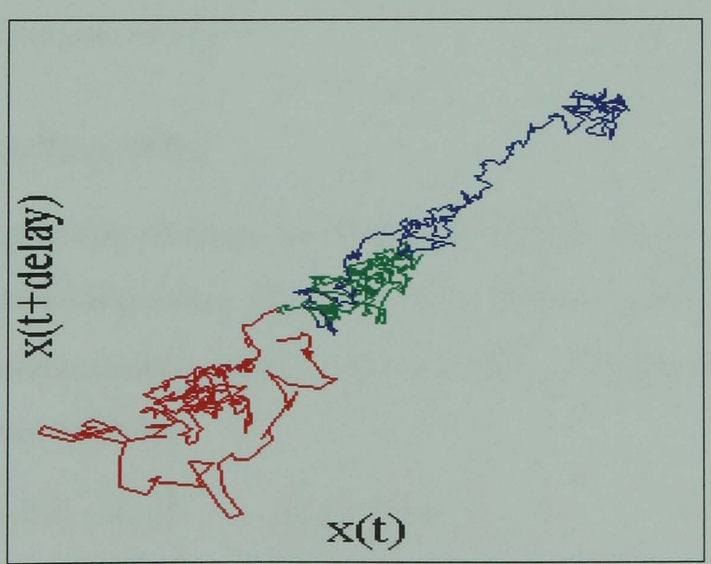


Figure 5.4. State space plot of the DJI* with noise and time delay of 15 months.

There is a limit to the information that can be derived from state space plots, as we can only observe two and thee dimensional graphs. To accommodate for these shortfalls a much more powerful and versatile technique was introduced called Recurrence Plots (RPs) and Recurrence Qualification Analysis (RQA).

5.3 Recurrence analysis

Recurrence analysis is a graphical, statistical and analytical method for the study of nonlinear dynamical systems. Recurrence plots were introduced by Eckmann et al. [77] in order to provide a comprehensive image of the dynamics of a given system at a glance. They are graphical rectangular plots consisting of pixels whose colours correspond to the magnitude of data values in a two-dimensional array and whose coordinates correspond to the locations of the data values in the array. These array values represent encoded distances between all delayed vectors of the reconstructed time series. The RP is an excellent tool that can be used to visualise the evolution and recurrence of trajectories in state space. Essentially, it is the graphical representation of a correlation integral explained in section 4.4.1, but unlike the correlation integral, the RP preserves both temporal and spatial dependence in time series. RPs are especially suitable for the analysis of complex, short and non-stationary time series and can be applied to numerous scientific fields. The RP does not use any assumptions about data and is not constrained to any type of statistical distribution.

The following sections will demonstrate the use of recurrence plots and recurrence qualification analysis to detect various properties of the financial time series including the start period of extreme events.

5.3.1 Recurrence Plots (RPs)

As shown in the previous sections, the dynamical system can be represented by the trajectory of the embedding vector $\{\vec{x}_i\}$ for $i = 2,..., N$ in a D - dimensional space. The RP plot is a two-dimensional graphical representation of the matrix of these vector distances within a cut-off limit:

$$R_{i,j} = \Theta\left(\varepsilon - \| \vec{x}_i - \vec{x}_j \|\right) \qquad i, j = 1,...N \tag{5.1}$$

where ε is a predefined threshold limit and Θ is a Heaviside function, as in the

correlation integral. The norm function geometrically defines the size and shape of the neighbourhood surrounding each reference point. The most common norms used to calculate distance are shown in figure 5-5. The diamond shape is the minimum norm, the square is the maximum and the circle is the middle (Euclidian) norm, which is used in the following examples.

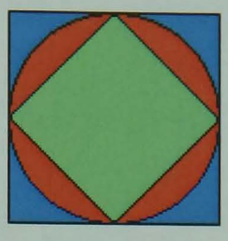


Figure 5-5. Common norms used to calculate distance in RPs.

The graphical representation in RPs is obtained by encoding the distances between the pairs of vectors into black (if the distance is below the threshold) and white dots (distance is equal or above the threshold). Often, the threshold parameter is not used and the distances are encoded into a dark (cold) and light (hot) colour scheme. These are known as global recurrence plots or distance plots. Usually the colour scheme used is shown on the right hand side of the RPs. The RP plot shows different structures depending on the nature of the underlying time series.

5.3.2 Interpretation of recurrence plots

By visually inspecting the typology and texture of a recurrence plot, properties of the system such as stationarity and determinism can be assessed. The RP obtained from purely random systems do not show distinguishable patterns. They appear as a cloud of evenly distributed coloured points with no apparent structure as in the case of white noise (figure 5-6 A)[1]. By contrast, the sine wave (figure 5-6 B) produces a clear pattern of many white/yellow diagonal lines that are an indication of a deterministic process. The more structured the RP is, the more deterministic the signal is.

Both these plots have a homogenous structure indicating that the data sets originated from a stationary process, whilst a non-homogenous structure indicates that which is non-stationary in the system (figure 5-6 C, D & F).

---

[1] The software used for creation of RP is the Visual Recurrence Analysis (VRA).

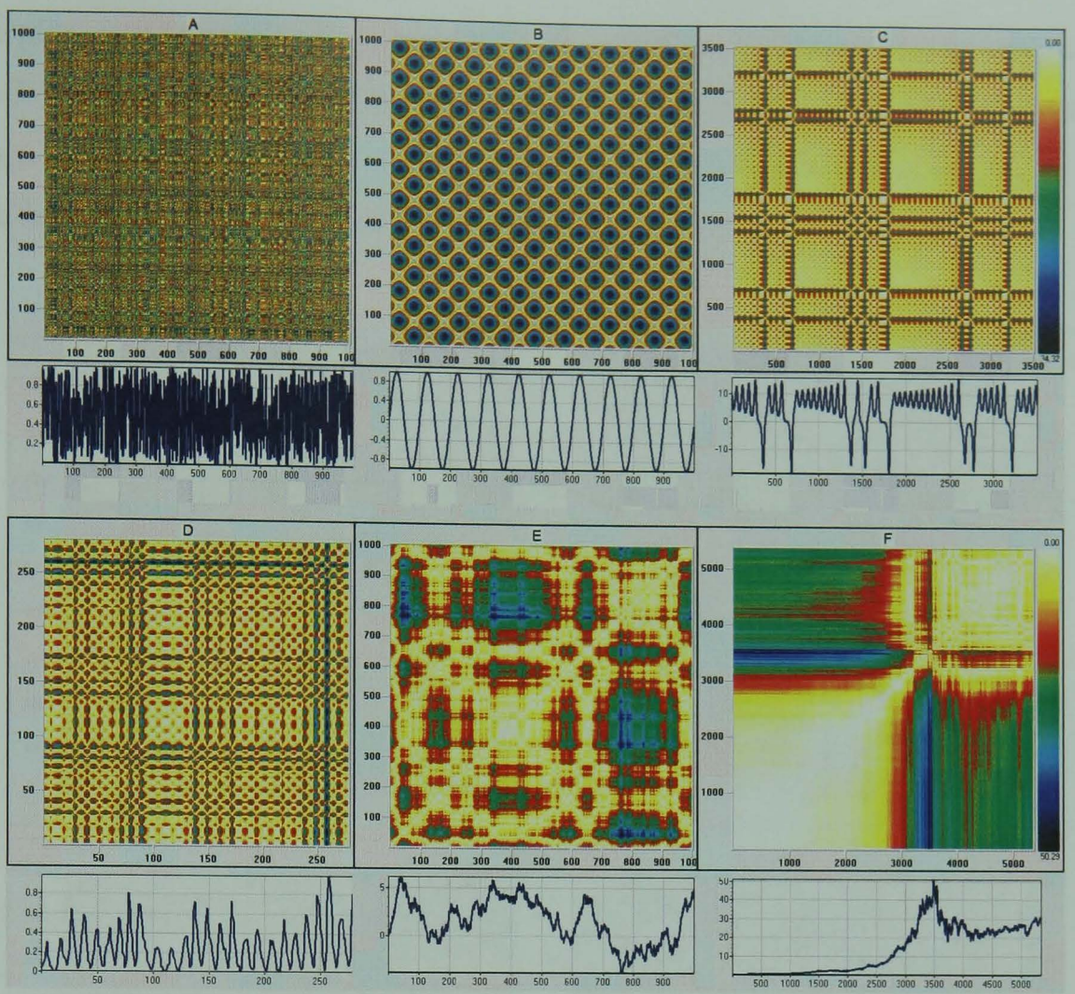


Figure 5-6. Recurrence plot examples and their corresponding time series for: A – white noise, B – sine wave, C – Lorenz time series, D – sun spots time series, E – Brownian motion, and F – Microsoft share price. The embedding was not used i.e. ($m = 1, \tau = 0$).

The diagonal lines are an indication of recurring cyclical or deterministic system behaviour and their length and distribution patterns can be used to classify the system further into being simple, complex or chaotic. The longer the diagonal lines, the more deterministic the signal is. It should be noted that there are no diagonal lines in the pure stochastic system (plot A). The distance between the lines is determined by the periodicity cycle of the sine wave. The line of identity (LOI), the main diagonal line from the bottom left hand corner to the top right hand corner, is clearly visible in all plots. This line can be ignored as it represents the zero distance of the vectors compared with themselves. Both parts of the RP image on each side of the LOI are a mirror image of each other. The RPs presented in figure 5-6 are an example of global distance recurrence plots. These RPs don't use the threshold value ε, as all distances between vectors are encoded using different colours. Often they reveal more structure and patterns and are supposed to show a full picture of the underlying dynamical system. However, in some cases this type of plots can hide important diagonal features

of the plot, as they emphasise vertical and horizontal lines. This is clearly illustrated in figure 5-7, which shows global distance RPs in the case of white noise (A1 and A2) and the logistic function (B1 and B2). The A2 and B2 RPs are enlarged squares (a zoomed in area from point 400 to 600) of A1 and B1 respectively.

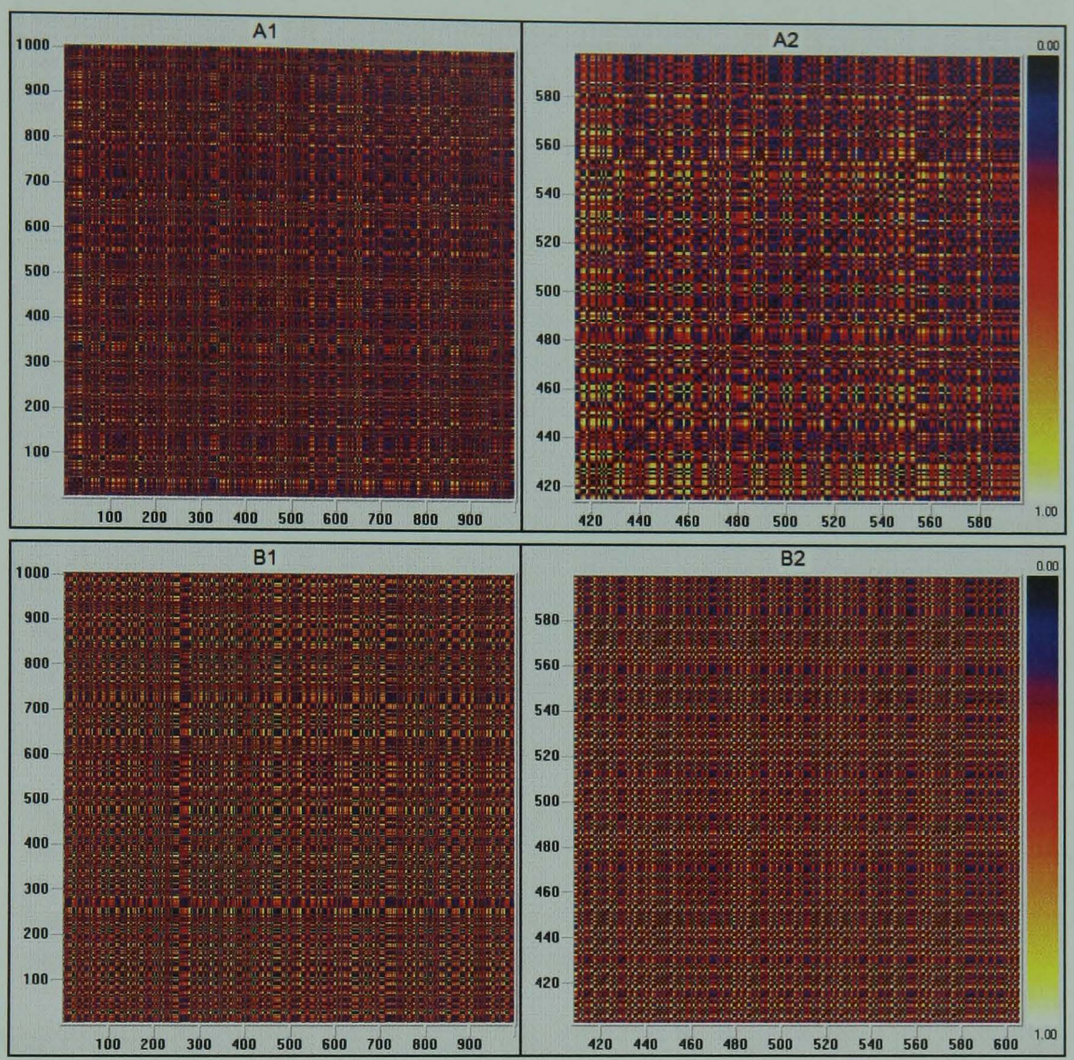


Figure 5-7. Global distance RPs of white noise (A1 and A2 (zoomed in area)) and the logistic function (B1 and B2 (zoomed in area)). The expected diagonal structure is not visible in the logistic function B1 & B2 plots, both panels A & B appear to be produced by the same or similar process. The embedding was not used i.e. ($m = 1, \tau = 0$).

By examining figure 5-7 it is not possible to distinguish between the purely random process (panel A) and the low dimensional chaotic system (panel B). It is only when the upper distance bound is considerably reduced that these two panels looks very different (figure 5-8). The short diagonal lines of the logistic function plot, that are an indication of a low-dimensional chaotic system, are clearly visible in panel B.

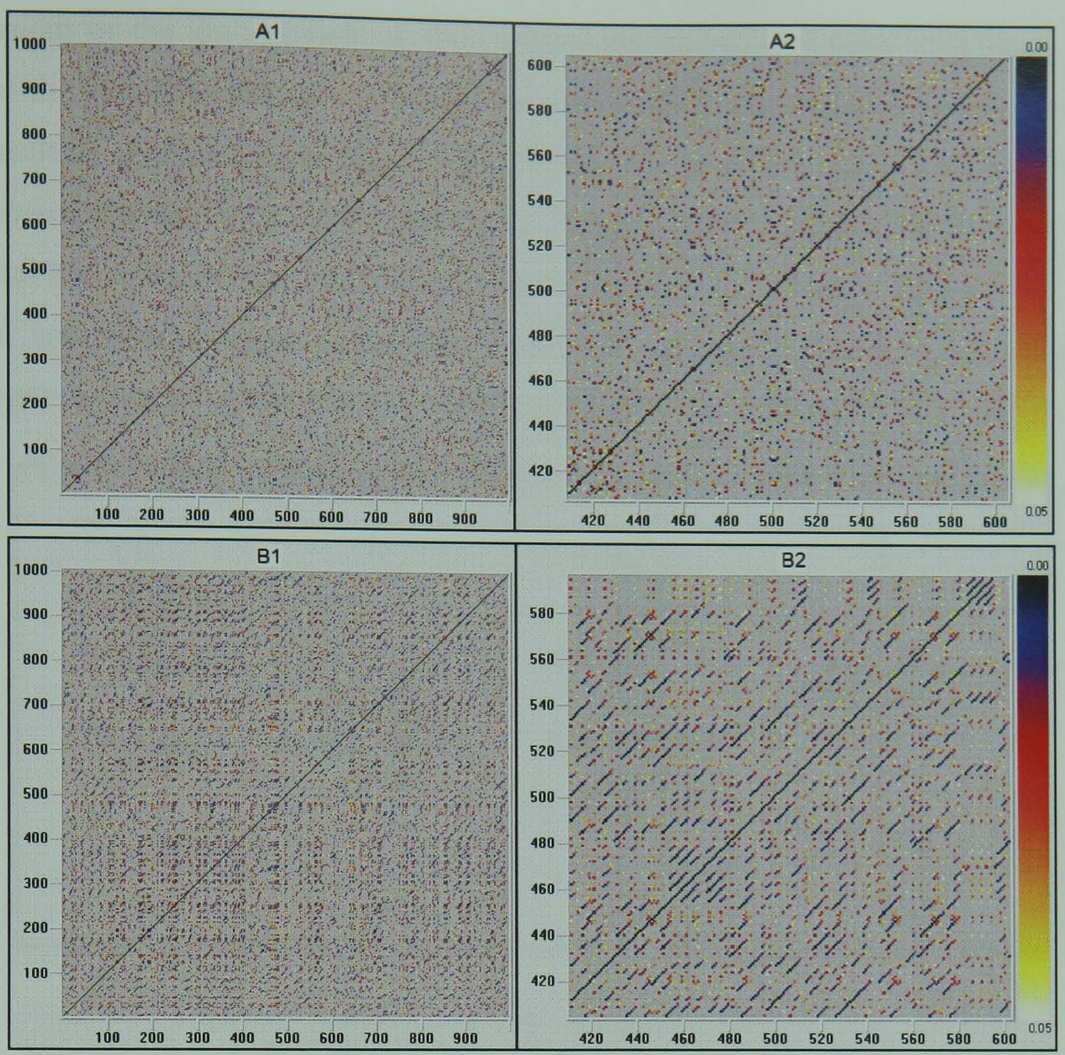


Figure 5-8. RPs of white noise (A1 and A2 (zoomed in area)) and the logistic function (B1 and B2 (zoomed in area)) produced using the 0.05 value as an upper distance bound (all distances above this value are shown in grey). The expected diagonal structure is clearly visible in the logistic function panel B. Short diagonal lines forming different patterns are common features of chaotic time series. The embedding was not used i.e. ($m = 1, \tau = 0$).

Two trajectories that move parallel to each other in the same direction produce a line parallel to the LOI (figure 5-9), and the trajectories that move in an opposite direction produce a line perpendicular to the LOI.

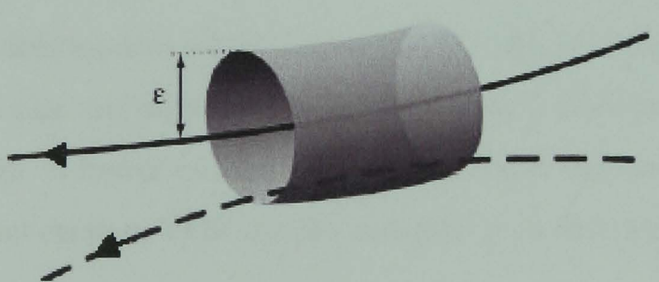


Figure5-9. A diagonal line in RP corresponds to trajectories passing in the same region of the state space at a different time, within a ε distance.[1]

[1] Source: [90].

The length of the line depends on how long these trajectories move together. In the case of the sine function, two points move continuously at the same distance apart, so the resulting lines have no gaps. This is not the case for chaotic systems as the trajectories, though they have periodicity diverge apart quickly, resulting in many diagonal and vertical lines of different length (figure 5-6 C, Lorenz time series). For chaotic systems, the length of the largest diagonal segment is inversely proportional to the largest Lyapunov exponent and can be estimated directly from the RPs[1] [80].

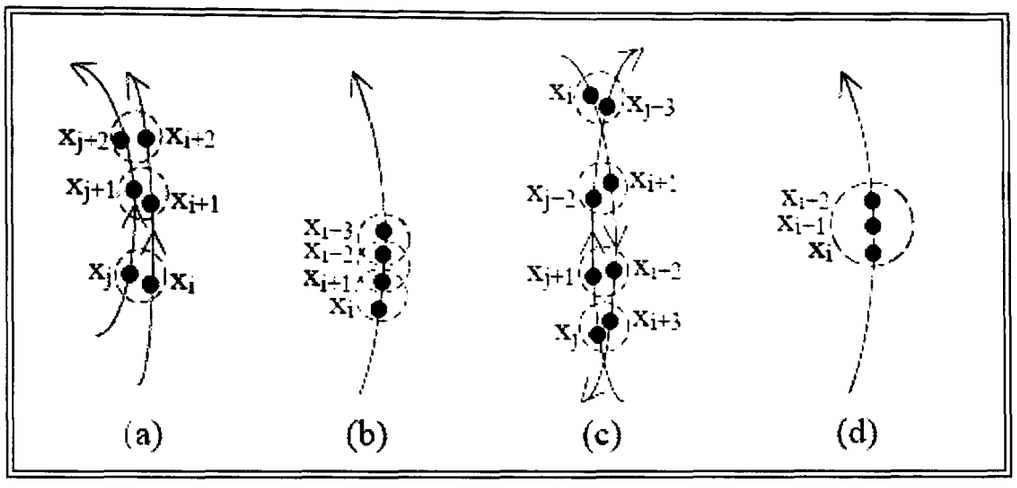


Figure 5-10 Trajectory examples that generate diagonal lines parallel to the LOI (a) and (b), lines perpendicular to the LOI (c), and horizontal and vertical lines (d).[2]

The chaotic time series exhibit non-stationarity which is clearly visible from plots C and D, compared to the uniform, symmetric and stationary time series shown in plots A and B. The RPs of chaotic systems also show a certain regularity but with more complex features. The vertical and horizontal lines are more pronounced indicating that the system state is stable or changes very slowly in time. This is typical behaviour of non-stationary processes with a drift.[3] The RP darkens or pales (depending on the colour scheme) away of the LOI.

Dark blue and black dots represent long distances and isolated states, and therefore the dark bands indicate rare or extreme events which are visible in figure 5-6 F (the Microsoft share price). A better example of similar extreme events are demonstrated in figure 5-12, the market crash in 1929 and the technology bubble burst in the year 2000.

---

[1] As mentioned before the largest Lyapunov exponent measures the average divergence rate of the nearby trajectories. Also, many other dynamical invariants can be estimated from the RPs: Hausdorff fractal dimension [79], multi-fractal properties of return time statistics [81], entropies etc.

[2] Source: [89].

[3] Also known as laminar or intermittency states.

It is remarkable how these two events that are seventy years apart, resemble each other. The white square patterns seen on the RPs are a sign of market periodicity, i.e. sideways moves. These white squares appear on different scales in a similar fashion, as we zoom into the highlighted square area shown in the panel above. The corresponding "zoomed in" area is shown in the B and C panels below. This illustrates the typical fractal property of financial time series.

Another interesting property can be observed from the plots. As we zoom into smaller squares we can see that the distance between the white/yellow squares (representing oscillations) are getting closer as their size (amplitude) is decreasing, which is a typical property of log-periodic oscillations (figure 5-11). It is a well known fact that these log-periodic oscillations are often observed before extreme events occur, such as in the case of an earthquake. Mirror image dynamics can be observed after a market crash.

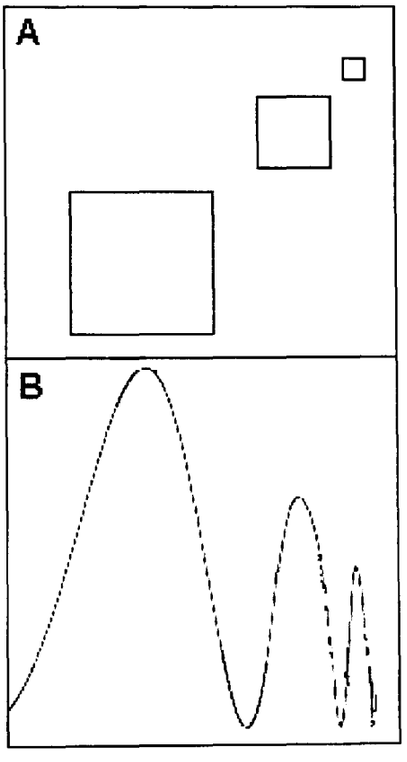


Figure 5-11. The white square areas (A) visible in RPs represent log-periodic oscillations (B) and are seen before market crashes.

