Chapter 11
Neural Networks and Wavelet De-Noising for Stock Trading and Prediction

Lipo Wang and Shekhar Gupta

Abstract. In this chapter, neural networks are used to predict the future stock prices and develop a suitable trading system. Wavelet analysis is used to de-noise the time series and the results are compared with the raw time series prediction without wavelet de-noising. Standard and Poor 500 (S&P 500) is used in experiments. We use a gradual data sub-sampling technique, i.e., training the network mostly with recent data, but without neglecting past data. In addition, effects of NASDAQ 100 are studied on prediction of S&P 500. A daily trading strategy is employed to buy/sell according to the predicted prices and to calculate the directional efficiency and the rate of returns for different periods. There are numerous exchange traded funds (ETF's), which attempt to replicate the performance of S&P 500 by holding the same stocks in the same proportions as the index, and therefore, giving the same percentage returns as S&P 500. Therefore, this study can be used to help invest in any of the various ETFs, which replicates the performance of S&P 500. The experimental results show that neural networks, with appropriate training and input data, can be used to achieve high profits by investing in ETFs based on S&P 500.

1 Introduction

Stock prices are highly dynamic and bear a non-linear relationship with many variables such as time, crude oil prices, exchange rates, interest rates, as well as factors like political and economic climate. Hence stock prices are very hard to model by even the best financial models. Future stock prices can be studied merely by historical prices.

Lipo Wang · Shekhar Gupta
School of Electrical and Electronic Engineering
Nanyang Technological University
Block S1, 50 Nanyang Avenue,
Singapore 639798
e-mail: elpwang@ntu.edu.sg

W. Pedrycz & S.-M. Chen (Eds.): Time Series Analysis, Model. & Applications, ISRL 47, pp. 229–247.
DOI: 10.1007/978-3-642-33439-9_11

With the globalization and ease of investment in international and national markets, many people are looking towards stock markets for gaining higher profits. There is a high degree of uncertainty in the stock prices, which makes it difficult for the investors to predict price movements.

Hence the study of prediction of stock prices has become very important for financial analysts as well as the general public, so as to gain high profits and reduce investment risk. With vast investments in the equity market, there has been a huge motivation for a system which can predict future prices. The Efficient Market Hypothesis (EMH) [1] states that no information can be used to predict the stock market in such a way as to earn greater profits from the stock market. There have been studies to show the accountability of the EMH [2], but some later studies have implied otherwise [3]. There are various views that oppose the EMH and indicate the predictability of some stock markets.

Stock prediction methods may be categorized into Fundamental Analysis and Technical Analysis [1]. Specific modeling techniques include multivariate regression [2] and artificial neural networks (ANN) [3,4]. This chapter is concerned only with ANN approaches. Rodrigues [5] used a relatively simple neural network to predict and trade in the Madrid Stock Market Index. Rodrigues [5] used nine lagged inputs to predict the prices and make buy/sell decisions, which gave evidence that ANN is a superior strategy to predict the index prices as compared to various other analyses. Although this model did not perform well in a bullish market.

Due to the ability of ANNs to form a complex model between training inputs and targets values, ANNs give an opportunity to model highly complex and dynamic relation in the stock prices [6,7]. There are many areas where the neural networks have been used, e.g., signal processing, speech recognition, control, and many types of neural networks have been created [5,6,24]. According to Chang et al [8], ANNs are believed to have limitations due to noise in and complexity of stock prices.

Neural network prediction systems can be divided into 2 categories, i.e., using (1) the past prices of the index and (2) fundamental data, such as exchange rates, gold prices, interest rates etc. [22-26]. For example, in the first category, [9] developed a neural network model based on past prices by using three neural networks. They were able to obtain around 16% returns per annum with a weekly prediction model, but the ANNs failed in daily prediction models. In the second category, ANNs in [10] performed much better compared to traditional stock valuation systems using financial models.

This chapter describes some of our attempts to successfully predict S&P 500 index [11], which is then developed into a trading model, in order to achieve relatively high rates of return over a long period of time. This involves the use of ANNs and wavelet de-noising in the input time series.

2 Artificial Neural Networks

Artificial neural networks attempt to mimic the biological counterparts. A neural network is composed of a number of interconnected processing elements (neurons) working in unison to solve specific problems. Learning involves adjustments to the synaptic connections between the neurons, as well as other parameters, such as the biases in neurons [5,6].

In a feed-forward neural network or a multilayer perceptron, there can be 3 layers, i.e., an input layer, an output layer, and a hidden layer. The number of hidden layers, as well as the number of neurons in each layer, can vary according to given requirements. A simple 3-layer neural network is shown below:

