"Babeş-Bolyai" University

Faculty of Mathematics and Computer Science

Department of Computer Science

LÓRÁNT BÓDIS

FINANCIAL TIME SERIES FORECASTING USING ARTIFICIAL NEURAL NETWORKS

Master Thesis

Supervisor:

Prof. Dr. Dan Dumitrescu

2004

Abstract

Financial and capital markets (especially stock markets) are considered high return investment fields, which in the same time are dominated by uncertainty and volatility. Stock market prediction tries to reduce this uncertainty and consequently the risk. As stock markets are influenced by many economical, political and even psychological factors, it is very difficult to forecast the movement of future values. Since classical statistical methods (primarily technical and fundamental analysis) are unable to deal with the non-linearity in the dataset, thus it became necessary the utilization of more advanced forecasting procedures. Financial prediction is a research active area and neural networks have been proposed as one of the most promising methods for such predictions.

Artificial Neural Networks (ANNs) mimics, simulates the learning capability of the human brain. NNs are able to find accurate solutions in a complex, noisy environment or even to deal efficiently with partial information. In the last decade the ANNs have been widely used for predicting financial markets, because they are capable to detect and reproduce linear and nonlinear relationships among a set of variables. Furthermore they have a potential of learning the underlying mechanics of stock markets, i.e. to capture the complex dynamics and non-linearity of the stock market time series.

In this paper, study we will get acquainted with some financial time series analysis concepts and theories linked to stock markets, as well as with the neural networks based systems and hybrid techniques that were used to solve several forecasting problems concerning the capital, financial and stock markets. Putting the foregoing experimental results to use, we will develop, implement a multilayer feedforward neural network based financial time series forecasting system. Thus, this system will be used to predict the future index values of major US and European stock exchanges and the evolution of interest rates as well as the future stock price of some US mammoth companies (primarily from IT branch).

Keywords: time series forecasting, prediction, technical analysis, neural networks, backpropagation, hybrid systems, capital markets, stock markets, interest rates.

Table of Contents

1 Introduction

The Artificial Intelligence (henceforth AI) is one of the most important and earliest research field of computer science. The classical and neo-classical methods used by scientist with the time became unpractical. The need for developing more efficient methods guided the researches in other directions, aiming for new concepts. Two prominent fields arose: connectionism – which involves neural networking and parallel processing – and evolutionary computing, respectively. Figure 1.1 shows the different AI technologies developed through the time. It can be seen that neural networks (or NNs) are intelligent technologies, it is used by several other methods and it's a relative new field.

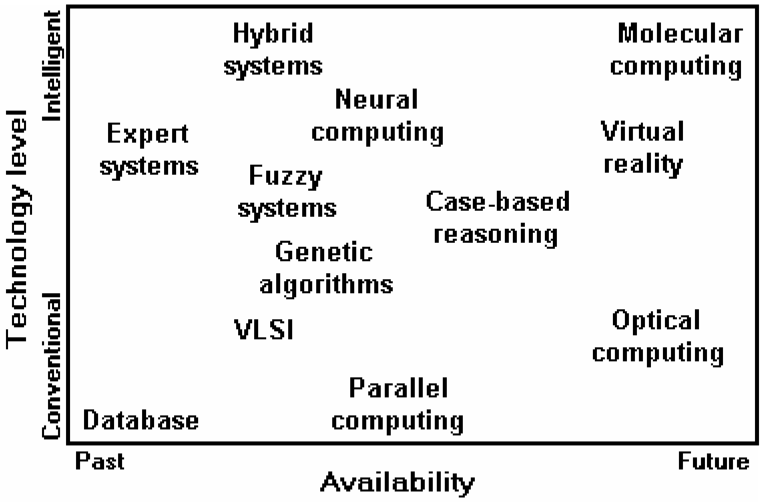

Figure 1.1. Place of neural networks between AI technologies.

Most symbolic AI systems are very static, which means that most of them can usually only solve one given specific problem. NNs were created to combat these problems. They are basically algorithms based on natural biological evolution, and they are more able to adapt to a wide range of problems. As their structure is general, they are very flexible, and they can applied for several different problems, making only very few modifications.

Artificial Neural Networks (ANNs) are modern computational entities, which have gain considerable importance in the last decades. They are widely used for various problems: pattern classification, clustering/categorization, function approximation, prediction (forecasting), optimization, retrieval by content and control. ANNs are formed by interconnected artificial neurons (elementary information processing units, called perceptrones) which are simulating the behavior of biological neurons. The neural network is based on the operational principle of the human brain, being able to learn and to generalize, in a highly efficient and fast way. This is because the architecture of neural networks is completely different from the architecture of von Neumann computers, implementing a connectionist model.

The field of artificial neural networks is an interdisciplinary area of research. A thorough study of artificial neural networks requires knowledge about neurophysiology, cognitive science/psychology, physics (statistical mechanics), control theory, computer science, artificial intelligence, statistics/mathematics, pattern recognition, computer vision, parallel processing, and hardware (digital/analog/VLSI/optical) [16].

For solving hard, complex optimization problems sometimes we need to combine different methods. Figure 1.2 depicts the relation between the most important AI technologies, pointing out in the same way the possible fields of application. Neural networks are used for learning and curve fitting, fuzzy logic is dealing with imprecision and uncertainty and genetic algorithms are used for search and optimization.

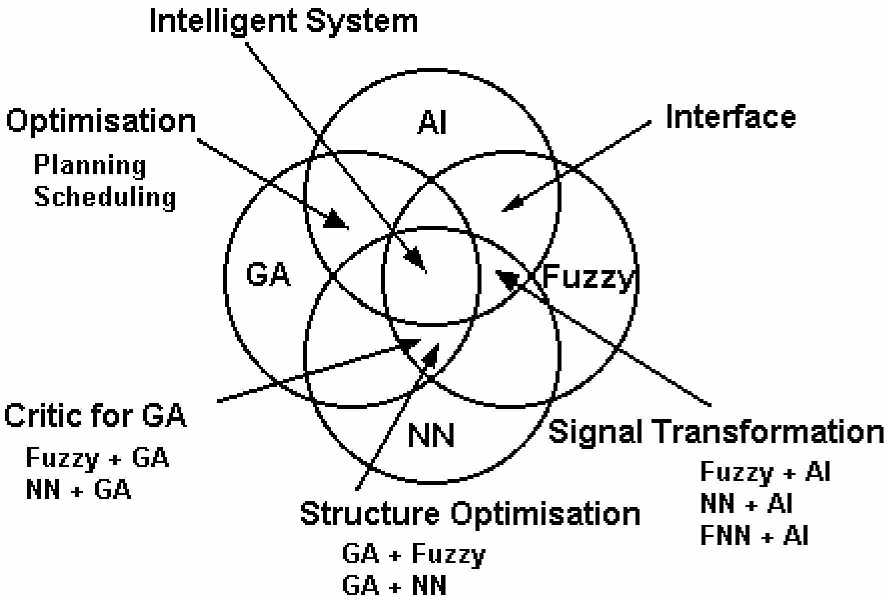

Figure 1.2. Place of Neural Networks between AI technologies.

In this thesis the neural networks are applied for stock market prediction, using the newest research results. In the last decades with the development of computer hardware and programming technologies there was an enormous interest toward this research area, because this problem requires a huge amount of computation time and playing on the stock market has proved to be a highly profitable investment. NNs have been very successful in a number of signal processing applications. Financial forecasting is an example of a signal processing problem, which is challenging due to small sample sizes, high noise, non-stationarity and non-linearity. Several journals (*International Journal of Forecasting, Decision Support Systems, Nonlinear Dynamics and Economics, Journal of Econometrics, International Journal of Theoretical and Applied Finance, Information Sciences, Advanced Complex Systems, Neurocomputing, Neural Networks, Applied Artificial Intelligence* etc.) has their topics the above mentioned two research domains, yearly hundreds of articles, reports and dissertations are published, as well as conferences (*AFIR, EANN, PASE, ICONIP* etc.) are organized.

The paper is organized as follows. Chapter 2 provides a presentation of the terminology concerning financial time series prediction. We will overview the economical backgrounds of forecasting, the indicators used for technical and fundamental analysis. What is more in this chapter are summarized the error and performance metrics as well as data preprocessing techniques, preparing in this way the transition to the next chapter.

Section 3 gives a comprehensive review of the existing financial, economical and capital markets applications, where the forecasting tasks were solved either by pure NN systems or by hybrid models. The NNs were utilized for a wide range of prediction problems, ranging from the prediction of stock markets, exchange rates to macroeconomic forecasting and portfolio management.

The implementation of the application and the description of methodologies are detailed in chapter 4. In order to test the performance of the NN based system, several experiment are conducted. Several demonstrations are carried out by using real-life historical data of various financial time series: stock markets, companies stocks, bond ratings. The section focuses on the presentation and interpretation of experimental results. The study ends with concluding remarks.

2 Financial Time Series

Financial prediction is one of the most exciting questions between economists. In the last decade financial time series forecasting has gained a considerable importance. This is because with the evolution of the computer hardware and development of more efficient programming technologies it is possible to make far more accurate predictions than in the previous decades.

Nowadays, prediction is widely used in different areas of financial economics and capital markets. In this chapter we will present the fields of capital markets where prediction has been made. Generally speaking we are interested in financial time series forecasting, which includes several areas of financial markets. As most of the forecasting attempts have been done for stock market indexes, we will focus on this area. Some traditional forecasting methods are detailed, presenting in the same time the technical indicators. Another important topic is the predictability of capital markets. Data preprocessing methods and error respectively performance measurements are also important part of this chapter, making in this way the transition to the next chapter.

2.1 Financial and Capital Markets

The capital market is the place where the money is traded. It is the most important component of the global economy, greatly influencing the industrial, the services and the commercial sectors of the macroeconomy. Financial and capital markets are considered high return investment fields, which in the same time are dominated by uncertainty and volatility. Thus, it has become essential to try to reduce the uncertainty and consequently the risk by forecasting the future evolution of financial time series.

In the followings we will describe several applications of forecasting in different areas of capital markets.

2.1.1 Market Forecasting

Market forecasting involves projecting such things like stock market indexes, companies' stock prices, treasury bill rates, exchange rates and net asset value of mutual funds. Stock market forecasting is one of the most important and most popular research fields within financial forecasting. Based on historical values one tries to approximate the function that generates the values of the stock market.

2.1.2 Trading Rules

Another important question within capital markets is the trading rules, which optimize buy/sell timing decisions. Timing is very important on capital markets. To illustrate this importance lets have the following example. If one dollar were invested in 1926 in 1-month U.S. Treasury bills, it would have grown to $14 by December 1996. If that dollar had been invested in the S&P 500, it would have grown to $1,370 during that period. If the dollar had been invested with monthly switching to either Treasury bills or the S&P 500, whichever asset would perform the best during that month, it would have grown to over $2 billion dollars during that period [34].

Applications include the development of technical trading rules, market-timing trading rules and buy/sell timing detection.

2.1.3 Option Pricing

Option pricing is also a significant application area of capital markets. The famous Black-Scholes option pricing model is used to compare the performance of newly developed methods. Estimation of volatility is also important, because in this way can be determined the risk of the investment.

2.1.4 Bond Ratings

Bond ratings are subjective opinions on the ability to service interest and debt by economic entities such as industrial and financial companies, municipals, and public utilities. They are published by major bond rating agencies, like Moody's and Standard & Poor's, who guard their exact determinants. Several attempts have been made to model these bond ratings, using methods such as linear regression and multiple discriminant analysis [34].

2.1.5 Portfolio Selection and Management

Portfolio construction is the part of the investment process that involves the determination of which assets to invest in and the proportion of funds to invest in each of the assets. At a minimum, effective portfolio optimization involves simultaneously maximizing the portfolio return and minimizing the portfolio risk [34].

2.2 Data

In every financial time series forecasting the future values are estimated based on the past values. Thus, the samples or observations are important elements of a forecasting system.

2.2.1 Types of Data

In order to get information about the market first we have to analyze data about the market. These data are classified into three main categories based on [17]: technical data, fundamental data and derived data.

2.2.1.1 Technical Data

Technical data represents all data that are referred to the predictor (the time series for which are going to predict the future values) only. For example in the case of the stock market, technical data include: price of the stocks at open and close, lowest and highest value of the stocks during a trading day and the volume of the traded shares.

2.2.1.2 Fundamental Data

Fundamental data: all data related to the intrinsic value of the company and to the macroeconomy: inflation, interest rates, trade balance, indexes of industries (e.g. heavy industry), prices of related commodities (e.g. oil, metals, currencies), net profit margin of a firm, prognoses of future profits of a firm.

2.2.1.3 Derived Data

These data can be originated by the combination and transformation of the technical and/or fundamental data. Derived data are used to calculate for example the return (usually one-step return, see section 2.2.3.4) and volatility. The volatility describes the variability of a stock and is used as a way to measure the risk of an investment.

2.2.2 Data Collection

Historical data representing the different stock markets index values and different company's share price, are available publicly and can be freely downloaded from Yahoo Finance web site [41]. For further links and a more comprehensive description of definitions used in the financial field, please consult [13]. [39] offers different types of time series data:

laser generated data, physiological data, currency exchange rate, computer generated series, astrophysical data, J. S. Bach's last (unfinished) fugue.

There are also various types of financial information sources on the Internet. The Wall Street Journal (www.wsj.com) and Financial Times (www.ft.com) maintain excellent electronic versions of their daily issues. Reuters (www.investools.com), Dow Jones (www.asianupdate.com), and Bloomberg (www.bloomberg.com) provide real-time news and quotations of stocks, bonds and currencies [40]. Excellent and vast amount of economic, financial data are available freely on [11], which is a database of over 2900 U.S. economic time series, divided in separate categories.

2.2.3 Data Preprocessing

Before data is used by an algorithm, it must go throw several transformations in order to prepare the raw data. The success of an algorithm greatly depends on the quality of input data. As different methods can handle only different samples, it is recommended to exploit certain data features with the purpose of finding out which preprocessing transformation works best.

2.2.3.1 Visual Inspection

Visualization and illustration (plot and histogram) of the dataset has an outmost importance. This makes possible to identify trends, missing values, noise and outliers. Figure 2.1 illustrates the close values of NASDAQ Composite index for 3000 days.

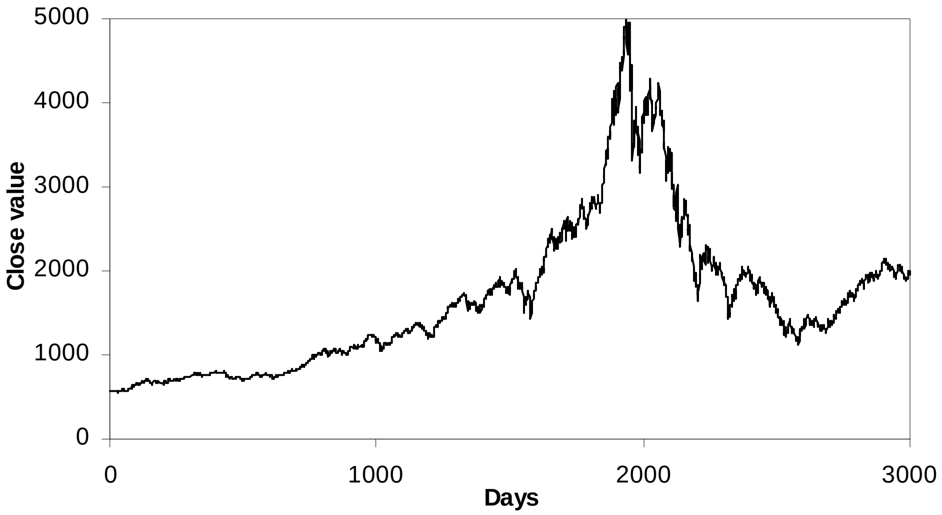

Figure 2.1. Daily NASDAQ Composite index from July 13, 1992 to June 4, 2004.

2.2.3.2 Normalization

Due to the chaotic nature of the data, the values of a time series can vary between wide ranges within a very short period of time, depending on the market volatility and company's performance. This can cause a great difficulty to neural networks, which can get disturbed by the large fluctuations in the price. Furthermore, the activation function used by the NN is bounded, causing in this way, inconsistencies in both the training and prediction phases. To avoid this pitfall, all the data in the input series are normalized to values between 0 and 1 (or –1 and 1, depending on the used transfer function).

There are various different types of normalization methods depending on between what values want to be the input data. It is common to use the following:

$$y = \frac{x - \bar{x}}{\sigma}$$

where $\bar{x}$ is the mean of x raw dataset, σ is the standard deviation both defined in the followings:

$$\bar{x} = \frac{1}{N}\sum_{t=1}^{N} x_t \text{ and } \sigma = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(x_t - \bar{x})^2}$$

2.2.3.3 Scaling

The following transformation scales the input *x* raw dataset to $y \in [-1,1]$:

$$y = \frac{2x - (\max + \min)}{\max - \min}$$

where max/min is the maximum/minimum value of *x* and *N* is the size of the dataset.

It is also possible to scale the *x* samples data to $y \in [0,1]$ by using the expression:

$$y = \frac{x - \min}{\max - \min}$$

2.2.3.4 Detrending

Detrending removes the growth of a series. In case of stocks, indexes, and currencies the difference of subsequent (e.g. daily) returns are converted into logarithmic values. For volume, dividing it by last *k* quotes average, e.g. yearly, can scale it down.