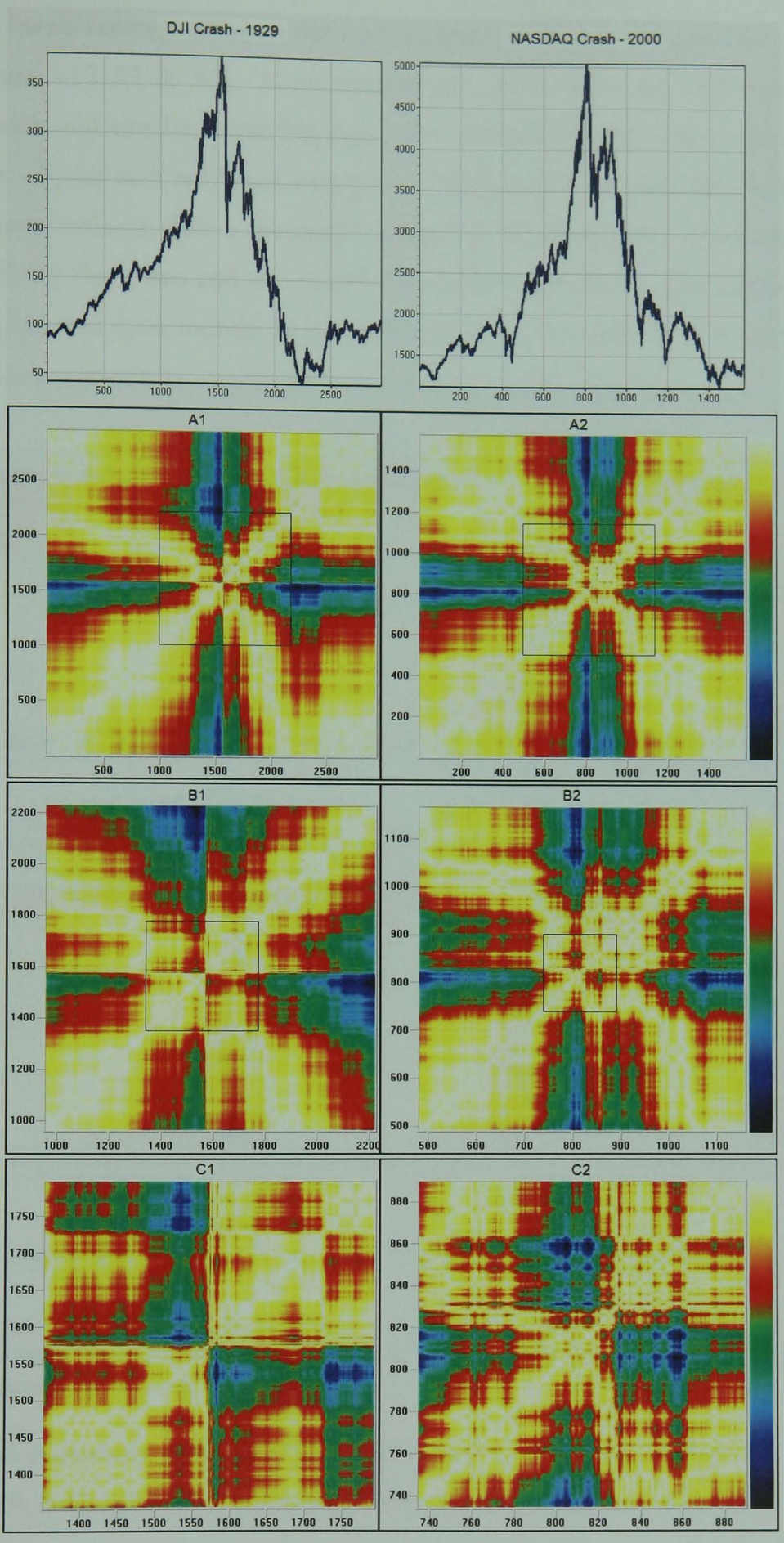


Figure 5-12. "Fingerprint" of DJI market crash in 1929 (A1, B1 & C1) and NASDAQ 2000 (A2, B2 & C2). The embedding parameters used are $m = 1, \tau = 0$.

The dark bands corresponding to the crash periods are clearly visible in the recurrence plots (figure 5-12 A1 & A2). More interestingly, these bands are visible prior to the market crash and can be a warning sign of an unstable market and a possible sharp downturn. As the dark red/black smudges or bands start to narrow, the likelihood of a crash is more eminent. This is illustrated in figure 5-13, showing the recurrence plots of the NASDAQ three, two and one month before the crash (top panel) and the index at the peak, two and three months later (bottom panel) . The yellow/white arrow shaped patterns are a sign of non-stationarity and a strong trend in time series.

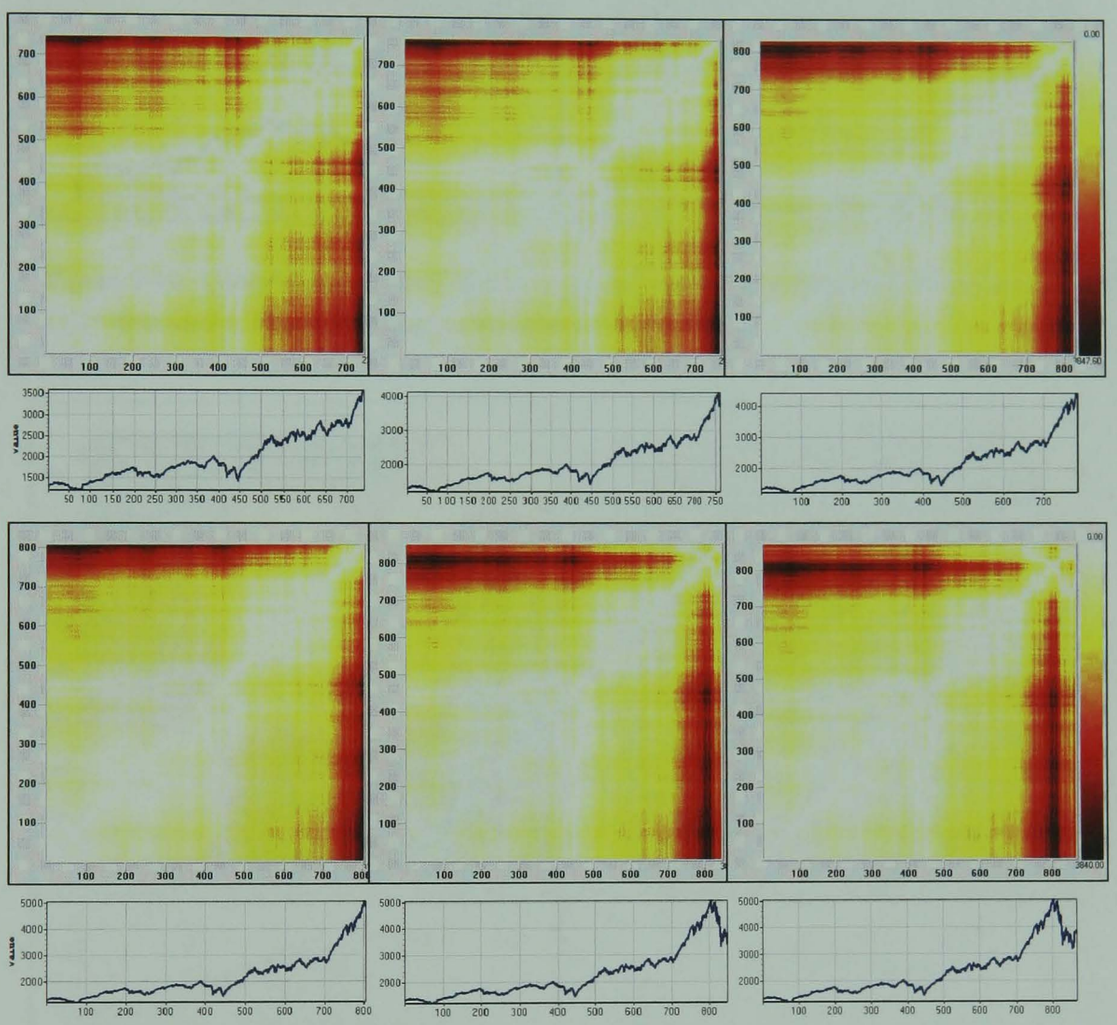


Figure 5-13. The RPs of the NASDAQ three, two and one month before the crash (top panel) and the index at the peak, two and three months later (bottom panel). The embedding parameters used are $m = 1, \tau = 0$.

The global recurrence plots of DJI*, DJI*-F and DJI*-FD time series are shown in figures 5-14, 5-15 and 5-16 respectively on the left side, whilst the 'thresholded' plots are shown on the right hand side. The RPs of the DJI* (figure 5-14) and filtered DJI*-F time series (figure 5-15) are very similar, except that the first one is sharper and more

detailed. The filtering effect produced a blurred and less detailed RP (DJI*-F).

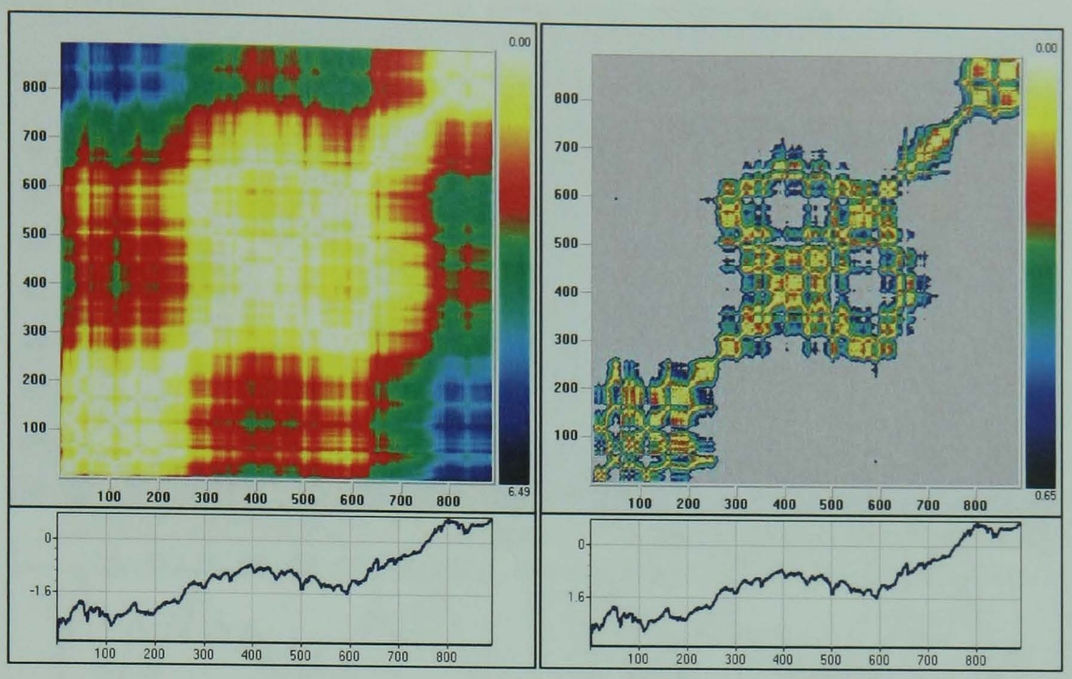


Figure 5-14. Global distance recurrence (on the left) and the 'threshold' plot (on the right) of the DJI* time series. The embedding parameters used are $m = 3, \tau = 1$.

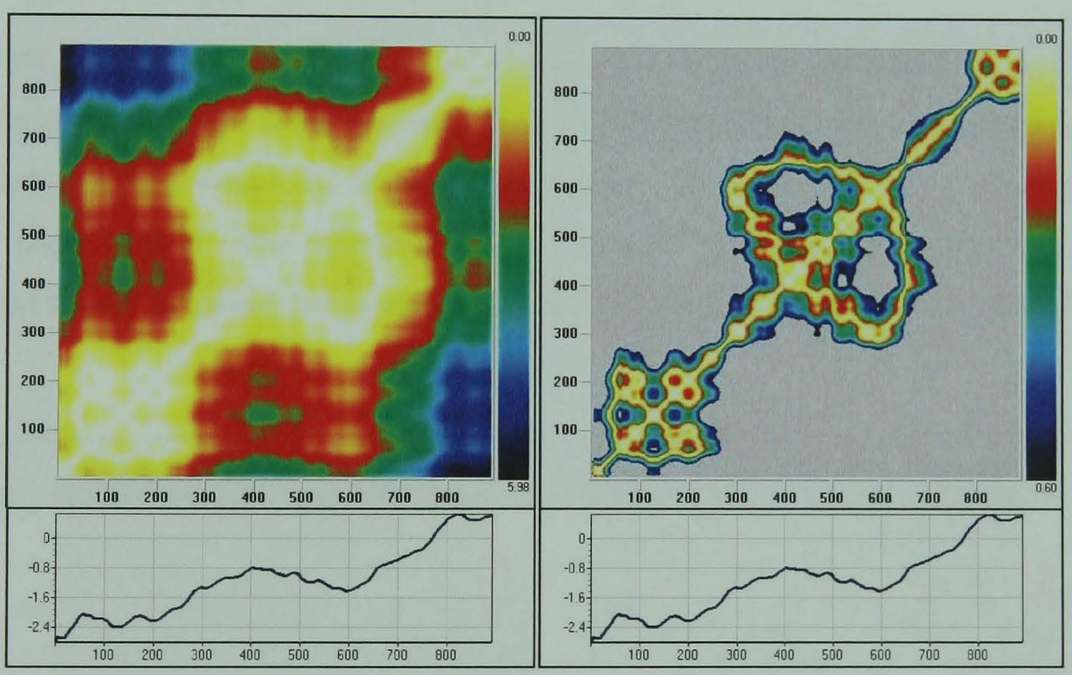


Figure 5-15. Global distance recurrence (on the left) and the 'threshold' plot (on the right) of the DJI*-F time series. The embedding parameters used are $m = 3, \tau = 1$.

The RPs show oscillatory periods represented as the white/yellow wide areas and the trending periods shown as narrow white/yellow bands along the LOI line. The DJI* and the DJI*-F time series exhibit larger drifts, while the DJI*-FD (figure 5-16) recurrence plot show more structure and complexity. There is a period (650-800) in this

plot that stands out. It represents the DJI behaviour from 1987 to 2000, where the operating market regime looks different compared to the other periods.

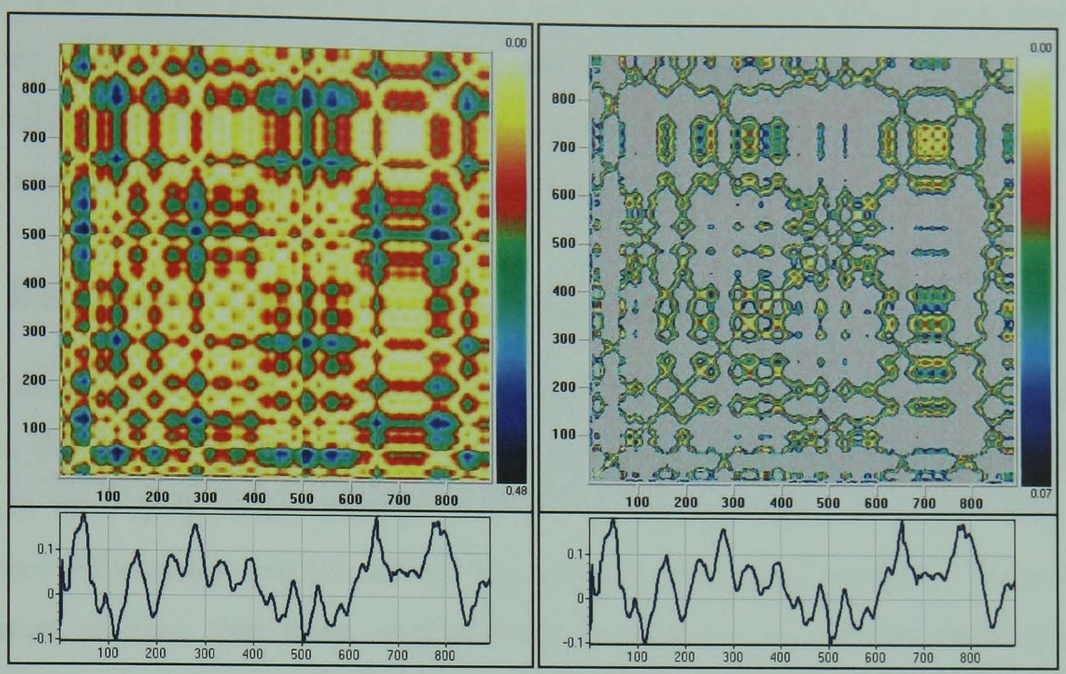


Figure 5-16. Global distance recurrence (on the left) and the 'threshold' plot (on the right) of the DJI*-FD time series. The embedding parameters used are $m = 3, \tau = 1$.

To show the relevance of the order of the data points in the financial time series two RPs are generated, one using the original DJI*-FD time series, and the second using a randomly shuffled sample of the same time series. The resulting plots are shown in figure (figure 5-17). The new re-shuffled time series preserved the same mean and standard deviation values, but completely destroyed the RP structure of the original time series.

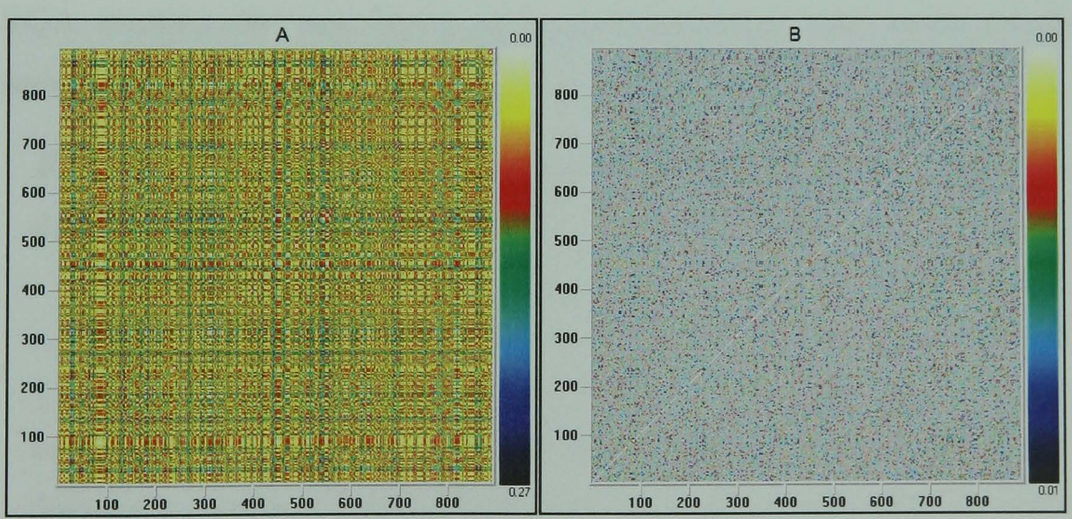


Figure 5-17. An example of the global distance RP of the randomised DJI*-FD time series (A) and the 'threshold' plot of the same time series (B).

The problem with RPs is that they can often be misinterpreted. The number of vertical and diagonal lines increases as the embedding dimension grows. These can be misinterpreted as the presence of an extreme event and determinism. A large time delay together with noise can amplify the number of vertical lines. A more reliable interpretation of RPs is obtained by means of the Recurrence Quantification Analysis, which mathematically quantifies many important features of the RP.

5.4 Recurrence Quantification Analysis (RQA)

To avoid sole reliance on the visual interpretation of RPs that may result in inconclusive results, the RQA was introduced by Zbilut and Webber [78] and developed further by Marwan et al. [87]. They introduced a set of statistical measures from which more objective information can be derived than purely from visual analysis. Some of the important features that these statistics can quantify are the deterministic structure, stationarity, periodicity and complexity of the plot. They can also help to analyse changes in the state of a dynamical system as it evolves over time.

5.4.1 RQA Measures

The starting point for RQA is the equation (5.1), rewritten in a more intuitive form, which identifies whether a point is recurrent or not:

$$R_{i,j} = \begin{cases} 1, & if \ \|\vec{x}_i - \vec{x}_j\| < \varepsilon \\ 0, & if \ \|\vec{x}_i - \vec{x}_j\| \geq \varepsilon \end{cases} \tag{5.2}$$

The RQA statistics that are based on the distribution of the length of diagonal structures in the RP are:

- ***Recurrence Rate*** (*REC*) or percentage recurrences count the number of black dots (recurrence points) in the RP.

$$REC = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{i,j}(\varepsilon) \tag{5.3}$$

  In order to calculate the above statistic it is necessary to specify a threshold distance ε, which is a crucial parameter for a correct interpretation of RPs. If the threshold is either too small or too large, the number of recurrence points would also be too small or too large, leading to erroneous conclusions. The

selection of ε depends mainly on the time series in question. Some authors propose a constant value proportional to the standard deviation of data [88], some as a proportion of the average distance of vectors [78] and others require that a minimum number of points enter the neighbourhood of each vector [77]. If the radius is equal or larger than the maximum distance in RP, then every point will be recurrent and both *REC* and *DET* (see the next measure) will saturate at 100%. The ε threshold can be selected to give a *REC* value of between 0.1% and 2.0% and the larger values of up to 5% can be used to obtain values for the *LAM* and *TT* measure presented below. This would ensure that the trajectories being compared are close in distance, but are above the noise floor.

The following formulas presented here are functions of ε but for the sake of simplicity, ε will not be specified, i.e. $P(\varepsilon,l)$ is written as $P(l)$ etc.

- ***Determinism*** (*DET*) represents the percentage of recurrent points that form diagonal lines of a minimum specified length $l_{\min}$. This measure is important, as it quantifies the amount of determinism (or predictability) in the time series and is equal to the ratio of recurrence points that form diagonal lines (of at least length $l_{\min}$) and all the recurrence points.

$$DET = \frac{\sum_{l=l_{\min}}^{N} lP(l)}{\sum_{l=1}^{N} lP(l)} \tag{5.4}$$

Where $P(l)$ represents the probability of finding a diagonal line of length l in the RP. This distribution is defined as:

$$P(l) = \sum_{i,j=1}^{N} (1 - R_{i-1,j-1})(1 - R_{i+1,j+1}) \prod_{k=0}^{l-1} R_{i+k,j+k} \tag{5.5}$$

The *DET* value for the stochastic processes tend to be low and high for periodic deterministic processes, but it is largely dependent on the choice of $l_{\min}$. If $l_{\min}$ is equal to one, than the *DET* is equal to one. A large value of $l_{\min}$ causes the histogram $P(l)$ to be too sparse and the derivation of *DET* can be unreliable.

The average diagonal line length is defined as:

$$L = \frac{\sum_{l=l_{\min}}^{N} lP(l)}{\sum_{l=l_{\min}}^{N} P(l)} \tag{5.6}$$

- ***Ratio*** *(RATIO)* is defined as the ratio between *DET* and *REC*. These two measures should be closely correlated. Any departure from the correlation i.e. the *REC* decreases whilst the *DET* increases or does not change, and is highlighted by the *RATIO* measure which can be used to discover certain types of transitions in the system's behaviour [92].

$$RATIO = N^2 \frac{\sum_{l=l_{\min}}^{N} lP(l)}{\left(\sum_{l=1}^{N} lP(l)\right)^2} \tag{5.7}$$

- ***The longest diagonal line*** ($L_{\max}$) apart from the LOI is defined as:

$$L_{\max} = \max(l_i)\ i = 1,...N_l \tag{5.8}$$

where $N_l = \sum_{l \geq l_{\min}} P(l)$ is the total number of diagonal lines. It has been shown that *Lmax* is sensitive to the stability of the system in question. The larger the *Lmax*, the more stable the system.

The inverse of $L_{\max}$ is a measure of divergence of nearby trajectories and is linked to the largest positive Lyapunov exponent.

$$DIV = \frac{1}{L_{\max}} \tag{5.9}$$

A higher value of *DIV* implies faster divergence and shorter diagonal lines, which are typical characteristics of chaotic systems.

- ***Entropy*** *(ENTR)* is a Shannon entropy of the frequency distribution of diagonal lines in RP and is recognised in the variety of RP structure.

$$ENTR = \sum_{l=l_{\min}}^{N} p(l) \ln(p(l)) \tag{5.10}$$

where $p(l) = P(l) / \sum_{l=l_{\min}}^{N} P(l)$ is the probability of finding a diagonal line of length l in RP. This measure quantifies the complexity of the deterministic structure in the RP in both a space and time domain. It compares the global distribution of recurrence points over the entire RP with the distribution of recurrence points over each diagonal line. The higher the differences the more structured the image is. A high entropy means the absence of structure (uniform distribution and randomness), whilst a low entropy implies structure (distinct patterns and predictability). The reconstructed time series would have entropy near its minimum for the optimal embedding parameters, and therefore, experimenting with different values that yield the lowest entropy can help to select a good embedding dimension and time delay. However, for a non-stationary time series with trend this measure could be misleading.

- ***Trend*** (*TREND*) is a linear regression coefficient (or slope of line) of best fit through a recurrence rate (*REC*) as a function of displacement (or time distance) from the main diagonal line.

$$TREND = \frac{\sum_{i=1}^{\tilde{N}} \left(i - \tilde{N}/2\right)\left(REC_i - \langle REC_i \rangle\right)}{\sum_{i=1}^{\tilde{N}} \left(i - \tilde{N}\right)^2} \tag{5.11}$$

The formula excludes the edges of the RP ($\tilde{N} < N$) because of an insufficient number of recurrence points in this region [91]. The TREND measures a fading of the patterns of RP away from the LOI and is used to detect drift and non-stationarity in a time series, in which case the value becomes strongly negative. If the density of recurrence dots decreases away from the LOI line, it is because there is a drift in the time series' mean. It should be noted that the LOI line corresponds to the same time and the dots away from the central diagonal line represent a deviation in time.

An analysis of the vertical structure in RPs can be used to detect chaos-chaos transitions as they occur. The latest RQA measures based on the distribution of the length of the vertical structures in the RP are:

- ***Laminarity*** (*LAM*) is analogous to *DET*, and is defined as the percentage of

recurrence points that form a vertical line of a minimum specified length v_{min}. This measure quantifies the occurrence of laminar (slow changing) states in a given trajectory.

$$LAM = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=1}^{N} vP(v)} \tag{5.12}$$

Where $P(v)$ represents the probability of finding a vertical line of length v in the RP. The histogram of a vertical line length is defined as:

$$P(v) = \sum_{i,j=1}^{N} (1 - R_{i,j})(1 - R_{i,j+v}) \prod_{k=0}^{v-1} R_{i,j+k} \tag{5.13}$$

- ***Trapping Time*** (TT) is analogous to L, representing the average length of vertical structures.

$$TT = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{v=v_{min}}^{N} P(v)} \tag{5.14}$$

TT estimates the mean time of a system being trapped in a particular state.

- ***The longest vertical line*** (V_{max}) is defined as:

$$V_{max} = \max(v_i)\, i = 1,...N_v \tag{5.15}$$

where $N_v = \sum_{v \geq v_{min}} P(v)$ is the total number of vertical lines.

Measures based on vertical structure are used to uncover chaos-chaos [87] and chaos-order transitions [90], [93].

5.4.2 RQA analysis of financial time series

The RQA measures above can be calculated for the entire RP window (whole time series) as well as many smaller and overlapping squares. The large window measures focus on global dynamics, whereas smaller windows (termed epochs) focus on local dynamics and changes in RQA measures over time. A graphical illustration of the

epoch construction is shown in figure 5-18.