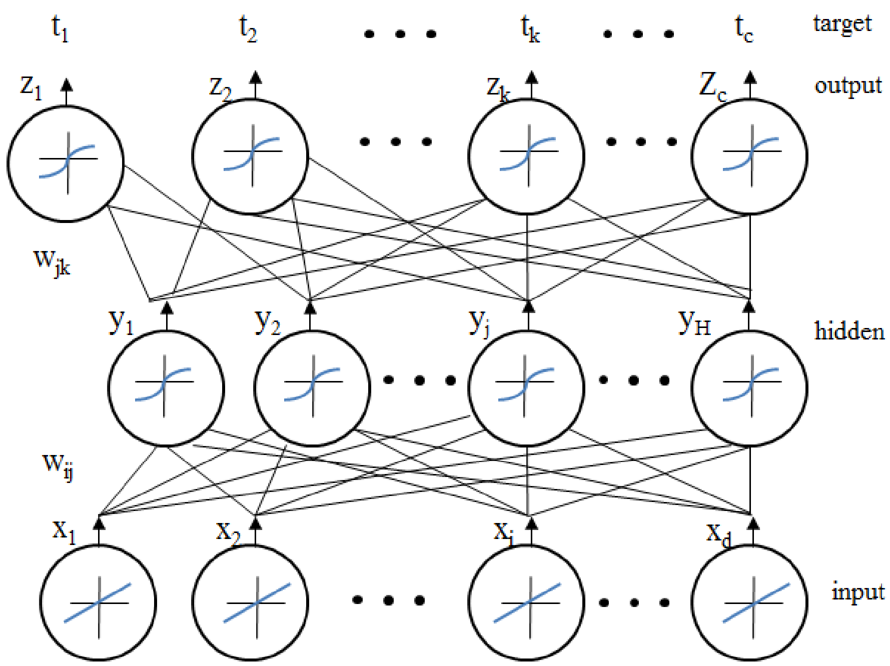

Fig. 1 A simple 3-layer neural network

The architecture of a network consists of the number of layers, the number of neurons in each layer, each layer's transfer function, and the weights by which the layers connect to each other. The best architecture to use depends on the type of problem to be solved by the network [5,6,12].

There can be two training methodologies used in the training of the neural networks, namely, incremental (online) training and batch training. In incremental training, the weights and biases are updated after each input is presented, while in batch training, weights and biases are updated after all of the inputs and targets are presented.

The inputs to a neuron include its bias and the sum of its weighted inputs. The output o_i of neuron i depends on the neuron's inputs and on the transfer function:

$$f(x) = \frac{1}{1 + \exp(-\beta x)} \tag{1}$$

$$o_j = f(\sum_i w_{ji} o_i + \theta_j) \tag{2}$$

where β is the gain, θ_j is the bias of neuron j, and w_{ji} is the connection weight from neuron i to neuron j. There are many training algorithms and different network structures which can be selected according the problem requirements. In our work, we use the Levenberg-Marquardt training algorithm. Sometimes, over-training of the network can lead to poor generalization. To prevent this from happening, a technique called early stopping is used with the help of validation data.

3 Wavelet Analysis and De-noising

Wavelets are mathematical functions used to decompose a given function or continuous-time signal into components of different scales. Usually one can assign a frequency range to each scale component. A wavelet transform is the representation of a function by wavelets. The wavelets are scaled and translated copies (known as "daughter wavelets") of a finite-length or fast-decaying oscillating waveform (known as the "mother wavelet"). Wavelet transforms have advantages over traditional Fourier transforms for representing functions that have discontinuities and sharp peaks, and for accurately deconstructing and reconstructing finite, non-periodic and/or non-stationary signals [13,14].

There are two types of wavelet transform, i.e., Discrete Wavelet Transform (DWT) and Continuous Wavelet Transform (CWT). In CWT, during computation, the analyzing wavelet is shifted smoothly over the full domain of the analyzed function [12]. Calculating wavelet coefficients at every possible scale can be very tedious and data generated can be difficult to analyze. By choosing the scales by the power of two, the analysis can become more accurate and faster. Hence, we use the DWT in this study.

There are numerous types of wavelets available for different types of series being analyzed. The Daubechies (db) wavelet is used in our analysis for the reduction of noise in the time series, which are later fed as an input to the neural network for training.

There have been research efforts on wavelet neural networks, in which wavelets are used as neuron transfer functions. But we will restrict ourselves to studying the effect of de-noising by wavelets and then feeding the reconstructed de-noised signal into the ANN, instead of training the neural network with wavelet coefficients [5,15].

4 Data and Experiments

This study attempts to predict future index prices, solely on the basis of past index prices, along with effects of NASDAQ 100 index on the prices of S&P 500. Another aspect of this chapter is to study the effect of wavelet de-noising on the raw S&P 500 time series. The number of inputs ranges from 10 day lagged values to 40 day lagged values of the S&P 500 index closing prices.

The main source of historical index prices is from the Yahoo Finance. We download 2 sets of data to study the effect and relevance of historical prices. The first set of data involves the closing prices of S&P 500 index from 9 January 1950 to 15 January 2010. The second set of data involves the closing prices of S&P 500 and NASDAQ 100 index from 7 January 1991 to 15 January 2010.

Due to public holidays, there are data missing on various days in the raw time series, which need to be adjusted to account for the missing values. A 5-day lagged average is used to fill in the missing data:

$$Missing\ Value\ (x_t) = \frac{x_{t-1} + x_{t-2} + x_{t-3} + x_{t-4} + x_{t-5}}{5} \tag{3}$$

The raw data are divided into 3 parts, i.e., for training, validation, and testing.

a. Wavelet De-noising

The wavelet toolbox of MATLAB is used to de-noise the raw S&P 500 time series. The first set of time series, i.e., from 9 January 1950 to 15 January 2010 is fed into the Signal Extension toolbox for the wavelet analysis, in order to make it compatible for stationary wavelet analysis (SWT) and de-noising. The data are then fed into the SWT De-noising 1-D toolbox, where the signal is decomposed by a db wavelet at Level 5. The de-noising tool is used to remove the white noise, with different thresholds for each level of decomposition and reconstruction of the de-noised signal. This signal is then divided into matrices of suitable sizes for training, validation and testing of the neural network.

b. Neural Network Architecture and Training

The feedforward back propagation neural network is used in this chapter, with a Levenberg-Marquardt training algorithm. The performance criterion used is the mean square error (MSE), with the TANSIG transfer function in MATLAB.