By replacing time series with the differences between subsequent values a linear trend is removed: $y = x_t - x_{t-1}$. Similarly, seasonalities can be eliminated by computing the differences between corresponding sequence elements: $y = x_t - x_{t-s}$, where s is the time interval that show similar patterns.

Another possibility for smoothing is to calculate the returns. Usually the **one-step return** (R_t) is used, which can be percentage or logarithmic return. Taking the logarithm of the price removes most of the market trend from the return calculation. The one-step logarithmic return: $y = \log(x_t) - \log(x_{t-1})$. The one-step percentage return [30] is calculated as follows:

$$y = R_t = \frac{x_t - x_{t-1}}{x_{t-1}}.$$

The subsequent differences and one-step returns of the above-introduced NASDAQ Composite index (see Figure 2.1) samples can be seen on Figure 2.2 and Figure 2.3.

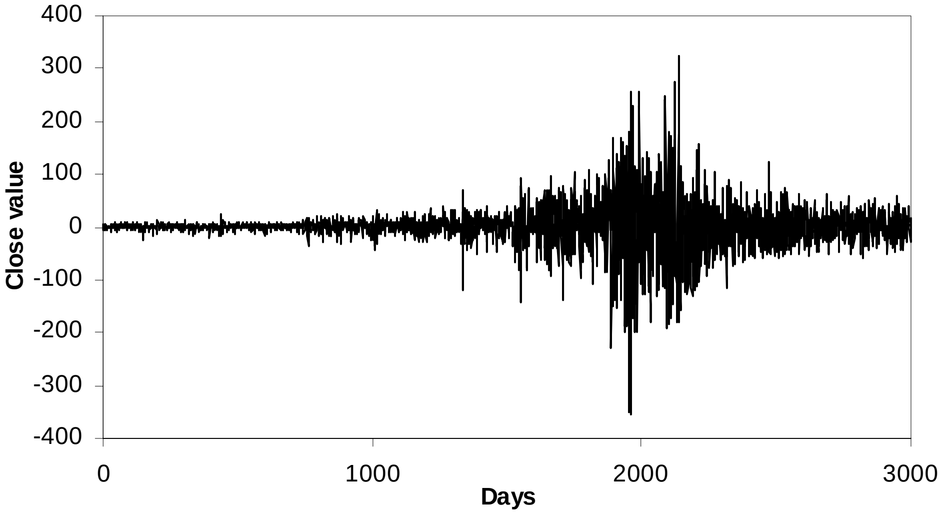

Figure 2.2. Differences between subsequent NASDAQ Composite index values.

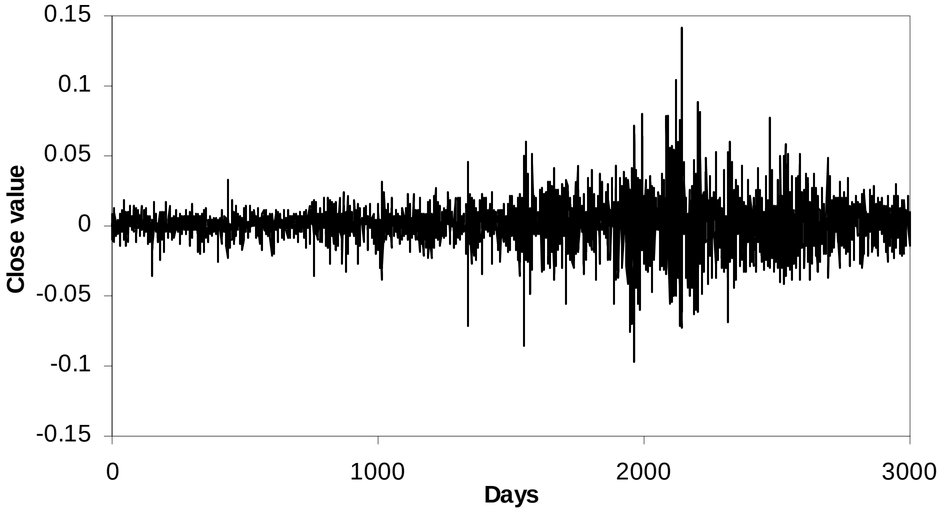

Figure 2.3. One-step percentage return applied for the NASDAQ time series.

The differences and one-step returns can be logarithmic scaled in order to remove the outliers. For this purpose it is possible to use the following formula [3]:

$$y = \frac{x \cdot \log(|x|)}{|x|},$$

where *x* takes the old *y* value. The logarithmic scaled differences and one-step returns of the NASDAQ time series are illustrated in Figure 2.4 and Figure 2.5, respectively.

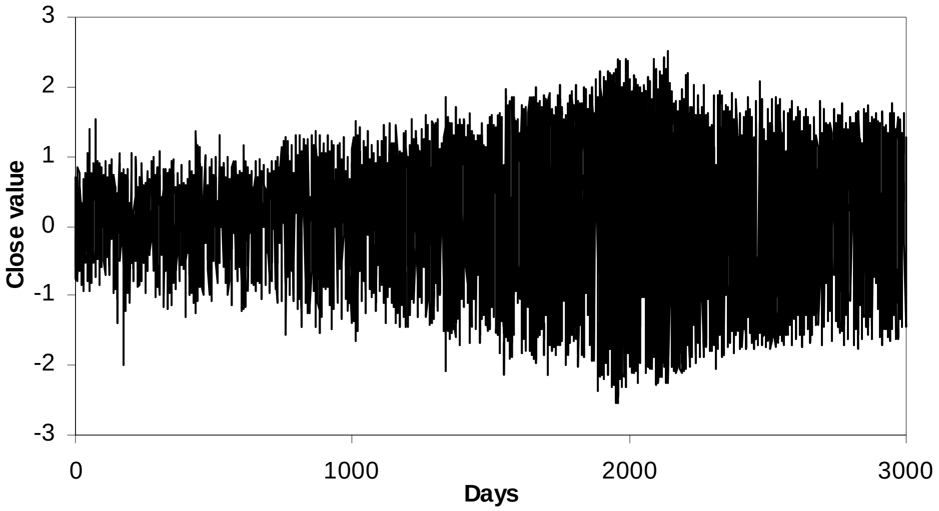

Figure 2.4. Logarithmic scaled differences of the NASDAQ Composite index values.

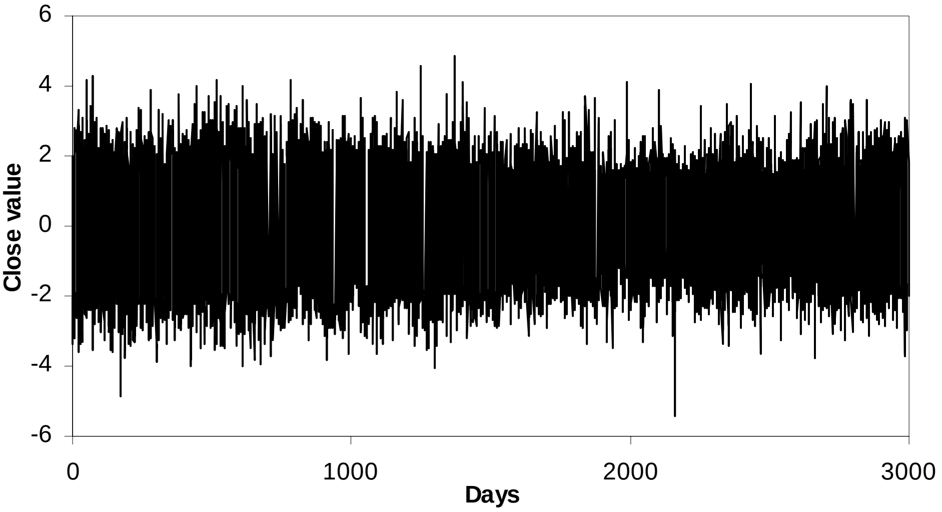

Figure 2.5. Logarithmic scaled one-step returns of the NASDAQ samples.

It is common to combine all the above-described transformations in order to normalize the samples, detrend, remove noise and to decrease the negative effect of the outliers.

2.2.4 Indicators

The indicators are used to describe more accurately the financial time series, by minimizing the trends, removing the outliers and maximizing the informational content of observations. Several consecutive samples of time series are combined, calculating different types of average values. The indicators can describe only one time series using in this way univariate analysis, but in the multivariate analysis several different time series are used.

Technical and fundamental indicators are used to describe the processes and environment which are having influence on the time series which are going to predict (that is why it is also called as the predictor time series). Researchers involved in financial time series prediction are using different types and different number of indicators, depending on the nature of the predictor (stock index, stock price, exchange rate, etc.), on the time rate of the used dataset (daily, weekly, monthly observations) and on the future time unit for which the prediction is made (one day ahead, one week ahead, one month ahead).

One of the most overwhelming list of indicators is found in [37], where a totally of 31 different financial and economic variables are used. Another comprehensive list of financial indicators is found in [7], containing as much as 32 various indicators.

The indicators are generating “buy” and “sell” signals, in order to help investors to make decisions. If there exist a lot of investors who believe in these signals, the signals become true and good [28]. In the following we will explain some indicators used for technical analysis.

2.2.4.1 Moving Averages

A moving average is an indicator that shows the average price value over a specified period of time. The most popular method to interpret a moving average is to compare the relationship between a moving average of a price with the price itself.

A buy signal is generated when the price rises above its moving average and a sell signal is generated when the price falls below its moving average. It is also generated a buy signal when the short term moving average is higher than the long term moving average and vice versa [28].

2.2.4.2 Moving Average Convergence Divergence Indicator

Moving Average Convergence Divergence (MACD) indicator is a trend-following momentum indicator that compares two moving averages (a "slow" and a "fast" one) of price history. Usually, the "fast" line is the difference between a 26-day and a 12-day exponential moving average, while the "slow" line is a 9-day exponential moving average of the fast line. A sell signal occurs when the "fast" line falls below the "slow" line and a buy signal is reported when the “fast” line rises above the "slow" line [28].

2.2.4.3 Relative Strength Indicator

Relative Strength Indicator (RSI) is an oscillator between 0 and 100. Buy signals are generated as soon as the RSI-value is below 20 and rises above 20 again, and sell signals are generated as soon as the RSI value is above 80 and drops below 80 again. RSI is calculated as follows:

$$RSI = 100 - \frac{100}{1 + \frac{\sum_{N} positive\_changes}{\sum_{N} negative\_changes}},$$

where *N* is the number of days. Normally the RSI value is computed for *N*=9 days or *N*=14 days [42].

2.2.4.4 Momentum

$$M = CCP - OCP,$$

where *CCP* is the current closing price and *OCP* is the old closing price for a predetermined period (usually 5 days).

2.2.4.5 Stochastic Oscillator

The stochastic oscillator is a momentum indicator, which measures the relationship between close prices and the previous high and low values for a specified number of days. The stochastic oscillator is defined by a fast and slow line, denoted with %K and %D, respectively. %K is defined as follows:

$$\%K = \frac{CCP - L9}{H9 - L9} \cdot 100,$$

where *L*9 is the lowest low value of the past 9 days and *H*9 is the highest high value of the past 9 days. In some situations it can be also applied for 14 days instead of 9. %D is also referred as the **moving average of stochastics**, given by the following formula:

$$\%D = \frac{H3}{L3} \cdot 100,$$

where *H*3 is the three day sum of (*CCP*-*L*9) and *L*3 is the three day sum of (*H*9-*L*9) [42].

2.2.4.6 Price Rate of Change Indicator

The Price Rate of Change (PROC) momentum or just simply Rate of Change (ROC) indicator is the ratio of current price and of an old closing price for a predetermined period (normally 5, 10 or 12 days for short-term indication of the price momentum and 26 days for mid-term).

$$PROC = \frac{CCP}{OCP} \cdot 100,$$

where *CCP* is the current closing price and *OCP* is the old closing price. Thus, the PROC indicator is an oscillator around 100, which generates a sell signal if approximates 100 from above and a buy signal when approximates from below.

There are far more indicators, but we are relying only on these in this project. For a more comprehensive list of indicators please see chapter 1.3.1 of the [28] diploma work.

2.3 Nonlinear Analysis. Predictability

Many theoretical researchers are concerned about the predictability of the financial time series events, primarily about the property of the dynamic changes of the stock markets. It is important to know if a time series is random or not, because this property will say if it is worth to make predictions or not.

2.3.1 Forecasting Hypotheses

There are two important hypotheses, which are characterizing the process of forecasting: the random walk hypothesis and the efficient market hypothesis. The **Random Walk Hypothesis** (RWH) states that the market prices wander in a purely random and unpredictable way. The **Efficient Market Hypothesis** (EMH) states that the markets fully reflect all of the available information and prices are adjusted fully and immediately once new information become available. In the actual market, some people do react to information immediately after they have received the information while other people wait for the confirmation of information. In the followings we will test the efficient market hypothesis.

2.3.2 Nonlinear Analysis

Hurst found that most natural phenomena follow a **biased random walk**, which is a trend with noise. The **Hurst Exponent** *H* is a measure of the bias in fractional Brownian motion. The **rescaled range analysis** (**R/S analysis**) is able to distinguish a random series from a fractal series, irrespective of the distribution of the underlying series (Gaussian or non-Gaussian). *R* captures the maximum and minimum cumulative deviations of the observations x_t of the time series from its mean $\bar{x}$, and it is a function of time:

$$R_n = \max_{1 \le t \le N}(x_{t,N}) - \min_{1 \le t \le N}(x_{t,N})$$

where $x_{t,N}$ is the cumulative deviation over *N* periods, given by the following formula:

$$x_{t,N} = \sum_{t=1}^{N}(x_t - \bar{x}) .$$

The *R/S* ratio of *R* and the standard deviation *S* of the original time series can be estimated by the following empirical law: $R/S = N^H$ when observed for various *N* values. For some values of *N*, the Hurst exponent can be calculated by: $H = \log(R/S)/\log(N)$, and the estimation of *H* can be found by calculating the slope of the log/log graph of *R/S* against *N* using linear regression.

H describes the probability that two consecutive events are likely to occur. A value of *H* different from 0.5 denotes the observations that are not independent. When $0 < H < 0.5$, the system is an antipersistent or ergodic series with random walks and high volatility. For the case $0.5 < H < 1$, *H* describes a persistent or trend reinforcing series which is characterized by long memory effects [43].

2.3.3 Randomness and Predictability

In [17] is implemented and used two empirical methods in order to test the randomness of the input data (time series): Run test and Brock, Dechert, Scheinkman (BDS) test. More advanced BDS algorithms are described in [18].

The predictability of stock market returns is examined in [25]. They are using metric entropy, which is capable of detecting nonlinear dependence within the return series. The suggested non-parametric and recursive unconditional mean model is compared with other more traditional models (linear, neural network system) used for the detection of dependence within the series.

2.4 Traditional Forecasting

Classical financial forecasting is used by economists for decades. This involves statistical techniques, the so-called technical and fundamental analysis; both of these are considered security analyses. Fundamental methods are based on economic data – retail sales, gold price, industrial production index, foreign currency exchange rate – while technical methods are trying to predict the future using only historical prices, i.e. the delayed time series data.

2.4.1 Theories

According to the **Firm Foundation Theory** the market is defined from the reaction of the investors, which is triggered by information that is related with the intrinsic value of firms. On the other hand, according to the **Castles in the Air Theory** the investors are influenced by information that is related to other investors' behavior [17].

2.4.2 Prediction Methods

Many models and techniques are used for prediction, ranging from traditional statistical analysis to highly formalized complex hybrid AI based learning system. These methods can be divided into 4 major groups (the machine learning based methods are presented in section 3.1) [17].

2.4.2.1 Technical Analysis

The technical analysis method is using technical data and price charts to detect trends, trading rules to predict future stock movements. Popular approach to predict the market, the extraction of trading rules are subjective and the practitioners' believe that the market is only "10 percent logical and 90 percent psychological". It is possible to predict on a daily basis.

In technical analysis, the main idea is that changes in stock prices can be predicted based on recent trends in stock price changes, i.e. by using only the delayed time series or historical data. In this approach it is assumed that all economic information is already contained in the current stock price.

[28] presents several different models based on technical data in order to predict the one-day ahead buy or sell decision with the aim of maximizing the gain. It is stated that technical analysis is more like voodoo, but the existing indicators are not able to outperform the so-called **Buy and Hold** strategy. On the other hand, forecasts could become true if there are many investors who believe in the indicators – as a kind of a self-fulfilling prophecy. In any case, it is still unknown whether short-term price movements in the stock market are predictable or not.

2.4.2.2 Fundamental Analysis

By using fundamental data the real value of the asset may be computed. The analysts apply a simple rule: if the intrinsic value of the asset is higher than the value it holds in the market, invest in it. If not, consider it a bad investment and avoid it. They believe that the market is defined "90 percent by logical and 10 percent by physiological factors". As the used fundamental data does not change on a daily basis, it is possible only to predict on a long-term basis.

The aim of the fundamental analysis is to predict the evolution of certain stocks on the basis of the company's financial situation. Any change on the global market can affect the performance of the company. Therefore it is important to study the overall economic situation and industrial section, and not only the company's situation. Some important factors concerning the global economic situation: economic growth, interest rate, inflation, exchange rates, unemployment rate, taxes. A few impact factors connected to the industrial sector: oil, energy prices, raw materials prices. Finally, some factors that relate the company's internal situation: dividends, cash flow, management quality.