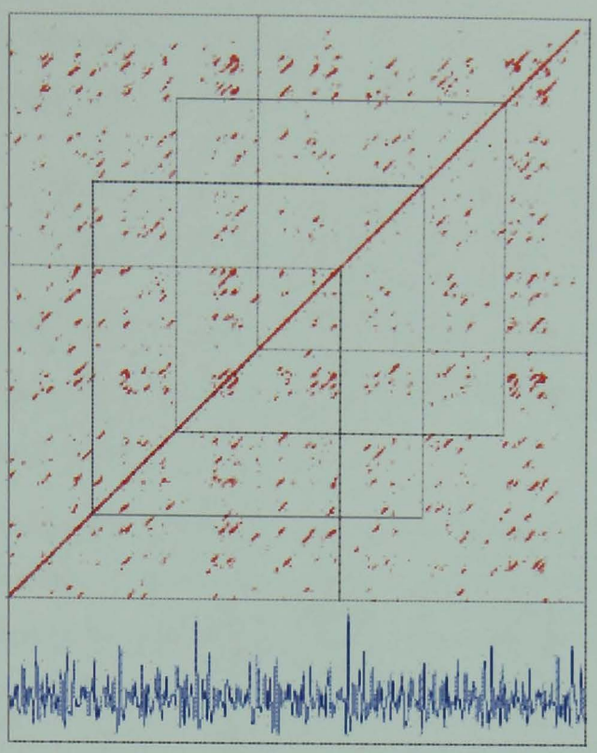


Figure 5-18. Example of a time series divided into four epochs.

The results of the RQA variables for DJI*, DJI*-F and DJI*-FD computed on the whole time series are presented in table 5-1. All three time series are comprised of 873 monthly recordings. The standard parameters required for the RQA calculations for the whole and the epoch-by-epoch basis are set as follows:

- The *embedding dimension* m of 3 was used for all three time series which are based on the correlation dimension and the false nearest neighbour estimates. As shown in sections 4.4.1 and 4.4.2 both these methods estimated a correct dimension in the case of logistic function.

- The *time delay* τ of 1 is used in all calculations. This rather low value is used for several reasons. Firstly, the average mutual information estimates for the time delay calculated in section 4.5 were found to be unreliable, even in the simple case of the logistic function. Secondly, the sampling frequency for the financial time series in question is sparse (on a monthly basis), and some authors suggest that for discrete systems a τ value of 1 is appropriate [78]. This low value is particularly important in the event that these time series are generated by chaotic systems, which are intrinsically unpredictable and lose memory of the initial state as the time delay is increased. For shorter sampling frequencies measured in

days, hours or minutes, a longer time delay would be more appropriate. □

- The RP matrix was derived using the *Euclidian distance* norm and *rescaled to a maximum* calculated distance. □
- For comparative reasons, the *radius* for each time series was chosen such to produce the same value of *REC* (2.0%) for all three time series. □
- *Line length* specifies the number of consecutive recurrent points required to define a line segment. Often, the line length is set at 2 points, but in this case a more conservative value of 5 was used. □

Table 5-1. RQA measures of DJI*, DJI*-F and DJI*-FD calculated for whole time series.

	DJI *	**DJI*-F**	**DJI*-FD**
Mean value	-1.114	-1.167	0.035
Standard deviation	0.912	0.900	0.063
Mean rescale distance	27.250	29.334	26.097
Embedding dimension (m)	3	3	3
Time delay (τ)	1	1	1
Line length	5	5	5
Radius (ε)	1.0863	0.5891	1.5934
REC	2.000	2.000	2.000
DET	39.337	72.664	44.698
RATIO	19.671	36.327	22.346
Lmax	42	450	65
ENTR	2.865	3.925	2.922
TREND	-7.127	-7.340	-2.012
LAM	35.982	70.760	32.543
TT	7.297	9.096	7.332
Vmax	33	23	16

All three time series show the presence of deterministic features, in particular the DJI*-F time series having a *DET* value of 73%. The reason for this higher value compared to the other two time series could be the filtering that has been applied to this time series. The smoother time series have consequent points closer in space, causing more recurrent points as illustrated in figure 5-10 case (b), or simply this signal is more deterministic when compared to the other two. Likewise, the high *LAM* values reveal a

significant structuring of recurring dynamical features in the vertical plane. As expected, both the DJI* and the DJI*-F time series exhibit a stronger trend than the DJI*-FD time series. The high value of *Lmax* for the DJI*-F indicates more stability present within the time series.

To show the relevance of the order of the data points in these time series, the same RQA statistics were calculated for randomly reshuffled samples of the same time series. The results are presented in table 5-2. These new re-shuffled time series preserved the same mean and standard deviation values, but completely destroyed the deterministic structure of the original time series causing all RQA measures to be virtually zero.

Table 5-2. RQA measures of DJI*, DJI*-F and DJI*-FD calculated for randomised time series[1].

	DJI*	**DJI*-F**	**DJI*-FD**
Mean value	-0.114	-1.167	0.035
Standard deviation	0.912	0.900	0.063
Mean rescale distance	39.129	38.143	36.882
Embedding dimension (m)	3	3	3
Time delay (τ)	1	1	1
Line length	5	5	5
Radius (ε)	1.0863	0.5891	1.5934
REC	0.009	0.002	0.010
DET	0.000	0.000	0.000
RATIO	0.000	0.000	0.000
Lmax	0	0	0
ENTR	0.000	0.000	0.000
TREND	-0.009	0.009	0.007
LAM	0.000	0.000	0.000
TT	0.000	0.000	0.000
Vmax	0	0	0

The RQA is carried out next on an epoch-by-epoch basis, with the aim that the changing values of RQA variables in the subsequent windows will allow for the detection of abrupt changes in the dynamical regime of the signal. The analysis was carried out using the same parameter setting used for the whole time series, except the

[1] The softwares used for RQA are Visual Recurrence Analysis (VRA) and Recurrence Quantification Analysis.

radius was set larger (3.5 for DJI* and DJI*-F and 2.47 for the DJI*-FD)[1] in order to gain values for most RQA measures in all epochs. Each epoch length was set to 120 points (10 years) with an overlap of 24 points (2 years) between consecutive windows.

The results of the epoch-by-epoch RQA analysis is presented in figure 5-19. The first row shows RPs of the whole DJI*, DJI*-F and DJI*-FD time series, which are shown in the fourth row respectively. For comparative purposes the second row shows the original DJI time series and the its logarithm is presented in the third row. The RQA measures *REC, DET, RATIO, Lmax, ENTR, TREND, LAM, TT* and *Vmax* are shown in the 5th to 13th row respectively.

Following the *DET* variations it is possible to distinguish several levels and to study how the forecasting capability would change according to the level of DET observed in the time series. The DJI* time series has higher values during the trending periods, with peaks in epoch 9 and 30. The filtered DJI*-F time series exhibits two plateaus of high *DET* that coincide with strong trending periods in *log*(DJI) series (epochs 9-13 and 26-30), whilst the DJI*-FD time series has a high value in *DET* during the oscillatory period of *log*(DJI) (epochs 14-18). This highlights the need for adequate pre-processing of time series depending on the type of forecasting model used.

It is noticeable that high values of *REC, DET, Lmax, ENTR* and the absolute value of TREND are associated with more deterministic periods, and most importantly, the lowest values of these measures (in particular *Lmax*) can be used to identify transition periods. Note that these measures take the lowest values at the beginning of the transition period in epochs 3-5 and 19-21, just before the start of a big upward rally in the DJI index. The *RATIO* measure is important as it can also detect transitions between states. This ratio has abrupt jumps in transition periods and settles down when a new 'steady' state is achieved. Two peaks in epoch 21 for DJI* and epochs 19-20 for DJI*-FD coincide with the above findings.

[1] The radius value for all three time series was set as the average of the three estimates: 5σ, 10% of the mean rescaled distance and the radius value that generates a *REC* of 10%.

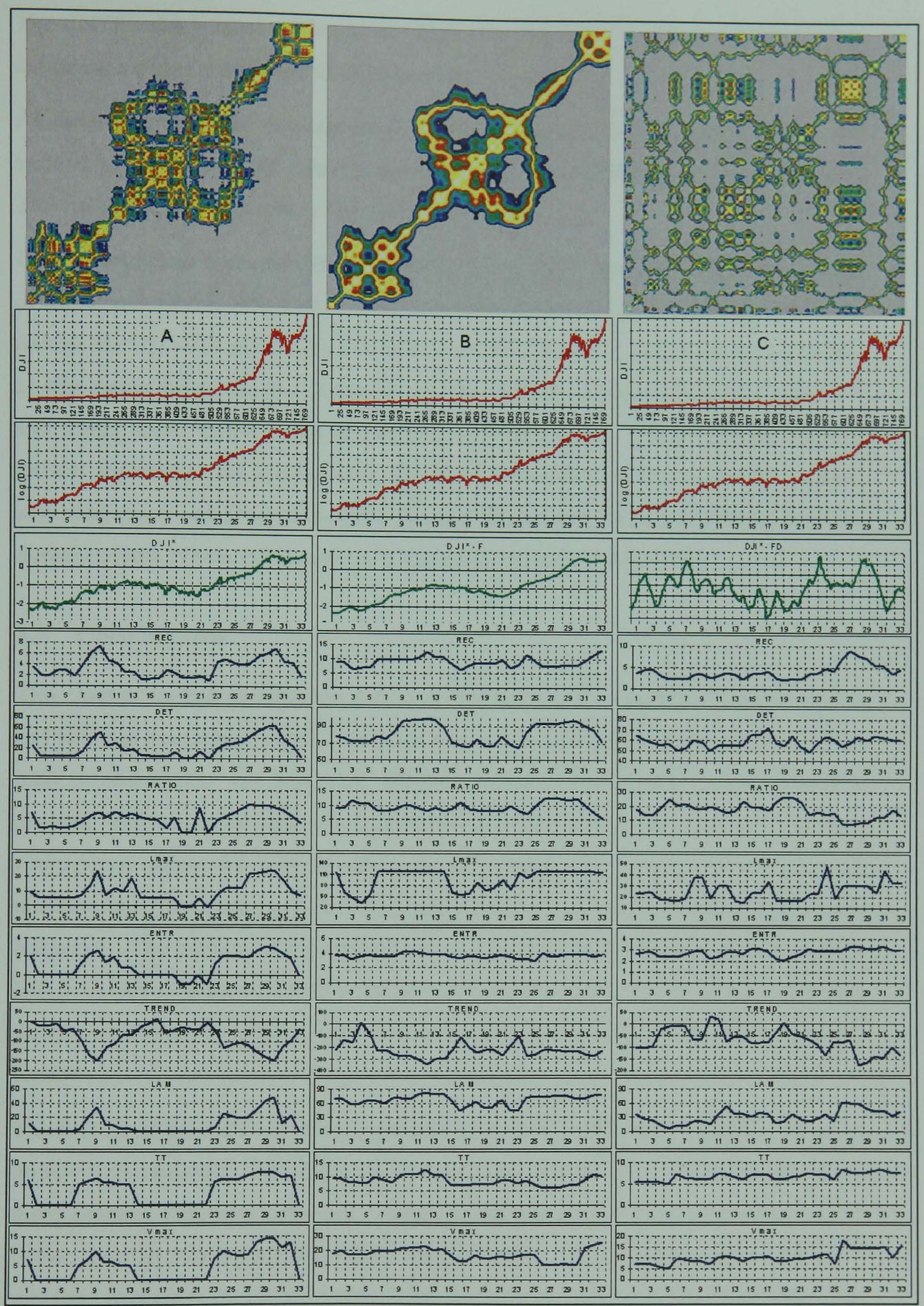


Figure 5-19. RQA measures (*REC, DET, RATIO, Lmax, ENTR, TREND, LAM, TT and Vmax*) for DJI* (panel A), DJI*-F (panel B) and DJI*-FD (panel C).

The ENTR value is more stable and also higher for the DJI*-F and DJI*-FD time series indicating a higher complexity of these two time series.

All three time series are non-stationary, which can be observed directly from RPs and verified by a changing and a strong negative value of the *TREND* measure over time. The stationary time series have *TREND* values close to zero.

Chaotic behaviour is characterised by a high value of *DET* and *REC* and a small value of *Lmax* and *ENTR* [86], [94] . There are no clear epochs that satisfy these criteria, though the DJI*-FD epochs 26-28 exhibit a similar behaviour, indicating the possibility of this region being chaotic. Interestingly, the vertical structure measures, *LAM*, *TT* and *Vmax* show visible peaks in the same epochs (26-28). These statistics show a distinct maxima at the chaos-chaos transitions and fall to zero in periodic regions, indicating chaos-order transitions [87]. But, *LAM*, *TT* and *Vmax also* attain large values during extreme events and in regions of intermittency (i.e. laminar states)[1].

5.5 Conclusions

This chapter applied visualisation and analytical tools in the analysis of several chaotic and financial time series. It has been shown that state-space and recurrence plots can be used to characterise and recognise different regimes within a time series. The RPs preserve both temporal and spatial dependence in the time series and don't use any assumptions about the data, i.e. they are not constrained to any type of statistical distribution. Two market crashes, 1929 and 2000, were analysed showing very similar behaviour. The markets fractal structure and log-periodic oscillations, typical of periods before extreme events occur, have been revealed through recurrence plots. Also, the dark bands that are a warning sign of extreme market levels are visible prior to market crashes.

The RQA analysis can be used as a powerful tool to detect hidden properties driven by nonlinear mechanisms. The RQA analysis indicated the presence of determinism in all three financial time series studied. Crucial transition periods were also detected just before the start of a big upward rally in the DJI index. The typical values of the RQA measures that are common to chaotic behaviour are not present in the results (figure 5-19), though there are some indications that they may exist during shorter periods of

[1] Long periods of relative stability, interrupted by fairly short periods of chaos.

time.

The procedure is robust to noise, and it does not impose any kind of restrictions on the data. Although the operation of the programs is straightforward, the selection of RQA parameters and the interpretation of RQA variables can be difficult especially in the case of complex financial time series.

The next chapter will examine how neural networks can be used to forecast low-dimensional chaotic time series as well as complex real-world financial time series.

Chapter 6

Nonparametric Time Series Forecasting

6.1 Introduction

Forecasts play a big part in life and we use them all the time to guide our decisions. Companies forecast using various measures to guide their decisions in marketing, strategic planning and control. Governments around the world routinely forecast major economic variables to guide their monetary and fiscal policies. Financial institutions have great interest in this area and use forecasting techniques in their trading and financial risk management. However, there is endless debate about the success of financial time series forecasts as they are very difficult to predict. Simple and widely available methods for forecasting usually have little success, but new and sophisticated techniques have a higher chance of uncovering and exploiting previously unnoticed patterns in financial data. This effect usually only lasts for a limited period as other market participants catch on. The existing models and their parameters need to be adjusted regularly and replaced with new ones from time to time.

The previous chapter showed that the financial time series examined do not fluctuate randomly, and on the contrary, they exhibit a certain amount of determinism. The methods that can be used to estimate model's parameters from the historical data observed were also highlighted, such as the embedding dimension, time delay, the forecasting horizon and the complexity of the model. In this chapter we will apply that knowledge and compare several non-parametric forecasting methods to our three financial time series. Their predictions will be used to derive and evaluate the trading models.

In general, financial prices are influenced by a myriad of external influences, but in the case of embedding, an attempt is made to discover the self-referential hidden rules within the price movement itself. This chapter will focus on "univariate" modelling, where a single variable is predicted solely on the basis of its own past. The next chapter will present some "multivariate" models, where the time series forecasts are based on

their own data, as well as other data variables. The univariate approach may seem simplistic, however, the previous chapter has demonstrated that the dynamics of the chaotic time series can be forecasted successfully by means of proper embedding.

6.2 Nonparametric and local modelling

This research focuses on nonparametric time series predictions, which do not make any assumptions about the functional form of the process that generate the observable time series. The models and their parameters are derived directly from data using various optimisation techniques, including neural networks models. In addition to NNs, other closely related local linear and nonlinear models are tested and their performance is compared. The local models fit many simple models to small portions of the data set, instead of fitting one complex model with many coefficients to the entire data set. These local models change their parameters adaptively depending on the geometry of the dynamical system's local neighbourhood. Essentially they are regression models whose parameters change over time. Most of these models use a similar approach; the only difference is in the choice of function that is used to map the input to the output values, and in some cases, the weighting function (kernels) that assigns the contribution of each neighbour to the prediction.

Essentially the prediction of next value x_{t+1} starts with last known state of the system, as represented by the delay vector $\vec{x}_t = \left(x_{t-(m-1)\tau},\ldots, x_{t-\tau}, x_t\right)$, where m is the embedding dimension and τ is the time delay. Then the algorithm searches the time series to find similar k states (nearest neighbours) that have occurred in the past, based on the distance between vector $\vec{x}_t$ and its neighbour, vector $\vec{y}_t$. If the observable time series was generated by some deterministic process, then x_{t+1} can be estimated by approximating the function $x_{t+1} = f(\vec{x}_t, \vec{y}_t)$ that maps the similar states $\vec{y}_t$ onto their immediate values.

As shown in the previous chapter, short-term predictions of low-dimensional chaotic time series are possible. This is demonstrated in the example of the logistic equation shown in figure 6.1. This approach can reliably forecast chaotic time series one step ahead, using a neural network model. The model was optimised over 2000 data points

and out-of-sample predictions were generated for the next 1000 points with only 200 predictions shown in the graph for reasons of visual clarity. An embedding dimension of 2 and a time delay of 1 were used to reconstruct the dynamics of time series.

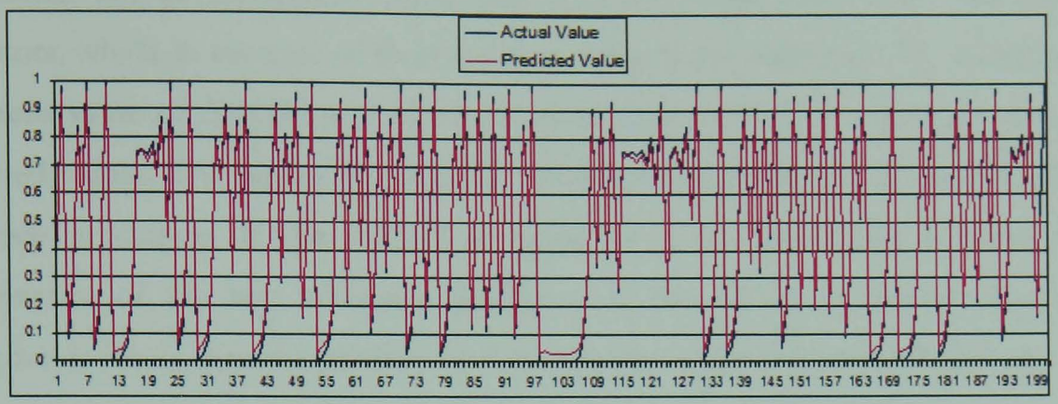


Figure 6-1. An example of a one step ahead out-of-sample forecast of the Logistic Map chaotic time series ($\lambda = 4$, x(0)=0.1) using the neural network model.

6.3 Performance measures

In order to evaluate the performance of forecasting models, the accuracy of the predictions generated are compared, with two reference predictors: the unconditional mean and the "random walk" predictor. These performance measures are known as the Normalised Mean Square Error (*NMSE*), where in the previous case, our prediction error is normalised with a squared error of the mean predictor:

$$NMSE_1 = \frac{\sum_{t=1}^{N}(x_t - x_t')^2}{\sum_{t=1}^{N}(x_t - \bar{x}_t)^2} \tag{6.1}$$

In the second case the prediction error is normalised with a "random walk" predictor which assumes no change, i.e. the next value will be the same as the last known one:

$$NMSE_2 = \frac{\sum_{t=1}^{N}(x_t - x_t')^2}{\sum_{t=1}^{N}(x_t - x_{t-1})^2} \tag{6.2}$$

Where x_t is the actual value, x_t' is the out-of-sample predicted value, and $\bar{x}_t$ is the average actual value over the out-of-sample period $t = 1, \ldots, N$.

The model performance measure is compared to the best performing reference

predictor and is defined as:

$$NMSE = \max(NMSE_1, NMSE_2) \tag{6.3}$$

The model can be considered useful only if it can outperform these two reference predictors, which in the case of financial time series is very difficult. By definition, the minimum value of *NMSE* is 0 representing an exact match between the actual and predicted values. The higher the *NMSE*, the worse the prediction is compared to the reference predictors. If the *NMSE* is equal to 1, the prediction is equal to the performance of the best reference predictor. If the *NMSE* is greater than 1, the prediction is worse than the prediction of at least one of the reference predictors.

The Root Mean Square Error (*RMSE*) will also be presented, which is defined as:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}(x_t - x_t')^2}{N}} \tag{6.4}$$

The model's predictions for the logistic function are very accurate, the *NMSE* for the prediction of 1000 points one step ahead is only 0.003705 and the *RMSE* is 0.0214. That means that our predictive model explains about 99.63% of the variance in the series, confirming that it is possible to obtain accurate short-term predictions for the chaotic and low-dimensional time series. As explained earlier, reliable long-term forecasts are impossible to achieve due to sensitivity to the initial conditions and different parameter values causing the prediction error to increase exponentially in time.

6.4 Financial time series prediction

The results from the previous chapter have shown that three financial series (DJI*, DJI*-F and DJI*-FD) are generated from a low-dimensional system and the Recurrence Quantification Analysis confirmed the presence of a strong deterministic structure within them. We have also seen that the chaotic properties of a time series are of practical value for predictions, as their short term forecasts can be achieved by using time lagged vectors.

In this section a forecasting test of the financial time series will be performed using different local prediction models as well as the neural network approach. The local predictors evaluated are: the *locally weighted linear*, the *locally linear*, the *radial basis*, the *kernel*

regression, the *nearest neighbour* and the *locally constant* model. The first test was performed using the fixed neighbourhood size which was set to 10. All model parameters were optimised using 772 points (months), and the out-of-sample test was created on the next 120 points (10 years). An embedding dimension of 3 and a time delay of 1 was used in all models. Table 6-1 shows results for the models that could be estimated successfully to produce a low out-of-sample error measure. In the case of the *kernel regression*, the *locally weighted linear* ant the *radial basis* models, the Gaussian weighting function (kernel) was used. There are only three examples (highlighted in bold) where the estimated models outperformed the reference predictors; the *locally weighted linear* model in the case of the DJI*-FD time series and the *locally linear* model in the case of the DJI*-F and DJI*-FD time series.

Table 6-1. The performance of different predictive models using the 10 nearest neighbours and applied to the DJI*-F, DJI*-F and DJI*-FD time series.

	k	DJI*		DJI*-F		DJI*-FD	
		NMSE	RMSE	NMSE	RMSE	NMSE	RMSE
Locally weighted linear	10	5.2330	0.10293	2.4436	0.01566	**0.6101**	0.00345
Locally linear	10	9.7880	0.14077	**0.7972**	0.00895	**0.5462**	0.00327
Radial basis	10	-	-	-	-	1.0638	0.00456
Kernel regression	10	-	-	-	-	1.7098	0.00578
Nearest neighbour	10	-	-	-	-	2.4838	0.00697
Locally constant	10	-	-	-	-	2.4838	0.00697

The neighbourhood size value of 10 may not be an optimal choice for all predictors and time series. These predictive models are very sensitive to this neighbourhood "bandwidth" parameter which may drastically affect their performance. For the very large values of k, the local predictor becomes a global one, since it considers many points in the data set. In this case the generated predictions will be more stable and have lower variance, but their accuracy may also be low due to a large bias. If on the other hand, the bandwidth is very small, the predictor simply interpolates between points and the predictions generated may quickly diverge from the actual values after an initial period of higher accuracy. Though this parameter is set to a fixed value, the size of the actual neighbourhood depends largely on the shape of the attractor, i.e. the

neighbourhood will be larger in the sparse regions of the attractor and smaller in the dense region.

One way to find a good estimate for this value is a simple trial and error approach. In this particular case it is done through experiments that test the performance of many models using the first 54 years of monthly data points and the different neighbourhood size value (1 to 30). The ones that produced the best performance measure on the validation set of 120 months were selected. As in the previous experiment, these models were tested on a further 10 years out-of-sample period (120 points), but on this occasion using the estimated neighbourhood size. Table 6-2 presents the results of this test showing a large improvement and much lower error performance measures. The number of cases where the models also outperformed the reference predictors doubled, from 3 to 6. The estimated neighbourhood size is shown in the *k* column for each model/time series.

Table 6-2.The performance of the predictive models using the estimated number of nearest neighbours from the validation set and applied to DJI*-F, DJI*-F and DJI*-FD time series.

	DJI*			**DJI*-F**			**DJI*-FD**		
	k	NMSE	RMSE	k	NMSE	RMSE	k	NMSE	RMSE
Locally weighted linear	22	1.9534	0.62888	10	2.4436	0.01566	18	**0.4568**	0.00299
Locally linear	22	1.5564	0.05613	10	**0.7972**	0.00895	13	**0.4791**	0.00306
Radial basis	-	-	-	-	-	-	16	**0.7394**	0.00380
Kernel regression	-	-	-	-	-	-	4	**0.9712**	0.00436
Nearest neighbour	-	-	-	-	-	-	4	1.2500	0.00494
Locally constant	-	-	-	-	-	-	4	1.2500	0.00494

A further test was made by applying the *neural network* forecasting model to the same time series. As in the previous example, the model was trained using the first 772 points and the out-of-sample test was created on the next 120 points (the last 10 years). The same embedding dimension of 3 and time delay of 1 were used. The prediction error results of this model are presented in table 6-3.

Table 6-3. The performance of the neural network model.

	DJI*		DJI*-F		DJI*-FD	
	NMSE	RMSE	NMSE	RMSE	NMSE	RMSE
Neural network	1.0835	0.04684	**0.0864**	0.00289	**0.3193**	0.00248

The neural network model outperformed the reference models in the case of DJI*-F and DJI*-FD time series. In the case of DJI* time series, the NN model was very close to the performance of the 'random walk' predictor reference model. This model had lower errors than any local model.

