PRICE PREDICTION IN IMKB USING NEURAL NETWORKS

MBA THESIS

SİNAN ALTUĞ

ANKARA, JUNE 1994

PRICE PREDICTION IN IMKB USING NEURAL NETWORKS

A THESIS SUBMITTED TO

THE FACULTY OF MANAGEMENT

AND

THE GRADUATE SCHOOL OF BUSINESS ADMINISTRATION

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF BUSINESS ADMINISTRATION

BY

SINAN ALTUG

JUNE, 1994

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Assoc. Prof. Gulnur Muradoglu

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Assist. Prof. Nejat Karabakal

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Assist. Prof. Serpil Sayin

Approved for the Graduate School of Business Administration

Prof. Subidey Togan

ABSTRACT

PRICE PREDICTION IN IMKB USING NEURAL NETWORKS

BY

SINAN ALTUG

Supervisor: Assoc. Prof. Gulnur Muradoglu

June 1994

The purpose of this thesis is to perform price prediction in Istanbul Stock Exchange (IMKB) using neural networks approach. The neural networks have been in use in the literature of plenty of time, however, this thesis is one of the first applications of neural network forecasting in the Turkish Financial Framework.

The study focuses on four stocks, each of which exhibited different trends for the period January 1991 - June 1993. Comparative analysis were carried out for each prediction and detailed statistical inquiry was performed. Even though the full potential of neural networks could not be utilized (basically because of data limitations), the results prove that neural networks perform significantly successful predictions.

ÖZET

İSTANBUL MENKUL KIYMETLER BORSASINDA YAPAY SİNİR AĞLARI KULLANILARAK FİYAT ÖNGÖRÜSÜ

SİNAN ALTUĞ

Yüksek Lisans Tezi, İşletme Fakültesi

Tez Yöneticisi : Doç. Dr. Gülnur Muradoğlu

Bu çalışmanın amacı İstanbul Menkul Kıymetler Borsası'nda yapay sinir ağları tekniği kullanılarak fiyat öngörüsü yapmaktadır. Yapay sinir ağları, öngörü literatüründe uzun bir süreden beri yer almasına rağmen, bu çalışma, tekniğin Türk Finans ortamındaki ilk uygulanışlarından biridir.

Çalışma Ocak 1991 - Haziran 1993 tarihleri arasında değişik salınımlar gösteren dört hisse senedi üzerinde odaklanmıştır. Uygulanan bütün analizler karşılaştırılmalı olarak yapılmıştır ve her öngörü için detaylı istatistiksel bilgiler sağlanmış ve incelenmiştir. Her ne kadar veri kaynakları sınırlamalarından dolayı yapay sinir ağlarının bütün gücü kullanılamamışsa da, sonuçlar bu tekniğin Türk Finans Ortamında oldukça başarılı olduğu görülmüştür.

Acknowledgments

I would like to first express my sincere gratitude to Assoc. Prof. Gulnur Muradoglu for her motivating encouragement and most valuable comments and constructive suggestions. I am also indebted to Mr. Guven Sak for his initiating discussions and recommendations. I would also like to thank sincerely to Assist. Prof. Nejat Karabakal for his important observations, mostly about the algorithm, and Assist. Prof. Serpil Sayin for her support, assistance and time.

Table of Contents

1. INTRODUCTION

Price prediction in financial markets using neural networks has increasingly become popular in the international literature. However, there are very few examples of it in Turkish financial environments. Furthermore, one of the latest trends in prediction, the use of neural networks, has not been applied at all, to the Turkish context. The purpose of this thesis is to perform price prediction in Istanbul Stock Exchange (IMKB) using neural networks.

The ability to forecast is a central requirement for rational decision-making, since the merit of any decision is always measured by its consequences in future. This applies in particular to the financial sector, for example in exchange rate trading or capital investment. The best known forecasting technique in this context is chart analysis, which only evaluates data from a specific time series in the past. In contrast, a fundamental analysis attempts to describe the actual dynamics of the market process. The success of chart analyses is handicapped by the low volume of input information, while that of a fundamental analysis is limited by the complexity of the market and the fact that it disregards the psychological factors in decision making.

The use of neural networks opens new possibilities for forecasting complex economic dynamics. For the prediction of stock prices, interest rates and exchange rates, this mathematical tool shows a new way to extract a dynamical structure from data of the past. Today it is becoming apparent that the discipline of neural computer science can provide a better basis for decision making.

I believe this thesis possesses a value, as to my knowledge, it is the one of the first studies of prediction with neural networks in the Turkish financial environment that has been made up to date.

The flow of the thesis is as follows: Chapter II is a brief review of the literature about forecasting with neural networks that has been conducted up to date; Chapter III is the general overview of the method used, together with the general definitions and the mechanism. Chapter IV is the Analysis part, containing the principal findings and results and the benchmarks for comparison. Chapter V is the Summary and Conclusions part. With the final interpretations of the finding and recommendations for the path of future studies.