The number of hidden layers varies from 1 to 2, with one input layer and one output layer. The number of neurons in the input layer varies from 10 to 40, with a search interval 5, that is, we try out 10, 15, 20, …, and 40 input neurons, in order to find the optimal number of input neurons. The output layer consists of only 1 neuron which is the predicted price on the next working day. The number of neurons in the hidden layers varies according to the number of neurons in the input layer.

After creating the neural network, the raw time series is divided into 3 sets of data, i.e., training, validation and testing data. These data sets in the form of p*1 matrices are divided into n*m matrices, where n is the number of inputs in the neural network and m is the number of training sets in batch training. For training, p = 15,040, n = 10, 15, 20, or 40, and m =15,000.

The training parameters are now selected. Since the Levenberg-Marquardt algorithm is a fast learning algorithm, the maximum number of epochs is limited to 100. The minimum gradient is selected to be 1e-6.

The validation data are an n*t matrix, where n = 10, 15, 20 or 40 (same as the training data) and t can vary according to the number of validations required, in our case, from 200 to 400, depending on the size of the training data. The validation is used for early stopping for the network, so as to prevent over fitting of the neural network and maintain its generalization. The testing for the trained network is done over different periods of time, ranging from 1 year to 2 years, i.e., from 250 to 420 prediction points. The input test matrix is of the size n*u, where n = 10, 15, 20 or 40 (same as that of the training and validation matrices, i.e., the number of inputs) and u varies from 250 to 420, i.e., the number of testing sets in the batch. The target test matrix is of the size 1*u.

c. Neural Network Simulations of Test Data

After the training of the neural network, the test data are used to generate the predicted outputs which are compared with the actual values. The test is done over various periods of time, ranging from 1 to 2 years, and with different market conditions, i.e., before, during, and after recessions.

d. Trading System

The predicted outputs during testing are exported into a trading system, to obtain the directional efficiency and rate of returns for the specified period of time. The trading system is developed for daily trading. When the predicted price for the next day is less than today's price, a sell decision is made. And when the predicted price for the next day is more than today's price, then a buy decision is made. Based on these trading rules, the rate of returns is calculated over a period of time.

Below is an example of the results of the trading system, which are shown in the cells G4:I6, i.e., directional efficiency, return, and the rate of return for the specified period of time. These results are calculated for different neural networks, with varying inputs, hidden layers, training data and are then compared in the next section.

	A	B	C	D	E	F	G	H	I
1	Actual	Predicted							
2	1385.7	1384.2			1.5				
3	1377.7	1387.2	Buy	Wrong	-9.5				
4	1377.2	1373.3	Sell	Correct	3.9		Efficiency	52.05993	%
5	1404.1	1375.1	Sell	Wrong	29		Return	322.5	
6	1360.7	1401.8	Sell	Correct	-41.1		Rate of Return	35.80443	%
7	1361.8	1365.6	Buy	Correct	-3.8				
8	1358.4	1358.7	Sell	Correct	-0.3				
9	1335.5	1354.2	Sell	Correct	-18.7				
10	1339.9	1331.3	Sell	Wrong	8.6				

Fig. 2 An example output of the trading system

5 Results and Discussions

We carry out various experiments to study the future index prices of S&P 500, i.e.,

a) effects of wavelet de-noising;
b) effects of gradual sub-sampling the past data;
c) effects of NASDAQ 100 on the S&P 500 index.

We will first discuss the effects of the above mentioned factors in detail, followed by selecting the best results based on the returns and risk for a long period of time.

The data set has been divided into 2 sets, i.e.,

a) Data Set 1: January 1950 – January 2010
b) Data Set 2: January 1991 – January 2010

a. Effects of Wavelet De-Noising

We now use the 1st set of data, i.e., the data from January 1950 till January 2010, for training and testing. The training and validation data are selected from January 1950 till April 2008. Then testing is done for 2 time intervals, i.e.,

a) Period 1: May 2008 till January 2010 (1.5 years)
b) Period 2: January 2009 till January 2010 (1 year).

This experiment is done with and without wavelet de-noising and the results with the maximum efficiency are shown as below:

Table 1 Results without wavelet de-noising in training from 1950 – 2008

Network Structure	10-10-1		10-20-1	
Period	1	2	1	2
Rate of Return	6.3%	26.5 %	5.5%	25.2%
Directional Efficiency	51.5%	53.4%	51.2%	52.3%

Table 2 Results with wavelet de-noising in training from 1950 – 2008

Network Structure	10-10-1		10-21-1	
Period	1	2	1	2
Rate of Return	5.2%	23.9%	4.4%	21.1%
Directional Efficiency	51.1%	52.3%	50.1%	50.4%

From the above results, it can be seen that the effect of wavelet de-noising is not satisfactory, as compared to the training of the neural networks by the raw time series. This may indicate that there exists minimum noise in the raw financial data.

b. Effects of Gradual Data Sub-Sampling

An experiment is carried out with a novel technique of gradual data sub-sampling [22], whereby data of historically distant past are given less significance and recent data are given more significance. This technique is applied on the 1st data set, i.e., from January 1950 till April 2008, during training. Originally there are 15,200 points of training data, which are reduced to 6,700 in the following way:

a) 900 training data are selected from the 1st 5,400 data;
b) 2,000 training data are selected from the next 6,000 data;
c) all the remaining 3,800 data are selected.