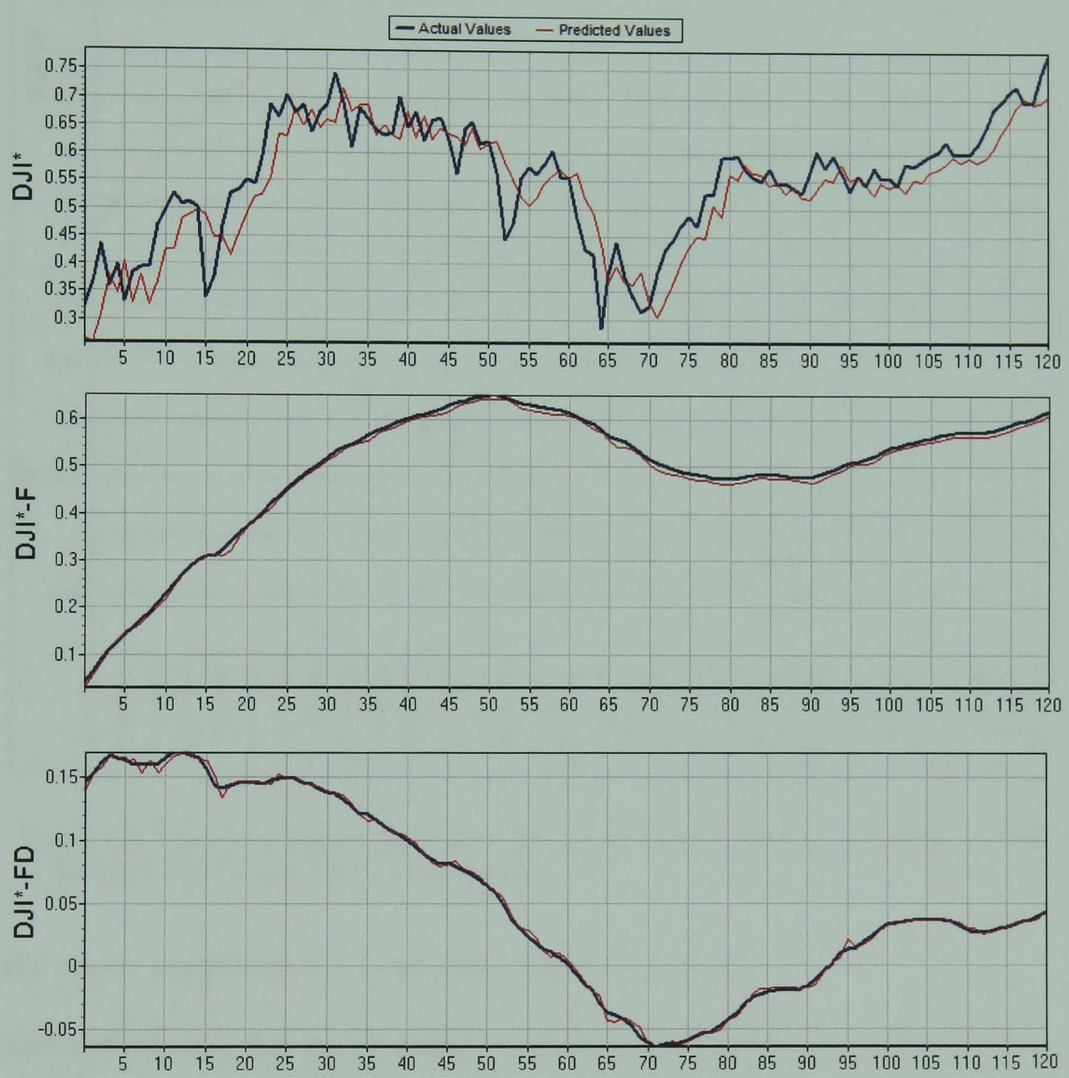


Figure 6-1. The actual and the predicted value of the *locally linear* non-parametric model applied to the DJI*, DJI*-F and DJI*-FD time series.

The actual and the prediction graphs of the best performing *locally linear* non-parametric model and the *neural network* model are shown in figure 6-1 and figure 6-2 respectively.

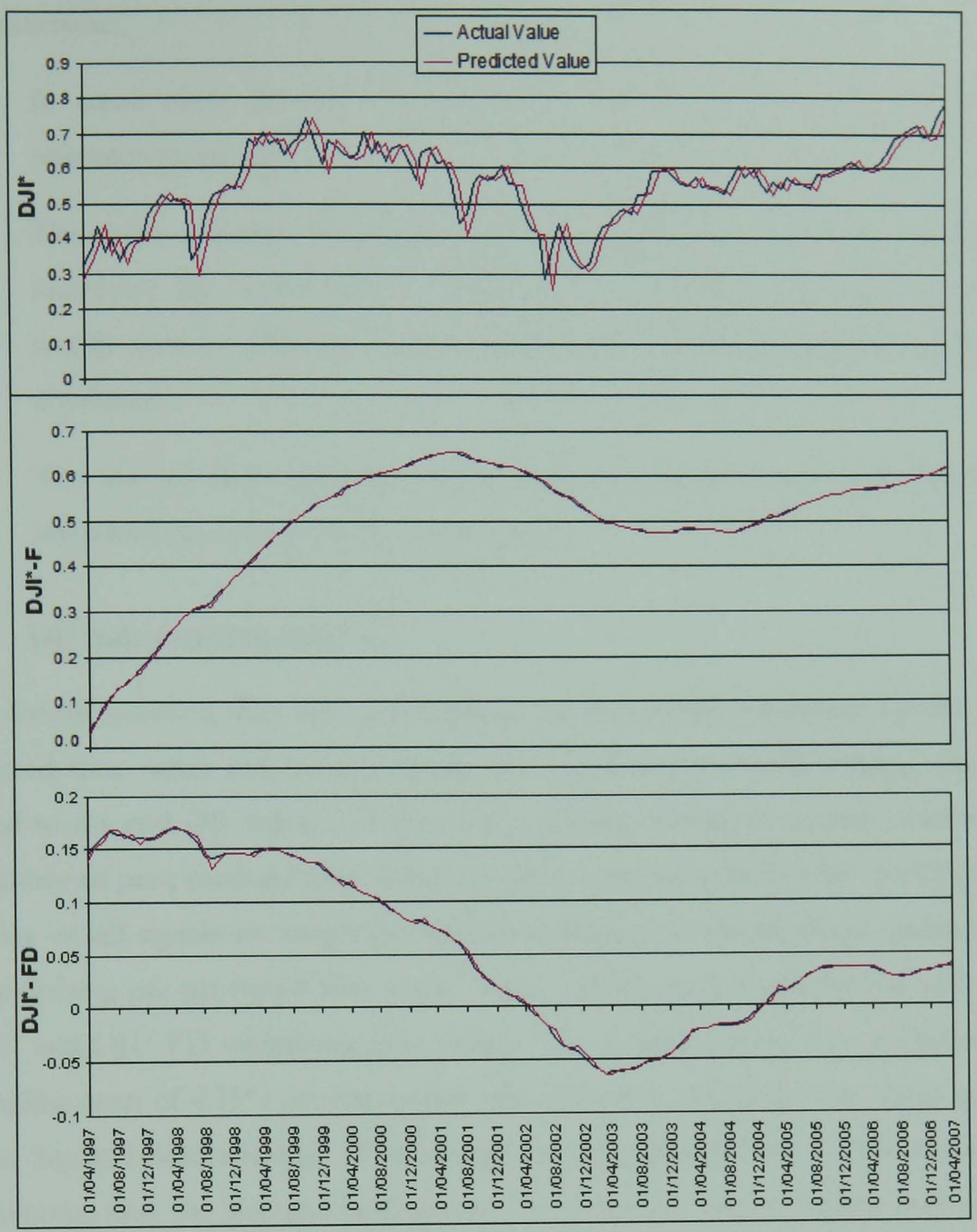


Figure 6-2. The actual and the predicted value of the *neural network* non-parametric model applied to the DJI*, DJI*-F and DJI*-FD time series

From the above experiments we can form several conclusions:

The neural network was the best predictive model, outperforming all the other models. The neural network DJI* predictions were as good as the reference predictors and the DJI*-F and DJI*-FD time series prediction error is astonishingly small, indicating that these time series have a large degree of determinism. Interestingly, the quality of these predictions go in line with the RQA measure of determinism (*DET*) found previously; 39.3% for DJI*, 72.7% for DJI*-F, and 44.7% for DJI*-FD (table 5-1, page 101). The best prediction results are achieved with DJI*-FD time series which has the highest

DET meassure.

- In some cases the local non-parametric models perform better than the reference predictors.
- The noise reduction technique (DJI*-F and DJI*-FD case) has significantly improved the predictability of the signals in question. The same underlying model with a different representation can differ widely in its prediction efficiency.
- The use of the validation set to estimate optimal model parameters has improved the out-of-sample performance.

6.4.1 DJI index trading models

The obvious question that arrives is whether the successful predictions of these pre-processed time series can be successfully translated into profitable trading strategies applied to the real DJI index. To test this, a trading strategy is created based on the predictions of pre-processed time series and the strategy is applied to the raw DJI index. The buy or sell signals are simply generated according to a 'month ahead' prediction of the underlying pre-processed time series. Three models were evaluated for each DJI*, DJI*-F and DJI*-FD underlying time series. The trading strategy was evaluated with the trading costs of 0.15% per transaction and without it, and both were compared to a simple buy-and-hold strategy. The trading cost depends mainly on the number of transactions, thus the buy-and-hold strategy produces the lowest trading cost of only 0.15% of the initial trade. The trading performance of these models is presented in figure 6-3 and table 6-4.

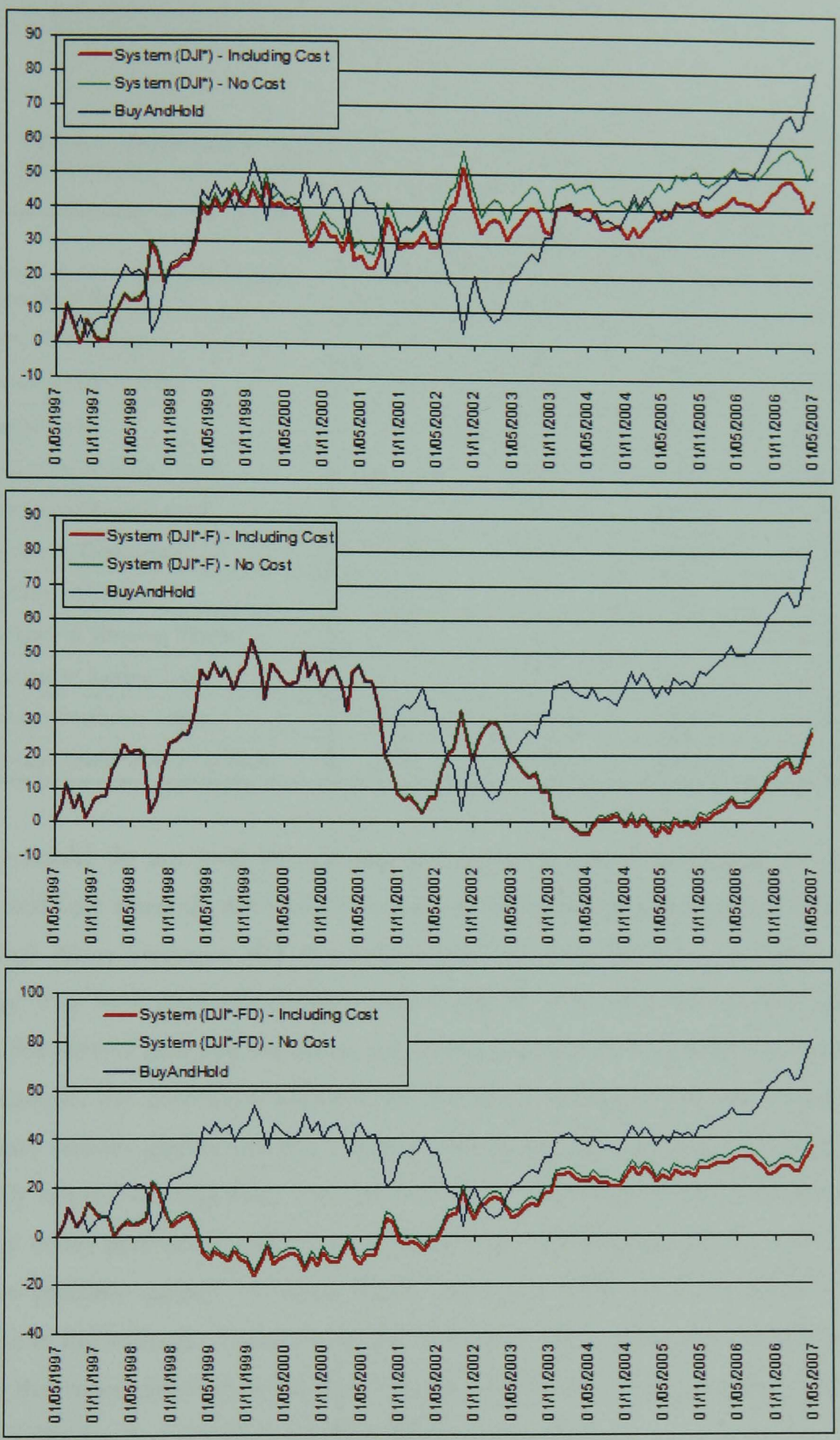


Figure 6-3. The DJI index trading models performances based on the DJI*, DJI*-F and DJI*-FD time series forecasts.

Table 6-4. The trading performance measures of the systems including the transaction cost. The B&H is the buy-and-hold trading strategy performance.

	DJI*	DJI*-F	DJI*-FD	B&H
Annualised Return	4.34%	2.66%	3.67%	8.14%
Cumulative Return	43.37%	26.63%	36.68%	81.45%
Annualised Volatility	14.50%	17.66%	14.39%	18.91%
Sharpe Ratio[1]	0.30	0.15	0.25	0.44
Maximum Monthly Profit	14.79%	13.04%	14.79%	13.04%
Maximum Monthly Loss	-8.24%	-17.47%	-10.03%	-17.47%
Maximum Drawdown	-25.23%	-51.92%	-38.18%	-50.77%
Number of Trades	35	5	13	1
Number of Winning Trades	18	2	7	N/A
Number of Losing Trades	17	3	6	N/A
Percent Profitable Trades	51.43%	40.00%	53.85%	N/A
Average Trade Gain/Loss Ratio	1.68	3.38	1.85	N/A

These results do not look too exciting, indicating that good predictions of the pre-processed time series do not translate equally well in predicting the original time series. Although these strategies did not outperform the long only buy-and-hold trading strategy, they have produced positive returns overall. The long-short trading strategies are by default less risky and usually are not correlated with the long only trading strategy. For example, the correlation between the DJI*-FD trading system return time series (red line, bottom graph) and the buy-and-hold return (blue line) is 0.17, whilst their monthly returns are negatively correlated (-0.53). This indicates that a similar trading strategy could perform equally well in the case of a declining market, so they can be used to preserve capital. As their risk is lower, the return can be improved by an increase in the leverage. Additionally, the filtering of the original time series introduces the lag that causes late buy/sell signals. Further experiments are required using different types of filtering that introduce smaller lags or none at all.

[1] In this case the Sharpe Ratio is a simple performance measure calculated by dividing the system return by its volatility (standard deviation).

6.4.2 DJI stocks neural networks trading models

In order to test predictions of financial time series further, using the embedding method, a larger test is performed whereby thirty constitutes of the DJI index are predicted using daily data, and trading strategies are devised accordingly. Four different trading systems, all utilising Probabilistic Neural Networks (PNN) and optimised by means of Genetic Algorithm (GA) were tested. The network structure, optimal lag and the future prediction period were found using a genetic algorithm. The GA tests competing models using a different number of lags and chooses the one with the best performance.

The main differences between these systems are the way the inputs and the outputs are pre-processed, and by the way the neural networks are optimised.

System A uses five logarithmic price inputs $\ln(p_t / p_{t-\tau})$, where the time delay τ for each input is selected using a GA from the range of 1 to 10. The output is the same logarithmic price predicted n days ahead, where n is found by a GA using the range 1 to 5.

System B is the same system as above, except five price differences $p_t - p_{t-\tau}$ were used instead, for both the input and output value. Similarly the optimal values of τ and n are found by GA.

System C used five raw price $p_{t-\tau}$ lagged values for the inputs and the price change percentage $(p_t - p_{t-\tau} / p_{t-\tau}) \times 100$ as the output.

System D is the same as system C except the GA optimiser finds the single set of parameters that performs best across all 30 stocks. Note that in the case of systems A, B and C each stock has its own set of optimal parameters.

All the systems were optimised using three years of daily data and a one year out-of-sample test was produced. This procedure was repeated ten times on sliding windows in order to produce a ten years out-of-sample test. All four systems apply a long-short trading strategy, generating buy/sell signals when the predicted output value crosses above/below a certain threshold found by GA. The trading costs of 1 cent per share and 0.1% per trade were used. The performance of these systems and the buy-and-hold

strategy is shown in figure 6-4.

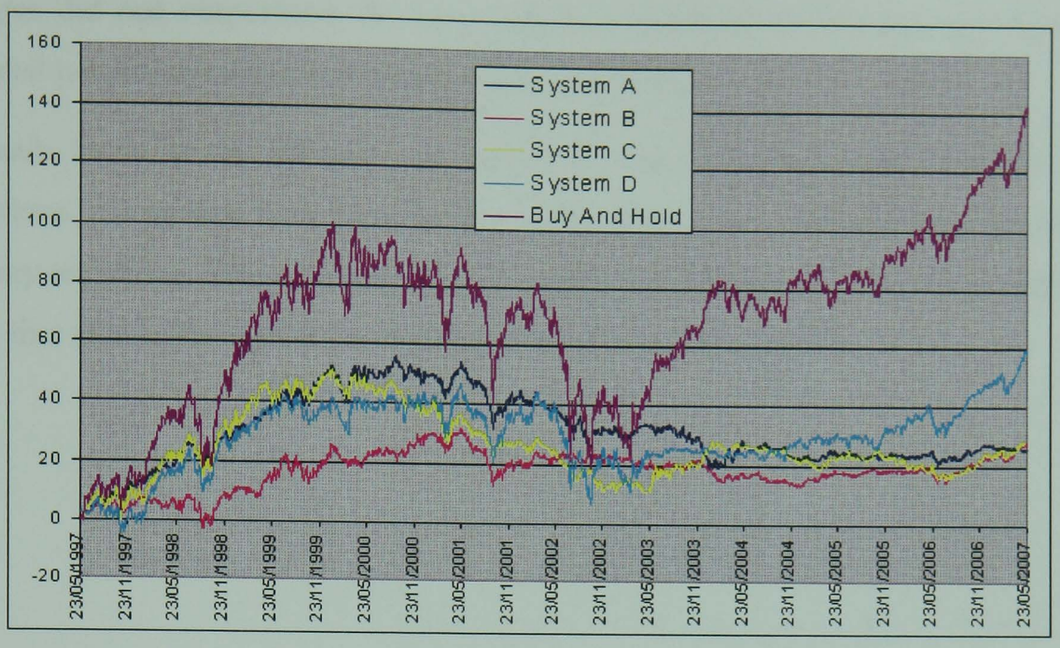


Figure 6-4. PNN trading systems applied to the portfolio of 30 DJI stocks.

From the graph we can see that all four systems produced a positive return, though the overall performance is not too exciting. This illustrates the difficulty of forecasting noisy financial time series by using past values only. Their poor predictability indicates that these time series are not produced by chaotic systems and that they have a strong stochastic component. Simple state-space forecasting becomes problematic for nonstationary and very noisy time series.

6.5 Conclusions

In this chapter several nonparametric predictive models, including neural networks were examined and compared to an "unconditional mean" and the "random walk" reference predictors. The neural network was the best predictive model, outperforming all the other models. The prediction error of the financial time series under study was very low, outperforming the "random walk" predictor and confirming the high value of determinism (*DET*) found through RQA analysis.

Four different trading systems, all utilising Probabilistic Neural Networks and optimised using Genetic Algorithm (GA) were applied to thirty constitutes of the Dow Jones Industrial index over ten years. Nonlinear dynamic theory will play a big part in analysing and the characterisation of financial markets, but its use for forecasting

utilising the univariate time delay embedding method is not clear. Although these strategies did not outperform the long only buy-and-hold trading strategy, they have produced positive returns overall with much lower risk.

The main focus in this chapter was mainly on univariate time series modelling and predictions. Using data from a single time series without any additional information can be a very restricting. The next chapter will employ a multivariate fundamental analysis of stocks that studies the interdependence between several time series.

Chapter 7

Neural Networks Models Based on Fundamental Analysis

7.1 Introduction

An analysis of a company's financial condition, its operations, the industry it is in, its competitors, and the general economic environment in order to determine a 'real' or *intrinsic* value of the company is termed in finance as *fundamental analysis.* The intrinsic value of a share is what the 'fundamentals' indicate it is worth. Most of the time this value is not the actual market share price. Typically, a fundamental analyst will buy/sell shares in a company if the intrinsic value is greater/less than the market price.

This research uses internal company information only, i.e. its assets, liabilities, income, profitability, operations and management performance figures collected from a Datastream source. In other words, it focuses on the "basics" of the business. From these current and the historical figures many *financial ratios* can be calculated which are used to estimate a company's future prospects, its potential returns and the risk behaviour of its shares.

An example of IBM financial reports are attached in the Appendix A.

7.2 Fundamental vs. technical analysis

Most trading strategies are based on some sort of fundamental, technical or combined analysis. Whilst *fundamental analysis* relies on variables representing the general state of the economy, industry and individual companies, *technical analysis* focuses exclusively on time series statistical properties derived from price, volume[1], and highest and lowest prices in the trading period. Technical analysts disregard a company's financial statements and rely instead on market trends and different recurring patterns present in time series.

The fundamental followers assume that prices should eventually reflect a fundamental

---

[1] In the case of stocks, volume can be defined as the number of shares traded on a specific day. Technical analysts often use it to confirm a specific pattern (i.e. trend), and believe that volume often precedes the price movement.

value and are not overly concerned with day-to-day stock price fluctuations that technical traders follow. For them, the value and price are not the same thing, and the day-to-day stock price fluctuation says more about volatility than value. A single price per share figure (market price) means little to the fundamental analyst. It only becomes useful when combined with other fundamental data like the number of shares outstanding[1]. The fundamental analysis approach may seem more rational compared to the technical approach which is based on the assumption that the markets are responding more to speculative forces rather than fundamental forces. Technical analysts, made up largely of followers of the efficient market hypothesis, believe fundamental analysis to be redundant, as prices already take into account company financial data, information that is widely available.

Many followers of these different analysis types criticise and dismiss each other's approach, finding evidence to support their beliefs. For instance, at the height of the 'dot com' era, almost all technology companies were hugely overpriced[2], and their fundamental valuation confirmed this. On the other hand, it is a common occurrence to see a stock with strong fundamentals whose share price falls for no apparent reason and the opposite situation, where a company share price rises over time on poor fundamentals. However, these could just be temporary discrepancies that correct themselves in the end, as was the case of the 'dot com' companies. It can be said that fundamental analysis is more relevant to a long term market investment style, whilst technical analysis is more to with day to day trading. It is also plausible to say that in normal circumstances, prices are driven more by fundamental forces and it is only when speculations start that technical forces becomes more dominant.

7.3 The advantages and disadvantages of fundamental analysis

Fundamental analysis provides a systematic approach to analysing companies and it provides additional information which may not generally be available to technical analysts. Using fundamental analysis a company can be compared with others in its industry. This type of analysis is also more suitable for long term forecasting and may indicate major price changes well in advance. By understanding the underlying

---

[1] A product of the two is the company market capitalisation value.

[2] At the Internet bubble peak, the Cisco Systems P/E ratio was above 130, implying huge earnings, which in ten years, assuming the same growth rate, would have exceeded US Gross National Product.

fundamentals, the investor may adopt a more aggressive position and can assist in deciding when to stay with a winning trade.

A major problem with fundamental analysis is that the figures found in financial statements cannot be fully trusted, due to manipulation in accounting. Some companies avoid disclosing certain information in their reports, thus making it difficult to make a fare assessment without certain relevant information. Even companies that comply with the generally accepted accounting principles (GAAP) are no exception. Probably one of the most misused fundamental figures is EBITDA[1], which is a very simple measure showing the profits that a company can generate if it doesn't have to pay interest, expenses or taxes and decides not to reinvest depreciation[2] and amortisation[3]. This variable is often misrepresented as profit from operations in the company's financial report headlines, but it should be used with great caution as organisations can use this measure or some variant[4] of it as an accounting gimmick to dress up a company's earnings. Therefore, the EBITDA is a misleading measure of profitability and a poor basis for valuation due to its lack of consistency of calculation amongst companies and from one period to the next. When using this metric, it is important for investors to focus on other performance measures to make sure the company is not trying to hide information through the EBITDA.

7.4 Value vs. growth fundamental analysis

Typical equity fundamental analysis tries to determine how much stocks and businesses are worth. There are many types of fundamental analysis, the main ones being *value* and *growth*, the former places more emphasis on the company's value whilst the latter on the potential growth. The difference between the two in practice can be very blurred, as growth investors focus on the company's value on an ongoing basis. The growth at reasonable price (GARP) fundamental model attempts to combine the two approaches. Contrary to popular belief, many studies have shown that value stocks produce better investment returns over time than growth stocks. This finding has been tested and

---

[1] Acronym for earnings before interest, taxes, depreciation and amortisation.

[2] An expense recorded to reduce the value of a long-term tangible asset. Since it is a non-cash expense, it increases free cash flow while decreasing reported earnings ([31]).

[3] The deduction of capital expenses over a specific period of time. Similar to depreciation, it is a method of measuring the consumption of the value of long-term assets like equipment or buildings [31].

[4] Some companies exclude some or all exceptional items such as incentive compensation, bonuses, pension deficit when calculating EBITDA.

documented in different financial world markets. Dimson, Nagel and Quigley [30], found a strong value premium in the U.K. market for the period 1955-2001 among small-caps as well as amongst large-caps. They also found that the dividend yield as a measure of value produces very similar results. This finding may be an explanation for the apparent success of the well-known 'Dogs of the Dow' investment strategy[1]. Dunis et al. [32] analysed the FTSE All-Share index returns during the 2000-2002 period, along five different value/growth techniques and found that the value stocks outperformed growth stocks in all examples. Capaul, Rowley and Sharpe [33] found similar evidence in the US market, while Fama and French [34], [35] expanded their study to twelve major world markets for the 1975-1995 period confirming the value premium internationally. They argued that the reason for the value premium is as a compensation for the risk missed by the capital asset pricing model and not because the market undervalues value (distressed) stocks and overvalues growth stocks.