2. LITERATURE SURVEY

Forecasting the stock price movements has always been a challenging problem. Price movements of stocks have been inspiring the theoreticians and the practitioners for a long time. Over the years, extensive alternative modeling approaches, differing notions of efficiency levels and conflicting empirical claims and counter claims have kept coming [Rawani and Mohapatra, 1993]. On the whole, evidence is consistent with the weak and semi-strong forms of efficiency of the market [Fama, 1991]. An efficient market is one where time series analysis will not provide any opportunity for making profit. Despite such theoretical results, practitioners speak of ample opportunities at various points of time where forecasting helps. Much contradictory evidence exists regarding the use of current information to predict stock price movements(returns). Fama [1991] notes that there is "no lack of evidence that short-horizon returns are predictable from other variables"[pp. 1578]. These variables include the E/P or (P/E) ratios[Bauman and Doven, 1988] . These and other variables have been used in multi-factor models to screen securities and stocks [Bauman and Doven, 1986]. Bulkley and Tonks[1992] report that stock prices are volatile, and profitable trading rules can be followed to earn significantly higher rates of return.

The currency price movements provide a data-rich environment. A novel forecasting technique that is capable of making a large number of computations in a data-rich environment (such as the one that exists in a stock market) and of adapting and learning to track the patterns underlying the price movements is the neural network technique. These studies therefore, justify the possibility of making forecasts in the stock market, which can yield results above the level that can be attained by chance alone.

In the past five years, neural networks have received a great deal of attention because of their ability to solve several classes of problems that are difficult and sometimes impossible to solve any other way. Neural networks are particularly well suited for finding accurate solutions in an environment characterized by complex, noisy, irrelevant, or partial information. They address these limitations by deriving nonlinear maps between high-dimensional, input pattern spaces and outputs [Eberhart and Dobbins, 1990]. The Appendix contains a technical description of the computational procedure used by neural networks to derive these maps.

Hecht-Nielsen [1992] makes the following statement for the predictive superiority of neural networks over general linear models:

"A primary advantage of mapping networks over classical statistical regression analysis is that the neural networks have more general functional forms than the well developed statistical methods can effectively deal with...Neural networks are free from depending on linear superposition and orthogonal functions--which linear statistical approaches must use...In summary, enough experimental evidence has now been gathered to state with some confidence that mapping networks are, in general, different than statistical regression approaches. The function approximations that arise from properly applied mapping networks (at least in distances where sufficient training data were available) are usually better than those provided by regression techniques" [pp. 27].

Several researchers have, in recent years, proposed the application of neural networks to stock market forecasting. Though very similar, the approaches differ slightly in the mechanism.

Various researchers made attempts for long term price forecasting, and several tried to make forecasting for very short term periods[Rawani and Mohapatra, 1993]; they even tried to forecast the closing price of a session that has started

Nearly all of the studies covers the percent change on the price of the stocks, rather than the exact price prediction. This is due to two basic reasons: First, prediction of the exact price does not give results as accurate as the percent-price prediction, as it necessitates more specific and "cleaner" inputs [Wong, 1990], and second, in the stock market framework, percentage changes provide more significant data [Kamijo-Taginawa, 1990],[Sharda-Patil, 1990].

The forecast output also differs from one study to another. For example, in numerous studies including the work conducted by Kamijo and Taginawa [1990] and also by the Siemens Research Team [Siemens, 1992], the prediction results are based on three-state outputs, that is, prediction was made for the: i.Increasing ii.Decreasing and iii.Not-changing states of the price. In some other studies, people made attempts for forecasting the bare percentage changes in the as their forecasted variable.

Another path of forecasting with neural networks has been performing predictions for the S&P 500 and Gold Futures Prices [Grudnitski-Osburn, 1993]. In this study, on the basis of a preliminary analysis of the data, it is determined that forecast parameters for the networks can be derived by relying on a relatively short history of 15 months of data patterns. Of the simulated 41 S&P and Gold trades, the sign of the next month's change were predicted correctly 75% of the time for S&P and 61% of the time for Gold Futures.

There are also studies forecasting trading strategies for the foreign exchange markets [Rawani-Mohapatra, 1993], In their paper, Rawani and Mohapatra [1993] use Walsh functions and Neural Networks comparatively for making separate forecasts of the foreign exchange market movements. A trading strategy is also presented, which is based on the consideration of the similarity of the current trends in these two forecasts.

Forecasting the treasury bills auction rates was the topic of another recent paper [Barucci-Landi, 1993]. The highly nonlinear nature of the short term interest rate movements make neural networks a better candidate for forecasting, compared to the classical econometrics. The results of this study is quantitatively better than the results that were obtained by the VAR models, as the authors comment .