2.4.2.3 Traditional Time Series Prediction

The traditional time series prediction methods analyze historic data and attempt to approximate future values of a time series as a linear combination of these historic data. In econometrics they are known as the simple and multivariate regressions. The results obtained with regression models were better for long-term prediction (annually) and outperformed the Buy and Hold strategy only with 1-2% depending on the transaction cost. Some benefits and drawbacks: widely accepted by economists, low computation time, difficult to capture non-linear patterns, their performance depends on few parameter settings.

2.5 Error and Performance Metrics

Error and performance metrics are involved to calculate the difference between the predicted and target value, and respectively to measure the performance of the forecasting system.

2.5.1 Error Metrics

Error measurements are widely used for several approximation problems. Some of them are very popular, while others are specialized. In the followings we are going to give a brief overview of the error measures, to which we will refer in the chapters.

The most popular error metrics is the Mean Square Error (*MSE*) and Mean Absolute Error (*MAE*) [25], both defined as follows:

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(x_t - \hat{x}_t)^2 \text{ and } MAE = \frac{1}{N}\sum_{t=1}^{N}\left|x_t - \hat{x}_t\right|,$$

where x_t and $\hat{x}_t$ represent the actual and predicted values, and *N* the size of the dataset.

The Root Mean Square Error (*RMSE*) is derived from the *MSE* and is given in the following formula [25], [2]:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(x_t - \hat{x}_t)^2},$$

where x_t, $\hat{x}_t$ and *N* are defined as above.

The Normalized Mean Square Error (*NMSE*) is also commonly used and is given by the following expression [42]:

$$NMSE = \frac{\sum_{t=1}^{N}(x_t - \hat{x}_t)^2}{\sum_{t=1}^{N}(x_t - \bar{x})^2},$$

where x_t, $\hat{x}_t$ and *N* are defined as above and $\bar{x}$ is the mean of x_t.

Other error measurements are the *MRPE* (Mean Relative Percentage Error) [9] and the *MAPE* (Mean Absolute Percentage Error) [25] given by the following expressions:

$$MRPE = \frac{1}{N}\sum_{t=1}^{N}\frac{\left|x_t - \hat{x}_t\right|}{x_t} \text{ and } MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{x_t - \hat{x}_t}{x_t}\right|.$$

POCID (Prediction of Change in Direction) [31] shows the percent of the wrong direction estimations and it is given by the following expression:

$$POCID = \frac{1}{N}\sum_{t=1}^{N}D_t,$$

where

$$D_t = \begin{cases} 1 & if \quad (x(t) - x(t-1)) \cdot (\hat{x}(t) - \hat{x}(t-1)) < 0 \\ 0 & otherwise \end{cases},$$

$\hat{x}(t)$ is the prediction of $x(t)$ and N is the size of the dataset. The error is illustrated in Figure 2.6.

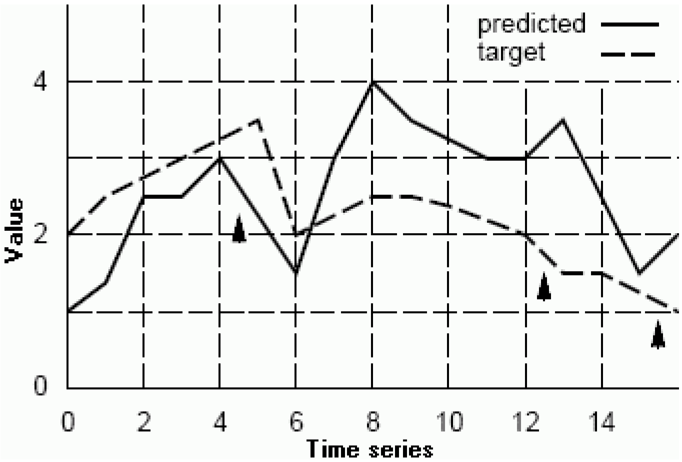

Figure 2.6. The POCID error occurs if the target and predicted value change in opposite directions. The errors are market with arrows.

2.5.2 Performance Metrics

Performance measurements can be considered as the inverse of the corresponding error metric. The error can be seen as a negative performance. Although, this is true, in the literature exist various specialized error and performance metrics. Some of them are well know, used for several computational problems, while others are specific to the financial community.

The inverse of the *POCID* error is the *SIGN* performance indicator, which shows the percent of the correct direction prediction. It is slightly different from the inverse of the POCID and is calculated as follows [5]:

$$SIGN = \frac{1}{N}\sum_{t=1}^{N}\left(HS(\Delta x_t \cdot \Delta\hat{x}_t) + 1 - HS(|\Delta x_t| + |\Delta\hat{x}_t|)\right),$$

where $\Delta x_t = x_t - x_{t-1}$, $\Delta\hat{x}_t = \hat{x}_t - x_{t-1}$ and *HS* is the modified Heaviside function:

$$HS(x) = \begin{cases} 1 & if \quad x > 0 \\ 0 & otherwise \end{cases}.$$

Another performance metric is the correlation coefficient (*CORR*). The values of *CORR* are between 0 and 1 and a higher value indicate a better performance. *CORR* is given by the next formula [25]:

$$CORR = \frac{\sum_{t=1}^{N}(x_t - \overline{x}_t)(\widehat{x}_t - \overline{\widehat{x}}_t)}{\sqrt{\sum_{t=1}^{N}(x_t - \overline{x}_t)^2 \sum_{t=1}^{N}(\widehat{x}_t - \overline{\widehat{x}}_t)^2}}$$

The *Theil coefficient* (inequality coefficient) is a metric that measures to what extent the results are better than those of a naïve prediction, comparing in the same time a model's performance with the Random Walk [17], [14]. The Theil coefficient is defined as follows:

$$Theil = \frac{\sqrt{\sum_{t=1}^{N}(x_t - \widehat{x}_t)^2}}{\sqrt{\sum_{t=1}^{N} x_t^{\ 2}}}$$

The Buy and Hold Return (R_{BH}) expresses the profit made when making an investment at the start of a time period and selling *n* time-steps into the future [30]:

$$R_{BH} = \frac{x_{t+n} - x_t}{x_t} \cdot 100$$

The Hit Rate (H_R) indicates how often the algorithm makes a correct prediction. The metric is given by the following formula [30]:

$$H_R = \frac{|\{t \mid R_t^k \widehat{R}_t^k > 0, t = \overline{1,N}\}|}{|\{t \mid R_t^k \widehat{R}_t^k \neq 0, t = \overline{1,N}\}|},$$

where the norm gives the number of elements in the series and $R_t^k / \widehat{R}_t^k$ is the actual/predicted *k*-step return at time *t* which is defined as follows:

$$R_t^k = \frac{x_t - x_{t-k}}{x_{t-k}}.$$

Other performance measurements are the Return on Investment and Realized Potential, which are used in [30] thesis.

3 Forecasting Using Neural Networks

Classical statistical methods are unable to efficiently handle the prediction of financial time series due to non-linearity, non-stationarity and high noise. Advanced intelligent techniques have been used in several financial trading systems to predict the future evolution of different capital markets, especially stock prices and stock exchange index values. The artificial neural network is a well-tested method for financial analysis on the stock market.

Neural networks have been successfully used in forecasting, because have been shown to be able to decode nonlinear time series data, which adequately describe the characteristics of the stock markets. Examples using neural networks in stock market applications include forecasting the value of a stock index, recognition of patterns in trading charts, rating of corporate bonds, estimation of the market price of options, and the indication of trading signals of selling and buying.

In this chapter we will present several different capital market and financial forecasting applications using neural networks and hybrid techniques.

3.1 Machine Learning Methods

Machine learning includes inductive learning methods, classifier systems (like nearest neighbor), neural networks etc. All these methods use a set of samples to generate an approximation of the underling function that generated the data. Unlike traditional forecasting systems, machine learning techniques are able to trace both linear and nonlinear patterns. As drawbacks can be stated that these methods require more computation time and their performance depends on a large number of parameters.

3.1.1 Neural Networks

The neural network belongs to the so-called “black-box” methods as they assume little about the structure of the economy. NNs are trained to approximate the thinking and behavior of some stock market traders. Different indicators are used as the inputs to a neural network and the index of stock is used to supervise the training process.

Mostly supervised NNs are used for financial forecasting, because the inputs (past values and several technical or fundamental indicators) and output (target value) are known and the objective is to discover a relationship between the two, i.e. to approximate the function that generates the values of the stock market. A basic knowledge of neural networks is highly recommended, as throughout this document we will refer to various terminology, components and methods related to NNs. A comprehensive tutorial on NNs is given in [16]; for a more detailed description please see [10]. Several practical applications and essays can be found on the Internet [12.a]. The MATLAB programming environment gives the possibility to implement in an elegant way NN based systems [26].

The most popular supervised NN involved in financial prediction is the Multilayer Perceptron, which, based on its architecture belongs to the class of feedforward networks (FFNNs). In a feedforward network the flow of information through the network is from the input to the output. Of course other types of feedforward (radial basis function net) and feedback (competitive networks, Kohonen's self organized maps, Hopfield networks and ART models) networks can be utilized as well. Based on the type of learning rules, the error-correction networks are the most commonly adopted in the following researches.

In order to develop a successful financial prediction system it is recommended to follow some advises and to avoid major pitfalls and common errors [35], [46]. In [35] it is stated that although the neural networks can hide a lot of pitfalls, they are more efficient that technical analysis. The [46] study is an excellent "screenplay" of financial forecasting implementation, listing all the "ingredients" needed by this system. Step by step are presented the procedures and the different possibilities in solving the "sub-problems" that built up the whole financial prediction system.

3.1.2 Soft Computing

Neural networks together with fuzzy logic and genetic algorithms are the most commonly used components of the so-called **soft computing** (SC) [34], because these technologies are often linked together. Neural networks are used for learning and curve fitting, fuzzy logic is dealing with imprecision and uncertainty and genetic algorithms (GAs) are used for search and optimization.

Soft computing – inspired from physical sciences – can handle very efficiently the uncertainty, imprecision and noisy information. SC is applied for two major disciplines of financial and investment trading: fundamental and technical analysis.

Another good survey on soft computing is found in [38], where it is overviewed the literature in applying the different soft computing techniques to the investment and financial field. The research papers are classified by different criterions: style of SC (time series, pattern recognition and classification, optimization, hybrid), performance against Buy and Hold strategy (outperforms, or not), and implementation of real world applications (money management, trading costs).

3.2 Stock Market Prediction

Many attempts have been made to model and forecast financial markets, using all the computational tools available for studying time series and complex systems: linear autoregressive models, principal component analysis, artificial neural networks, genetic algorithms, and others. The first NN based market forecasting was implemented by White in 1988. He used FFNNs to decode previously undetected regularities in asset price movements, such as fluctuations of common stock prices.

3.2.1 Forecasting the KLSE Index

Yao et al. successfully applies the neural networks to forecast the indices of Kuala Lumpur Stock Exchange (KLSE) [42]. A more detailed version of this article was later published as [44], where the prediction is done for the Kuala Lumpur Composite Index (KLCI).

A feedforward neural network with backpropagation (BP) is used to capture the relationship between the stock prices of today and the future. The NN based forecasting consists of 5 steps:

1. The information that will be used as input and output of the network are collected;
2. Data is normalized and scaled in order to reduce the fluctuation and noise;
3. Model of the neural network is built;
4. Variations of the models, i.e., different models and configurations with different training, validation and testing data sets are experimented;
5. The best model measured is chosen for use in forecasting.

The daily data from Jan 3, 1984 to Oct 16, 1991 (1911 data) are used. It is possible to make predictions for this market, as the 0.88 Hurst Exponent denotes a long memory effect in time series. The data is segregated in time order. Usually one third of the collected data is used for testing and two third of the data for training. Two fifth of the testing data is used for validation. The data of the earlier period are used for training, the data of the later period are used for validation and the data o the latest period are used for testing. This procedure can have some **recency problems**, and in order to solve this problem the data are randomly chosen.

The input of the network are the following: I_{t-1}, I_t, MA_5, MA_{10}, MA_{50}, *RSI*, *M*, *%K*, *%D*, where I_t is the index of the *t*-th period and MA_j is the moving average of the *j*-th period and I_{t-1} is the delayed time series. The output is I_{t+1}. The other indicators were already defined in chapter 2.2.4.

The main consideration when building a suitable neural network for the financial application is to make a trade-off between convergence and generalization. It is important not to have too many nodes in the hidden layer because this may allow the neural network to learn by example only and not to generalize. They have adopted an iterative process to build up several models. After experiments the 5-3-2-1, 6-4-3-1 and the 5-4-1 have proved to be the most efficient, managing to reach a small NMSE (approx. 0.0322) and a high rate of correctness of gradients (85%) respectively. For the five input model the I_t, MA_5, MA_{10}, *RSI*, and *M* are used, and in the case of the six input model the I_{t-1} was added to the previous ones. The learning rate (0.005) and the momentum (0.1) of the NN were very small as well. Figure 3.1 illustrates the results obtained by using the methods and parameters described above.

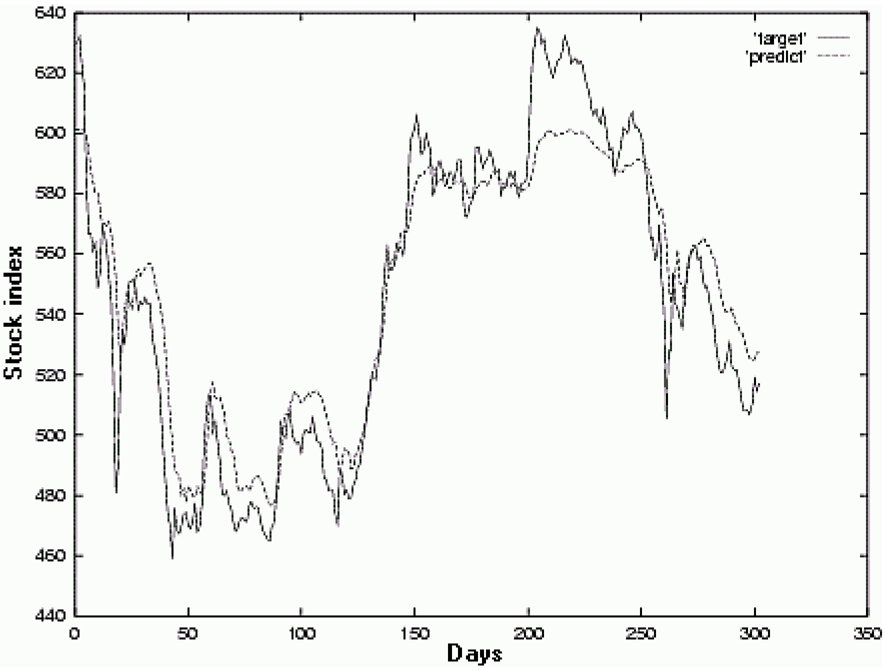

Figure 3.1. Daily stock price index prediction of KLSE from July 30, 1990 to Oct 16, 1991.

In [42] is also proposed two kinds of trading strategies. One uses the difference between predictions, another uses the difference between the predicted and the actual levels:

Strategy 1: $\text{if}\left(\hat{x}_{t+1} - \hat{x}_t\right) > 0$ then buy else sell

Strategy 2: $\text{if}\left(\hat{x}_{t+1} - x_t\right) > 0$ then buy else sell

The best return is obtained with the 6-4-3-1 model and the annual return rate is approximately 26%.

3.2.2 Forecasting Various Stock Market Indices and Stock Prices

[23] is an overview of the experiments done in the field of financial markets prediction using neural networks. It is summarized some experimental result that were tried with the "4Thought" software package, which was implemented using the feedforward multi-layer perceptron model.

The aim of [22] is to predict the Standard & Poor's 500 (S&P 500) stock exchange index value based on the past values and some technical indicators. It is reviewed the literature by categorizing the neural network based stock market prediction in two classes: prediction using past stock values only and prediction using past stock values and other fundamental indicators. We find out that several different neural network models have been used to solve this problem: dual module neural network, neural sequential associator, recurrent neural network, radial basis function network and modular neural network. Some statistical methods are presented for data analysis and preprocessing (ex. normalization) and then the author focuses on two case studies: bull run since Jan 1994 and crash of October 1986.

The [45] comparative study is a methodology analysis of the existing neural network applications in stock market predictions. It is stated that in the past 10 years the NN applications were mainly developed in the productions/operations and finance fields. The most frequent applications in the finance field are the forecasting stock prices, returns, and stock modeling, while the most popular methodology is the backpropagation algorithm. The design of an efficient NN topology it is still remains an open problem. In the case of the stock price prediction there were several tries with different models: 24-24-1, 10-10-10-1 or 40-1. The used algorithms for this problem besides the BP are ADALINE and MADALINE, and it is used several input indicators: current stock price, the absolute variation of the price in relation to previous day, direction of variation, direction of variation from two days previously etc.

In [6] has been investigated financial time series forecasting using a FFNN and daily trade data from the Shanghai Stock Exchange. To improve speed and convergence the researches used a conjugate gradient learning algorithm and multiple linear regression (MLR) for the weight initialization.

The stocks of different major US companies from the IT branch as well as the index values of some US stock exchanges (S&P 500, Dow Jones Industrials, NASDAQ 100) are predicted one-day ahead with the help of the MLP neural networks [5]. The NN had different topologies (8-2-1 for S&P 500, 13-2-1 for Dow Jones Industrials, 4-25-1 for NASDAQ), sigmoidal activation function, 0.05 fixed learning rate, 0.5 momentum. The NN architecture generation consists of training nets of different topologies (the input units are varying between 2 and 15, and the hidden layer units between 2 and 25) and observing their performance. The trained (using the training and validation dataset) NNs are tested on a so-called checking data. Only the NNs that had a small error (MSE is divided by the standard deviation of the time series) are used to predict the price changes. In the case of the companies' assets, almost every (and only about half for the stock exchanges) of these good nets have a sign prediction rate above 50%. The maximum sign prediction rate reached is 54% for the asset of SUN Microsystems, using a 9-7-1 NN topology, 500 samples for training, 300 for validation and 200 for checking (from a total of 2024 observations).