There are a few different models used to calculate the value of a company, the major ones being the cash flow discount model and the dividend discount model.

Some practitioners complain that there is a lack of clear and well-defined models similar to the Black and Scholes model to calculate company value. However, this model is extremely difficult to develop for many reasons. First, there are so many potential fundamental variables that can play a part in the model and choosing the right ones is a daunting task. Even worse, most of these variables are rough estimates, like earning estimates and it is almost impossible to create a reliable model based on poor data. In addition, there are so many qualitative factors such as a company's management and its competitiveness, which is very difficult to quantify. It is most likely that we will see many different models developed in the future, applied to different markets and companies. However, one can always develop a solid fundamental model for investment purposes without the need to know the 'real' value of the company.

7.5 Fundamental financial ratios

Some general state of the economy models may use fundamental variables such as interest rates, inflation rates, gross domestic product and money supply; however this

---

[1] This is a very simple strategy in which an investor buys and rebalances a yearly portfolio consisting of the ten highest dividend paying Dow Jones Industrial (DJI) stocks. These companies tend to have lower prices compared to other Dow components and are called the dogs of the Dow. This strategy was first popularised by Michael O'Higgins in his 1991 book 'Beating The Dow' and historically has been outperforming DJI index.

research will only concentrate on quantitative information from company financial reports.

The fundamental variables and the ratios considered in this research are listed below. They cover various aspects of fundamental analysis including profitability, financial leverage / liquidity and operating efficiency. These fundamental variables are arbitrary measures and do not necessarily represent the optimal set of variables.

- **Price to Book Ratio (PB)** is one of several variables that can be used to separate value stocks from growth stocks. This ratio compares the market valuation of a company (share price) to the latest quarter book value (total assets minus intangible assets and liabilities). Value stocks tend to have a low P/B ratio whilst growth stocks tend to have a high PB ratio. The equivalent reciprocal measure is known as the Book to Market (BM) ratio and is often used.

- **Gross Profit Margin (GPM)** is calculated as a ratio of gross profit (company's total revenue minus the cost of producing its goods or services) and the revenue (sales). A company with a higher gross profit margin than its competitors is more efficient. It is a stable measure over time and any large fluctuations from a historical average can be an indication of fraud or accounting irregularities.

- **Net Profit Margin (NPM)** is a similar profitability measure calculated as a ratio of net profit after taxes (net income) and revenues. It is an indicator of company effectiveness and cost control. Profit margins are industry specific and it is a good sign if a company's profit margin is higher than that of its competitors. Some retailers and low budget airlines have very low profit margins as a result of their pricing strategies.

- **Asset Turnover (AT)** is a ratio of total sales (revenue) and total assets. This measure represents a firm's efficiency in using its assets in generating sales. Companies with low profit margins tend to have a high asset turnover and vice versa. The higher the figure the better and as is the case with most financial ratios, investors must make comparisons within the same industry.

- **Return on Assets (ROA)** is calculated as the ratio of a company's fiscal year earnings (net profit) and its total assets (the shareholders' capital plus short and long-term borrowed funds). It is a useful indicator and a measure of a

company's profitability, asset intensity and management of a business and can be decomposed as a product of the two ratios above, net profit margin (ratio of earnings and sales), and asset turnover (ratio of sales and assets). This measure is industry specific as companies in manufacturing and telecommunication sectors (usually have a ROA below 5%) are very asset-intensive compared to advertising agencies and software companies (the ROA can be above 20%).

- **Return on Equity (ROE)** is calculated as the ratio of a company fiscal year's earnings (net income) and the shareholders' equity (assets that have actually been generated by the business representing total assets minus total liabilities). Similar to ROA it is a profitability measure that reveals how much profit a company generates with the money shareholders have invested. But at the same time it is an asset management and financial leverage measure as it can be decomposed as a product of the ROA and the financial leverage ratio (total assets/total equity). Companies without debt will have ROE and ROA figures that are equal. Investors prefer companies having a high and growing ROE.

- **Inventory Turnover (IT)** is a measure showing how fast the inventory is 'turned over' (sold and replaced) in a period or fraction of a year measuring how long an average item remains in inventory. It is calculated either as the ratio of company annual sales and inventory or the ratio of the cost of goods sold (COGS)[1] and average inventory. The cost of sales is considered to be a more accurate input than sales alone because it is recorded by the company rather than the marketplace, as in the case of sales. The average inventory is used instead of the year ending inventory level to minimise seasonal factors. A low turnover implies inefficiency in a product line or marketing effort, poor sales or excess inventory, whilst a high ratio implies the opposite. Dividing 365 days by this ratio gives the number of days it takes a business to clear its inventory. This number varies greatly by industry.

- **Debt to Equity (DTE)** ratio is a measure of a company's financial leverage, indicating what proportion of equity and debt the company is using to finance its assets. It is calculated as a ratio of long-term debt and common shareholders' equity. Sometimes the company's total liabilities (including short-term and long-

[1] Cost of sales.

term debt) are used instead of long-term debt in the calculation. Companies with a high ratio can be risky to invest in, especially in times of high interest rates resulting in volatile earnings. The accepted level of debt to equity has changed over time, and depends on both economic factors and society's general feeling towards credit. However, it is a delicate balance, especially if the cost of debt financing is larger than the returns that the company generates which can lead to bankruptcy. The measure is very dependent on the industry in which the company operates.

- **Dividend Yield (DY)** is a financial ratio that shows how much a company pays out to its shareholders in the form of dividends each year relative to its share price. It is calculated as a ratio of annual dividends per share and the price per share. Similar to the PB ratio, this measure can be used to classify a company into a value or growth group. Young and growth-oriented companies tend to have lower dividend yields, whilst mature and well-established companies tend to have a higher one. For most small growing companies this ratio is missing, as they do not pay dividends at all.

- **Operating Margin (OM)**[1] is a ratio used to measure a company's management (operating) efficiency and pricing strategy. It is calculated as a ratio of net income[2] and net sales[3] showing how much a company makes (before interest and taxes) as a proportion of sales. It is best to look at the change in operating margin over time and to compare the company's figures to those of its competitors. A higher operating margin gives management more flexibility in determining prices and extra security during tough economic times.

- **Current Ratio (CR)**[4] is a measure of a company's financial strength and its ability to pay short-term obligations. It is calculated by dividing total current assets by total current liabilities. The higher the ratio, the more liquid the company is. A high ratio may suggest that a company has too much cash and is doing a poor job of investing it and expanding the business. A ratio under one

---

[1] Also known as operating profit margin or net profit margin.

[2] Operating income or operating profit is a measurement of the money that company generated from its own operations and is equal to gross profit less operating expenses.

[3] Sales amount after the deduction of returns, allowances and any discounts.

[4] Also known as liquidity ratio or cash asset ratio.

is not a good sign but it does not necessarily mean that the company will go bankrupt. An acceptable current ratio varies by industry. There are a few variations and derivatives of this measure, one being the Quick Ratio (acid-test ratio) which is simply current assets minus inventories divided by current liabilities. Also, the components of current ratio (current assets and current liabilities) are used to derive working capital, which is the difference between the two. A ratio of working capital and sales (working capital turnover ratio) provides information as to how effectively a company is using its working capital to generate sales.

- **Price to Cash Flow (PCF)** is a measure of the market's expectations of a firm's financial strength, calculated as a ratio of share price and operating cash flow per price[1]. It is often used to classify companies along the value-growth line where a low PCF indicates a value stock. Operating cash flow (OCF) refers to how much cash a company generates out of revenues[2] and excludes costs that are not a part of the on-going operation.

- **Earnings per Share (EPS)** is an indicator of a company's profitability and is calculated as a ratio of net earnings and the number of average outstanding shares. This is probably the most common single variable used by analysts in determining a share's price. There are two types of EPS, diluted and basic, depending whether the stock options, convertibles and warrants are included in the outstanding number of shares. The EPS is calculated for each major category in the income statement: continuing operations, discontinued operations, extraordinary items, and net income. A trailing EPS is calculated for the previous financial year, while the current and forward EPS are estimates for the current and the coming year. An important aspect of EPS that is often overlooked is the amount of capital that is required to generate earnings (net income) in the calculation. Two companies may have the same EPS number, but the one with less equity (investment) would be more efficient at using its capital to generate income. Another issue that makes a comparison of EPS amongst different companies difficult is the fact that they have a different number of outstanding shares. One way to check the quality of the EPS is to

[1] It can also be calculated by dividing a company's market capitalisation by the company's operating cash flow.

[2] OCF = Earnings before interest and taxes + Depreciation - Taxes

compare it to the operating cash flow per share (OCFPS)[1]. The reported EPS figure is of high quality if it is smaller than the OCFPS, otherwise it means that the company is generating less cash than is suggested by the EPS. There may be reasons for this discrepancy, such as for young companies, economic cycles or financing investment for future growth, but if the company is to survive, the discrepancy cannot last long, as was the case for many dot com companies that went bust.

- **Sales per Share (SPS)**[2] is used to evaluate a company's business activities in comparison to its share price. It is calculated by dividing total revenue by the weighted average of shares outstanding. The higher the ratio, the more active the company is. A derivative of this ratio is the Price to Sales ratio (PS) calculated by dividing the current stock price by the sales per share and is often used for the evaluation of young and unprofitable companies.

The main measures used to distinguish between value and the growth stocks are the book to market or price to book ratio, price to earnings ratio, price to cash flow, dividend yield and market capitalisation. The traditional value investor would look for companies with any one or all of the following attributes: low price to book value, low price to earnings ratio, low price to cash flow ratio and high dividend yield and the growth investor would look for the opposite.

7.6 Use of price-to-book ratio in portfolio selection

Several studies found a strong correlation between the price to book ratio[3] and future stock performance [32]-[39]. These tests have been applied to major world markets on a large number of companies and data histories and in some cases the data was free from survivor bias[4]. This research will test whether the effect of the PB ratio on the portfolio return confirms these findings when applied to a randomly selected portfolio of forty eight companies drawn from a pool of European stocks that are quoted on the US market. The fundamental ratios of these companies were taken from a DataStream source representing yearly data from 1990 to 2005.

---

[1] Operating cash flow divided by the number of shares used to calculate EPS.

[2] Also known as the 'sacrifice ratio' and 'revenue per share'.

[3] A PB reciprocal value Book to Market (BM) is often used.

[4] The portfolio includes historical data of closed companies.

In order to assess the relationship between the price-to-book ratio and the portfolio return a simple trading strategy is devised based on the single PB ratio value. Each year the portfolio is divided into four groups based on the PB percentile ratio value (i.e. 0-25, 25-50, 50-75 and 75-100) and the performance of each group is recorded. The graphs shown in figure 7-1 confirm the positive correlation between a low PB ratio and future stock performance. The portfolio belonging to the lowest PB ratio percentile (0-25, i.e. value stocks) is found to be the best performer. This portfolio return is based on a trading strategy with a single rule, to enter a long trade (buy shares) if the PB ratio is between 0 and 25, and if otherwise, to exit a long trade. Note that this is only in the case of a long strategy, as short selling[1] is not used. The other three portfolio returns were generated in the same manner, using different PB thresholds. The portfolio consisting of a high PB (75-100, i.e. growth stocks) is the second best performer. Less well known is the fact that the long term return of traditional value stocks is better than the return of new and highly promising growth stocks, confirmed in this experiment and in agreement with findings in [32]-[39].

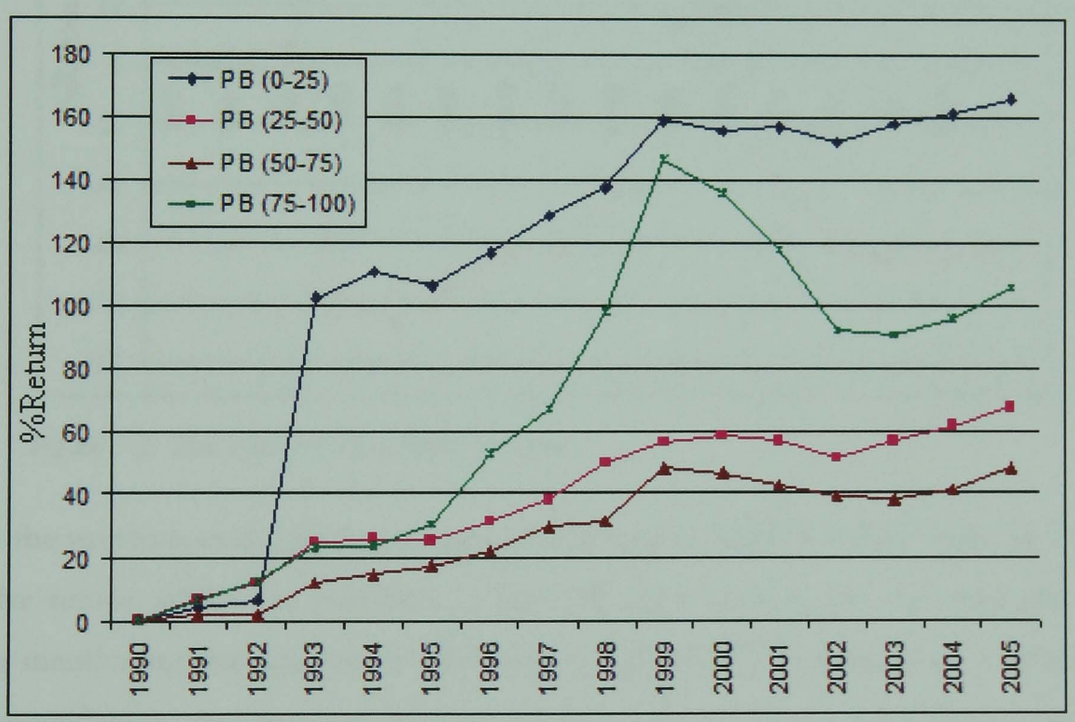


Figure 7-1. The price-to-book trading strategy performance.

The transaction costs were not included, though they should not be high as portfolios are rebalanced on a yearly basis. It is important to note that we were only interested in

[1] Selling the borrowed stock expecting that price will fall and profit is made when the stock is bought back at a lower price.

the comparison between different portfolio strategies and their returns, rather than their absolute performance. In some cases the buy-and-hold strategy with a considerably higher risk profile produces the best returns compared with other strategies.

In order to examine a long-short investment strategy based on the PB ratio, we expand the trading strategy above to include two rules: to buy the low value PB stocks and to sell short the high PB value stocks. The PB range value used for the long entry is 0-25, and for the short entry 75-100. Long and short exits are generated if the PB value is in the 25-75 range. These strategy portfolio returns are presented in figure 7-2. A reverse trading strategy of buying high PB stocks (75-100) and selling low PB stocks (0-25) produce a negative return and is the mirror image of the blue portfolio curve.

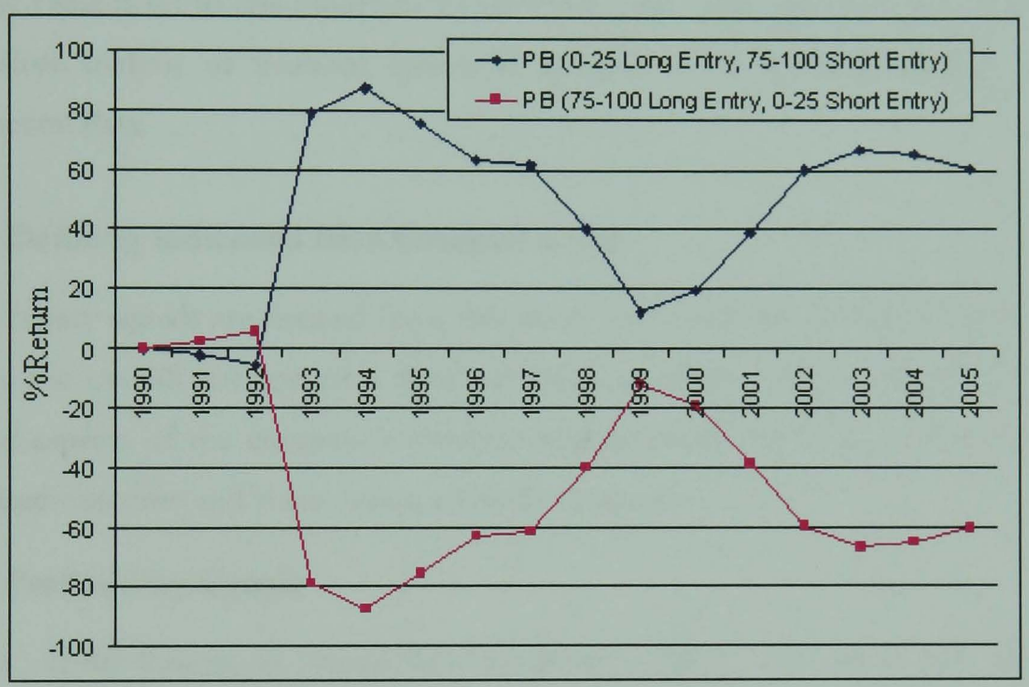


Figure 7-2. The effect of a low/high PB value

As in the previous example, the strategy that invests in low PB value stocks produces a positive return, whilst the portfolio of high PB value stocks does the opposite. It is worth mentioning that analysts usually neglect these low PB value firms, making their recommendations and forecasts rarely available. Instead, they tend to favour high PB momentum driven growth companies with strong recent performance, despite their inferior overall performance. The reason for this could be due to marketing, as it is easer to catch investors' attention by recommending exciting growth stocks that may promises high short-term returns.

7.7 Trading systems utilising financial ratios

It has been shown that by utilising only the price to book ratio in the simple trading strategy, there would have been an improvement in portfolio performance (figure 7-1). A natural extension to this approach is to add some of the financial ratios described earlier. As the value of the company and its share price can be difficult to ascertain, emphasis is not put on calculating the intrinsic value of a company, instead the aim is to identify patterns and key drivers that have a positive/negative influence on company returns and to devise a trading strategy accordingly.

Instead of comparing the above ratios between companies in the same industry they are compared to their previous values and if the change reflects a positive outcome the indicator value is set to one otherwise to zero [37], [38]. This approach can be seen as a quantitative analysis of financial ratios; an application of technical analysis utilising fundamental data.

7.7.1 Defining indicators from financial ratios

Twelve binary signals are derived from the above ratios and the sum of them is used to measure the overall strength of a firm's financial position. These indicators measure different aspects of the company's financial health, where the value of one represents the desired outcome and a zero value a negative outcome.

- **Profitability signals**

I-1. If the Return on Assets (ROA) is positive (ROA > 0), set to one, otherwise, set to zero.

I-2. If the ΔROA is positive, set to one, otherwise set to zero, where ΔROA is defined as the difference between the current and the previous year's ROA.

I-3. If the ΔROE is positive, set to one, otherwise set to zero, where the ΔROE is defined as the difference between the current and the previous year's Return on Equity (ROE).

I-4. If the EPS is larger than the SPS, set to one, otherwise set to zero, where the EPS represents the Earnings per Share and the the SPS the Sales per Share.

I-5. If the PCF is positive, set to one, otherwise set to zero, where the PCF is the Price to Cash Flow.

- **Operating efficiency signals**

I-6. If the ΔAT is positive, set to one, otherwise set to zero, where the ΔAT is defined as the difference between the current and the previous year's Asset Turnover (AT).

I-7. If the ΔIT is positive, set to one, otherwise set to zero, where the ΔIT is defined as the difference between the current and the previous year's Inventory Turnover (IT).

I-8. If the ΔGPM is positive, set to one, otherwise set to zero, where the ΔGPM is defined as the difference between the current and the previous year's Gross Profit Margin (GPM).

I-9. If the OM is larger tan the NPM, set to one, otherwise set to zero, where the OM represents Operating Margin and the NPM Net Profit Margin.

- **Leverage, liquidity and dividends signals**

I-10. If the ΔDTE is negative, set to one, otherwise set to zero, where the ΔDTE is defined as the difference between the current and the previous year's Debt to Equity (DTE) ratio. The reduction of debt indicates a positive signal.

I-11. If the CR is positive, set to one, otherwise set to zero, where the CR is the Current Ratio.

I-12. If the DY is positive, set to one, otherwise set to zero, where the DY is the Dividend Yield.

The twelve indicators above are summarised in table 7-1.

Table 7-1. Twelve performance indicators derived from financial ratios. If the condition is satisfied the indicator value is set to 1 otherwise to 0.

	Signal category	Condition	Variable(s) used	Value
I-1	Profitability	ROA > 0	Return on Assets	1
I-2		ΔROA > 0	Return on Assets	1
I-3		ΔROE > 0	Return on Equity	1
I-4		EPS > SPS	Earnings per Share, Sales per Share	1
I-5		PCF > 0	Price to Cash Flow	1
I-6	Operating efficiency	ΔAT > 0	Asset Turnover	1
I-7		ΔIT > 0	Inventory Turnover	1
I-8		ΔGPM > 0	Gross Profit Margin	1
I-9		OM > NPM	Operating Margin, Net Profit Margin	1
I-10	Leverage, liquidity and dividends signals	ΔDTE < 0	Debt to Equity	1
I-11		CR > 0	Current Ratio	1
I-12		DY > 0	Dividend Yield	1

7.7.2 Use of financial indicators in portfolio selection

The total rank (I-R) is derived as the sum of the I-1 to I-12 binary indicators representing the overall company quality. To test whether this rank can be used as a means of selecting a portfolio of stocks, a few simple trading strategies are evaluated based on the rank value. The test results of three trading strategies are presented in figure 7-5. The portfolios are rebalanced on a yearly basis based on the I-R rank value.

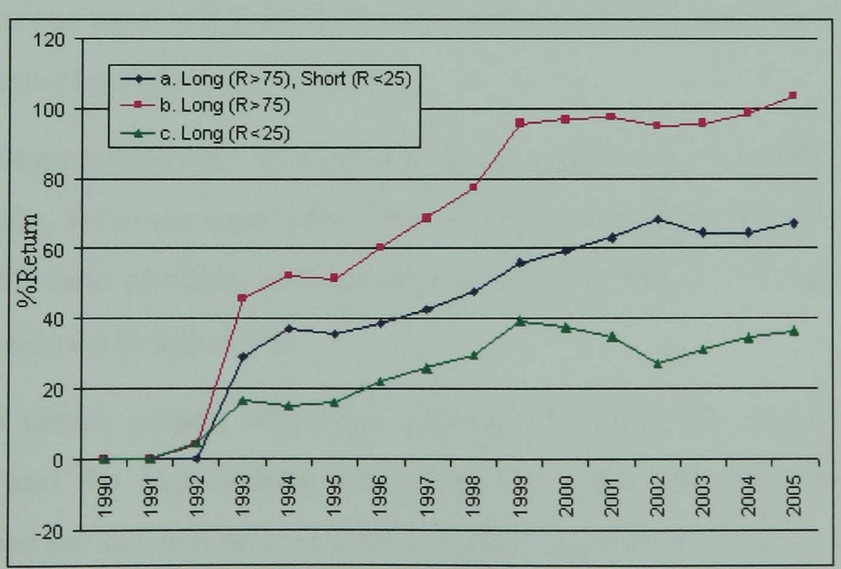


Figure 7-5. Rank based strategy portfolio's performances 1991-2005.

The three trading strategies are defined as follows:

a. The long-short strategy, comprising of buying high-ranking stocks and selling low-ranking stocks. The rank range value used for the long entry is above the 75th percentile, and for the short entry, below the 25th percentile. Long and short exits are generated if the rank value is in the 25th - 75th percentile range.

b. The long only trading, strategy, comprising of buying high-ranking stocks where the rank is above the 75th percentile.

c. The long only trading strategy, comprising of buying low-ranking stocks where the rank is below the 25th percentile.

We can see from the graphs that the rank representing the sum of the twelve binary indicators has a significant effect on the different portfolio returns. The long portfolio of high-ranking stocks (b) generates much larger returns than the long portfolio of low-ranking stocks (c), whilst the less risky long-short strategy (a) is right in the middle.

7.7.3 Neural network models based on fundamental ratio indicators

The previous examples did not use any optimisation or statistical models, instead thy use the PB ratio and the rank indicator derived from fundamental financial ratios directly in order to create different portfolios and test their performances. In the next tests a neural network model is used to see if it can further improve portfolio performance. These models require data to be divided into training and the test set, so we use the first ten years (1990-2000) for the training set and five years (2001-2005) for assessing the out-of-sample performance.

In order to compare these new models with the previously presented PB and rank based portfolios results, these are shown with the last five years (2001-2005) results. The last five years of PB ratio portfolio performance is shown in figure 7-4 and the rank based portfolio is presented in figure 7-5.

We can see a similar pattern where low-ranking PB portfolios outperform the high-ranking ones and the high-ranking I-R portfolios outperform the low-ranking ones. These portfolios are also less risky as their volatility and draw-downs are much smaller.

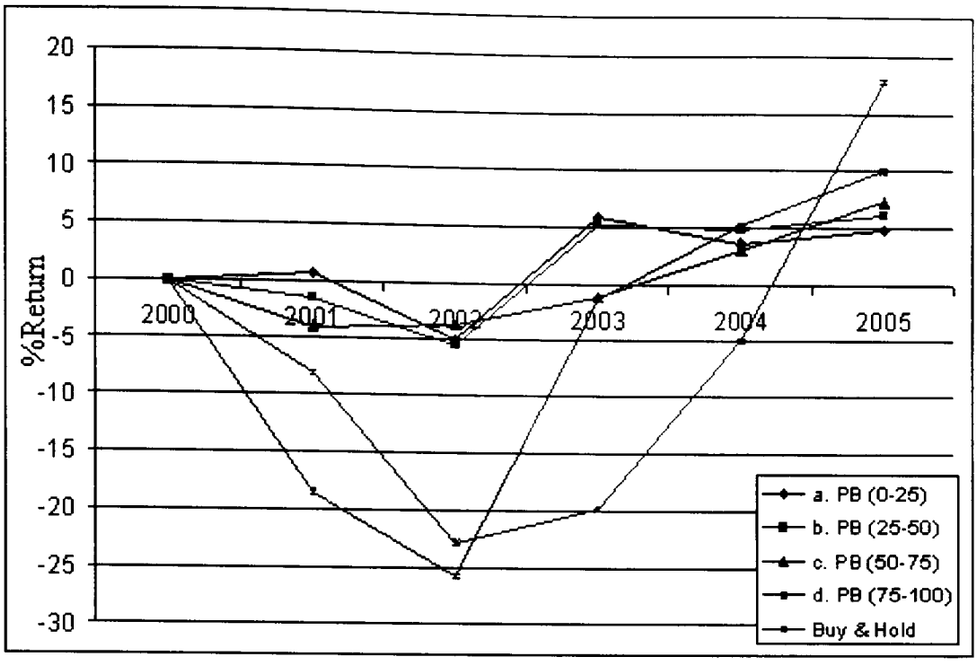


Figure 7-4. The price-to-book trading strategy performance (2001-2005).