In another recent study, Refenes [1993] compares the neural network forecasting with "classical" smoothing techniques, namely, second order exponential smoothing, third order exponential smoothing and autoregression. The second order exponential smoothing corresponds to an ARIMA (0,2,2) model with a single parameter, and the third order compares to the ARIMA(0,3,3). The results are very similar,being very slightly in favor of the neural network forecasts.

Some papers do not attempt any comparison. For example Hoptroff [1993] gives four apparently successful applications of neural networks in business and economic forecasting, but does not compare the results with any alternatives. On the other hand De Groot and Wurtz [1991] do make an apparently fair comparison of neural networks with standard non-linear time-series models, such as threshold and bilinear models, for the classical sunspots data. They also show how neural networks can be used to model a deterministic chaotic time series-with an added white noise component. The results suggest neural networks are worth considering for time series exhibiting non-linear characteristics.

Another path on which the studies proliferated was to use the artificial neural networks in the decision making process of picking stocks. Basically this kind of study focuses on the ability of neural networks of discriminating between stocks that provide positive returns from those that provide negative returns. The artificial neural network employs a particular pattern recognition algorithm to learn the relationships between a company's stock return one year forward and the most recent three to four years of financial data for the company and its industry, as well as the data for several macroeconomic variables [Kryzanovski, Galler and Wright, 1993].

This study is an introduction of artificial neural networks to the Turkish financial environment. As stated above, the forecasting is constructed upon a percent change in price prediction model, including the factors like the index, past price movements, and also macroeconomic variables like currency exchanges etc. The performance index was selected as the root mean square error, which is basically the deviation of the result of the forecast from the real value.

3. METHODOLOGY

3.1 Definitions

3.1.1 Neural network:

A neural network is an implementation of an algorithm inspired by research into the brain. In fact, one branch of neuroscience uses computers to model cognitive functions.

However, the neural networks discussed here have little to do with biology. Rather, they are a technology in which computers learn directly from data, thereby assisting in classification, function estimation, data compression, and similar tasks.

Having been used experimentally for decades, neural networks are reputedly a solution in search of a problem. More recently, though they began moving into practical applications. The basic structure will be explained in more detail in Chapter III.

3.1.2 Neuron:

Neural networks are built of "neurons". Other terminology for neurons include nodes and processing elements. A neuron is a multi-input single-output artificial structure that was developed for simulating the function of the biological neuron in computer applications. The weighted sum of the inputs (to the neuron) are compared with a threshold value which is sometimes called as the Bias, Neurons are usually arranged in layers and the neurons in a layer are often connected to many neurons in the other layers or neurons in the same layer. Each neuron processes the input it receives via these connections and provides a continuous analog value to other neurons via its outgoing connections. As in biological systems, the strengths of these connections can change (and in fact do change in response to the strengths of the inputs and the type of the transfer function used by the neurons). Deciding how the neurons in a network are connected, how the neurons process their information and how the connection strengths are modified all go into creating a neural network.

3.1.3 Feed Forward Neural Networks:

The networks where data flows only in forward direction are called feed-forward networks. These type of networks are very popular due to their relative simplicity and stability. The back propagation network, which was used in this work, is also a type of feed-forward neural network which uses the algorithm that was first proposed by P. Werbos.

3.2 *The Neural Network for price prediction*

The neural network as a whole can be thought as a system which simply connects a set of inputs to a set of outputs in a probably nonlinear way. The typical