This technique ensures that the historical trends are not ignored and, at the same time, the system is more related to the current market situations. Thus the number of lagged values also varies to show the effect of number of inputs as well as wavelet de-noising. Again, the testing is done for Period 1 and Period 2 stated above. The most efficient results are shown below:

Table 3 Results without wavelet de-noising with gradual data sub-sampling

Network Structure	10-5-1		15-26-1		20-22-1	
Period	1	2	1	2	1	2
Rate of Return	8.9%	34.4%	15.3%	44.7%	3.5%	12.7%
Directional efficiency	52.6%	50.0%	53.9%	52.8%	52.2%	51.6%

Table 4 Results with wavelet de-noising with gradual data sub-sampling

Network Structure	10-18-1		15-21-1		20-24-1	
Period	1	2	1	2	1	2
Rate of Return	6.0%	36.5%	1.6%	21.5%	9.5%	5.3%
Directional efficiency	51.5%	53.2%	50.4%	49.6%	50.9%	47.4%

Again from the above results, it can be seen that wavelet de-noising is not effective in predicting the future index prices. After obtaining these results, we decide not to include wavelet de-noising in further experiments in this chapter, since the results are much better with the raw S&P 500 time series. This may mean that the financial data used here are not noisy.

Our results also show an increase in the efficiency with gradual data sub-sampling. A comparison between the original data series and the series with gradual data sub-sampling is shown in Table 5:

Table 5 Effect of gradual data sub-sampling without wavelet de-noising

Network Structure	10-10-1 (Original data)		10-5-1(Gradual Data Sub-sampling)	
Period	1	2	1	2
Rate of Return	6.3 %	26.5 %	8.9%	34.3%
Directional efficiency	51.5%	53.4%	52.6%	50.0%

It is evident from Table 5 that the rates of return with gradual data sub-sampling are much better than those with the original time series, for both Periods 1 and 2.

We now discuss results for data set 2, i.e., the data from January 1991 till January 2010, with data and without data sub-sampling. The testing period is kept the same as the previous section, so as to compare all the models under the same market conditions.

In this approach the training and validation of the neural network are done with data sets from January 1991 till April 2008. Again, testing is done for Period 1 and Period 2 stated above. The results are shown in Table 6:

Table 6 Results with original data (no sub-sampling) from 1991 – 2010

	Period 1		Period 2	
Network	Rate of Return	Directional efficiency	Rate of Return	Directional efficiency
40-16-4-1:	11.7%	55.2 %	53.7 %	58.7 %
40-120-80-25-1:	14.8 %	51.9 %	36.3 %	49.6 %
40-20-1:	17.8%	54.8%	30.1%	52.4 %
40-13-1:	19.8 %	54.0 %	32.8 %	53.6 %

We now describe the results for the data set 2, i.e., from January 1991 till January 2010, with gradual data sub-sampling. Originally there are 4,500 data sets for training and validation, which are reduced to 2,900 data sets as follows:

a) 800 data sets are selected from the first 2,400 data;
b) all the remaining 2,100 data sets are selected.

The above training set is used to train the network with different lagged inputs. Again, the testing is done for Period 1 and Period 2 stated previously. The most efficient results are shown below:

Table 7 Results with gradual data sub-sampling for Data Set 2

Network Structure	40-22-1		30-30-1		10-22-1	
Period	1	2	1	2	1	2
Rate of Return	36.0%	47.2%	24.5%	45.6%	24.5%	30.7%
Directional efficiency	55.7%	55.6%	55.2%	56.3%	53.3%	51.6%

Table 7 shows that the results with 40 lagged input values are the best.

Now we will further reduce the historical data and give even more preference to the more recent data, by reducing the training and validation inputs from 2,900 to 1,800 data sets. The most efficient results are shown in the table below:

Table 8 Results with further sub-sampling for Data Set 2

Network Structure	10-9-1	
Period	1	2
Rate of Return	47.2%	44.1%
Directional efficiency	57.6%	54.8%

The results in the above table are quite good.

We compare the results of training with the original data, sub-sampled data, and further sub-sampled data in the following table:

Table 9 Comparison between various training data sets

Network Structure	40-13-1 (Original data)		40-22-1 (Data Sub-sampling)		10-9-1 (Further Sub-sampling)	
Period	1	2	1	2	1	2
Rate of Return	19.8%	32.8%	36.0%	47.20 %	47.2%	44.1%
Directional efficiency	54.0 %	53.6%	55.7%	55.6 %	57.6%	54.8%

It can be seen that the effect of gradual data sub-sampling has a great influence on the directional efficiency and the rate of return. With reducing the training and validation data set from 4,500 to 1,800, the rate of returns increased from 19.8% to 47.2% for period 1 and from 32.8% to 44.1% for period 2. Hence, the effect of data sub-sampling is of major importance.

c. Effects of NASDAQ 100 Index

In this section the training data also include the NASDAQ 100 lagged index values along with the lagged values of S&P 500 index (Data Set 2, i.e., from January 1991 till January 2010). 4500 data points are used for training and validation, and there are 20 inputs to the neural network, i.e., 10 lagged values each for S&P 500 and NASDAQ 100. The testing is again done for Period 1 and Period 2 stated previously. The most efficient results are shown below:

Table 10 Results with effect of NASDAQ 100

Network Structure	20-7-1	
Period	1	2
Rate of Return	34.8%	72.4%
Directional efficiency	57.9%	59.6%

The above table shows that NASDAQ 100 data can be used to predict the S&P 500 index prices and these results are comparable to the effect of gradual data sub-sampling.