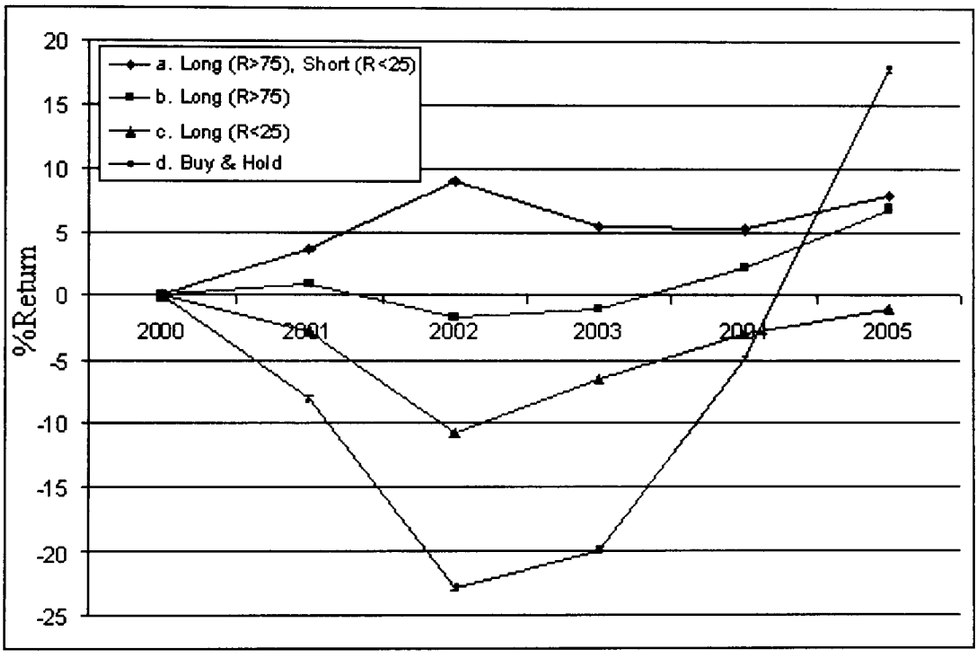


Figure 7-5. Rank based strategy and portfolio's performances 2001-2005.

In the next experiment we feed the above twelve individual binary inputs into an advanced statistical regression model using the neural network and genetic algorithm optimisation approach in order to predict a stock return one year ahead. A simple long-short trading strategy is created, whereby the stock is bought if the predicted return is positive and sold if negative. In order to reduce the model's complexity and to avoid

over-optimisation, the genetic algorithm was instructed to use a maximum of five inputs. These inputs are estimated to be the most relevant and have the largest influence on the stock return. All the models are trained using ten years of historical data and test results are produced on the following five years out-of-sample data. Four different models are evaluated, depending on the approach in which the model parameters are estimated:

a. A simple buy-and-hold strategy

b. A linear model where the neural networks regression parameters (weights) are estimated without using the 'nonlinear' hidden layer neurons[1]. Also, the model parameters are derived for each individual stock independently of other stocks, so that each stock has its own model with the 'optimal' sets of parameters.

c. The same linear model as in b. except that one set of model parameters are estimated using the data from all stocks.

d. A nonlinear model where the neural networks regression parameters (weights) are estimated using up to five nonlinear hidden neurons. The model parameters are derived for each individual stock independently.

e. The same nonlinear model as in d. except that one set of model parameters are estimated using the data from all stocks.

The results of the four models are presented in figure 7-6. From the graph we can conclude that the performance of the models in which the parameters are estimated using data of a particular stock only (case b. and d.) performed better than the global models, that used one set of estimated parameters derived using data from all stocks (case c. and e.). Also, nonlinear models slightly outperformed the linear models, but more importantly all the statistical models did much better than the PB and I-R rank models tested previously, showing a more than double increase in portfolio returns.

[1] A neural network model without the nonlinear hidden layer (i.e. input nodes are directly connected to the output nodes) is equivalent to a linear regression model.

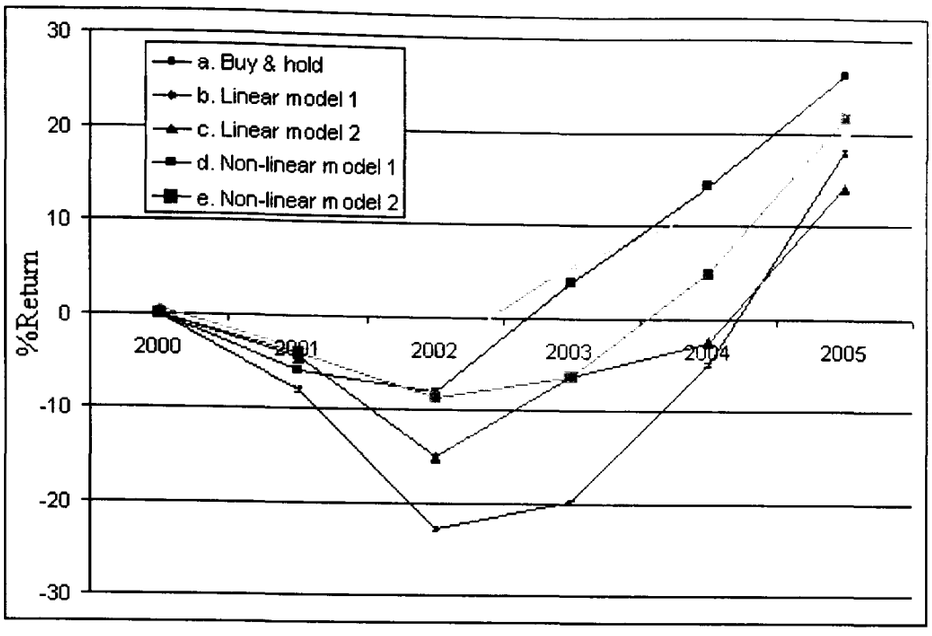


Figure 7-6. Neural networks model portfolio performances using the binary signals.

Converting fundamental variables into the binary inputs may eliminate potentially useful information. To examine this issue another test has been carried out replacing the binary signal inputs with a yearly percentage change of the raw fundamental ratios (ROA, PB, PCF, DY, SPS, CR, DTE, GPM, OM, NPM, ROE, AT and IT). The portfolio performance of the previously described neural network model applied to these variables is presented in figure 7-7. The model regression parameters (weights) are derived for each individual stock independently of the other stocks.

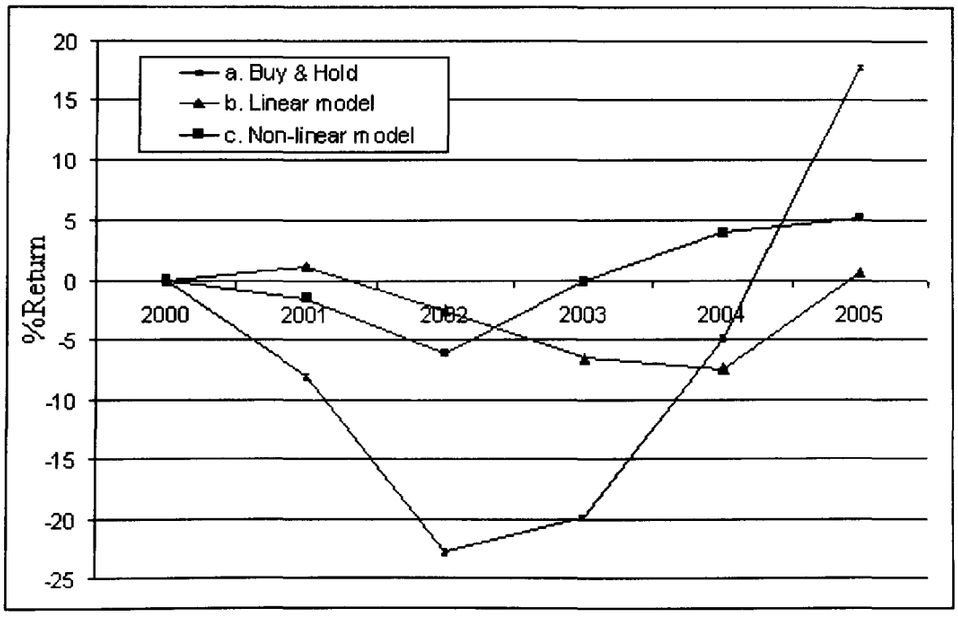


Figure 7-7. Neural network models portfolio performance using 'raw' inputs.

The question arises whether similar quantitative strategies can be used instead of human analysis. These "artificial analysts" may or may not do as well as people, but can make up for it in volume and save on operating costs. The next chapter will test the use of technical analysis in trading, and will highlight its advantages and disadvantages when compared to models based on fundamental analysis.

As in the previous tests, two different models were examined, linear and nonlinear. The nonlinear model (c) that used up to five neurons in the hidden layer has a slightly better performance than the linear model (b). However, this portfolio performance test was considerably worse if compared to the previous tests where the binary inputs were used instead. The reason for this could be that by encoding the fundamental inputs into the binary signals using prior knowledge to derive the positive/negative outcomes, representing direct 'hints' to the system which can simplify and improve the model generalisation. On the other hand a good model should be able to extract those rules from the data directly, but it would require a much larger and more representative data set than was used in this case. In addition, the inputs used representing the yearly percentage change may mot be the best choice.

7.8 Conclusions

From the tests performed in this chapter we can conclude the following:

- Portfolios made of stocks with a low price-to-book ratio (value stocks) outperformed the portfolios with a high price-to-book ratio (growth stocks).
- The market does not fully process the fundamental financial information and incorporate it into prices in a timely manner. Higher returns can be achieved by utilising the models based on fundamental measures found in the company's financial statements. The overall rank measure derived from fundamental ratios has a significant effect on the different portfolio returns.
- The use of neural network improved the performance of the models that are based on ranking information only.
- More complex nonlinear models outperformed the linear one, indicating their better suitability for modelling financial time series and the possible presence of nonlinearity in them.
- Individual stock models in which parameters are optimised using data of the particular stock, performed better than global models that used one set of estimated parameters for all stocks.
- Models that used pre-processing of data to include prior knowledge performed better.

Chapter 8

Neural Networks Models Based on Technical Analysis

8.1 Introduction

One of the most controversial and oldest schools of thought in the analysis of future prices is technical analysis. Over the last century much research has been performed by statisticians, financiers and economists in the search for lucrative formulas that may earn them fame and fortune. Technical analysis has, to some degree, been used over the last two centuries and has increased significantly since the 1980s due to technological developments. Early implementation goes back to the 1700s, where the plotting of price charts in the Japanese rice market was recorded. In the late 1800s and early 1900s the famous "Dow Theory" was coined, which is the foundation of technical analysis.

This chapter will put to the test the use of technical analysis combined with neural networks and genetic algorithms in portfolio trading.

8.2 Technical analysis

Technical analysis is based on the interpretation of patterns, trends, cycles, and formations that develop on charts with the primary aim of identifying trends and major turning points in the market. Followers of this strategy believe that prices[1] and trading volume offer all necessary information for trading [115]. They have over a thousand technical indicators[2] to choose from in order to monitor changes and derive trading models. There are many different approaches to technical analysis, though most of them rely upon the assumption that the stock market moves in trends and that history repeats itself [114]. These trends are assumed to last until the supply-demand balance changes and noticeable patterns appear, indicating the beginning or end of a trend. Supply and demand determines whether the buying or selling pressure is gaining a predominant position. The fact that too many investors base their expectations on

[1] Prices used are opening, closing, highest and lowest price in the period.

[2] Mathematical/statistical formulas derived from price and volume data.

historical prices and act on these expectations has a direct influence on future prices.

8.3 The advantages and disadvantages of technical analysis

Though technical analysis has many indicators, the ultimate source of data is very small, due to the assumption that all other information is reflected in the market price. This simplifies the modelling process as it does not require the gathering and modelling interaction of many fundamental variables. Both technical analysis and the efficient market hypothesis coincide in the assumption that all information is reflected in the price, yet the assumption leads to different conclusions. The EMH states that markets are unpredictable, whilst technical analysts believe that they can predict the future price by observing the historical price and other trading variables. Technical analysis assumes that price behaviour is influenced by an underlying mass psychology, which leads to predicting a rise or fall in price. For that reason, many technical analysts are also market timers, who believe that technical analysis can be applied just as easily to the market as a whole, as well as to an individual stock. Technical analysts don't analyse a company's fundamental qualitative data. In some cases, when fundamental information contradicts price level expectations[1], the technical analyst is most likely to make the right decision. Also, turns in prices can be detected earlier by looking at the charts rather than following fundamentals and the news.

The major criticism of technical analysis is that the method makes use of a huge number of highly subjective tests in order to determine the behaviour of future market price. The interpretation of the various patterns, formations and cycles is viewed by many as art rather than science, and it is very possible that different analysts may interpret the same chart quite differently. Many trading systems based on technical analysis under perform the simple buy and hold trading strategy after trading costs are considered, though this is also true for many other investment approaches.

8.4 Trading models based on technical indicators

The use of technical indicators as an investment tool has been extensively studied by

---

[1] A good example of this is the question raised by a technical analysis follower on www.trendfollowing.com. "Chipmaker Marvell Technology Group (NASDAQ:*MRVL*) breezed past Thomson First Call consensus estimates in each of the past 13 quarters, by 2% to 11%. Earnings bounded 50% or more and sales 31% and higher in the past eight quarters. Double-digit earnings and sales growth are expected through next year. Profit margins have also been strong, while cash flow has been growing. So why is the stock 52% below its all-time high?"

Brock, Lakonishok and LeBaron [7]. They applied 26 simple trading rules to 90 years of daily Dow Jones Industrial Average data and found that every single rule outperformed the benchmark of holding cash. Sullivan, Timmermann and White [22] expanded this study by applying 8,000 parameterisations of trading rules over 100 years of DJI and Standard and Poor's (S&P) daily data and concluded that the best technical trading rules are capable of generating superior performance even after removing the effect of data-snooping. However, they also found that for the last 10 years of test data (1987-1996) the trading rules did not outperform the benchmark.

Many algorithmic trading systems rely on underlying technical analysis. However, it is preferable not to rely on a single technique to provide market forecasts, but rather to use a variety of techniques in order to obtain multiple signals. Neural networks (NN) can be trained by both technical and fundamental indicators to produce trading signals. One advantage of NN systems is that they remove the need for human interpretation of charts or a series of rules for generating entry/exit signals.

Very few trading models based only on simple technical analysis manage to produce positive returns consistently. A better use of technical indicators can be made as data pre-processing elements for a NN system. It is well known that some types of indicators perform better when the market is in an upward or downward trend, while others do so when the market is in an oscillatory state with many turning points. Two different sets of indicators are required to model these two scenarios and create trend following or reversal trading strategies. Choosing the right indicators and their appropriate parameters by a trial and error process can be complicated and time consuming, with the added difficulty of choosing the appropriate model for the specific time period. It is difficult, if not impossible, for technical analysts to detect the relationship between many different indicators and strategies in order to derive a complex trading model.

In order to find optimal solutions to the above problems, a combination of neural networks and genetic algorithm are applied. The graphical representation of the system used is presented in figure 8-1.

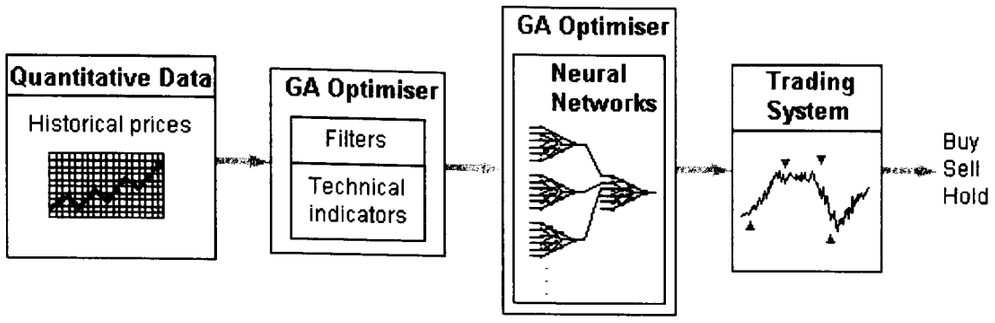


Figure 8-1. An example of a computerised trading system.

Generally, investments tend to be more biased towards buying rather than selling. This could possibly be due to the relatively recent rapid growth in western economies and even to the predominantly optimistic side of human nature. In times of crisis and economic slowdown very few investment companies manage to avoid making big losses, let alone making a profit. In order to avoid making big losses and possibly making a profit during 'bear markets'[1], a less risky long-short trading strategy is applied in most tests. Though this strategy will generally not outperform the market during a 'bull'[2] period, it will definitely reduce the risk profile and the maximum drawdown of the system.

Once these models and their parameters are estimated using the training set, their performance is validated on the out-of-sample test. Applying these models to a few time series would not be proof enough, as the results could be explained by chance. Instead they are applied to a large number of stocks which should enhance the credibility of the results. Several experiments have been conducted on a large number of stocks and their results are presented in the following sections. All the results were obtained on out-of-sample testing where model parameters were optimised on 'training' period data and tested on new data 'unseen' by the model.

8.4.1 The NN trading model applied to major US stocks

The aim of the first test is to select a representative set of US market stocks and examine their predictability through simulated trading models. The model was applied to all stocks in the NASDAQ 100, Standard & Poor's 100 (S&P 100) and the Dow

[1] Declining market for a prolonged period.

[2] Rising market for a prolonged period.

Jones 30 Industrial Index (DJI 30) daily data, representing 230 stocks in total[1]. If the model has any predictive power then we would except to see an overall positive portfolio performance sliding upwards, similar to figure 1-1 C.

Figure 8-2 shows the systems performance applied to a portfolio of NASDAQ 100 stocks and compared to the index's return for the period (20/08/1998 to 16/08/2001). Similarly, the system's performance applied to all S&P 100 and 30 DJI stocks are shown in figure 8-3 and figure 8-4 respectively. This test period is particularly interesting as the time series are roughly split in half between the bull and the bear market. It covers the final years of the rise and 'bubble burst' of the NASDAQ stock market, so that most time series charts look like an inverted letter V. The NN based trading models showed a steady and positive performance over both periods. This demonstrates the great adaptability of these models to changing market conditions.

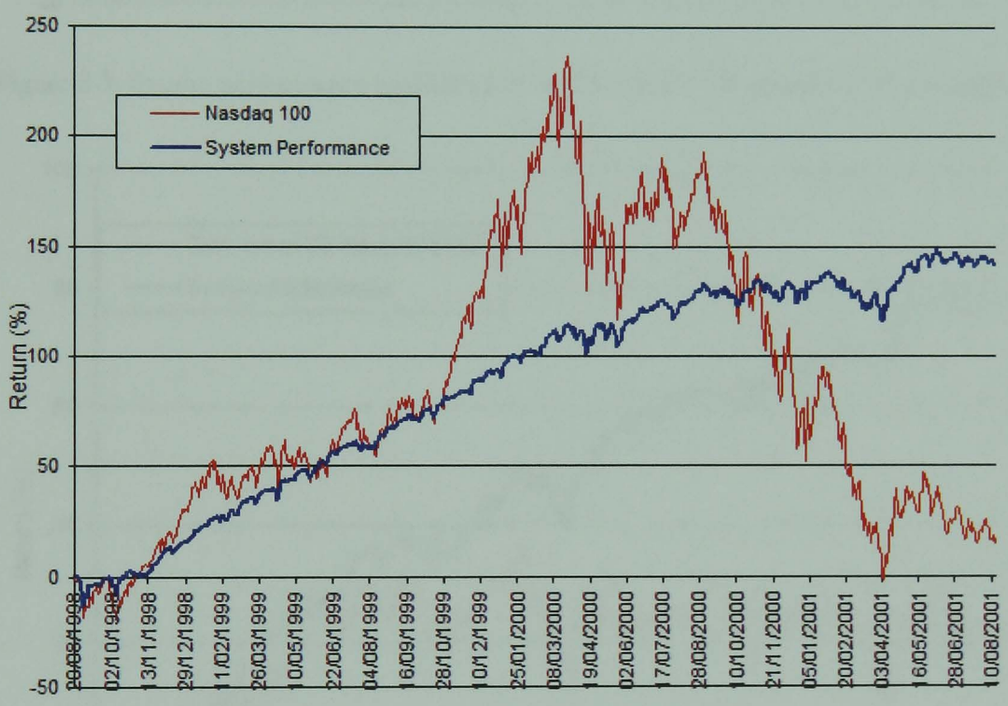


Figure 8-2. System performance applied to all stocks in the NASDAQ 100 compared to the index return.

[1] The results of this test were published in Risk & Reward magazine, Sep 2001 [116].

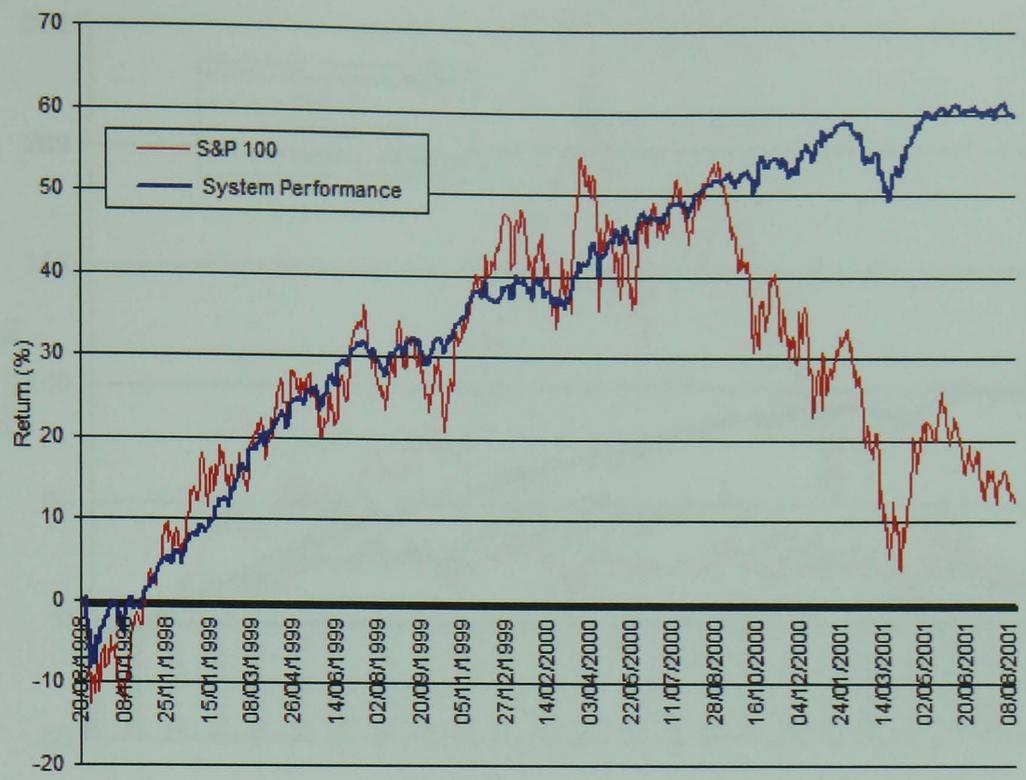


Figure 8-3. System performance applied to all stocks in S&P 100 compared to the index return.

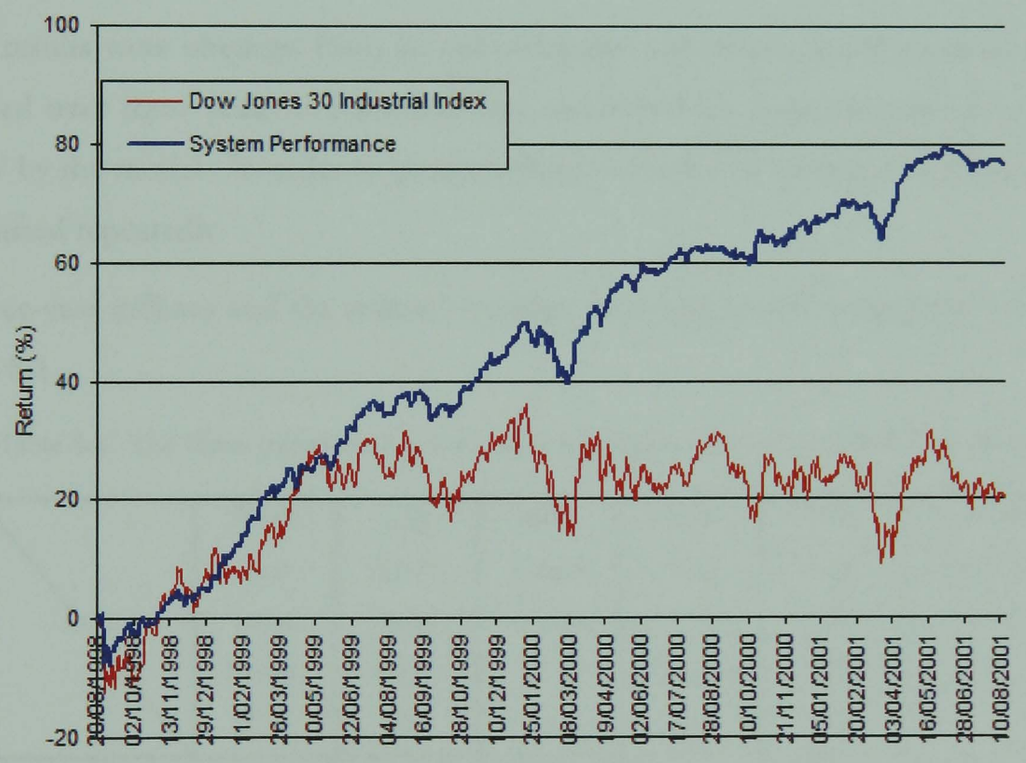


Figure 8-4. System performance applied to all stocks in DJI compared to the index return.