Multilayer perceptron (MLP) based FFNN networks and Generalized FeedForward (GFF) networks were tried to predict Istanbul Stock Exchange (ISE) index [9]. GFF networks are a generalization of the MLP networks where connections can jump over one or more layers. Various indicators were used as inputs: previous day's index value (close price), previous day's TL/USD exchange rate (average of buying and selling), previous day's simple overnight interest rate (weighted average). Other 5 dummy variables representing the working days of the week were used: value of 1 for the corresponding day of week and 0 in case of the other 4 variables (ex. if the current day is Monday then the first variable's value is 1). The dataset included a period of 417 days starting from July 2, 2001 to February 28, 2003, using 90% for testing. The results were evaluated by using the coefficient of determination (R^2) performance metric and the *MRPE* error measurement: the GFF with 1 hidden layer had the best performance.

Daily and weekly predictions are performed of the Swedish stock exchange (SXGE) utilizing Error Correction Neural Network (ECNN) with backpropagation learning algorithm [30]. For the NN implementation the Simulation Environment for Neural Networks (SENN) software package was used. The daily high, low, close values and traded volume served as internal inputs. External financial data were used as further inputs: S&P 500, SX 16, Nikkei 225, Dow Jones Stock Index, German DAX, Gold Price ($/oz), 3-month interest rate Sweden, 5-year interest rate Sweden, Swedish SEK/USD FX-rate, Swedish SEK/DEM FX-rate. As a result 56.8% of hit rate is achieved for 1-day forecast and 52.9% in case of the 1-week forecast.

3.3 Foreign Exchange Rate Forecasting

In the [43] paper Yao et.al. successfully applied the above-described (section 3.2.1) system with minor changes to exchange rate forecasting. This is possible because the evolution of exchange rates shares mainly the same properties as stock markets. The study shows that without the use of extensive market data or knowledge, useful prediction can be made and significant paper profit can be achieved with simple technical indicators.

The daily exchange rates of five major currencies reported to USD are studied between May 18, 1984 and July 7, 1995 (a totally of 2910 observations). From the daily rates the weekly closing prices are used as the prediction target of the experiment. Having the highest Hurst exponent of 0.55, the CHF/USD and DEM/USD exchange rates are representing a trend reinforcing time series, thus a prediction is possible. Different time delay (lagged) values and moving averages (5, 10, 20, 60 and 120) were used as inputs of the NN. With an NN model of 5-3-1, the AUD/USD exchange rate performs the lowest NMSE value at test, while the CHF/USD has the highest gradient value of 56%. The obtained result can be seen on Figure 3.2. A paper profit is used in order to simulate the real profit.

As a limitation of the implemented system can be seen the segregation of data in time order, because this may cause recency problems. In this way can be explained the fact that better testing results are obtained in the period near the end of the training sets.

The NN based model is compared with the Box-Jenkins methodology, or Autoregressive Integrated Moving Average (ARIMA) Model. This model provides a systematic procedure for the analysis of time series that was sufficiently general to handle virtually all empirically observed time series data patterns. From the point of view of economists, returns are more important than gradient. Thus, the best return using strategies for ARIMA models is only 6.94%, while for neural network models is 28.49%.

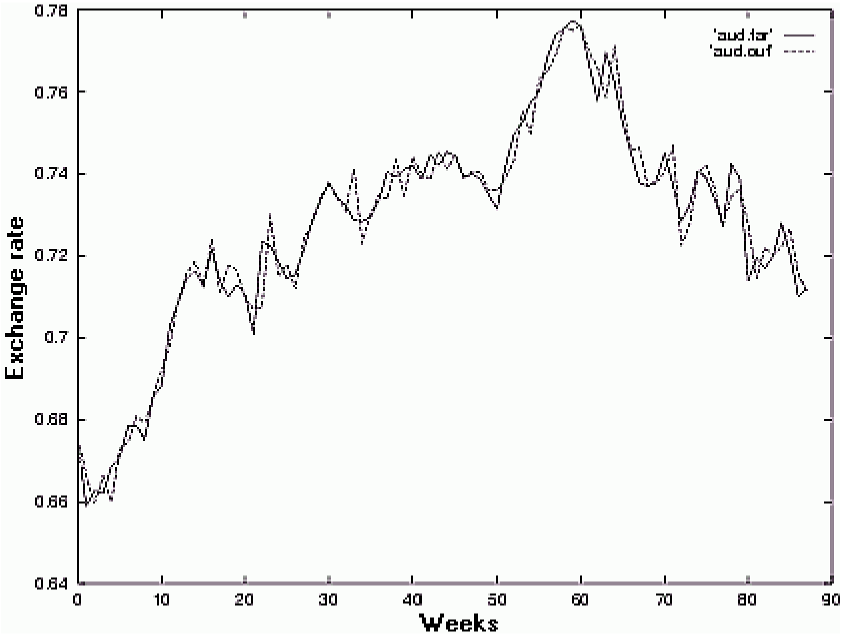

Figure 3.2. Weekly AUD/USD exchange rate prediction between Nov 5, 1993 and July 7, 1995.

In [2] it is proposed a data compression technique for input dimension reduction in order to solve in a more efficient way the one-step ahead prediction of the USD/GBP exchange rate. For this purpose an autoassociator MLP network has been applied to reduce input data dimension. The autoassociator MLP network is a three-layer network in which the number of input nodes is equal to the number of output nodes. The output vector is identical to the input vector so that the MLP network is trained to reproduce the input vector. Using a number of hidden nodes less than the number of input units, the aim is to compress the input vector without losing any information within it. The dimension reduction of the input vector is important because the speed and convergence of the NN greatly depends on the number of hidden units and the dimension of input vector. The best input vector had the following components: 4 days lag including the same day's price of USD/GBP, a set of three RWIs (Random Walk Indicator) for one, two and three look back periods and the differences between the same day's and last day's price of USD/JPY and GBP/DEM. The RWI is defined by the following formula:

$$RWI = \frac{\Delta p_t(N_\tau)}{\overline{\Delta p_t}\sqrt{N}},$$

where $\Delta p_t(N_\tau)$ is the price difference over the time interval N_τ, τ is the sample interval (in this case it's value is 1), $\overline{\Delta p_t}$ is the average differences of the series and N is the size of the dataset.

3.4 Portfolio Selection and Management

In [24] Lazo et al. presents the development of a hybrid genetic-neural system for construction and management of assets investment portfolio. The system has 3 modules: a genetic algorithm for the selection of the stocks that are going to form the portfolio, a neural network for the prediction of the returns on the assets in the portfolio, and a genetic algorithm for the determination of the optimal weights for each asset. The portfolio is managed by means of weekly updates over a period of 49 weeks.

The study is based on the series of returns for 137 Brazilian assets traded at the São Paulo Stock Exchange (BOVESPA) from July 1994 to December 1998. One part of the data (July 1994 to December 1997) was used to train the model, and the other part (January 1998 to December 1998) to test it, i.e., to manage the portfolio.

In the first step a set of 12 assets are chosen from the 137 stocks traded at the BOVESPA using GAs. The representation of the chromosome consists of 137 genes, where each gene identifies an asset, but only the first 12 genes of the chromosome are evaluated, which represent the assets with a higher return, a lower risk, and a low correlation with the other assets that form the portfolio.

The aim is to select those stocks that have the highest return, the lowest risk, and a low correlation with the remaining assets. Thus the fitness function is defined by the following expression:

$$\max \prod_{i=1}^{12} \prod_{\substack{j=1 \\ j \neq i}}^{12} R_i (1 - \sigma_i)(1 - \rho_{ij}),$$

where R_i is the return of asset *i*, σ_i is the risk of the asset *i* and ρ_{ij} is the correlation coefficient between asset *i* and asset *j*.

The used genetic operators were the crossover and mutation. The GA was run for several different parameter settings: the population size was 100, 500 and 3500; the crossover rate was set 0.5, 0.1, 0.25 and 0.55; and the mutation rate varied from 0.06 to 0.8. Two portfolios were built up, which were optimized for minimum risk and maximum return respectively. Both of them achieved a higher profit than the market portfolio.

In the next step the return was predicted using monthly, weekly and daily data to evaluate different types of neural network: Backpropagation Neural Nets, Bayesian Neural Nets, Hierarchical Neuro-Fuzzy Networks, and Backpropagation Neural Networks with Kalman Filters. This last model has proved to have the lowest prediction error rate, having 10 inputs (5 previous values of the asset returns series and 5 previous values of the stock market returns series), several neurons in a single hidden layer, and one output. The activation (transfer) function is hyperbolic tangent and several error measurements were used to evaluate the quality of the prediction: mean absolute deviation, mean square error, normalized mean square error. The results of the weekly return prediction can be seen on Figure 3.3.

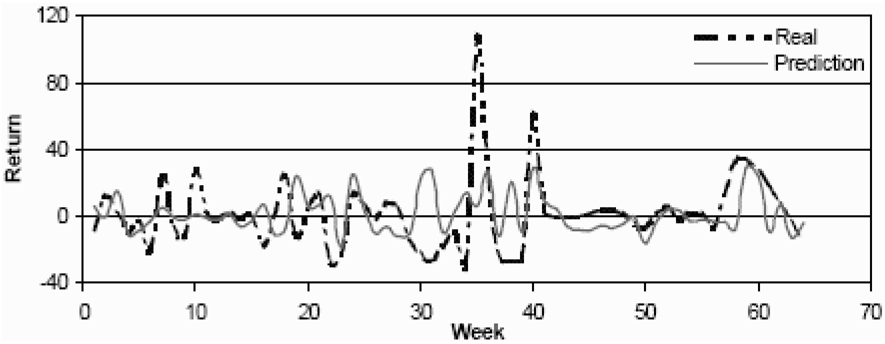

Figure 3.3. Weekly return prediction of the 12 asset portfolio.

The management of the constructed portfolio was done by GAs. Have to be decided the weight of each asset in the portfolio, i.e. what percentage of the fund should be invested in each of the portfolio's assets. The representation of the chromosome consists of 12 genes, where each gene identifies the weight of the asset. The aim is to minimize the portfolio's risk for a given return. Thus, the evaluation function is given by the following expression, which represents the standard deviation of the portfolio's return:

$$\min \sum_{i=1}^{12} w_i^2 \sigma_i^2 + \sum_{i=1}^{12} \sum_{\substack{j=1 \\ j \neq i}}^{12} w_i w_j \sigma_{ij} \ ,$$

where w_i / w_j is the weight of asset *i/j*, σ_i is the risk (standard deviation) of asset *i*, and covariance of asset *i* with asset *j*.

The basic crossover and mutation were used as genetic operators. The GA was tested with the following parameter values: population size is 40000, crossover rate 0.5 and 0.68, and the mutation rate was varying from 0.06 to 0.25. The portfolio was managed over a period of one year with weekly updates of the asset weights. Figure 3.4 presents the performance of the portfolio that was managed through the proposed hybrid system and compares it with the return provided by the market portfolio (BOVESPA Index). It can be noticed that the managed portfolio closely follows the market index, surpassing it in some cases.

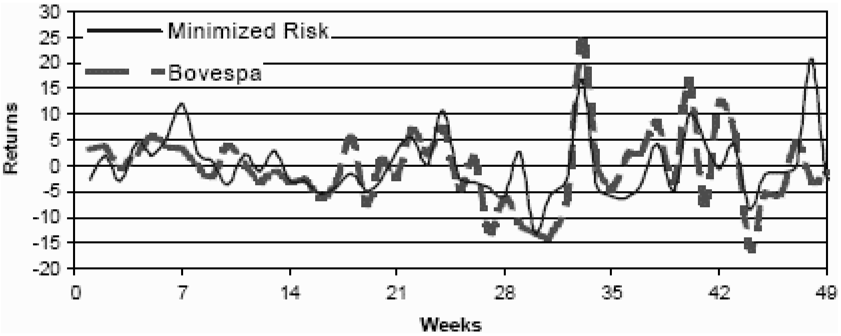

Figure 3.4. Performance of the portfolio managed through 49 weeks.

3.5 Multivariate Time Series Modeling

In financial forecasting it is very important to identify the correct indicators, those indicators, which are highly correlated with the predictor (financial time series that is going to be predicted). For this purpose can be used traditional statistical methods [7], data mining techniques [37] but also neural networks, as well [3].

3.5.1 Using NNs to Identify Indicators and to Predict Stock Index

In [3] paper NNs are applied first to identify the relevant market indicators then to forecast one-month ahead the Swiss Performance Index (SPI) future values. The SPI is a trend reinforcing market with a 0.78 Hurst Exponent, which means that is not a random walk, hence it should be predictable to a certain level.

Possible indicators are: S&P 500 stock index, DEM/USD exchange rate, average bond interest rate of CHF and USD. The predictor (SPI index time series) is scaled to [0.2,08], while the indicators are scaled in the range [-1,1]. The scaling formula for the predictor and the indicators is:

$$y = \frac{x(T_{\max} - T_{\min})}{x_{\max} - x_{\min}} + T_{\min} - \frac{x_{\min}(T_{\max} - T_{\min})}{x_{\max} - x_{\min}},$$

where x is the raw input data, y is the scaled data, $T_{\min}$ target minimum (0.2 for predictor, -1 for indicators), $T_{\max}$ target maximum (0.8 for predictor, 1 for indicators) and $x_{\min} / x_{\max}$ is the raw minimum/maximum value. The monthly differences of the two stock indices are logarithmic scaled as they have some outliners:

$y = \frac{0.2x \cdot \log(|x|)}{|x|} + 0.5$ for the SPI and $y = \frac{x \cdot \log(|x|)}{|x|}$ in case of the S&P 500.

The scaling formula for the DEM/USD exchange rate is $y = 3.39x - 5.71$ and for the differences bond CHF is $y = 1.83x + 0.05$.

The linear dependencies between the indicators and predictor are measured with the cross correlation function. The nonlinear non-parametric multivariate dependencies are measured with a 3-1-1 architecture NN, using backpropagation learning rule, and 0.5 fixed learning rate and momentum. The influence of the interest rate on stock market is well known: if the interest rates are high it is better to invest in bonds than in stocks. As a result, the price of the stocks decreases. The influence of the DEM/USD exchange rate is strong on the Swiss market, because the Swiss industry is sensitive to exports.

The inputs of the 3-2-1 topology (10 connections) NN: DEM/USD exchange rate, monthly difference of S&P 500 and monthly difference of the CHF interest rate. More connections (degree of freedom) allow the NN to adapt to noise, but the overlearning decreases the generalization capabilities, the NN does not recognize the general structure of the market. According to a rule of thumb, it is recommended that the number of connections in a network would be 1/10th of the training data. In order to prevent overfitting the learning stops if the error over the validation set increases. The learning rate and momentum are 0.8 for the first 400 iterations, 0.5 for 200 cycles and 0.2 for the last 200 epochs. Starting with a high learning rate and momentum stops the NN from being trapped in a local minimum.

The network would learn the trend only with lagged SPI values, but this would result in a trend following system. ANN can be used to predict trend breaks, as the linear trend following models fail in this area. Monthly data from January 1987 to December 1994 are used, 84 samples for training and 10 for validation. For testing purposes it is used every ninth sample, reaching in this way a 78% of accuracy in forecasting the trend. The out-of-sample trend prediction performance is 70%.

3.5.2 Leader/Follower Technique

In [31] was investigated the application of feedforward neural networks (implemented with the SPRANNLIB library) for stock price predictions. When the standard technique was used to train the network on MSE a 0.012 NMSE value and a 46.28% of correct direction (SIGN value) was realized while 65.59% SIGN value and the same NMSE value was reached when the network was trained on POCID (Prediction of Change in Direction, see 2.5.1). The standard technique means that only data (10 values of the company’s history) regarding to the predictor’s stock prices were used as inputs. The **Leader/Follower** technique states that the stock price of Ahold company (which is being used through the study) is influenced by other

30 major companies. Thus, Ahold is a follower of other leaders, i.e. the changes occurred on the leaders' stocks are reflected some days later on the Ahold's stock. According to the technique the NN inputs have to be the stocks of the leaders. In this case resulted a 67.71% of SIGN and 0.012 NMSE value was reached when the NN was trained on MSE and 68.74% SIGN and 0.018 NMSE was reached when the network was trained on the POCID error function (Figure 3.5 illustrates the performance of the prediction).