In the next section, we will compare the best results from each of the above sections and discuss which model would be the best suited for prediction of S&P 500 index, and therefore, making the maximum profit in the long run. Finally, we make comparisons of our results with other published work and discuss the feasibility of our model for investment.

d Discussions of Various Models for Prediction

Now we will discuss about the best results from the above 3 sections and compare them. Below is the table which consists of the best results from each section. The comparison is between:

a) The results obtained after the wavelet de-noising of the original data from January 1950 till January 2010.
b) The results obtained after the further data sub-sampling for the data set from January 1991 till January 2010.
c) The results obtained by including the NASDAQ 100 index to study the index movements for S&P 500, for data set from January 1991 till January 2010.

Table 11 Comparisons between Wavelet, Data sub-sampling and NASDAQ 100

Network Structure	10-10-1 (Wavelet)		10-9-1 (Further Data Sub-sampling)		20-7-1 (NASDAQ 100)	
Period	1	2	1	2	1	2
Rate of Return	5.2%	23.9%	47.2%	44.1%	34.8%	72.4 %
Directional Efficiency	51.1%	52.3%	57.6%	54.8%	57.9%	59.6 %

From the above comparisons, we draw the following conclusions:

a) Wavelet de-noising does not help much to predict the future index prices.
b) The technique of gradual data sub-sampling has been proved to be effective in prediction of index values. 10 lagged values as inputs are optimal.
c) NASDAQ 100 index has been proved to be effective for the prediction of S&P 500 index.

Period 1 (May 2008 till January 2010) used in testing includes the time period when the economies were in and coming out of recession, and there was a very slow improvement in the overall index movements. Period 2 (January 2009 till January 2010) includes the time when the economy was recovering and coming out of the recession, and when the increase in index was relatively stronger than that in Period 1.

Since the wavelet de-noising has been proved to be an ineffective technique to train the neural network with de-noised inputs, we will compare the results of data sub-sampling and NASDAQ 100 effects for the above mentioned 2 prediction periods.

The rate of return is as high as 47% for the 1^{st} period and 44% for the 2^{nd} period for the data sub-sampling technique, and around 34% and 72%, respectively, for the 2 periods with NASDAQ 100 index. This shows that the technique of data sub-sampling can provide much more stable results over a long period of time, in different markets conditions.

On the other hand, the technique of including S&P 500 index with NASDAQ 100 also leads to good results, which varies significantly for the 2 periods of study. The returns are around 34% for Period 1 and 72% for Period 2. This shows that this model is very effective to predict prices in good market conditions and leads to high returns, but is not very effective to deliver stable returns over a long period of time.

Table 12 Risk and return comparisons of the 2 models

	Returns	Risk
Model 1: Data Sub-sampling	Stable Returns	Low Risk
Model 2: NASDAQ 100	High Returns	High Risk

Hence an investor who would like to take less risk and want steady returns would consider using the gradual data sub-sampling technique, while an investor who wants enormously high returns can invest using the model with NASDAQ 100 index, if he thinks that the market conditions are relatively good.

Table 13 Best models for investment

Network Structure	10-9-1 (Further Data Sub-sampling)		20-7-1 (NASDAQ 100)	
Period	1	2	1	2
Rate of Return	47.2%	44.1%	34.8%	72.4%
Directional Efficiency	57.6%	54.8%	57.9%	59.6%

Next we show the actual vs. predicted prices for both the models. Both figures indicate that the actual and predicted prices coincide with each other quite well, which also makes it evident that these 2 model are able to predict well and provide good returns.