Figure 8-5 shows system performance for all stocks in all three indexes.

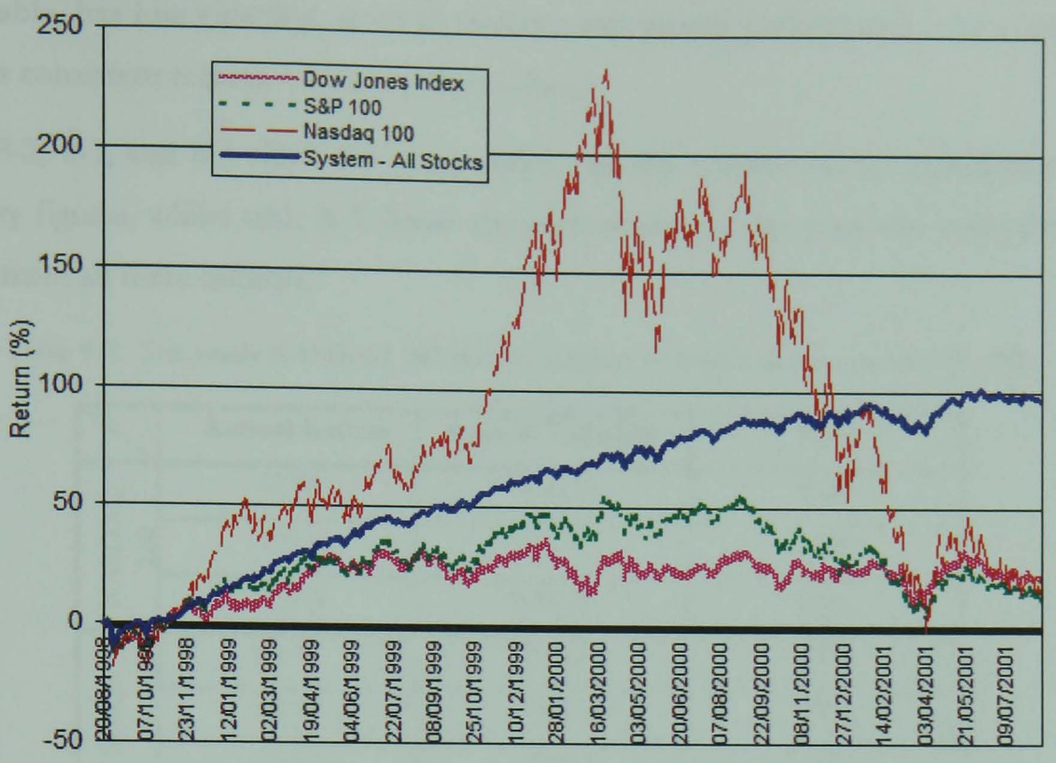


Figure 8-5. System performance applied to all stocks in all three indexes compared to the indexes' return.

All the results were obtained from an out-of-sample test where model parameters were optimised over three years of historical data and tested on three months of new data, 'unseen' by the model. In order to produce three years out-of-sample data these periods were shifted repeatedly.

The three-year indexes and the system's returns, including annual volatilities, are shown in table 8-1.

Table 8-1. The three-year indexes and the system's returns with annual standard deviations

	Three-Year Index Return (%)	Avg. Yearly Index Return (%)	Index Annual Volatility (%)	Three-Year System Return (%)	Avg. Yearly System Return (%)	System Annual Volatility (%)
S&P 100	11.97	3.99	28.21	59.65	19.88	9.77
NASDAQ 100	12.43	4.14	96.01	141.59	47.19	24.13
DJI 30	20.68	6.89	23.05	76.37	25.45	11.29

From the above charts and tables, it can be observed that the system's performance is

very stable, has low volatility, small drawdown and steady performance. As a result, it delivers consistent returns with a low risk of loss.

Table 8-2, 8-3, and 8-4 show the yearly index and the system returns including annual volatility figures, whilst table 8-5 shows the performance of the portfolio comprising all stocks from all three indexes.

Table 8-2. The yearly NASDAQ 100 and the system's returns including annual volatility.

	Annual Return	Annual Volatility	Year
NASDAQ 100	62.98	49.13	1989
	109.3	121.99	2000
	-159.8	101.26	2001
System	63.47	20.42	1989
	62.44	23.23	2000
	15.67	28.11	2001

Table 8-3. The yearly S&P 100 and the system's returns including annual volatility.

	Annual Return	Annual Volatility	Year
S&P 100	27.72	24.72	1989
	23.36	30.72	2000
	-39.11	28.77	2001
System	30.43	11.35	1989
	20.67	8.91	2000
	8.54	9.77	2001

Table 8-4. The yearly DJI and the system's returns including annual volatility.

	Annual Return	Annual Volatility	Year
DJI 30	27.32	21.13	1989
	1.35	24.69	2000
	-7.98	23.05	2001
System	36.90	12.39	1989
	25.25	11.46	2000
	14.22	9.86	2001

Table 8-5. The yearly system's returns for all stocks in the three indexes including annual volatility.

Annual Return	Annual Volatility	Year
45.82	14.36	1989
38.87	13.45	2000
11.79	15.82	2001

The system generates both long and short trades. Transaction costs are not included in the results. The average number of trades per stock for the three-year period is shown in table 8-6.

Table 8-6. Trade statistics.

	All Trades	Long Trades	Short Trades
Average number of trades	192	96	96
Average number of winning trades	110	57	53
Average number of losing trades	82	39	43
Percent profitable trades	57.29%	59.38%	55.21%

The overall system performance is summarized in the table 8-7.

Table 8-7. Portfolio statistics.

Average yearly return	32.16%
Average monthly return	2.68%
Three-year system return	96.47%
Average annual volatility	14.54%
Sharpe ratio with zero risk-free rate	2.35
Best month	8.60%
Worst month	-5.99%
Profitable months	77.77%
Worst peak-to-valley draw-down	-12.32%
Length of worst peak-to-valley draw-down in months	2.16
Recovery time from worst draw-down in months	0.87
Maximum time to new peak in months	3.03

From the graphs presented and the statistics shown in the tables, it can be concluded that the system clearly outperforms all indexes, achieving it with much lower risk, indicating that there is a strong element of predictability in these financial time series. It can also be observed that the system performs well in both bull and bear markets, as well as in a sideways market, showing a great degree of flexibility and adjustability to changing market conditions. The system exploits both the trend and the cyclical mean-reverting behaviour, aiming to eliminate market directional risk by offsetting long and short positions. The NASDAQ portfolio was the best performer with an average yearly return of 47% before trading costs and 32% net of trading costs. The reason for the NASDAQ portfolio's superior performance is most probably the presence of high volatility in this time series.

8.4.1.1 The effect of trading costs

The system undoubtedly has a predictive power, indicating that the time series under study are not random. The key question is whether these systems would still be profitable after applying the associated trading costs. The average number of trades per stock per year is 64, making this trading strategy turnover-intensive. A high turnover trading strategy can erode a portfolio's return dramatically. Assuming an average transaction cost of 0.1% per trade, the yearly equivalent cost would be 0.1% x 64 = 6.4%. If we add another 6% for the bid/ask spread cost and another 2-3% for financing and stock borrowing costs, this trading strategy can easily hit yearly costs of 15%. This would reduce a portfolio's returns considerably, questioning the viability of trading the S&P 100 portfolio as similar returns can be achieved from risk free investment. Table 8-8 shows the estimated net return for all three index portfolios and the overall portfolio.

Table 8-8. The estimated portfolio yearly returns after applying trading costs.

	NASDAQ 100	**S&P 100**	**DJI 30**	**All Stocks**
Estimated yearly return after trading costs	32%	5%	10%	17%

These results were achieved on a fixed portfolio comprising all stocks in the three US indexes. The portfolio returns could be improved by advanced stock screening

methods and portfolio optimisation. This would create a smaller, but a more profitable portfolio.

8.4.2 The NN trading model applied to the 320 largest NASDAQ stocks

In the previous experiment the NN/GA trading model applied to NASDAQ stocks was found to perform particularly well. In order to test this trading system's performance further, the model was applied to an expanded universe of stocks, comprising of the 320 largest NASDAQ companies based on their capitalisation value. The transaction costs were applied directly during model optimisation in order to reduce the number of trades and consequently the overall trading cost. The system's performance was analysed using the equity curve under different transaction costs and compared to the NASDAQ 500 index return.

The two year test period (01/03/2002 – 25/03/2004) follows the test period in the first experiment. The results of the out-of-sample test are shown in figure 8-6. As in the first experiment, the long-short strategy was used where the universe of 320 stocks have either a long or short position.

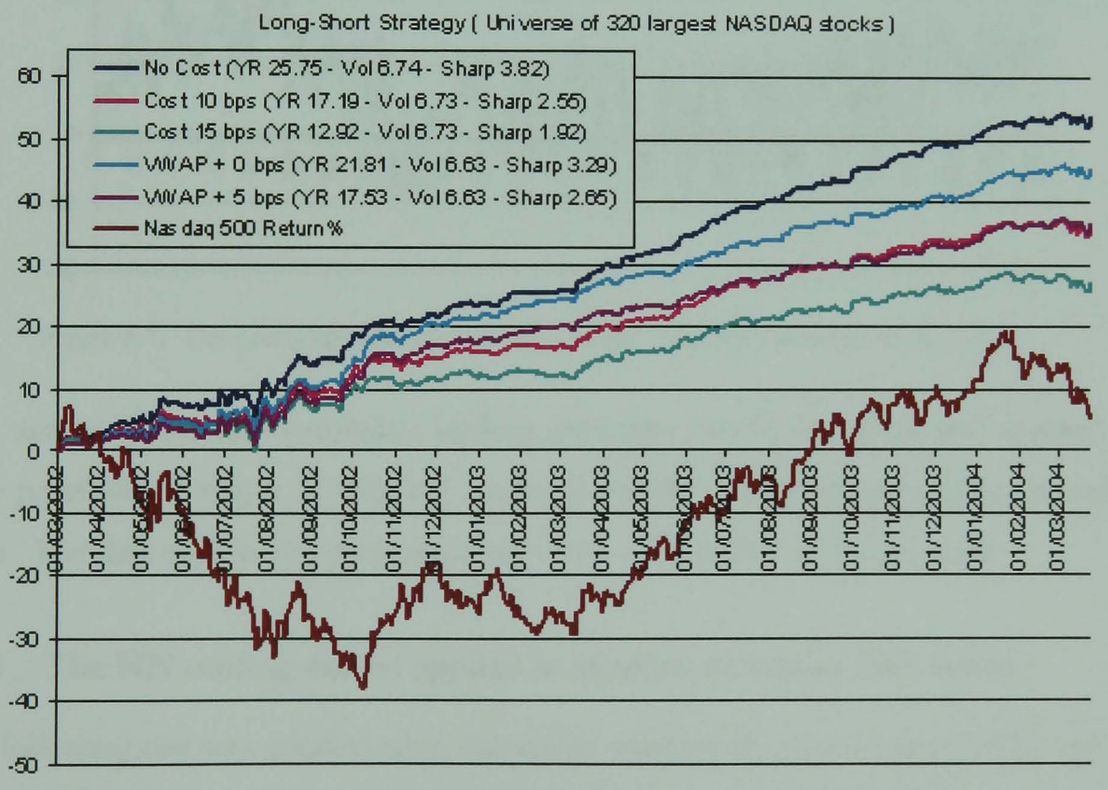


Figure 8-6. The trading system's performance applied to the 320 largest NASDAQ stocks using different costs.

The blue equity curve represents the trading system's performance without costs and the other four curves use different costs: 10 basis points (0.10%) per trade, 15 basis points per trade, simulated Volume Weighted Average Price (VWAP)[1], simulated VWAP plus 5 basis points. The simulated VWAP in this test was calculated as the average of the opening, high, low and closing price. The VWAP plus 5 basis points cost or the 10 basis points cost is a representative estimate of the real transaction cost. The information shown in the legend represents: Yearly Return (YR), Yearly Volatility (Vol) and the Sharpe Ratio (Sharp).

The trading model dynamically adjusts the number of long and short positions and on average the portfolios were around 15% long. Figure 8-7 shows a graphical example of a long invested value vs. a short invested value over time.

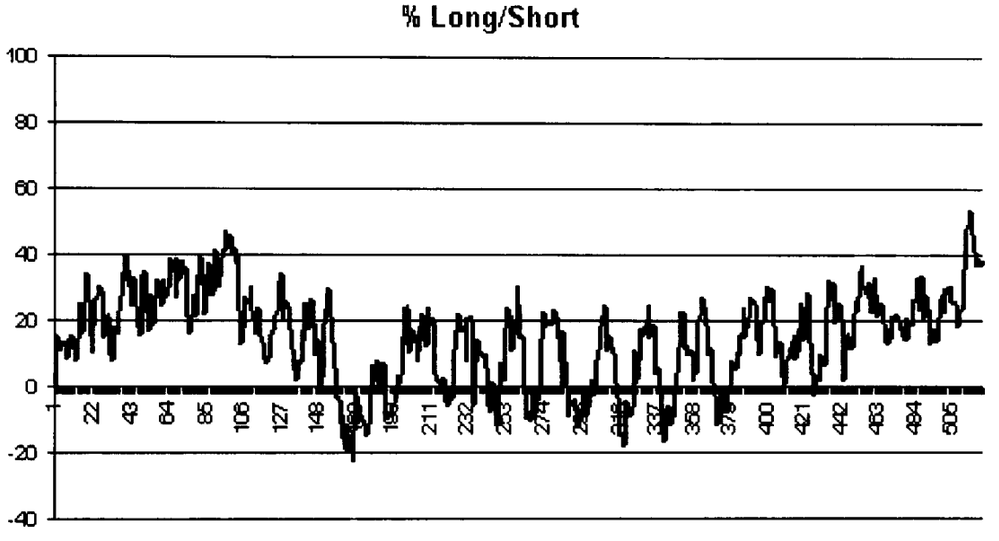


Figure 8-7. The percentage ratio of a long portfolio value vs. a short portfolio value

This test confirms that profitable trading strategies can be designed and applied to a large portfolio of major NASDAQ stocks, even after applying reasonable transaction costs. The data used in the performed tests was obtained from Yahoo.com.

8.4.3 The NN trading model applied to random vs. equity time series

The following test was conducted to determine whether the return achieved by applying the neural network trading model to a real price series differs significantly from that generated by applying it to a pseudo price series. The fact that the system return

[1] Investors that trade a large volume of stocks during day often opt to use the VWAP price which is calculated by Bloomberg (and other companies) at the end of day as the average trading price weighted by the trading volume.

achieved using the original price series is significantly different to the return achieved using the randomly generated time series provides strong support against the efficient markets hypothesis.

Two different 'anonymous' data sets, consisting of seventy-four time series each, were provided by an established hedge fund. The above NN model was applied to both portfolios and tested on five and a half years of out-of-sample data sets (03/12/98 to 09/06/04). The NN trading model was applied to both sets. The model applied to the first set produced an average yearly return of 4.07%, whilst the second set produced an average yearly return of 26.61% over the five year period. The six monthly portfolio accumulative returns were recorded and shown in figure 8-8.

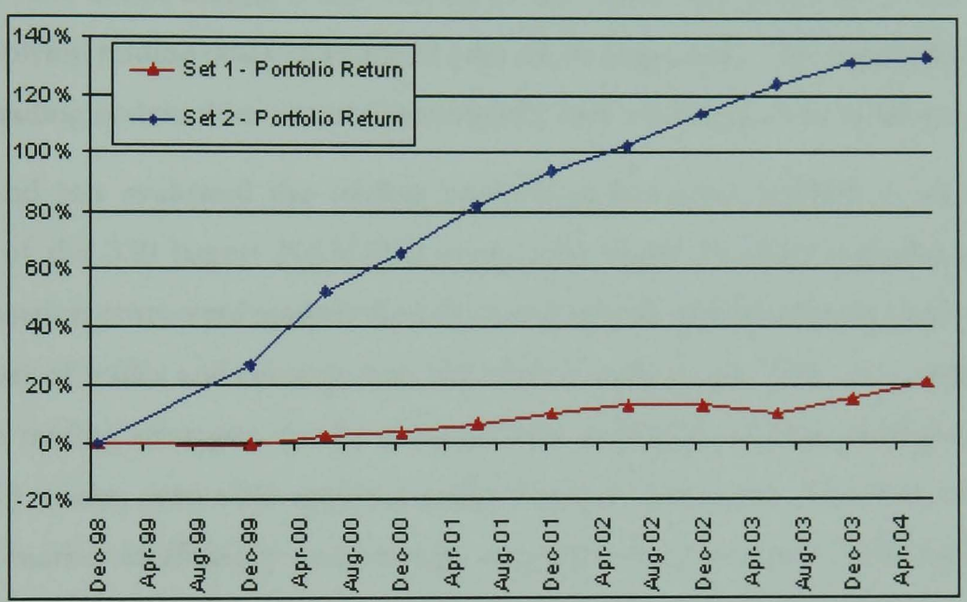


Figure 8-8. The system portfolio return for 74 European equities (Set 2) and for 74 randomly generated time series (Set 1).

The transaction costs of 0.1% per trade and 1 cent per share (a rough total cost of 15 basis points) were applied to both data sets. After producing the results, it was revealed that the first portfolio (Set 1) was randomly generated and the second portfolio (Set 2) was made up of transformed European equity prices. The real European equity portfolio return is significantly different to that of a randomly generated portfolio. It can be concluded with a high degree of certainty that these time series are not random and have some serial dependence and patterns, allowing for the creation of profitable trading systems. In addition, it seems that European markets are less efficient and more predictable than US markets.

8.4.4 Conclusions

In this chapter three tests were performed. The first test was designed to assess the predictability of a large number of US stocks through the use of a neural network trading system without applying transaction costs. This test confirmed that the neural network trading system clearly outperforms all indexes, showing that there is a strong element of predictability in these financial time series. Based upon the results of this test we can reject the null hypothesis. The NN trading system performs well in both bull and bear markets, as well as in a sideways market, demonstrating a great degree of flexibility and adjustability toward changing market conditions. The NASDAQ portfolio was the best performer, with an average yearly return of 47% before trading costs and 32% net of trading costs. However, the yearly S&P portfolio return was 20% before applying trading costs and only 5% net of trading costs. This highlights the point that the trading system does not perform equally well in all sectors or industries.

The second test evaluated the trading model's performance applied to an expanded universe of the 320 largest NASDAQ companies based on their capitalisation value. The transaction costs were applied directly during model optimisation in order to reduce the number of trades and consequently the overall trading cost. This test confirmed that profitable trading strategies can be designed and applied to a large portfolio of major NASDAQ stocks, even after applying realistic transaction costs. This test, once more, confirms market inefficiency and strongly supports the rejection of null hypothesis in this thesis.

The most convincing test against market inefficiency was the European equity market test. The European equity portfolio return was significantly different to that of a randomly generated portfolio. It can be concluded with a high degree of certainty that those time series for the period tested were not random, allowing for the creation of profitable trading systems.

Chapter 9

Summary, Conclusions and Recommendations

9.1 Introduction

This chapter presents an overall summary of the research. Discussions and conclusions of the most prominent findings are made and some areas for future research are identified. The main areas of this research have been the study and characterisation of financial time series in order to design an advanced self-evolving trading system based on neural networks and genetic algorithms, and subsequently establish whether stock returns are predictable enough to create profitable trading strategies.

9.2 Hypotheses rejections/acceptances

The test results performed in this research showed that profitable trading models utilising advanced nonlinear trading systems and applied to equity markets can be created after accounting for reasonable transaction costs, thus **rejecting** the null hypothesis (H1). This indicates that equity markets are not fully efficient and random, but to the contrary, they are nonlinear and to some degree predictable. This predictability is not uniformly spread; some markets, some time series within the same market and some time periods within the same time series are more predictable than others. It is possible to devise trading strategies that exploit the predictability in each of these levels or with a combined approach.

The H2 hypothesis is **accepted**, i.e. neural networks represent superior forecasting models compared to other non-parametric approaches tested: nearest neighbour, kernel regression, locally constant, locally linear, locally weighted linear, and radial basis models. The neural network model also outperforms the "random walk" and unconditional mean reference models in the case of filtered DJI*-F and DJI*-FD time series achieving a small level of error, indicating that these time series have a large degree of determinism. The neural network DJI* predictions were as good as the reference predictors.

9.3 Summary conclusions

The main conclusions from this research can be summarised as follows:

9.3.1 Market characteristics and fractal properties

This research has found that financial markets are more volatile than is generally assumed. These volatile periods are clustered in time, and the volatility (standard deviation) is constantly changing over time. The rescale range analysis and the Hurst exponent indicated the presence of memory in financial time series and showed that prices are not independent of each other and do not rise or fall by the mathematical rules of chance. The Hurst exponent was also used to characterise time series and utilised in trading strategies accordingly.

9.3.2 Extreme events

The existence of fat tails in the distributions of financial returns is widely known, but the great risk they represent is underestimated by most market practitioners. Economic bubbles and crashes are not new phenomena, they go way back in history, but seem to be soon forgotten, with few lessons learned. They are inevitable and probably will always exist, mainly due to investors' greed and to 'herd' behaviour within human nature. These are major factors influencing the nonlinear behaviour of financial markets. If market bubbles and crashes are inevitable at times, mechanisms should be in place in order to control them better, slow them down and minimise their effect. This can only be done through the development and implementation of adequate risk models by all or at least the majority of market players. Currently very few financial organisations have adequate risk models to prevent and cope with these extreme situations. With a better understanding of market behaviour, new models need to be developed to include historic data covering market bubbles and crashes. Fractal market models and their variants show a promising approach in analysing extreme events. This research has shown that recurrence plots can be used to discover fractal properties within financial time series and to anticipate the onset of extreme events. Market crashes play such a big part in the finance, and despite this, it is a subject hardly covered in business schools and finance programs.

A detailed analysis of nonlinear dynamical systems and the calculation of their invariants have been presented in chapter four. It has been shown that the characterisation of

dynamical time series and the estimation of their invariants work well on stable systems with low noise, but when presented with real financial data these methods are fraught with difficulties. This is mainly due to serial correlations, noise and non-stationarity present in financial time series. Ideally both the embedding dimension and time delay should be estimated together in one procedure that yields the best predictive model. These parameters required for the successful modelling of nonlinear systems could be found experimentally through the use of neural networks. Both of the methods used to estimate the largest Lyapunov exponent confirmed the presence of chaos in the time series. Even the successful short-term predictions of these time series give a strong indication that they can be chaotic. However, the RQA analysis did not confirm this.

9.3.3 Nonlinear analysis and financial predictions

The RQA analysis found a strong presence of structure, recurrence and determinism in the financial time series studied. The structure and the measures were completely destroyed by randomly re-shuffling the financial time series in a way that the mean and standard deviation were preserved. Crucial transition periods were also detected just before the start of a big upward rally in the DJI index. The typical values of RQA measures that are common to chaotic behaviour were not present in the results overall, though there are some indications that they may exist during shorter time periods. It has been shown that RQA analysis can be used as a powerful tool to detect hidden properties driven by nonlinear market mechanisms. The measure of determinism estimated by RQA coincides with the forecasting ability of the successive prediction models, i.e. a higher *DET* value yields a better prediction.

Nonlinear dynamic theory plays a big part in the analysis and characterisation of financial markets, but its use in forecasting, utilising the univariate time delay embedding method, is not clear. Using data from a single time series without any additional information can be very restricting. Despite this, trading systems based on a single time series and its past data have produced positive portfolio returns. Even limited additional information in the form of opening, closing, high and low prices, including volume, can drastically improve a model's performance. The superior performance of multivariate nonlinear models is demonstrated in this research.

9.3.4 Fundamental forecasting models

This research has shown that higher returns can be achieved by utilising models based on fundamental measures found in a company's financial statements. The overall rank measure derived from fundamental ratios has a significant effect on the different portfolio returns. Portfolios comprising stocks with a low price-to-book ratio (value stocks) outperformed those made up of high price-to-book ratio stocks (growth stocks). The use of neural network models further improved these models. More complex nonlinear models also outperformed simple linear ones, indicating their suitability in modelling financial time series, and a presence of nonlinearity in data. Individual stock models, where parameters are optimised using data from the particular stock only, performed better than global models that used one set of estimated parameters which are derived using data from all stocks. Also, the fundamental models that used a pre-processing of data to include prior knowledge performed better.

9.3.5 Technical forecasting models

Several tests utilising neural networks and a technical analysis approach confirmed that profitable trading strategies can be designed and applied to a large portfolio of major US stocks, even after applying reasonable transaction costs. These systems outperformed the benchmark (indexes) returns with much lower risk, indicating that there is a strong element of predictability in these financial time series. They also performed well in both bull and bear markets, as well as in a sideways market, showing a great degree of flexibility and adjustability to changing market conditions.

The tests applied to European equities and a randomly generated portfolio were even more convincing. The return achieved in the case of the equity price series is significantly different and larger than the return obtained using a randomly generated time series, providing strong support against the efficient market hypothesis and a rejection of the null hypothesis.

These multivariate statistical models performed significantly better when compared to the univariate models. The quality of predictions and ultimately the system performance largely depended on the stocks analysed, i.e. the more volatile NASDAQ stocks showed a better performance.