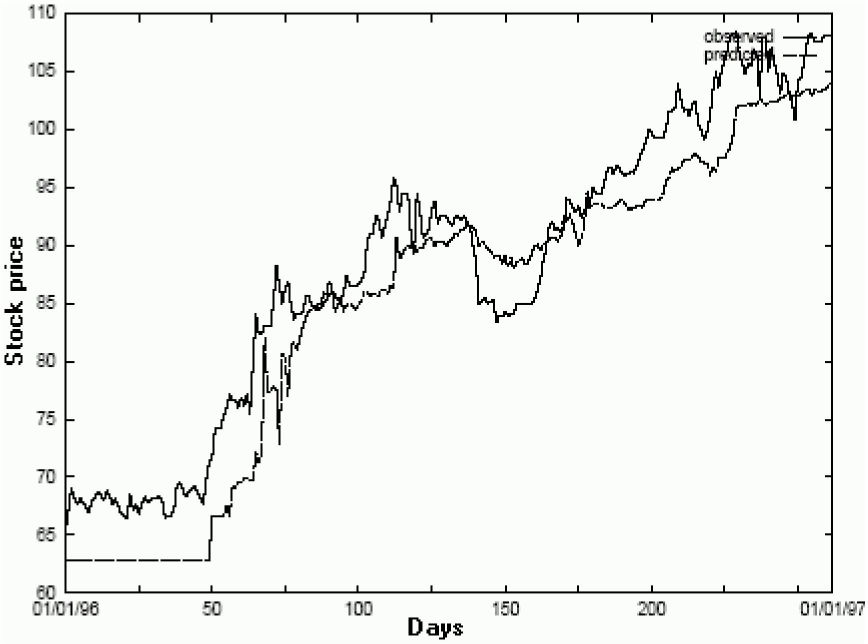

Figure 3.5. Result of the one day ahead Ahold stock prediction using the Leader/Follower technique. The neural network was trained on the POCID error.

3.5.3 Using Statistical Methods and Dual NN System

A dual FFNN system is implemented [7] to predict the returns of S&P 500 stock exchange. In order to test the performance of the proposed model a decision rule is used to determine buy/sell recommendations. Daily and monthly predictions are involved both on single and dual neural network systems. The results showed that the dual NN system (one trained to recognize positive and the other negative returns) gives higher returns with fewer trades. Several selection techniques, selection criteria and ranking techniques are used to select the input indicators of the neural networks. For the daily predictions initially 32 inputs (daily and monthly data as well) are used and 6 in a reduced version (only this is used for the dual NN model), while for the monthly predictions 29 and 8 inputs are used initially and in a reduced case, respectively.

3.5.4 Combining Data Mining with NN

The prediction of stock market return is very complicated because the changes are influenced by many market factors and their structural relationships are not linear. In [37] data mining techniques are used to uncover the relationships of numerous financial and economic variables, the so-called recent relevant variables. The variables then are used as inputs of the probabilistic NN and FFNN to predict the value of future excess stock return of the S&P 500 stock exchange. The training data is divided into 4 periods, and for each of them separate input variables are used. The number of inputs is ranging from 10 to 16, counting a total of 31 financial and economic variables, which have their monthly values from March 1976 until December 1999 (a total of 286 months). The NN model which is using this kind of variables is outperforming the Buy and Hold strategy, the linear regression and the neural network models that use constant relevant variables.

3.5.5 Examining Interrelations

Feedforward NNs (MLPs with 5 units in hidden layer) are involved in the examination of the dynamic interrelations among major world stock markets of Canada, France, Germany, Japan, United Kingdom (UK), the United States (US) [33]. It is found that the NN predicts daily stock returns better than the traditional ordinary least squares and general linear regression models.

3.6 Forecasting Using Various NN Models and Soft Computing Methods

In this section we will present other types of capital market applications (described in section 2.1) using various NNs models (not just FFNNs) and hybrid soft computing methods (see section 3.1.2), pointing out in the same way the used methods and solved problems.

3.6.1 Stock Index Prediction Using Various NN Models

Although FFNNs are having a great popularity within the financial – more explicitly stock market – forecasting there were several attempts to involve and develop other NN architectures as well.

Probablistic Neural Networks (PNNs) are applied for the prediction of Singapore Stock Price Index in terms of its fractional change from the current value [19]. In the article the PNN model is compared with several BP networks as well as Recurrent Neural Network (RNN), Arrayed Probabilistic Network (APN), and Case Based Reasoning (CBR). 6

indicators are used as inputs: index, cumulative stock index return, including dividends, dividend yield, trading volume, Price/Earnings (P/E) ratio. The training data consist of daily samples between January 1, 1985 and December 31, 1995.

The Time-Delay Recurrent Neural Networks (TDRNN) are applied to predict the Korean Stock Market Index in [20]. TDRNNs are capable of learning temporal correlation between current data and past events by using dynamic time delays and recurrences (used in speech recognition or language processing). The TDRNN is an extension of the Time-Delay Neural Network (TDNN) and Adaptive Time-Delay Neural Network (ATNN). The network has four layers (input, two hidden layers and output) and an internal state layer as showed in Figure 3.6. The activation of the second hidden units at time $t-1$ are copied into the internal state units which contain temporal context information of the input sequence and is used as the additional inputs at time *t*. The TDRNN has modifiable weights and time delays along interconnections between the input unit and the first hidden unit. The configuration of the interconnections from the input layer and from the internal state layer to the first hidden layer is called a **Delay Box**. The TDRNN with an NMSE value of 0.015 in prediction of the stock index outperformed the TDNN and ATNN (TDNN realized 0.069 while ATNN 0.068).

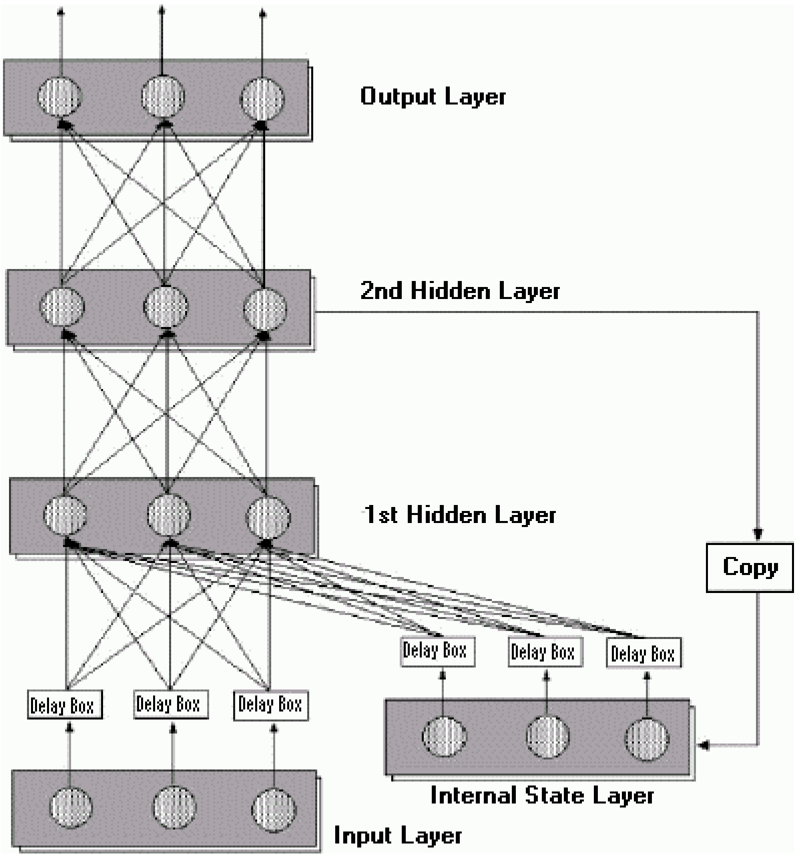

Figure 3.6. Architecture of the time-delay recurrent neural networks (TDRNN).

In [21] were applied Support Vector Machines (SVMs) for daily prediction of the Korea Composite Stock Price Index (KOSPI). 12 indicators are used as inputs: %D, %K, momentum, ROC, RSI etc. The SVM outperformed the backpropagation algorithm and the case-based reasoning method. The SVM produced a 57.83% sign prediction rate in the case of 581 holdout data and 64.75% on training data (1637 samples). The BP algorithm (50-100 epochs, 0.1 learning rate and momentum) reached a 54.73% sign prediction rate when was tested on the holdout data, and 58.52% when was tested for out-of-sample data.

In [15] is proposed a hybrid system where the supervised multilayer perceptron and the unsupervised Kohonen network are integrated in order to predict the chaotic stock series. The so-called Hybrid Time Lagged Network (HTLN) has been successfully tested on data from the Kuala Lumpur Stock Exchange (KLSE). What is more, the HTLN is compared with two other standard networks applied for stock prediction: supervised MLP network known as Time Lagged Feed-forward Network (TLFN) and unsupervised Kohonen network known as Highly Granular Unsupervised Time Lagged Network (HGUTLN).

3.6.2 Forecasting Using Hybrid Soft Computing Methods

There are several attempts to solve the forecasting problem using hybrid systems, by combining the capabilities of different optimization techniques like neural networks (NNs), genetic algorithms (GAs), fuzzy logic, and other learning and classifier systems.

We have already seen in part 3.5 that neural networks are combined with other statistical and machine learning methods in order to forecast multivariate financial time series. Several techniques are used with the purpose of identifying the financial and economical indicators: traditional statistical methods [7], data mining techniques [40], [37], but also neural networks [3].

There are several attempts to solve the forecasting problem by using hybrid soft computing (presented in section 3.1.2) systems: the power of neural networks is combined with other machine learning tools. The most prevalent solution is to combine ANNs with GAs, i.e. to use GAs to evolve the topology and/or weights of the network. The Genetic Programming (GP) can be applied to generate more efficient learning functions.

3.6.2.1 Web Based Stock Prediction Systems

Stock market forecasting can be combined with other research fields. A very useful combination is with data mining [40]. Major European, Asian and US stock markets are predicted using information contained in articles published on the Internet. These information are not only the daily closing price of the markets but also textual data, which can assure an improved prediction. The prediction is done in several steps. After downloading the articles from the given sites, the number of occurrences of the keyword tuples (it is given manually by a domain expert) are counted, which are then transformed in weights. From the weights and the closing values of the training data, probabilistic rules are generated and from these the today's closing price is predicted, which are published on the project's web page.

Another web based stock prediction system is implemented in [36], where a fuzzy neural network based hybrid method is used for the prediction of stock prices. The input data are the moving averages of the weekly stock prices. Web search techniques are used to collect the data from the Internet, and then fuzzy rules are extracted to make future predictions.

3.6.2.2 Genetically Evolved NNs

In [27] is presented a hybrid forecasting system using feedforward and reccurent neural networks as well as genetic programming. The developed method is tested on the S&P 500 stock exchange and on other types of time series. It is stated that the recurrent neural networks are more efficient than FFNNs but above all it is more important to choose the best inputs and preprocessing techniques as this can result in an enhanced prediction. Genetic programming is used to evolve more efficient transfer function. The results of various algorithms and NN models are compared.

Genetically evolved neural networks are used to predict the Straits Times Index (STI) of the Stock Exchange of Singapore (SES) [32]. The used inputs are the open, high, low, close prices, trading volume of STI, as well as the indices of world major stock exchanges: Dow Johns Industrials Average (DJIA), NASDAQ (IXIC), Heng Seng Index (HSI) and Nikkei 225 (N225). A total of 360 observations are used between August 1, 1998 and January 31, 2000. The test was done on the training data and an accuracy of 81% is obtained in predicting the market direction (see Figure 3.7).

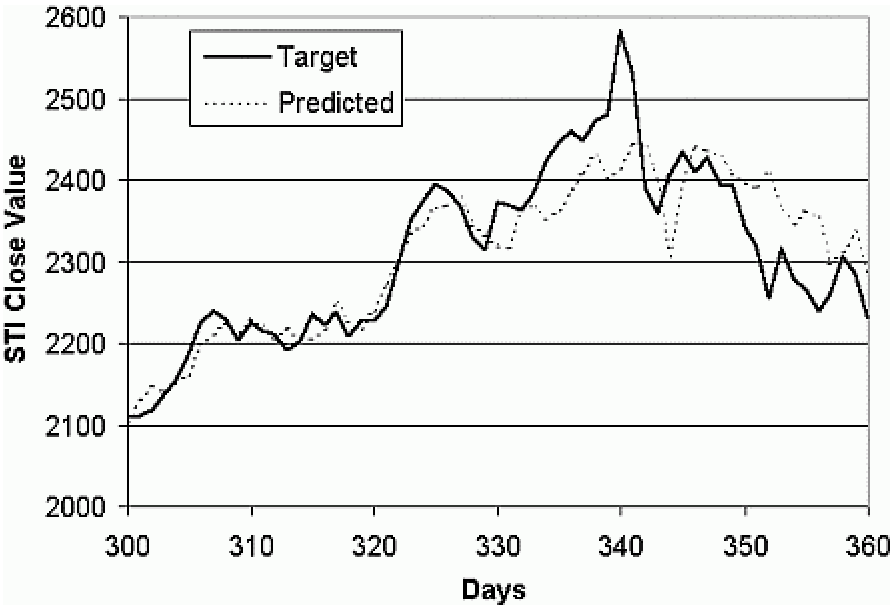

Figure 3.7. Forecasting the STI by a genetic-neural system between August 1, 1998 and January 31, 2000 (same train and test period).

In the [17] study it is predicted the daily excess returns of FTSE 500 and S&P 500 indices over the respective Treasury Bill rate returns using autoregressive and feedforward neural network models. The optimal NN model was evolved with GAs. It is implemented and used two empirical methods in order to test the randomness of the input data (time series): Run test and Brock, Dechert, Scheinkman (BDS) test. It is proved that the index values are not random, so it is possible the prediction.

In [4] the author presents a hybrid approach to stock market forecasting that integrates both GAs and ANNs and cooperatively exploits them to forecast the next-day price of stock market indexes. It is used the so-called extended classifier system (XCS), which relies on technical analysis indicators to determine the current market status, in conjunction with feedforward ANNs explicitly designed for financial time series prediction. Thus the new system it is called neural XCS (NXCS), which outperformed the Buy and Hold strategy.

3.6.3 Other Financial and Capital Market Applications

Quite recently some researches have used hybridized SC techniques for automated stock market forecasting and trend analysis. Principal component analysis was utilized to preprocess the input data, a NN for one-day-ahead stock forecasting and a neuro-fuzzy system for analyzing the trend of the predicted stock values.

3.6.3.1 Trading Rules

There were several attempts to find efficient trading rules. The fuzzy logic and NNs were applied to buy/sell timing detection and stock portfolio selection. The GAs were used as a method for formulating market-timing trading rules, as well singular value decomposition (SVD) and principal component NN with 3, 4, 5 and 10 nodes in the hidden layer. In another research the GAs were utilized to learn technical trading rules for the S&P 500 index using daily prices from 1928 to 1995 [34].

A buying and selling timing prediction system is implemented for the Tokyo Stock Exchange Prices Index (TOPIX) by using an 11-10-3 NN topology [29]. The output patterns (see Table 3.1) correspond to specific TOPIX curve patterns: buying signal (current price is at bottom), selling signal (current price is at top) and no change (otherwise). The inputs are: deviation from MA25/MA75/MA200, vector curve of MA6/MA25/MA75, psychological line in 12/25 days base, relative strength index in 9/12 days base and volume ratio in 25 days base. The psychological line is calculated by dividing the number of days of price ups by certain past period. Relative strength index is calculated by dividing the sum of price ups by the sum of price ups and downs over certain past period. 260 weekly data were used for training and 119 weekly data for test. Total ratio of correctness for training is 97% and for testing is 63%, outperforming in this way the traditional technical analysis.

Table 3.1. NN Output patterns for buying and selling timing prediction.

TOPIX Past Present	Future	NN output pattern
↗	↘	1-0-0: selling signal
↗	↗	0-1-0: no change
↘	↘	
↘	↗	0-0-1: buying signal

3.6.3.2 Macroeconomic Forecasting

The neural networks are applied for a great variety of financial forecasting problems. Besides stock market forecasting, which is the most important research field, it is also used for macroeconomic forecasting, customer targeting etc. Augmented neural networks are used to forecast quarterly growth of Canada's real GDP [14]. Several fundamental data concerning the country’s economy are utilized as indicators: quarterly growth rate of Finance Canada's index of leading indicators of economic activity (one-quarter lag), employment growth (contemporaneous and one-quarter lag), Conference Board's index of consumer confidence, first difference of the real long term interest rate (nine-quarter lag), first difference of the federal government budgetary balance as a share of GDP (three-quarter lag). The neural network model outperforms the linear regression method by reaching a small (0.077) mean squared error.

3.7 Improving Neural Networks

In this section we will consider some ideas that can help us to improve the performance of FFNNs and to remove also some limitations.

3.7.1 Advanced Cost Functions

Cost (also referred as error or performance) functions are used to estimate the difference between the target value and the output of the network. The deviation depends on the specific problem and on the used cost function which estimates the error. Normally the quadratic error and absolute error measurements are used as performance functions.

In [30] the author proposes another error function to efficiently deal with the outliers:

$$\frac{1}{a}\ln\cosh(a(x_t - \widehat{x}_t)),$$

where $\widehat{x}_t$ is the output of the neuron *t*, x_t is the corresponding target value and $a \in [3,4]$ has proven to be suitable for financial applications. The $\ln\cosh(\cdot)$ error function is a smooth approximation of the absolute error function.

Asymmetric cost functions are suggested for financial time series predictions in [8]. The majority of the used error functions are symmetrical (primarily the quadratic cost function). The quadratic error measure penalize a forecast more for extreme deviations rather than for small ones, while absolute error measures give identical weight to every error regardless of scale. In business management forecasting the cost arising from over- and underestimation are often not symmetrical and definitely not quadratic in form. The used *LINLIN* cost function for the stock keeping unit problem is as follows:

$$LINLIN = \begin{cases} a\,|\,x_t - \widehat{x}_t\,| & if \quad x_t > \widehat{x}_t \\ 0 & if \quad x_t = \widehat{x}_t \\ b\,|\,x_t - \widehat{x}_t\,| & if \quad x_t > \widehat{x}_t \end{cases},$$

where $\hat{x}_t$ is the predicted value, x_t is the corresponding target value and $a, b,\ a \neq b$ gives the slopes of the branches. In real-life financial applications it is important to penalize under- or overestimation with different values.