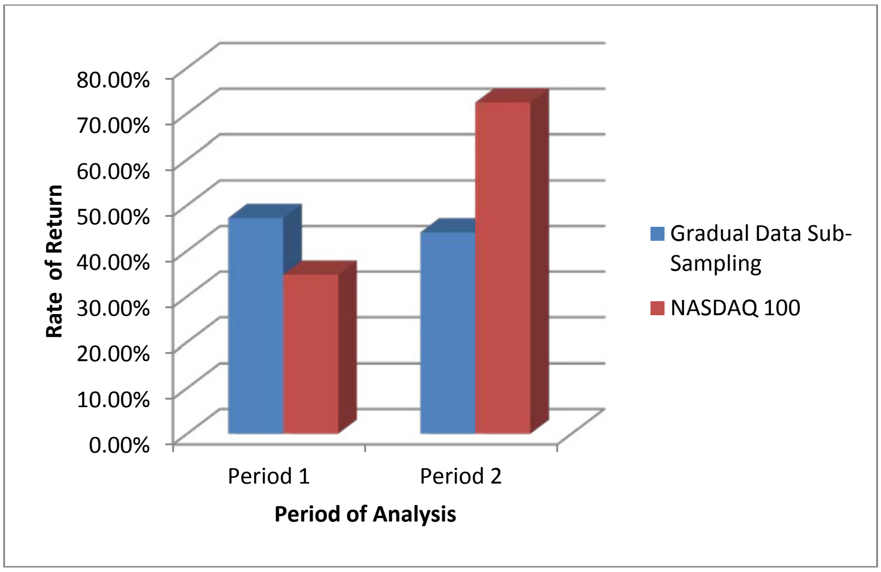

Fig. 3 Comparison of rate of returns for Period 1 and Period 2

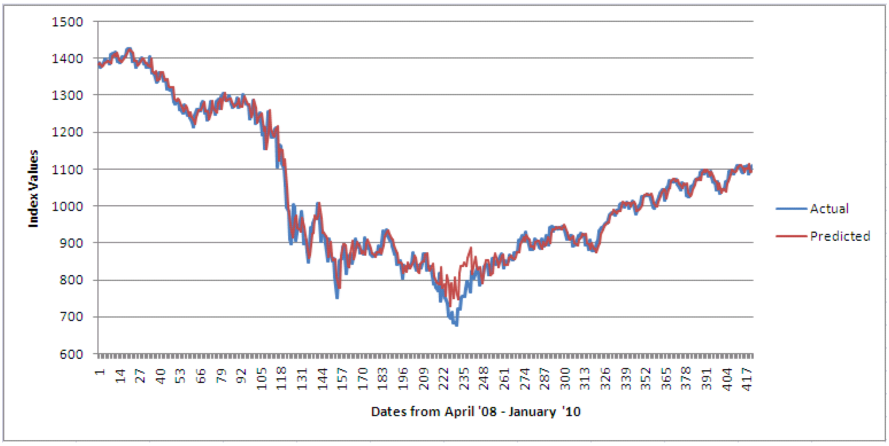

Fig. 4 Actual vs predicted values for gradual data sub-sampling (10-9-1)

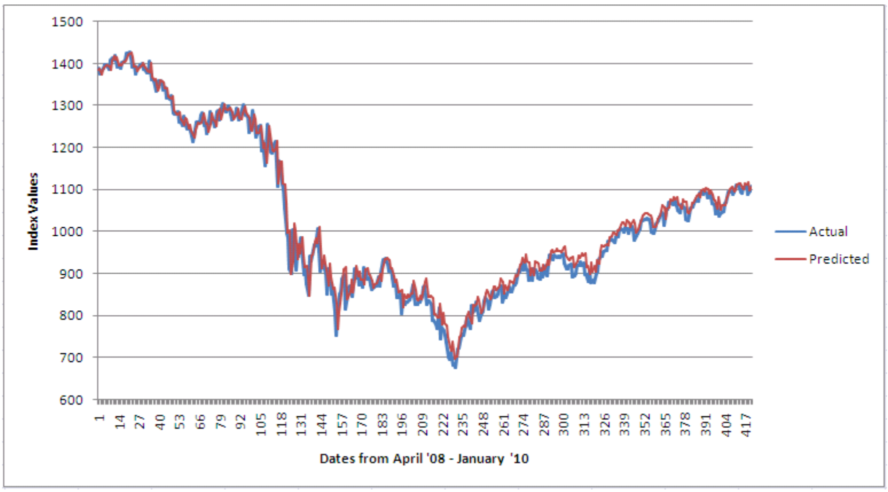

Fig. 5 Actual vs. predicted values with NASDAQ 100

e. Comparisons with Other Papers

Another set of experiments is done to compare the results of the gradual data sub-sampling technique and effects of NASDAQ 100 on S&P 500 with various other published papers.

For this part, the 2 neural networks with the best efficiencies from the above sub-sections are used, i.e.:

a) A network trained with the gradual data sub-sampling technique with 10 lagged values.
b) A network trained with the inclusion of both the NASDAQ 100 index and S&P 500 index, 10 lagged values each.

The results of our research are compared with 3 sources, i.e.,

a) Trading Solutions [16], an online website for a trading software package.
b) A model integrating a piecewise linear representation method and a neural network [18].
c) A system of 3 neural networks (3-ANN) [9], one network each for bullish, bearish and choppy markets.

The period of comparison is from January 2004 till December 2004. The results are as follows:

Table 14 Comparisons of rate of return by various techniques for January - December 2004

Technique	Trading Solutions [16]	IPLR [18]	3 – ANN System [9]	Gradual Data Sub-sampling	Effect of NASDAQ
Rate of Return	11.0%	35.7%	18.4%	25.4%	16.3%

It can be seen that the Trading Solutions, the 3-ANN system and the effect of NASDAQ are not very effective for this period. The rate of return for this period is the highest for the IPLR system, followed by the new technique of data sub-sampling.

We now show the rates of return for our gradual data sub-sampling for a number of periods:

Table 15 Rates of return with gradual data sub-sampling for different periods

Time of Testing	January 2004 – December 2004	April 2008 - January 2010	January 2009 – January 2010
Rate of Return	25.4%	47.2%	44.1%

Although the rate of return is not very high with our technique of gradual data sub-sampling for the period of January – December 2004, but for the period from April 2008 till January 2010, the rate of return is as high as 47.2%. Data is not available for us to comment on the rate of return from the IPLR technique for the period from April 2008 till January 2010. But we can still compare the situation of the markets in the two periods.