9.3.6 Overall conclusion

The dynamics of financial markets will never be an exact science, however, from the experiment results presented in this research we can conclude that the markets are predictable to a different degree at different times and this presence of predictable components can lead to a rejection of the null hypothesis, which also implies the rejection of the Random Walk Hypothesis. It must be stressed that no trading system can predict future prices accurately, but what can be estimated is the probability of future price movements. This approach would improve the odds of making a profitable trade in much the same way that a casino makes money from its roulette wheel. The number that will come up next is unknown, but the odds are such that the casino will make money in the long run. After all, predictability is the main reason and driving force behind the existence of many investment funds. If financial markets were perfectly efficient and random, there would be little reason to trade and they could eventually collapse. In fact, most natural and man made systems/processes are far from being perfectly random and efficient, though they may seem to be. Even what is designed to be random, in reality it is rarely the case. Being able to forecast prices does not imply that the markets are not functioning well, and not being able to forecast prices does not imply the opposite. These market inefficiencies and profit opportunities must be big enough to compensate for the cost of trading and information gathering. Opportunities are in general created by competitive advantage, based upon superior information and financial innovation and technology.

The future of trading is moving slowly towards technology. More and more traders will be replaced or supported by computerised trading systems. Many of these will be largely based on artificial intelligence (AI) tools such as neural networks. It is vital to choose the correct combination of these processing technologies within a model, as this can simplify solutions and improve performance. A good understanding of AI tools is necessary, but a good knowledge of trading systems and techniques is equally important. In current market conditions it may be anticipated that traditional and hedge fund managers will learn to rely on such new trading and analytical tools.

9.4 Future recommendations

The obvious extension to this research is to create new trading models that combine

fundamental and technical analysis. The selection of fundamental factors could be extended to include macroeconomic variables. Technical analysis should explore other ways of data pre-processing, such as wavelets analysis. Employing architectures that also allow for stretching and compressing of time series (fractal feature) can be of particular value.

In addition to the above suggestions, the next stage of this research could be the application of Natural Language Processing (NLP) tools to the analysis of the vast amount of financial textual information. Also, the use of artificial agents in order to model market behaviour can be explored.

The models may be expanded to other financial instruments, such as currencies, commodities and certain derivatives.

References

[1] Achelis B. S. (1994), *Technical Analysis from A to Z*, Probus Publishing Company.

[2] Azoff M. R. (1994), *Neural Network Time Series Forecasting of Financial Markets*, Wiley Finance Editions.

[3] Barnes R. M. (1997), *Trading in Choppy Markets*, Irwin Professional Publishing.

[4] Bachelier L. (1900), *Théorie de la Spéculation*. Doctorate dissertation. Translation in Cootner (1964) [54] .

[5] Bishop C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford University Press.

[6] Blau W. (1995), Momentum, *Direction, and Divergence*, John Wiley & Sons, Inc.

[7] Brock W., Lakonishok J., and LeBaron B. (1992), *Simple Technical Trading Rules and the Stochastic properties of Stock Returns*, Journal of Finance 47, pp.1731-1746.

[8] Deboeck G. J. (1994), *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets*, John Wiley & Sons Inc.

[9] Diebold F. X. (1998), *Elements of Forecasting*, South-Western College Publishing.

[10] Dunis C. (1996), *Forecasting Financial Markets*, John Wiley & Sons Ltd.

[11] Freedman R. S., Klein R. A., and Lederman J. (eds.) (1994), *Artificial Intelligence in the Capital markets*, Probus Publishing Company.

[12] Goonatilake S., and Treleaven P. (1995), *Intelligent Systems for Finance and Business*, John Wiley & Sons Ltd.

[13] Lederman J., and Klein R. A. (eds.) (1995), *Virtual Trading*, Probus Publishing Company.

[14] Loofbourrow J. W., and Loofbourrow T. H. (1994), *What Artificial Intelligence Brings to Trading and Portfolio Management* in Freedman R. S., Klein R. A. and Lederman J. (eds.) *Artificial Intelligence in the Capital Markets*, Probus Publishing Company, pp. 3-28.

[15] Luca C. (1997), *Technical Analysis Applications in the Global Currency Markets*, Prentice Hall.

[16] Mandelbrot B. B. (1999), *A Multifractal Walk down Wall Street*, Scientific American, February 1999, pp. 50-53.

[17] Peters E. E. (1991), *Chaos and Order in the Capital Markets*, John Wiley & Sons, Inc.

[18] Peters E. E. (1994), *Fractal Market Analysis*, John Wiley & Sons, Inc.

[19] Refenes A. N. (ed.) (1995), *Neural Networks in the Capital Markets*, John Wiley & Sons Ltd.

[20] Rosenberg M. R. (1996), *Currency Forecasting*, McGraw-Hill.

[21] Samuelson P. A. (1981), *Economics*, McGraw-Hill.

[22] Sullivan R., Timmermann A., and White H. (1997), *Data-Snooping, Technical Trading Rule Performance, and the Bootstrap*, Technical Report, University of California, San Diego.

[23] Trippi R. R. (ed.) (1995), *Chaos & Nonlinear Dynamics in the Financial Markets*, Irwin Professional Publishing.

[24] Trippi R. R. and Turban E. (eds.) (1995), *Neural Networks in Finance and Investing*, Irwin Professional Publishing.

[25] Weigend S. A., and Gershenfeld N. A. (eds.) (1994), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley Publishing Company.

[26] Saci K., and Holden K. (2005), *Evidence on Growth and Financial Development using Principal Components*, School of Accounting, Finance and Economics, Liverpool John Moores University.

[27] Masters T. (1995), *Advanced Algorithms for Neural Networks: A C++ Sourcebook*, NY: John Wiley and Sons.

[28] Specht D. F. (1991), *A Generalized Regression Neural Network*, IEEE Transactions on Neural Networks, 2, Nov. 1991, pp. 568-575.

[29] Williamson D. (2005), *The Royals Mail's Magical Profit and Cash Flow Figure.*

[30] Dimson E., Nagel S., and Quigley G. (2003), *Capturing the Value Premium in the United Kingdom*, Financial Analysts Journal, Vol. 59, No. 6: pp. 35-45.

[31] Financial Dictionary, http://financial-dictionary.thefreedictionary.com.

[32] Dunis C. L., and Reilly M. D. *Alternative Valuation Techniques For Predicting UK Stock Returns*, CIBEF – Centre for International Banking, Economics and Finance.

[33] Capaul C, Rowley I., and Sharpe W. F. (1993), *International Value and Growth Stock Returns*, Financial Analyst Journal 49, pp. 27-36.

[34] Fama E. F., and French K. R. (1998), *Value Versus Growth: The International Evidence*, The Journal of Finance 53, pp. 1975-1999.

[35] Fama E. F., and French K. R. (1995), *Size and Book-to-market Factors in Earnings and Returns*, The Journal of Finance 50, pp. 131-155.

[36] Lakonishok J., Shleifer A., and Vishny R. W. (1994), *Contrarian Investment, Extrapolation, and Risk*, The Journal of Finance 44, pp. 1541-1578.

[37] Piotroski J. D. (2000), *Value Investing: The Use of Historical Financial Statement Information to Separate Winners from Losers*, Journal of Accounting Research, Vol. 38. Supplement, pp. 1-41.

[38] Mohanram P. S. (2005), *Separating Winners from Losers among Low Book-to-market Stocks using Financial Statement Analysis*, Review of Accounting Studies (10), pp. 133-170.

[39] Chan L. K. C., Yasushi H., and Lakonishok J. (1991), *Fundamentals and Stock Returns in Japan*, The Journal of Finance 46, pp. 1739-1789.

[40] Hurst H. E. (1951), *Long Term Storage Capacity of Reservoirs,* Transactions of the American Society of Civil Engineers, Vol. 116, pp. 770-799.

[41] Mandelbrot B. B. (1971), *When Can Price be Arbitraged? A Limit to the Validity of the Random Walk and Martingale Models,* Review of Economics and Statistics, Vol. 53, pp. 225-236.

[42] Mandelbrot B. B. (1972), *Statistics Methodology for Non-periodic Cycles: from the Covariance to R/S Analysis,* Annals of Economics and Social Measurement, Vol. 1, pp. 259-290.

[43] Riedi R. H. (2003), *Multifractal processes,* in Doukhan P., Oppenheim G. and Taqqu M. S. (eds.) *Theory and Applications of Long-Range Dependence*, Birkhäuser, pp. 625-716

[44] William F. (1951), *The Asymptotic Distribution of the Range of Sums of Independent Random Variables* , The Annals of Mathematical Statistics, Vol. 22, pp. 427-432.

[45] West B. J., and Goldberger A. L. (1987), *Physiology in Fractal Dimensions*, American Scientist 75.

[46] Mandelbrot B. B. (1967), *How long is the coast of Britain? Statistical self-similarity and fractional dimension,* Science 156, pp. 636-638.

[47] Mandelbrot B. B. (1982), *The Fractal Geometry of Nature,* New York, W. H. Freeman & Co.

[48] Barnsley M. (1988), *Fractals Everywhere*, Academic Press

[49] Fama E. F. (1965), *The Behaviour of Stock Market Prices,* Journal of Business 38

[50] Mandelbrot B. B. (1964), *The Variation of Certain Speculative Prices,* in P. Cootner, ed., *The Random Character of Stock Prices.* Cambridge, MA, M.I.T. Press.

[51] Ross S. A. (1976), *The Arbitrage Theory of Capital Asset Pricing*, Journal of Economic Theory 13.

[52] Sharpe W. F. (1963), *A Simplified Model of Portfolio Analysis*, Management Science 9.

[53] Mandelbrot B. B., and Hudson R. L. (2004), *The (mis)Behaviour of Markets*, London, Profile Books.

[54] Cootner P. H. (1964), *The Random Character of Stock Market Prices*, Cambridge, MA, M.I.T. Press.

[55] Taleb N. N. (2001), *Fooled by Randomness*, New York, TEXERE Publishing Limited.

[56] Takens F. (1981), *Detecting Strange Attractors in Fluid Turbulence*, Dynamical Systems and Turbulence, edited by D. A. Rand and L. S. Young, Lecture Notes in Mathematics, Vol. 898. pp. 363-381, Berlin, Springer-Verlag.

[57] Akaike H. (1970), *Statistical Predictor Identification*, Ann. Inst. Stat. Math. 22, pp. 203-217.

[58] Ruelle D. (1970), *Small random perturbations of dynamical systems and the definition of attractors*, Communications in Mathematical Physics 137, pp. 82.

[59] Packard N. H., Crutchfeld J. P., Farmer J. D., and Shaw R. S. (1980), *Geometry from a Time Series*, Physical Review Letters 45(9), pp. 712-716.

[60] Grassberger P. (1983), *Generalized Dimension of Strange Attractors*, Physical Letters A 97, pp. 227-230.

[61] Grassberger P. (1985), *Generalizations of the Hausdorff Dimension of Fractal Measures*, Physical Letters 107, pp. 101-105.

[62] Grassberger P., and Procaccia I. (1983), *Characterization of Strange Attractors*, Physical Review Letters 50, pp. 346-349.

[63] Theiler J. (1990), *Estimating Fractal Dimension*, Journal of the Optical Society of America A 7(6), pp. 1055-1073.

[64] Kennel M. B, Brown R., and Abarbanel H. D. I. (1992), *Determining Minimum Embedding Dimension Using a Geometrical Construction*, Physical Review A 45(6), pp. 3403-3411.

[65] Ruelle D. (1989), *Chaotic Evolution and Strange Attractors*, Cambridge, England, Cambridge University Press.

[66] Wolf A., Swift J. B., Swinney H. L., and Vastano J. A. (1985), *Determining Lyapunov Exponents from a Time Series*, Physica D 16, pp. 285-317.

[67] Shannon C. E., and Weaver W. (1964), *The Mathematical Theory of Communications*, Urbana: University of Illinois Press.

[68] Alligood K., Yorke J., and Sauer T. (1995), *Chaos – An Introduction to Dynamical Systems*, Springer-Verlag.

[69] Weigend A. S., Huberman B. A., and Rumelhart (1990), *Predicting the Future: Connectionist Approach*, International Journal of Neural Systems 1, pp. 193-209.

[70] Gencay R., and Dechert W. D. (1992), *An Algorithm for the n Lyapunov Exponents of a n-Dimensional Unknown Dynamical System*, Physica D 59, pp. 142-157.

[71] Hsieh D. A. (1991), *Chaos and nonlinear dynamics: Application to financial markets*, The Journal of Finance, Vol. 46:5, pp. 1840-1847.

[72] Brock W. A., Hsieh D. A., and LeBaron B. (1991), *Nonlinear Dynamics, Chaos, and Instability: Statistical Theory and Economic Evidence*, The MIT Press, Cambridge MA.

[73] Sauer T., Mingzhou D. , Grebogi C., Ott E., and Yorke J. A. (1993), *Estimating correlation dimension from a chaotic time series: when does plateau onset occur?*, Physica D, Vol. 69, Issue 3-4, pp. 404-424.

[74] Theiler J., and Eubank S. (1992), *Don't Bleach Chaotic Data*, Technical Report LA-UR-92-1575, Los Alamos National Laboratory, Los Alamos, NM.

[75] Theiler J., Linsay P. S., and Rubin D.M. (1994), *Detecting Nonlinearity in Data with Long Coherence Times*, in Weigend S. A., and Gershenfeld N. A. (eds.) (1994), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley Publishing Company.

[76] Fraser A. M., and Swinney H. L. (1986), *Independent Coordinates for Strange Attractors from Mutual Information*, Physical Review A 33, pp. 1134-1140.

[77] Eckmann J. P., Kamphorst S. O., and Ruelle D. (1987), *Recurrence plots of dynamical systems*, Europhysics Letters, Vol. 4, No. 9, pp. 973-977.

[78] Zbilut J. P., and Webber Jr. C. L. (1992), *Embedding and delays as derived from quantification of recurrence plots*, Physics Letters A 171, pp. 199-203.

[79] Barreira L., and Saussol B. (2001), *Hausdorff dimension of measure via Poincaré recurrence*, Communications in Mathematical Physics 219 (2), pp. 443-463.

[80] Saussol B., Troubetzkoy S., and Vaienti S. (2002), *Recurrence, dimensions, and Lyapunov exponents*, J. Statist. Phys. 106 (3–4), pp. 623–634.

[81] Hadyn N., Luevano J., Mantica G., and Vaienti S. (2002), *Multifractal properties of return time statistics*, Physical Review Letters 88 (22), 224502

[82] Markowitz H. M. (1952), *Portfolio Selection*, Journal of Finance 7.

[83] Markowitz H. M. (1959), *Portfolio Selection: Efficient Diversification of Investments*, New Haven, CT, Yale University Press.

[84] Kantz H. (1994), A *robust method to estimate the maximal Lyapunov exponent of a time series*, Physical Review A, 185, pp. 77.

[85] Rosenstein M. T., Collins J. J., (1993), *A practical method for calculating largest Lyapunov exponents from small data sets*, Physica D, 65, pp. 117.

[86] Iwanski J. S., and Bradley E. (1988), *Recurrence plots of experimental data: To embed or not to embed?*, Chaos 8, pp. 861-871.

[87] Marwan N., Wessel N., Meyerfeldt U., Schirdewan A., and Kurths J. (2002), *Recurrence plot based measures of complexity and its application to heart rate variability data*, Physical Review E, 66(2), 026702.

[88] Casdagli M. C. (1997), *Recurrence Plots Revisited, Physica D* 108, 12-44.

[89] Aparicio T., Pozo E.F., and Saura D. (2005), *Detecting Determinism Using Recurrence Quantification Analysis: Three Test Procedures,* Department of Economics Analysis, University of Zaragoza.

[90] Marwan N., Romano M. C., Thiel M., and Kurths J. (2007), *Recurrence plots for the analysis of complex systems*, Physics Reports 438 pp. 273-329. Also available online at www.sciencedirect.com.

[91] Marwan N. (2003), *Encounters With Neighbours – Current Developments of Concepts Based on Recurrence Plots and Their Applications.*, Ph.D. Thesis Institute of Physics, University of Postdam, ISBN 3-00-012347-4.

[92] Webber C. L., and Zbilut J. P. (1994), *Dynamical assessment of physiological systems and states using recurrence plot strategies*, Journal of Applied Physiology, 76 (2), pp. 965-973.

[93] Trulla L. L., Giuliani A., Zbilut J. P., and Webber Jr. C. L. (1996), *Recurrence quantification analysis of the logistic equation with transients*, Physical Letters A 223 (4), pp. 255–260.

[94] Atay F. M., and Altintas Y. (1999), *Recovering smooth dynamics from time series with the aid of recurrence plots*, Physical Review E 59, pp. 6593-6598.

[95] Osborne A. R., and Provenzale A. (1989), *Finite Correlation Dimension for Stochastic Systems with Power-Law Spectra*, Physica D 35, pp. 357-381.

[96] White H. (1989), *Learning in neural networks: A statistical perspective*, Neural

Computation 4, pp. 425-464.

[97] Rumelhart, D. E., Hinton, G. E., and Williams R. J. (1986). *Learning internal representations by error propagation*, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, editors Rumelhart, D. E. and McClelland, J. L., Volume 1, MIT Press, Cambridge MA., pp. 318–362.

[98] Rumelhart, D. E., and McClelland J. L. (1986), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. I & II, MIT Press.

[99] Kohonen T. (1989), *Self-Organization and Associative Memory*, Springer-Verlag, Berlin.

[100] MacKay D. J. C. (2003), *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press.

[101] Hecht-Nielsen R. (1989), *Neurocomputing*. Addison-Wesley, Reading, MA.

[102] Poggio T., and Girosi F. (1990), *Regularization algorithms for learning that are equivalent to multilayer networks*, Science Vol. 247, pp. 978-982.

[103] Moody J. E. (1992), *The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems*, Advance in Neural Information Processing Systems, Morgan Kaufman, san Mateo, CA.

[104] Stork D., and Hassibi B. (1993), *Second order derivatives for network pruning: Optimal Brain Surgeon*, In Sejnowski T. J., Hinton G. E., and Touretzky D. S., editors, Advances in Neural Information Processing Systems 5, San Mateo, CA, Morgan Kaufmann Publishers Inc., pp. 164-171.

[105] LeCun Y., Denker J., and Solla S. (1990), *Optimal brain damage*, in Advances in Neural Information Processing Systems (Touretzky D., ed.), San Mateo, CA, Morgan Kaufmann Publishers Inc, pp. 396–404.

[106] Smolensky P., and Mozer M. (1989), *Skeletonization: A Technique for Trimming the Fat from a Network via Relevance Assessment*, in Advances in Neural Information Processing Systems 1 (Touretzky D., ed.), San Mateo, CA, Morgan Kaufmann Publishers Inc, pp. 107–115.

[107] Falham S. E., and Lebiere C. (1991), *The cascade-correlation learning architecture*, Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University.

[108] Krogh A., and Hertz J. (1992), *A simple weight decay can improve generalization*, in Advances in Neural Information Processing Systems 4 (J. Moody, S. Hanson, and

R. Lippmann, eds.), San Mateo, CA: Morgan Kaufmann Publishers, pp. 950–957.

[109] Holland H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Harbor.

[110] Goldberg E. D. (1989), *Genetic algorithm in search, optimization, and machine learning*, Reading, Massachusetts, Addison-Wesley.

[111] Koza R. J. (1992), *Genetic programming, on the programming of computers by means of natural selection*, Cambridge, Massachusetts, MIT Press.

[112] Man F. K., Tang S. K., and Kwong S. (1999), *Genetic algorithms: Concepts and designs*, Heidelberg, Springer-Verlag.

[113] Fisher B., and Scholes M. (1973), *The pricing of options and corporate liabilities*, Journal of Political Economy 81, pp. 637-654.

[114] Pring M. J. (1985), *Technical analysis explained*, New York, McGraw-Hill Book.

[115] Murphy J. J. (1986), *Technical analysis of the futures markets: A comprehensive guide to trading methods and applications*, New York, Prentice-Hall Company, Inc.

[116] Milanović V. (2001), *Two heads are better than one*, Risk & Reward, Strategic thinking in alternative assets, FOW and MAR/Hedge.

[117] Williams R. W., and Herrup K. (2001), *The Control of Neuron Number*, Originally published in The Annual Review of Neuroscience 11 (1988), pp. 423–453.

[118] Lawrence S., Giles C., and Tsoi A. (1996), *What size neural network gives optimal generalization?*, Tech. Rep. UMIACS-TR-22, Institute for Advanced Computer Studies, University of Maryland, College Park, MD

[119] Chenoweth T., and Obradović Z. (1996), *A multi-component nonlinear prediction system for the S&P 500 index*, Neurocomputing. Vol. 10, Issue 3, pp. 275-290.

Appendix A

Examples of yearly IBM financial statements[1].

IBM Income Statement — Annual data, all numbers in thousands

PERIOD ENDING	31-Dec-06	31-Dec-05	31-Dec-04
Total Revenue	**91,424,000**	**91,134,000**	**96,293,000**
Cost of Revenue	53,129,000	54,602,000	60,261,000
Gross Profit	**38,295,000**	**36,532,000**	**36,032,000**
Operating Expenses			
Research Development	6,107,000	5,842,000	5,673,000
Selling General and Administrative	20,259,000	21,314,000	19,384,000
Non Recurring	(900,000)	-	(1,169,000)
Others	-	-	-
Total Operating Expenses	-	-	-
Operating Income or Loss	**12,829,000**	**9,376,000**	**12,144,000**
Income from Continuing Operations			
Total Other Income/Expenses Net	766,000	3,070,000	23,000
Earnings Before Interest And Taxes	13,595,000	12,446,000	12,167,000
Interest Expense	278,000	220,000	139,000
Income Before Tax	13,317,000	12,226,000	12,028,000
Income Tax Expense	3,901,000	4,232,000	3,580,000
Minority Interest	-	-	-
Net Income From Continuing Ops	9,416,000	7,994,000	8,448,000
Non-recurring Events			
Discontinued Operations	76,000	(24,000)	(18,000)
Extraordinary Items	-	-	-
Effect Of Accounting Changes	-	(36,000)	-
Other Items	-	-	-
Net Income	**9,492,000**	**7,934,000**	**8,430,000**
Preferred Stock And Other Adjustments	-	-	-
Net Income Applicable To Common Shares	**$9,492,000**	**$7,934,000**	**$8,430,000**

[1] Source Yahoo.

IBM Balance Sheet

Annual data, all numbers in thousands

PERIOD ENDING	31-Dec-06	31-Dec-05	31-Dec-04
Assets			
Current Assets			
Cash And Cash Equivalents	8,022,000	12,568,000	10,053,000
Short Term Investments	2,634,000	1,118,000	517,000
Net Receivables	28,655,000	26,193,000	30,365,000
Inventory	2,810,000	2,841,000	3,316,000
Other Current Assets	2,539,000	2,941,000	2,719,000
Total Current Assets	**44,660,000**	**45,661,000**	**46,970,000**
Long Term Investments	18,449,000	14,602,000	16,418,000
Property Plant and Equipment	14,439,000	13,756,000	15,175,000
Goodwill	12,854,000	9,441,000	8,437,000
Intangible Assets	2,202,000	1,663,000	1,789,000
Accumulated Amortization	-	-	-
Other Assets	10,629,000	20,625,000	20,394,000
Deferred Long Term Asset Charges	-	-	-
Total Assets	**103,233,000**	**105,748,000**	**109,183,000**
Liabilities			
Current Liabilities			
Accounts Payable	18,006,000	17,292,000	24,524,000
Short/Current Long Term Debt	8,902,000	7,216,000	8,099,000
Other Current Liabilities	13,182,000	10,644,000	7,175,000
Total Current Liabilities	**40,090,000**	**35,152,000**	**39,798,000**
Long Term Debt	13,780,000	15,425,000	14,828,000
Other Liabilities	17,690,000	22,073,000	24,810,000
Deferred Long Term Liability Charges	3,167,000	-	-
Minority Interest	-	-	-
Negative Goodwill	-	-	-
Total Liabilities	**74,727,000**	**72,650,000**	**79,436,000**
Stockholders' Equity			
Misc Stocks Options Warrants	-	-	-
Redeemable Preferred Stock	-	-	-
Preferred Stock	-	-	-
Common Stock	31,271,000	28,926,000	18,355,000
Retained Earnings	52,432,000	44,734,000	42,464,000
Treasury Stock	(46,296,000)	(38,546,000)	(31,072,000)
Capital Surplus	-	-	-
Other Stockholder Equity	(8,901,000)	(2,016,000)	-
Total Stockholder Equity	**28,506,000**	**33,098,000**	**29,747,000**
Net Tangible Assets	**$13,450,000**	**$21,994,000**	**$19,521,000**

IBM Cash Flow

Annual data, all numbers in thousands

PERIOD ENDING	31-Dec-06	31-Dec-05	31-Dec-04
Net Income	**9,492,000**	**7,934,000**	**8,430,000**
Operating Activities, Cash Flows Provided By or Used In			
Depreciation	4,983,000	5,188,000	4,915,000
Adjustments To Net Income	2,323,000	1,736,000	1,681,000
Changes In Accounts Receivables	(512,000)	2,219,000	2,613,000
Changes In Liabilities	(729,000)	(219,000)	(476,000)
Changes In Inventories	112,000	202,000	(291,000)
Changes In Other Operating Activities	(662,000)	(2,186,000)	(1,567,000)
Total Cash Flow From Operating Activities	**15,007,000**	**14,874,000**	**15,323,000**
Investing Activities, Cash Flows Provided By or Used In			
Capital Expenditures	(4,362,000)	(3,842,000)	(4,368,000)
Investments	(3,013,000)	(346,000)	112,000
Other Cashflows from Investing Activities	(4,174,000)	(235,000)	(1,090,000)
Total Cash Flows From Investing Activities	**(11,549,000)**	**(4,423,000)**	**(5,346,000)**
Financing Activities, Cash Flows Provided By or Used In			
Dividends Paid	(1,683,000)	(1,250,000)	(1,174,000)
Sale Purchase of Stock	(6,399,000)	(6,506,000)	(5,418,000)
Net Borrowings	(122,000)	609,000	(1,027,000)
Other Cash Flows from Financing Activities	-	-	-
Total Cash Flows From Financing Activities	**(8,204,000)**	**(7,147,000)**	**(7,619,000)**
Effect Of Exchange Rate Changes	201,000	(789,000)	405,000
Change In Cash and Cash Equivalents	**($4,545,000)**	**$2,515,000**	**$2,763,000**