3.7.2 Dealing with Non-Stationarity

If the training set is to small, the noise makes it harder to estimate the appropriate mapping. On the other hand if the training set is too large, the non-stationarity of the data will mean more data with statistics that are less relevant for the task. In order to solve this problem we divide the original historical data in several subsets, where the training and testing datasets are alternating each other. It is recommended that, after training the NN on a certain number of samples (let us say 100) to test the NN's performance on the subsequent segment of data (let's say the next 30 observations). The entire training/test set window is moved forward with a given number of units (for example 30) and the process is repeated.

3.7.3 Evolving Neural Networks with Genetic Algorithms

When NNs are involved to solve computational problems, the key to success is the optimal NN architecture. Network design has an outmost importance within neural computing community.

Several components of the neural network can be evolved with the help of the evolutionary computation. Finding the optimal topology requires the identification of the number of hidden layers, the number of neurons in each layer, and the connections between neurons from different layers. Genetic algorithms can be also applied for the training of neural networks, i.e. to evolve the NN's weights [12.c]. Finally, the generation of learning rules is possible with the support of genetic programming methods. For a comprehensive list of publications in this topic please see the [1] indexed bibliography.

4 Implementation. Experimental Results

In this chapter it is presented a self-developed computer application and is related the experimental results. The software package implements a neural network based financial time series forecasting system, using the methods, techniques described in the above chapters. With the help of this application, we are trying to make predict the index values of several major US and European stock markets and to forecast the stock prices of famous gigantic US companies (especially from the IT field).

4.1 Data Choice and Preprocessing

Several different financial time series were used for experimental purposes. These are real datasets, representing the indices of various stock markets, the prices of companies stocks and bond ratings. The historical data were downloaded from the Yahoo Finance website [41]. The daily/weekly/monthly observations are stored in Excel CSV comma delimited spreadsheets and are sorted by date in descending order. In most of the cases for each time unit we have access to the open, high, low and close prices and to the total volume of transactions. As usually we will use only the close prices in order to built up different indicators and to use as inputs of the neural network. The primarily aim when collecting data is to find for different financial time series representative number of samples. The more sample we can use the more accurate results we can achieve.

4.1.1 Financial Time Series Data

The stock exchanges, companies and bond ratings involved in experiments for which we are trying to predict the future values (one-day ahead, one week ahead and one-month ahead, respectively) are taken from various fields of activity and geographical locations. In the US the NASDAQ Composite (also will be referred later as NASDAQ), Standard & Poor's 500 (S&P 500), Dow Jones Averages Industrials (DJAI or just simply DJI) are the most important stock exchanges, which have influence also on the global market. In Europe the German and the Swiss stock exchanges (DAX respectively SSMI) are very representative for the Euro zone and outside of the Euro zone.

As we had access to representative amount of data only for US companies, we will carry out experiments on these time series. IBM and Microsoft are very representative companies in the IT field, which usually are dominated by high volatility, so to make an accurate prediction is more challenging, than to predict other financial time series. Both companies are components of the Dow Jones Industrials stock market.

Treasury securities are good alternatives for money investment. Unlike stocks, they are more stable, with lower volatility but also the returns are more moderate. There is an obvious relation between the price of the stocks and the value of bond ratings. If the interest rates are high it is better to invest in bonds than in stocks. As a result, the price of the stocks decreases. We will see how can our system predict the future bond ratings for two representative treasury bills: 13-Week Bill for short term investment and 10-Year Note for long term investment.

4.1.2 Statistical Analysis

The economists and especially practitioners are using statistical indicators in order to get more information about the time series. Usually it is calculated the minimum and maximum value, the mean of the observations and standard deviation. These are helping to identify the nature of the time series, to see the trends, and to decide what kind of preprocessing procedures to use.

4.1.3 Normalization, Detrending and Noise Reduction

By normalization original values are mapped to fewer (new) ones. Usually the time series are transferred to [0,1] or [-1,1] range, depending on the nature of transfer function used by the NN system.

Most of the time series are dominated by trends. Trends are influencing the searching direction, which can lead to false results. Detrending removes the growth of a series. The trend is removed or at least drastically minimized by calculating the k-step returns (see 2.5.2)– most of the time one-step return (see formula in section 2.2.3.4) – or the differences between two or more subsequent values.

Noise reduction it is also very important, as the outliers can influence in a bad way the quality of the results. In order to reduce the noise, it is used the logarithmic values of the time series.

We will use only normalization, as we are interested in the forecasted exact value. The predictor (the predicted dataset) is scaled the in the range [0,1], because in the NN model we will use the sigmoid activation function, which produces values in this range. If we want to forecast only the trend, the direction of the time series, we calculate the difference or the k-step returns in order to detrend and then take the logarithmic values to reduce the noise.

For a more detailed explanation of data preprocessing please consult chapter 2.2.3.

4.2 Nonlinear Analysis. Predictability

Financial time series (mostly stock indexes) have a complex dynamics, which is very hard to describe or even impossible by using only linear methods, analysis. Nonlinear analysis is able to find relations between several time series, which is important in order to predict future values.

There have been several questions about the predictability of financial time series. It is still unclear if the stock indexes are following a random walk or they are integrating regularities. The Hurst Exponent (see chapter 2.3), which is calculated with the help of the rescaled range analysis, can be a useful property to decide on the type of the time series. The values of the Hurst Exponent for the time series involved in the experiment are listed in Table 4.1.

Table 4.1. Hurst exponent values of the time series.

Time Series	Hurst Exponent
NASDAQ Composite	0.642
Dow Jones Industrials	0.537
S&P 500	0.585
DAX	0.641
SSMI	0.637
IBM	0.507
Microsoft	0.460
13-Week Bill	0.692
10-Year Note	0.562

It can be observed that the lowest values have the time series representing the stock prices of the Microsoft and IBM companies. This means that these time series are representing antipersistent processes, dominated by high volatility. In fact, this is true, because the companies from the IT field were influenced by the fast grow of the technology, producing sudden booms, and by the sudden drops caused by the saturation of the market, processes which are very characteristic to this branch of industry. On the other hand, contrary to this, the 13-week Treasury Bill shows the highest Hurst Exponent. This is also natural, because the bond markets are less susceptible to changes, the volatility and the risk is much lower, but the interest rates are representing lower incomes than the incomes which can be achieved on stock markets. If we are looking at the stock markets, the NASDAQ Composite and the DAX have the highest Hurst Exponent, representing trend-reinforcing processes.

As a conclusion, we are expecting to have the best predictions in the case of the bonds and NASDAQ stock market, while in the case of companies the predictions will be less accurate.

4.3 Indicators

It is important to use an optimal number of inputs, as two few inputs does not contain enough information about the time series, while on the other hand to many inputs generates higher noise, the time series will be distorted, biased in this way.

In the case of the univariate analysis the moving averages (please refer to section 2.2.4) showed the best results, which were applied for 5, 10 and 50 days. MA120 had a negative effect on the results, which is explained by the noise introduced in the NN system. We also used as input the momentum indicator, the current value and a lagged value in order to predict the next value of the time series. In the application we also listed in addition the ROC and RSI indicators. It is possible to use any combination of the 8 different indicators.

4.4 NN Model Description

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

One of the most important part of the application is the design of the neural network system, i.e. constructing an optimal topology and connections. Another important task is to find the correct settings of parameters, which are giving the best results. In the following we will describe the type of the NN and values of the various parameters used for prediction.

4.4.1 Topology

For this purpose we have used supervised **Feedforward Neural Network**s (FFNN), with error correction learning rule, based on the **MultiLayer Perceptron** (MLP) architecture. As we already presented in the chapter 3 several other NNs, like recurrent NNs, time delay NNs, NNs using radial bases functions were used to solve forecasting problems, but the FFNNs were the most often used architectures. This is because, they are very efficient in solving patter recognition problems and forecasting problems, they are fast and easy to implement.

It is very hard, if it is not even impossible to find for every problem the “best” network topology. In order to design an efficient topology we have to find the corresponding values for the following parameters: the number of layers (more exactly the number of hidden layers, as the input and output layer is given), number of neurons in each layer, and the connections between neurons from different layers. Several methods exist in order to optimize the topology of a neural network. Most often genetic algorithms are used to evolve the optimal topology.

As we have to have for sure one input layer and one output layer, we have to decide how many hidden layers to use, which depends on the number of hidden units. More hidden neurons increase the NNs capability to solve more complex problems, to extract higher-order statistics from input data. It is more efficient and convenient to distribute the hidden neurons in more hidden layers, rather than to use only one hidden layer. Usually two hidden layers should be enough to solve any kind of linear or nonlinear problem.

Previous results have shown that, for prediction problems the 6-4-3-1 topology is very efficient [42]. We have 6 neurons in the input layers as we use 4 indicators, and 2 lagged values of the time series. It is important to use an optimal number of input units, because the more inputs are used the higher the noise. The input layer is the only layer that does not contain transfer functions. The output is also obvious, because we want to have as a result the forecasted value. Two hidden layers are better than one, and a totally of 7 hidden units is quite enough to describe any kind of nonlinear event. We also used in the case of the hidden and output layer an extra unit, representing the bias, having a constant input value of 1. This has the role of the threshold, having a negative weight.

The network is fully connected, i.e. every unit is connected to each unit from the next layer. More connections (degree of freedom) allow the NN to adapt to noise, but the overlearning decreases the generalization capabilities, the NN does not recognize the general structure of the market. According to a rule of thumb, it is recommended that the number of connections in a network would be 1/10th of the training data. As we are intending to use 2000 training samples for daily forecasting and at least 108 for monthly forecasting, this means that the total number of connections (58) is not high, so the network will not overlearn.

4.4.2 Training

Learning (or training) of the NN is the procedure of the adjustment of weights. The training procedure “fits” the network to a set of samples (training set). The purpose of this fitting is that the fitted network will be able to generalize on unseen samples and allow us to infer from them.

The samples from the training data are fed into the network, and the outputs are compared with the target values. The difference between the target and output value is the error, which is propagated back in the network by updating the weights of the connections. Consequently, we use the **backpropagation (BP) algorithm** [12.b] to implement the **error-correction learning rule**, i.e. gradient descent weight/momentum, weight/bias learning function [26]. With the help of the back-propagation, inadequacies in the output are fed back through the network so that weights can be updated in order to improve the NN’s learning capability.

The used **activation (transfer) function** is the **sigmoid (logistic) function**, and the **error (cost or performance) function** is the **mean square error function**.

4.4.3 Parameters

The number of **epochs** in case of the daily and weekly data was 1000 and in case of the monthly data 3000 epochs.

In order to prevent overfitting we use **cross-validation**, i.e. the learning stops if the error over the validation set increases. Periodically, while training on the learning data set the network is tested for performance on the cross validation set. If the network is starting to overtrain on the training data, the cross validation performance will begin to degrade. Thus, the cross validation data set is used to determine when the network has been trained as well as possible without overtraining (maximum generalization).

We use fixed and **variable (dynamic) learning rate**s as well. The learning rate is 0.8 for the first 250 iterations, 0.5 for 250 cycles, 0.2 for the next 250 cycles and 0.1 for the last 250 epochs. Starting with a high learning rate stops the NN from being trapped in a local minimum. We also used **momentum term** with a value of 0.1. The dynamic learning rate and momentum term are enhancing the NN performance regarding learning time and accuracy.

4.5 Testing the System. Results and Evaluation

The system was tested on nine different financial time series using historical data, which are publicly available. Daily, weekly and monthly observations were used to predict future value: one-day ahead, one-week ahead and one-month ahead, respectively.

For each of the nine time series we will report different error and performance measurements in order to compare the results: mean square error (for the training and testing datasets), accuracy of predicting the future value and accuracy of predicting the price changes (or sign prediction rate). All these measurements are described in section 2.5. The NN is trained on the MSE (defined in 2.5.1), which predicts the future values of the time series. In practice it is more relevant to have information about the direction of price changes.

As the most concluding results we got for the NASDAQ Composite index, we will illustrate with figures the predicted values for this and other relevant time series.

4.5.1 Daily predictions

In order to have a more precise evaluation of the results it is recommended to use as much data as possible for training and testing. In the case of the daily observations we used for every time series a totally of 3000 samples. These data are segregated in time order. According to a rule of thumb [42], [43] two third (in this case 2000) of the data is used for training and the remaining one third (i.e. 1000 samples) is used for testing the performance of the system. Two fifth (400 samples) of the testing data it is also used for cross-validation. It is also common to use 80% of the original data for training, 10% for validation and 10% for testing the performance of the forecasting system.

In the case of forecasting it is crucial what kind of data we use to train the neural network. The NN will learn and discover trends based on the training data. Figure 4.1 illustrates the evolution of daily NASDAQ Composite index, which was already showed in section 2.2.3.4. In this case we are lucky, because the first 2000 data used for testing contains positive and negative changing trends, so the NN is capable of a more accurate prediction. If we would use only the first 1500 observations, the NN would not have any information concerning the abrupt increase of the index values. It is also important to use relevant amount of data for testing purposes. Table 4.2 enumerates for all of the nine time series the number of samples used for training and testing, denoting the time intervals, as well.

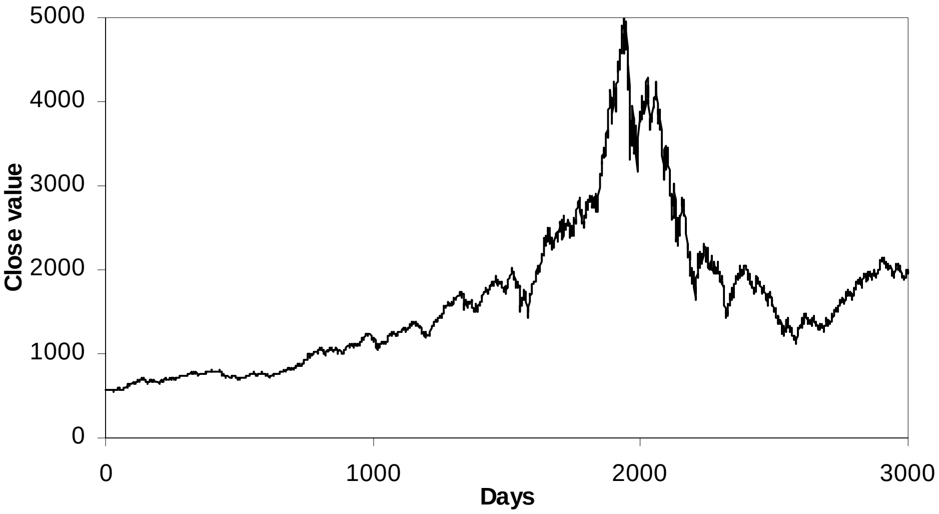

Figure 4.1. Evolution of daily NASDAQ Composite index between July 13, 1992 and June 4, 2004.

Table 4.2. Segregation of daily historical data.

Time Series	Training samples	Training period	Testing samples	Testing period
NASDAQ	2000	13-Jul-92/9-Jun-00	1000	12-Jun-00/4-Jun-04
DJI	2000	13-Jul-92/9-Jun-00	1000	12-Jun-00/4-Jun-04
S&P 500	2000	10-Jul-92/8-Jun-00	1000	9-Jun-00/4-Jun-04
DAX	2000	10-Jul-92/22-Jun-00	1000	23-Jun-00/4-Jun-04
SSMI	2000	19-May-92/30-May-00	1000	31-May-00/4-Jun-04
IBM	2000	13-Jul-92/9-Jun-00	1000	12-Jun-00/4-Jun-04
Microsoft	2000	13-Jul-92/9-Jun-00	1000	12-Jun-00/4-Jun-04
13-Week Bill	2000	8-Jun-92/1-Jun-00	1000	2-Jun-00/4-Jun-04
10-Year Note	2000	11-Jun-92/6-Jun-00	1000	7-Jun-00/4-Jun-04

The results of the predictions are listed in Table 4.3. The *Training MSE* denotes the mean square error (defined in section 2.5.1) achieved by the NN during the last epoch of training. The *Test MSE* and *Test NMSE* reports the MSE value and NMSE value (see 2.5.1) achieved during testing session. This are important information, as one can see if the NN system learns or not. If the difference is big between the two values, it means that the neural network is unable to generalize, to learn the general structure of the market. The *Value* column contains the percentage (accuracy) of the correctly predicted prices, and finally the *Sign* stand for the percentage of the correct direction estimations.

Table 4.3. Results of the one-day ahead prediction using daily data.

Time Series	Training MSE	Test		Value (%)	Sign (%)
		MSE	NMSE		
NASDAQ Composite	2.71E-5	2.15E-4	0.011	97.11	51.30
Dow Jones Industrials	2.29E-5	7.40E-4	0.117	96.16	52.51
S&P 500	2.92E-5	1.20E-3	0.089	95.99	50.00
DAX	2.77E-5	9.59E-4	0.032	95.90	50.30
SSMI	2.49E-5	1.58E-3	0.078	94.88	49.90
IBM	8.15E-5	9.39E-5	0.024	98.18	49.60
Microsoft	1.24E-4	4.88E-3	0.532	68.85	48.90
13-Week Bill	2.12E-5	1.58E-3	0.019	89.76	66.23
10-Year Note	3.33E-5	1.20E-4	0.016	98.49	56.41

In every case the system achieves a low MSE value during the training phase and for the testing samples. This means, that the NN is learning in a very good rate. What is more important to see how the NN manages to predict the future values, i.e. in what percentage the value is estimated. For this performance measurement we also get very good results, and as we expected based on the Hurst Exponents (see section 4.2), the treasury bills and the NASDAQ Composite (see Figure 4.2) have the best results. The Microsoft gets the worst results, which is also due to fact that has the lowest Hurst Exponent.