Below are the rates of return with the effect of NASDAQ 100:

Table 16 Rate of returns with effect of NASDAQ for different periods

Time Period	January 2004 – December 2004	April 2008 - January 2010	January 2009 – January 2010
Rate of Return	16.3%	34.8%	72.4%

Below are 3 graphs for the trend of S&P 500 index for the 3 periods:

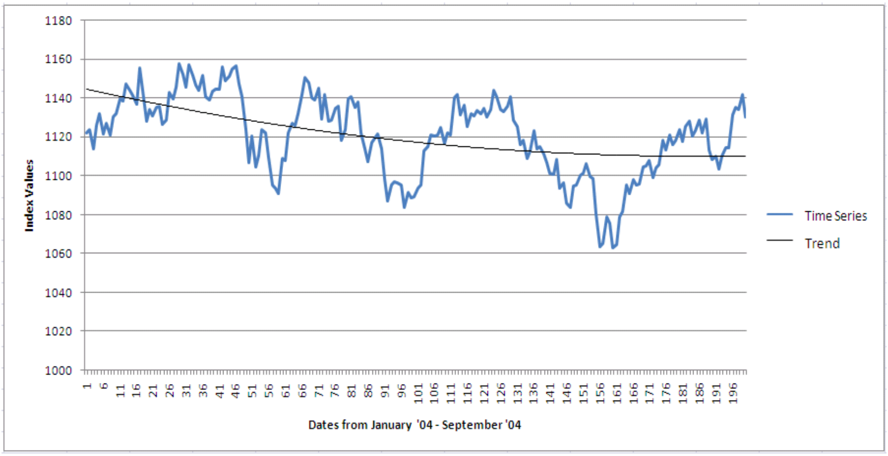

Fig. 6 Trend for the period from January - September 2004

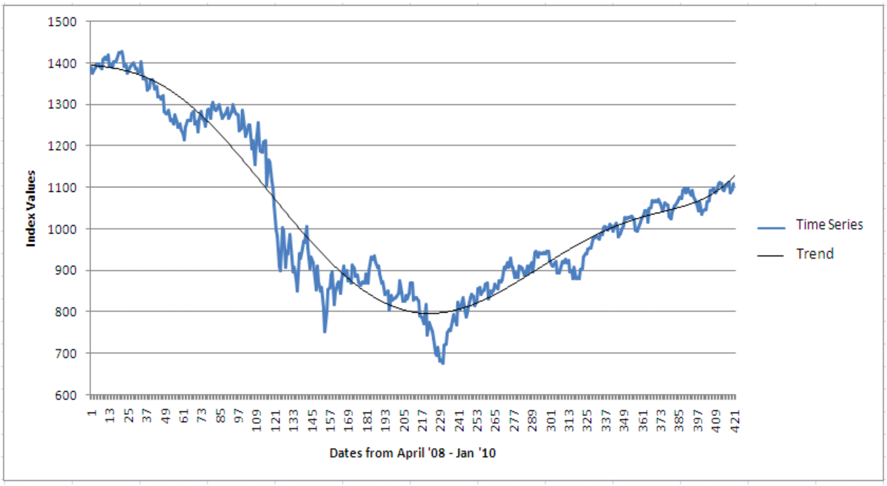

Fig. 7 Trend for the period from April 2008 - January 2010

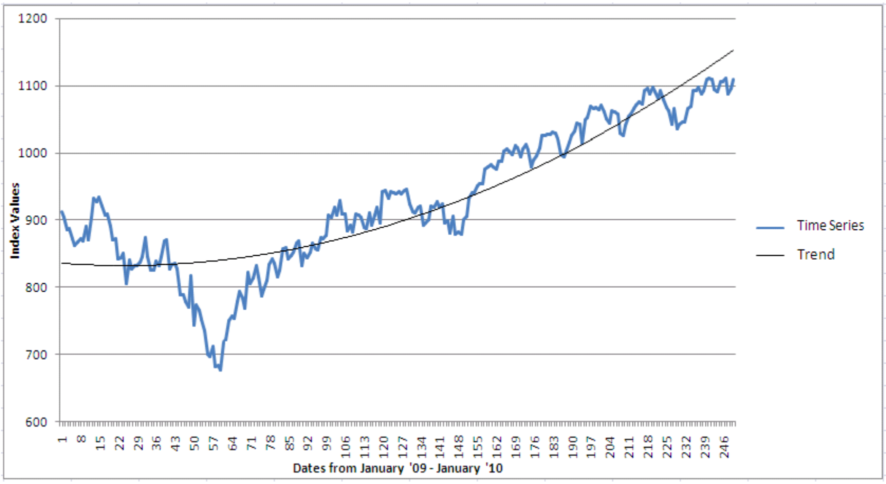

Fig. 8 Trend for the period from January 2009 - January 2010

From the above 3 figures, it can be concluded that:

a) January 2004 – September 2004 was a relatively stable period.
b) April 2008 – January 2010 involved a huge dip in price index (recession) followed by a recovery.
c) January 2009 – January 2010 had an increasing trend.

Table 17 Trading models vs. investor risk profile

Model / Risk Profile	Data Sub-sampling (Moderate to High and Stable Returns)	NASDAQ 100 (Stable Returns, High Returns in bullish market)	IPLR (High Returns in Stable Markets, Unknown Returns in other market conditions)
Low	✓		
Moderate		✓	
High			✓

From the analysis above, it can be seen that our system with the innovative technique of gradual data sub-sampling is relatively stable and efficient in delivering good rates of return in all market situations and high rates of return in special market situations involving more movement in the index values. Also, the effect of NASDAQ is suitable to deliver high rates of return, i.e., around 72.4%, in good market conditions.

5 Conclusions and Discussions

It is evident from this chapter that feed forward back propagation neural networks are useful in predicting the future index prices and the research also proves that suitable network structure and appropriate data sub-sampling could lead to maximization of rates of return and hence deliver large profits to investors.