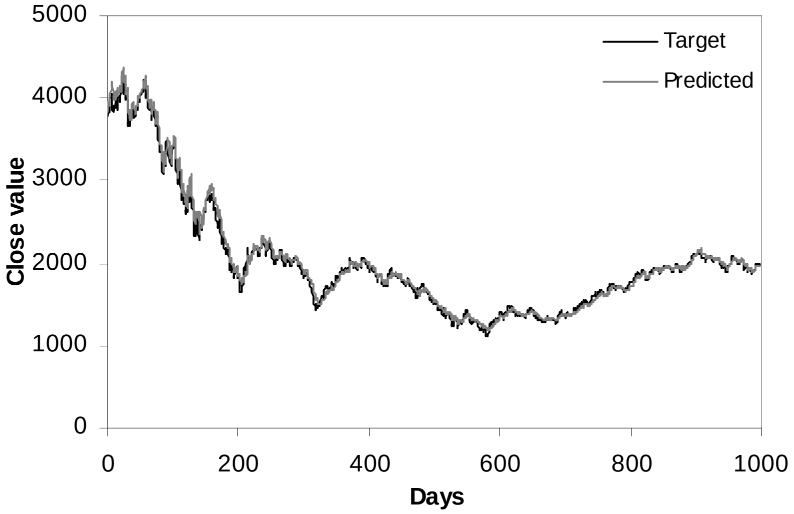

Figure 4.2. One-day ahead forecasting results for NASDAQ Composite index from June 12, 2000 to June 4, 2004.

In practice between economists it is more important to predict the price changes. Unfortunately here we get quite disappointing results, but it is a common issue in the literature. The best result is obtained by the 13-Week Bill, which was expected, as the treasury bills represent a save investment with lower risk but also with lower return compared to stock markets. Within the stock markets the DJI receives the highest sign prediction rate. Figure 4.3 and Figure 4.4 depict the daily prediction results for the DJI time series and 13-Week Bill time series, respectively.

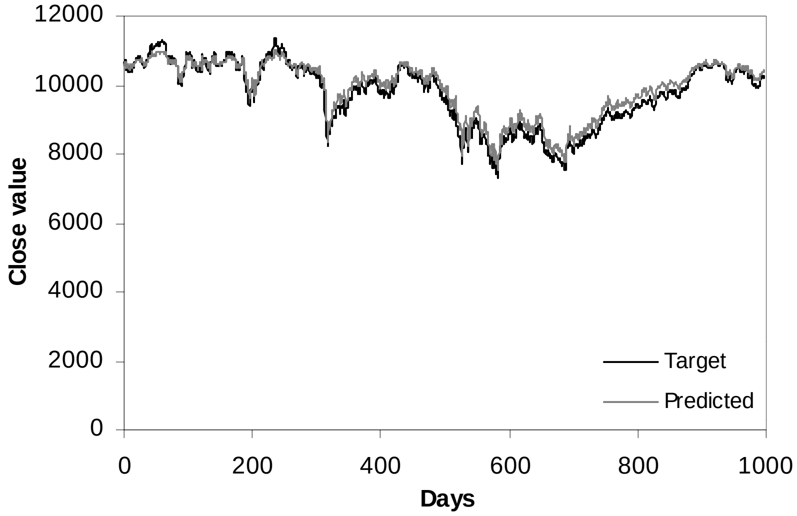

Figure 4.3. Daily forecasting results for Dow Jones Industrials index between June 12, 2000 and June 4, 2004.

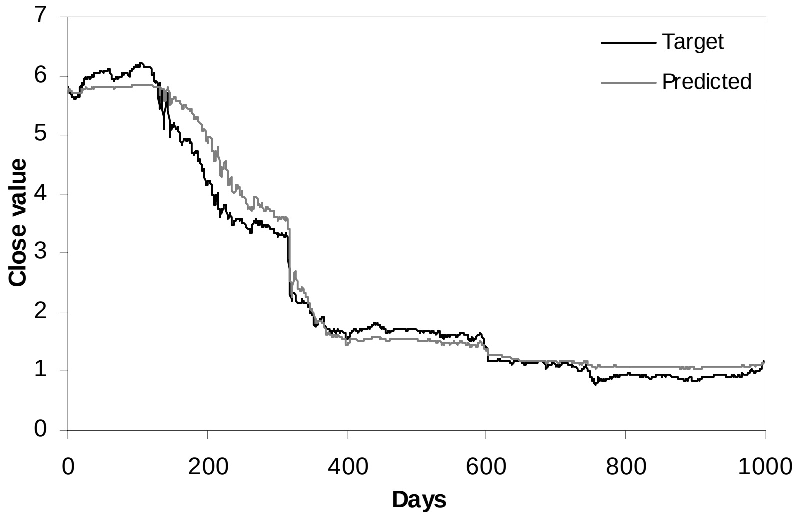

Figure 4.4. Daily forecasting results for the 13-Week Bill between June 2, 2000 and June 4, 2004.

The *Sign* values in the table take into account only the cases where the directions are exactly predicted. Most of the times, only on a small percent of the sample depend if the price is incrementing or decreasing, which is due to the noise in the time series. If we were applying an error tolerance of 2% (of the current sample value) we would get much better results, for example in case of the NASDAQ index prediction we would have 67.23% of sign prediction rate. In some experiments the neural network's performance is tested by using the training dataset. We also tested our NN on the training samples, and the NASDAQ time series reached a sign prediction rate of 57.29% (88.50% with 2% error tolerance).

4.5.2 Weekly predictions

We had much fewer weekly and monthly data, so in these cases we use the maximum number of samples, the size of the dataset varying from one time series to another.

Also in case of the weekly data is crucial what kind of data are used for training. Figure 4.5 illustrates the evolution of weekly NASDAQ Composite index. The first 700 samples are used for training. In this case it is quite difficult for the NN to predict the high increment of the prices, as it does not have any information concerning this. In most of the experiment we used more samples for training than 1/3 of the historical data, rounding the values to the next hundred. Table 4.4 lists for every time series the number of samples used for training and testing the neural network, reporting the sampling periods, as well.

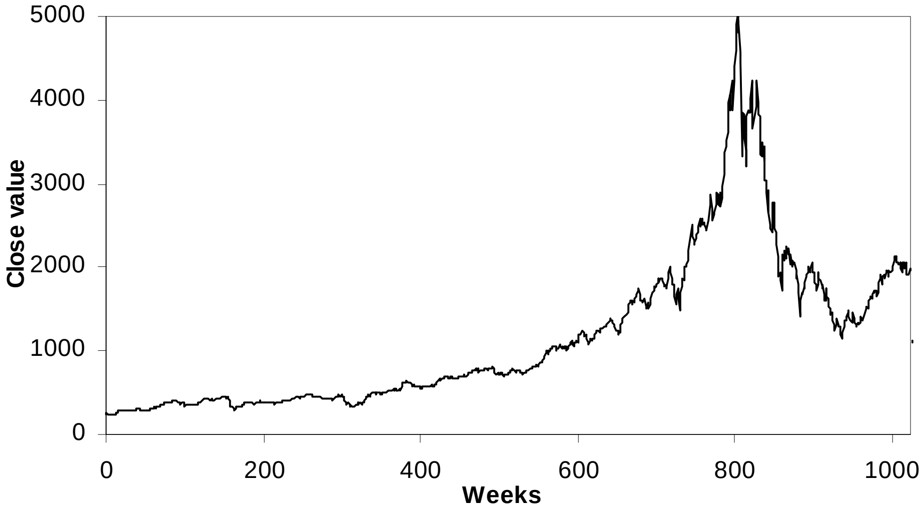

Figure 4.5. Evolution of weekly NASDAQ Composite index between October 11, 1984 and June 1, 2004 (1025 samples).

Table 4.4. Segregation of weekly historical data.

Time Series	Training samples	Training period	Testing samples	Testing period
NASDAQ	700	11-Oct-84/2-Mar-98	325	9-Mar-98/1-Jun-04
DJI	2000	2-Dec-46/25-Mar-85	1000	1-Apr-85/1-Jun-04
S&P 500	800	20-Oct-82/9-Feb-98	328	17-Feb-98/1-Jun-04
DAX	500	26-Nov-90/19-Jun-00	206	26-Jun-00/31-May-04
SSMI	500	9-Nov-90/26-Jun-00	205	3-Jul-00/1-Jun-04
IBM	1474	2-Jan-62/26-Mar-90	739	2-Apr-90/1-Jun-04
Microsoft	700	13-Mar-86/2-Aug-99	251	9-Aug-99/1-Jun-04
13-Week Bill	1542	4-Feb-60/14-Aug-89	771	21-Aug-89/1-Jun-04
10-Year Note	1500	2-Jan-62/24-Sep-90	713	1-Oct-90/1-Jun-04

Table 4.5 reports the results of the weekly forecasting for the nine financial time series. One can observe that we get better sign predictions than with daily data, but the predicted values are less accurate. This is quite obvious, because on one hand the system has fewer input data. On the other hand for a long term prediction we are not really interested in the concrete index value, but it is more important the trend, the changes which are going to happen in the near future. It is very hard to have a better value accuracy, because the time interval is large, so the fluctuations and volatility of the prices is also higher.

Using the cross-validation method, the neural network outputs the result very fast, only after approximately 250 iterations in the case of NASDAQ Composite.

Table 4.5. Results of the weekly forecasting.

Time Series	Training MSE	Test MSE	Value (%)	Sign (%)
NASDAQ Composite	1.33E-5	1.85E-2	86.90	57.59
Dow Jones Industrials	1.87E-5	5.82E-3	92.56	52.01
S&P 500	1.53E-5	7.28E-3	92.90	51.23
DAX	8.75E-5	1.50E-2	82.85	49.51
SSMI	8.95E-5	1.39E-3	95.75	53.69
IBM	1.19E-4	1.27E-4	95.05	54.95
Microsoft	4.90E-4	1.40E-2	52.03	52.21
13-Week Bill	7.57E-5	4.42E-4	88.54	54.23
10-Year Note	4.73E-5	4.09E-4	95.56	52.74

As we already seen in the previous section, also in this case the treasury bills and the NASDAQ time series had the best performance. The forecasting result for NASDAQ Composite index is illustrated in Figure 4.6, while Figure 4.8 proves the performance of the 13-Week Bill. IBM is another good performer, achieving a low MSE value, high value and sign prediction rate, which can be seen on Figure 4.7.

Also for weekly predictions, the Microsoft time series gets the lowest overall results, but this time also DAX has low sign prediction accuracy.

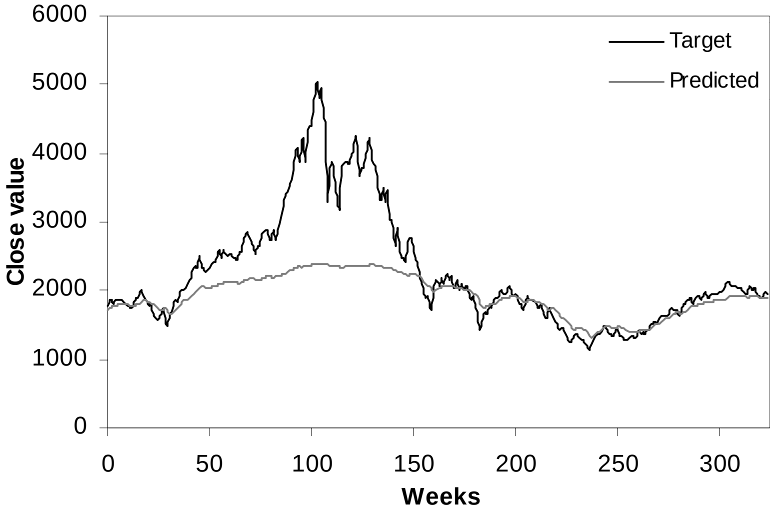

Figure 4.6. Weekly stock price prediction of NASDAQ Composite from March 9, 1998 to June 1, 2004 (325 observations).

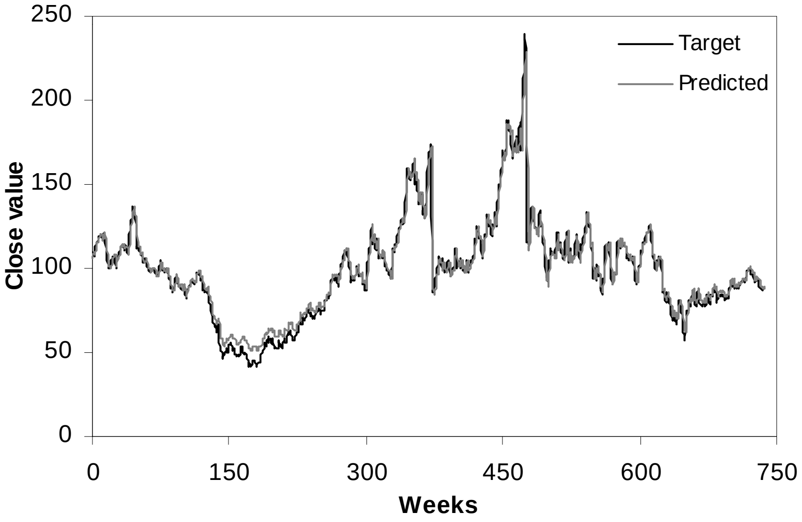

Figure 4.7. Prediction of the weekly IBM stocks from 2-Apr-90 to 1-Jun-04.

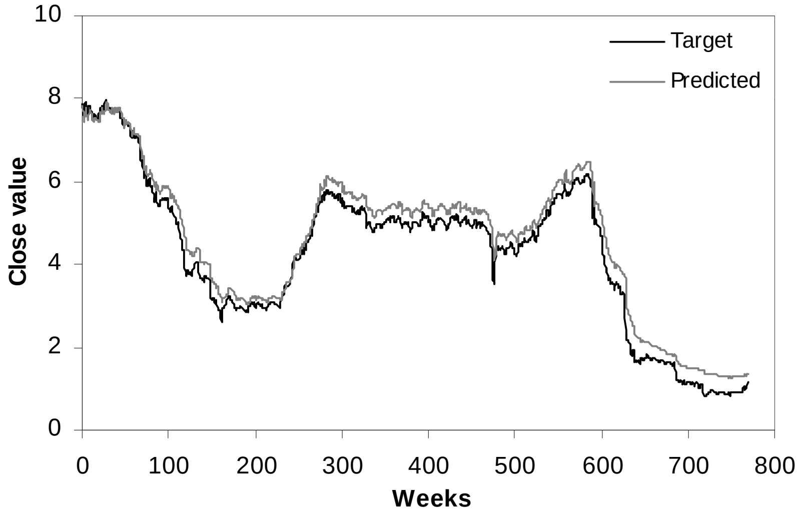

Figure 4.8. Weekly prediction of the 13-Week Bill between 21-Aug-89 and 1-Jun-04.

4.5.3 Monthly predictions

It is much harder to make accurate monthly predictions as we have much fewer data (see Figure 4.9) and also the time-step – for which the future prediction is made – is quite long. Table 4.6 summarizes the number of observations used for training and testing purposes, specifying in the same way the time intervals. For some time series (DJI, DAX, S&P 500) we used a few more samples than 1/3 of the historical data for training purpose.

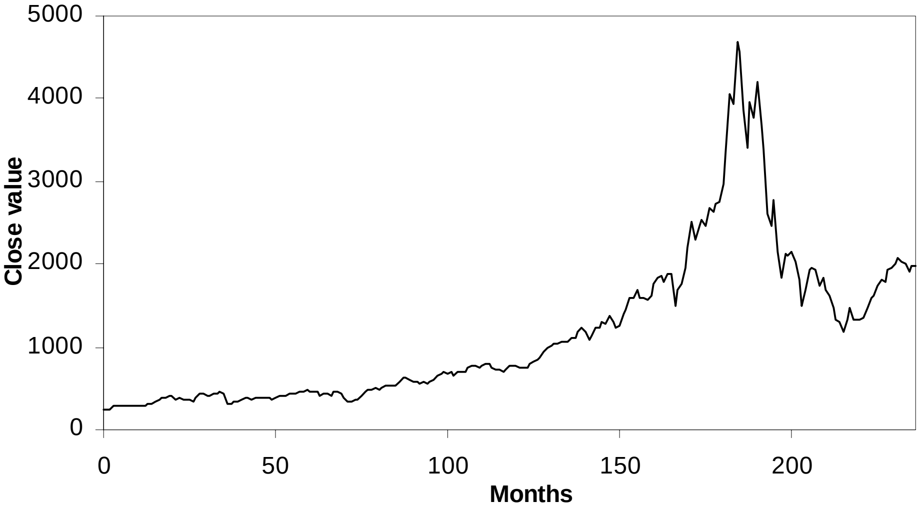

Figure 4.9. Evolution of monthly NASDAQ Composite index between October 11, 1984 and June 1, 2004 (237 samples).

Table 4.6. Segregation of monthly historical data.