Historical data of the index, as well as historical data of other indices like NASDAQ 100, along with the innovative technique of data sub-sampling, are helpful in determining future index prices.

Trial-and-error was used at various stages of our experiments to select the most suitable network architecture with optimum number of hidden layers, lagged values, and neurons.

This trading system could be used by various financial products, such as Exchange Traded Funds (ETFs), which replicate the performance of the S&P 500 index, thereby giving investors an opportunity to invest following the S&P 500 trends. We emphasize that neural networks in this chapter are not used to get the exact future index prices, but is used to determine the directional trend of the index.

Finally, this study shows that the technique of gradual data sub-sampling and the effect of NASDAQ 100 on the prediction of S&P 500 index prove to be very beneficial, leading to high rates of return.

There are rooms for improvement and further research. We have studied here the effect of only NASDAQ 100 on S&P 500 prediction. Various other factors, such as crude oil prices, gold prices, other indices, exchange rates, interest rates, etc., could be used to study their impact on the price movements of S&P 500.

The feed forward back propagation neural network has been used in this chapter. There are various other types of neural networks, e.g., the cascaded back propagation neural networks, radial basis function, etc., which could be used to study the index prices.

The gradual data sub-sampling technique can be studied together with other systems, such as IPLR. A real time trading system could be developed, in which prices are downloaded live from the server. These prices are then fed into the system to predict prices at various intervals in a particular day, which could further be used for intra-day trading or high frequency trading.

References

1. Malkiel, B.G.: A Random Walk Down Wall Street. W. W. Norton & Company, London (1999)
2. Fama, E.F., Fisher, L., Jensen, M., Roll, R.: The Adjustment of Stock Market Prices to New Information. International Economic Review X, 1–21 (1969)
3. Thaler, R.: Does the Stock Market Overreact? Journal of Finance 40(3), 793–805 (1985)
4. Yahoo Finance, `http://finance.yahoo.com`
5. Wang, L.P., Fu, X.J.: Data Mining with Computational Intelligence. Springer, Berlin (2005)

6. Zurada, J.M.: Introduction to Artificial Neural Systems. Jaico Books, Mumbai (1992)
7. MATLAB Version 7.0.1 Help Manual, The MathWorks Inc.
8. Azoff, E.M.: Neural Network Time Series Forecasting of Financial Markets. John Wiley & Sons, New York (1994)
9. Haddad, A.N., Akansu, R.A.: Multiresolution Signal Decomposition: Transforms, Subbands, Wavelets. Academic Press (1992)
10. Kaiser, G.: A Friendly Guide to Wavelets. Birkhäuser (1994)
11. Tak, B.: A New Method for Forecasting Stock Prices using Artificial neural network and Wavelet Theory. UMI Dissertation Services (1995)
12. Timmermann, M., Pesaran, H.: Forecasting stock returns: An examination of stock market trading in the presence of transaction. Journal of Forecasting 13, 335–367 (1994)
13. Chang, P.C., Wang, Y.W., Yang, W.N.: An investigation of the hybrid forecasting models for stock price variation in Taiwan 21(4), 358–368 (2004)
14. Chi, S.C., Chen, H.P., Cheng, C.H.: A forecasting approach for stock index future using grey theory and neural networks. In: Proc. IEEE Int. Joint. Conf. Neural Netw., vol. 6, pp. 3850–3855 (1999)
15. Femdndez-Rodriguez, F., Gonzdlez-Martel, C., Sosviall-Rivero, S.: On the Profitability of Technical Trading Rules based on Artificial neural networks: Evidence from the Madrid Stock Market. Economics Letter 69(1), 89–94 (2000)
16. Yoo, P.D., Kim, M.H., Jan, T.: Financial Forecasting: Advanced Machine Learning Techniques in Stock Analysis. In: Proceedings of the 9th IEEE International Conference on Multitopic, pp. 1–7 (2005)
17. Siddique, M.N.H., Tokhi, M.O.: Training neural networks: Backpropagation vs Genetic Algorithms. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN, pp. 2673–2678 (2001)
18. Chang, P.-C., Fan, C.-Y., Liu, C.-H.: Integrating a Piecewise Linear Representation Method and a neural network Model for Stock Trading Points Prediction. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 39, 80–92 (2009)
19. Fu, C.: Predicting the Stock Market using Multiple Models. In: ICARCV (2006)
20. Chenoweth, T., Obradović, Z.: A Multi-Component Nonlinear Prediction System for the S & P 500 Index. Neurocomputing 10, 275–290 (1996)
21. NeuroDimension, Inc., Trading Solutions (2008), `http://www.tradingsolutions.com/`
22. Gupta, S., Wang, L.P.: Stock Forecasting with Feedforward Neural Networks and Gradual Data Sub-Sampling. Australian Journal of Intelligent Information Processing Systems 11, 14–17 (2010)
23. Zhu, M., Wang, L.P.: Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–5 (2010)
24. Wang, L.P. (ed.): Support Vector Machines: Theory and Application. Springer, Berlin (2005)
25. Rather, A.M.: A prediction based approach for stock returns using autoregressive neural networks. In: 2011 World Congress on Information and Communication Technologies (WICT), pp. 1271–1275 (2011)
26. de Oliveira, F.A., Zarate, L.E., de Azevedo Reis, M., Nobre, C.N.: The use of artificial neural networks in the analysis and prediction of stock prices. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2151–2155 (2011)