Time Series	Training samples	Training period	Testing samples	Testing period
NASDAQ	158	11-Oct-84/3-Nov-97	79	1-Dec-97/1-Jun-04
DJI	800	2-Jan-30/1-Aug-96	94	3-Sep-96/1-Jun-04
S&P 500	181	20-Oct-82/1-Oct-97	80	3-Nov-97/1-Jun-04
DAX	128	26-Nov-90/1-Jun-01	36	2-Jul-01/1-Jun-04
SSMI	108	9-Nov-90/1-Oct-99	56	1-Nov-99/1-Jun-04
IBM	340	2-Jan-62/2-Apr-90	170	1-May-90/1-Jun-04
Microsoft	150	13-Mar-86/3-Aug-98	70	1-Sep-98/1-Jun-04
13-Week Bill	354	4-Feb-60/3-Jul-89	179	1-Aug-89/1-Jun-04
10-Year Note	340	2-Jan-62/2-Apr-90	170	1-May-90/1-Jun-04

Although, according to the literature it is very hard to make one-month ahead predictions, we got satisfactory results, which are summarized in Table 4.7.

Table 4.7. Results of the monthly predictions.

Time Series	Training MSE	Test MSE	Value (%)	Sign (%)
NASDAQ Composite	3.77E-5	1.16E-2	85.81	58.44
Dow Jones Industrials	8.43E-6	1.06E-2	89.82	56.52
S&P 500	3.78E-5	6.99E-3	91.35	53.85
DAX	5.01E-4	3.10E-2	63.59	58.82
SSMI	3.28E-4	2.20E-3	94.87	55.56
IBM	4.73E-4	4.59E-4	89.72	56.55
Microsoft	1.39E-3	1.05E-2	67.79	51.47
13-Week Bill	4.28E-4	1.41E-3	76.56	57.63
10-Year Note	1.82E-4	1.08E-3	91.73	57.14

As we expected, the high sign accuracy are achieved by the treasury bills and the two stocks with the highest Hurst Exponent: NASDAQ and DAX. SSMI has the highest value accuracy. The prediction of NASDAQ stock market is illustrated on Figure 4.10, while Figure 4.11 depicts the results of the 10-Year Note.

One can observe the big differences between the MSE values for the training session and the MSE values reached during testing. This can be explained by the major differences between the patterns in the two datasets. Every time series has much lower values in early periods, much higher values between 1998-2000 when the whole economy was characterized by a boom. The near past was dominated by an economic recession, when suddenly the values of the economic indicators had dropped. In the present the prices are increasing as a sign of the economic grow. This can be also observed on Figure 4.9 illustrating the evolution of NASDAQ index in the last 20 years.

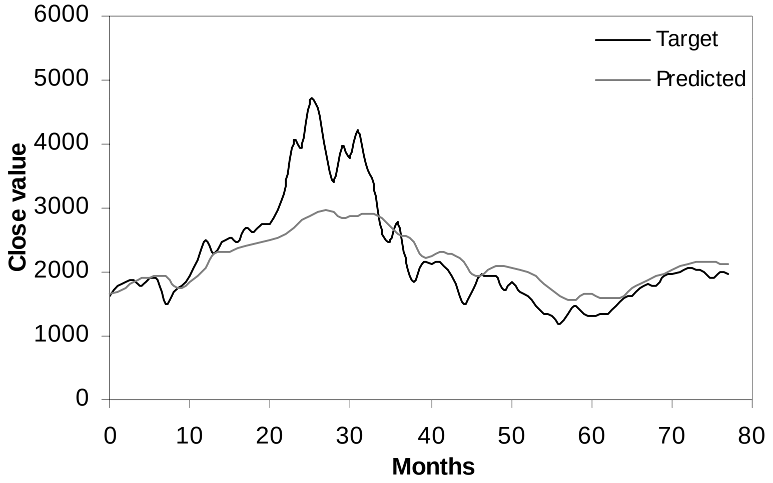

Figure 4.10. One-month ahead forecasting of NASDAQ Composite index from 1-Dec-97 to 1-Jun-04.

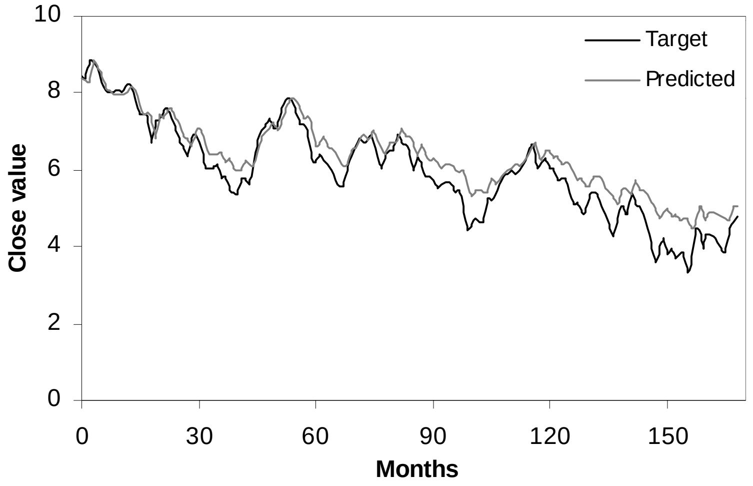

Figure 4.11. Monthly prediction of the 10-Year Note between 1-May-90 and 1-Jun-04 (170 samples).

As we have very few data for training and testing purposes we used 3000 epochs in order to train the neural network. In the case of the monthly prediction it is even more important how many and what kind of samples are used for training. In the above experiments we used two third of the data for training, but increasing the number of testing samples we can get much better results. In the case of the NASDAQ index we used 158 samples for training, and if we look at Figure 4.9 we can see that this does not includes the high deviation. If we are using 200 samples, the abrupt rise and fall is included, the NN system will have the possibility to learn this pattern in order to make a more accurate prediction. In this case we get 3.58E-4 MSE during training, 2.15E-3 MSE value for the test dataset, 88.49% accuracy in predicting the concrete index value and 68.57% accuracy in forecasting the direction of price change.

5 Conclusions, Recommendations and Future Work

Neural networks have proved to be efficient forecasting systems. Feedforward neural networks with backpropagation learning algorithm are popular and efficient methods used for pattern recognition and prediction. Financial forecasting is a challenging task, which can be tackled with multilayer perceptron based FFNNs.

Stock markets are complex, noisy environments, dominated by uncertainty and high volatility. Classical linear analysis is inefficient in capturing the complex dynamics of stock market time series. We have presented several NN based models and hybrid methods, which are capable to detect the underlying mechanics of stock markets. The implemented financial forecasting system was tested successfully for a variety of time series, ranging from stock markets, to companies' stocks and treasury bills. The performance of the system predominantly depends on the quality of data, which are used with the purpose of training the neural network. Thus, data choice and collection, as well as data preprocessing are important parts of the prediction process.

ANNs integrate various different components and parameters, which have to be selected and set optimally. Relying on previous results we used a predefined network topology, standard cost function and activation function. The various financial time series involved in experiments are publicly available. These were segregated in time order for training and testing purposes. Daily, weekly and monthly predictions were conducted. As we had much fewer weekly and especially monthly samples, the predicted future values are less accurate, while it is easier to predict the trend, the sign, i.e. the direction of price changes. The results were satisfactory, outperforming in this way several previous NN based predictors.

It is recommended to consider NNs as a powerful complement to standard econometric methods, rather than a substitute. Combining neural networks with linear regression models can result in a more efficient prediction system. Although, NN based prediction systems could be very useful and they demonstrated several times their importance in the forecasting process, several pitfalls have to be avoided. Moreover, the results should be interpreted with increased caution. As the dynamics of the stock markets are very complex, a professional forecaster should use the prediction system as a hint, and the final decision should be made based on other relevant information, as well.

As a future work it would be interesting to evolve the NN topology and connections by using genetic algorithms. Genetic algorithms are global optimization methods, which have proved to be efficient in generating the architecture of ANNs. It is also recommended to try out other error (cost) functions and activation functions.

In another important future work it is intended to construct a multivariate forecasting system, where several different financial and economic data are involved to predict the future values of a time series. We would like to use – as a complement to the stock market data, interest rate and companies' stock prices – the exchange rate of various foreign currencies as well as gold prices, which it was not possible to make in this thesis due to lack of relevant financial data.

References

1. Jarmo T. Alander: An Indexed Bibliography of Genetic Algorithms and Neural Networks, Report 94-1-NN, Department of Information Technology and Production Economics, University of Vaasa, Finland, 2001.
2. Leonidas Anastasakis, Neil Mort: Neural network-based prediction of the USD/GBP exchange rate: the utilisation of data compression techniques for input dimension reduction, in Proceedings of the Nostradamus Prediction Conference (Zlin, Czech Republic, September 2-3, 2000), 2000.
3. Thomas Ankenbrand, Marco Tomassini: Multivariate Time Series Modelling of Financial Markets with Artificial Neural Networks, in D. W. Pearson, N. C. Steele and R. F. Albrecht (Eds.), Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms – ICANNGA'95 (Alès, France, April 18-21, 1995), Springer-Verlag, Wien, New York, pp. 257-260, 1995.
4. G. Armano, M. Marchesi, A. Murru: A hybrid genetic-neural architecture for stock indexes forecasting, Information Sciences xxx, pp. xxx–xxx, 2004.
5. Filippo Castiglione: Forecasting price increments using an artificial Neural Network, Advanced Complex Systems 1, pp. 1-12, 2000.
6. Chan Man-Chung, Wong Chi-Cheong, Lam Chi-Chung: Financial Time Series Forecasting by Neural Network Using Conjugate Gradient Learning Algorithm and Multiple Linear Regression Weight Initialization, Computing in Economics and Finance 61, 2000.
7. Tim Chenoweth, Zoran Obradović: A Multi-Component Nonlinear Prediction System for the S&P 500 Index, Neurocomputing 10(3), pp. 275-290, 1996.
8. Sven F. Crone: Training Artificial Neural Networks for Time Series Prediction Using Asymmetric Cost Functions, Proceedings of FGML-Workshop (Hannover, Germany, October 9-11, 2002), 2002.
9. Birgul Egeli, Meltem Özturan, Bertan Badur: Stock Market Prediction Using Artificial Neural Networks, in the Proceedings of the 3rd International Conference on Business (Honolulu, Hawaii June 18-21, 2003), 2003.

10. Emile Fiesler, Russell Beale (Eds.): Handbook of Neural Computation, IOP Publishing Ltd and Oxford University Press, 1997.
11. Federal Reserve Economic Data, Federal Reserve Bank of St. Louis, Economic Research Division, 2004: research.stlouisfed.org/fred2.
12. generation5: At the forefront of Artificial Intelligence, 2004: www.generation5.org.
    a. www.generation5.org/nnintro.shtml, 2000.
    b. www.generation5.org/bp.shtml, 2002.
    c. www.generation5.org/nn_ga.shtml, 2000.
13. Gaston Gonnet: Stock Market Prediction, Institute for Scientific Computing, ETH Zürich, Switzerland, 2002: linneus20.ethz.ch:8080/4.html.
14. Steven Gonzalez: Neural Networks for Macroeconomic Forecasting: A Complementary Approach to Linear Regression Models, Working Paper, Department of Finance Canada, 2000.
15. S.C. Hui, M.T. Yap and P. Prakash: A Hybrid Time Lagged Network for Predicting Stock Prices, International Journal of the Computer, the Internet and Management 8(3), 2000.
16. Anil K. Jain, Jianchang Mao, K. Mohiuddin: Artificial Neural Networks: A Tutorial, IEEE Computer 29(3), pp. 31-44, 1996.
17. Efstathios Kalyvas: Using Neural Networks and Genetic Algorithms to Predict Stock Market Returns, Master Thesis, Department of Computer Science, University of Manchester, 2001.
18. Ludwig Kanzler: Very Fast and Correctly Sized Estimation of the BDS Statistic, Department of Economics, University of Oxford, Oxford, UK, 1999.
19. Steven H. Kim, Se Hak Chun: Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index, International Journal of Forecasting 14, pp. 323-337, 1998.
20. Sung-Suk Kim: Time-delay recurrent neural network for temporal correlations and prediction, Neurocomputing 20, pp. 253-263, 1998.
21. Kyoung-jae Kim: Financial time series forecasting using support vector machines, Neurocomputing 55, pp. 307-319, 2003.
22. Amol S. Kulkarni: Application of Neural Networks to Stock Market Prediction, Technical Report, 1996.
23. Leonid Kuvayev: Predicting Financial Markets with Neural Networks, Review Paper, Seminar in Capital Markets, 1996.

24. Juan G. Lazo Lazo, Marley Maria R. Vellasco, Marco Aurélio C. Pacheco: A Hybrid Genetic-Neural System for Portfolio Selection and Management, in Proceedings of the 6th International Conference on Engineering Applications of Neural Networks – EANN2000, (Kingston Upon Thames, UK, July 17-19, 2000), 2000.
25. Esfandiar Maasoumi, Jeff Racine: Entropy andpred ictability of stock market returns, Journal of Econometrics 107, pp. 291-312, 2002.
26. The MathWorks: Neural Network Toolbox for use with MATLAB, Version 4.0.2 (Release 13), 2002.
27. Peter C. McCluskey: Feedforward and Recurrent Neural Networks and Genetic Programs for Stock Market and Time Series Forecasting, Master Thesis, Department of Computer Science, Brown University, Providence, Rhode Island, 1993.
28. Thomas Meyer: Comparison of Training Periods for Stock Market Prediction, Diploma Project, Institute of Scientific Computing, Swiss Federal Institute of Technology (ETH) Zürich, 2000.
29. Hirotaka Mizuno, Michitaka Kosaka, Hiroshi Yajima: Application of Neural Network to Technical Analysis of Stock Market Prediction, Studies in Informatics and Control 7(2), pp. 111-120, 1998.
30. Karl Nygren: Stock Prediction – A Neural Network Approach, Master Thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2004.
31. F. W. Op't Landt: Stock Price Prediction using Neural Networks, Master Thesis, Leiden University, The Netherlands, 1997.
32. Paul Kang Hoh Phua, Daohua Ming, Weidong Lin: Neural Network with Genetic Algorithms for Stocks Prediction, in Paul Kang Hoh Phua, Chen Guan Wong, Dao Hua Ming, Wendy Koh (Eds.), Proceedings of the 5th Conference of the Association of Asian-Pacific Operations Research Societies (Singapore, July 5-7, 2000), 2000.
33. Yochanan Shachmurove, Dorota Witkowska: Utilizing Artificial Neural Network Model to Predict Stock Markets, Working Paper, Institute for Economic Research, University of Pennsylvania, BibEc: Working Papers in Economics, 2000.
34. Arnold F. Shapiro: Capital Market Applications of Neural Networks, Fuzzy Logic and Genetic Algorithms, in Proceedings of the 13th Annual International AFIR Colloquium, Maastricht, The Netherlands, 2003.
35. Kevin Swingler: Financial Prediction, Some Pointers, Pitfalls, and Common Errors, Technical Report, Center for Cognitive and Computational Neuroscience, Stirling University, Stirling, UK, 1994.

36. Yu Tang, Fujun Xu, Xuhui Wan and Yan-Qing Zhang: Web-based Fuzzy Neural Networks for Stock Prediction, in Proceedings of the 2nd International Workshop on Intelligent Systems Design and Application, (Atlanta, Georgia, August 7-8, 2002), Dynamic Publishers Inc, pp. 169-174, 2002.
37. Suraphan Thawornwong, David Enke: The adaptive selection of financial and economic variables for use with artificial neural networks, Neurocomputing 56, pp. 205-232, 2004.
38. Bruce Vanstone, Clarence Tan: A Survey of the Application of Soft Computing to Investment and Financial Trading, in Proceedings of the 8th Australian and New Zealand Conference on Intelligent Information Systems – ANZIIS2003, (Sydney, Australia, December 10-12, 2003), 2003.
39. Andreas Weigend: The Santa Fe Time Series Competition Data, Stanford University, 1994: www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html.
40. B. Wuthrich, V. Cho, S. Leung, D. Permunetilleke, K. Sankaran, J. Zhang, W. Lam: Daily Stock Market Forecast from Textual Web Data, in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics – SMC'98 (San Diego, CA, October 11-14, 1998), Volume 3, pp. 2720-2725, 1998.
41. Yahoo Finance, 2004: http://finance.yahoo.com.
42. Jingtao Yao, Hean-Lee Poh: Forecasting the KLSE Index Using Neural Networks, in Proceedings of the IEEE International Conference on Neural Networks – ICNN'95 (Perth, Australia, November 27 – December 1, 1995), Volume 2, pp. 1012-1017, 1995.
43. Jingtao Yao, Hean-Lee Poh, Teo Jašić: Foreign Exchange Rates Forecasting with Neural Networks, in Shunichi Amari (Ed.), Progress in Neural Information Processing: Proceedings of the International Conference on Neural Information Processing – ICONIP'96 (Hong Kong, September 24-27, 1996), Springer-Verlag New York, Volume 2, pp. 754-759, 1996.
44. Jingtao Yao, Chew Lim Tan, Hean-Lee Poh: Neural Networks for Technical Analysis: A Study on KLCI, International Journal of Theoretical and Applied Finance 2(2), pp. 221-241, 1999.
45. Marijana Zekić: Neural Network Applications in Stock Market Predictions – A Methodology Analysis, in B. Aurer, R. Logožar, Varaždin (Eds.), Proceedings of the 9th International Conference on Information and Intelligent Systems, pp. 255-263, 1998.
46. Stefan Zemke: On Developing a Financial Prediction System: Pitfalls and Possibilities, in Proceedings of the 19th International Conference on Machine Learning – ICML'02 (Sydney, Australia, July 8-12, 2002), 2002.