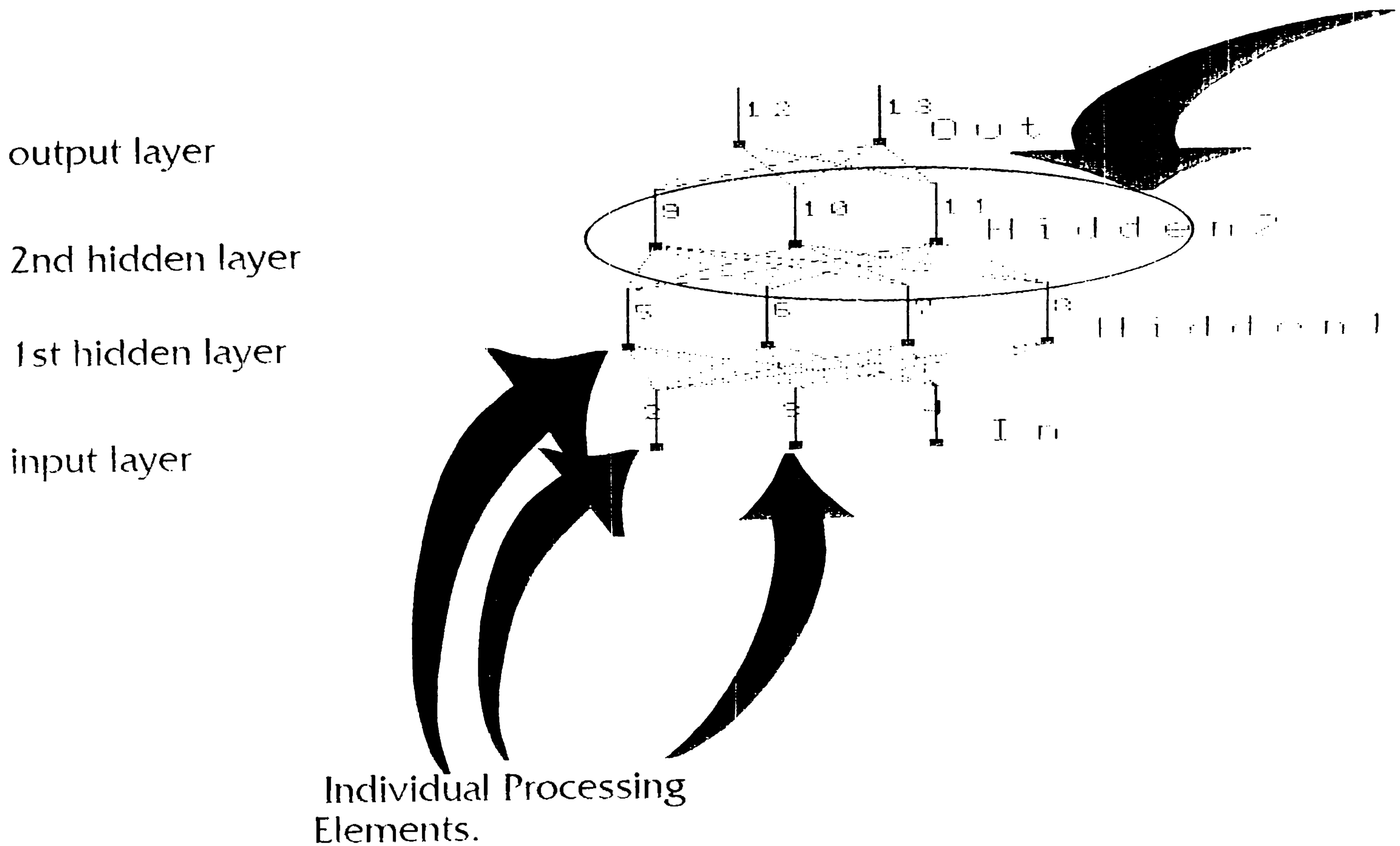


Figure 1: Illustration of a Neural Network.

illustration of a neural network is given Figure 1. The links between inputs and outputs are typically made via one or more hidden layers of neurons. The numbers of neurons, layers and the weights that are attached to the mappings from neuron to neuron arc are chosen to give the best possible fit to a set of training data using some certain training algorithm which can take a large number (e.g. tens of thousands) of iterations to converge. This is meant to mimic the sort of way the brain "learns". The approach is non-parametric in the character in that no subject domain knowledge is used in the modeling process (except the choice of which variables to include).

When applied to forecasting, the whole process can be completely automated on a computer *'so that people with little knowledge of either forecasting or neural networks can prepare reasonable forecasts in a short space of time''* [Hoptroff, 1993].

The neural network, as a whole can be interpreted as a complex model of the input output behavior of a single stock exchange trader (the individual model) [Siemens, 1993]. The trader has his own information set, which are fed as the input variables to the neural network. As a result of this accumulation of knowledge, the trader focuses his attention selectively, and gives more emphasis to some certain pieces of information, which he thinks are more important in the price determination of a stock; the other indicators are given more importance by the trader. As a second step, the indicators that the trader has registered combine to form an overall impression, for example, the price of a certain stock will rise or fall, and the amount.

Neural networks are very good choices for constructing a prediction mechanism in the market for the reason that their forecasting is based on past experience. Using the past data, they can build the future figures.

The input variables, together with the corresponding outputs are presented to the network. The neural network then computes the output, by the process described above. This predicted output is compared with the actual output, and the error is "backpropagated" to change the parameters of the network (i.e. the weights of the inputs of each neuron) so that the prediction improves. This process continues until the error between the predicted and the desired outputs converges to a previously set value. The details of the process will be rationalized in the coming pages.

3.3 The Mechanism

The following is a brief explanation of the prediction mechanism. It is visualized in Figure 2 The steps will be analyzed in full detail in the analysis chapter.

- Choose all the relevant factors (inputs)-everything that can be relevant to the price fluctuations of the analyzed stock. The choice of the so called "relevant factors" will be discussed briefly in the coming chapters.

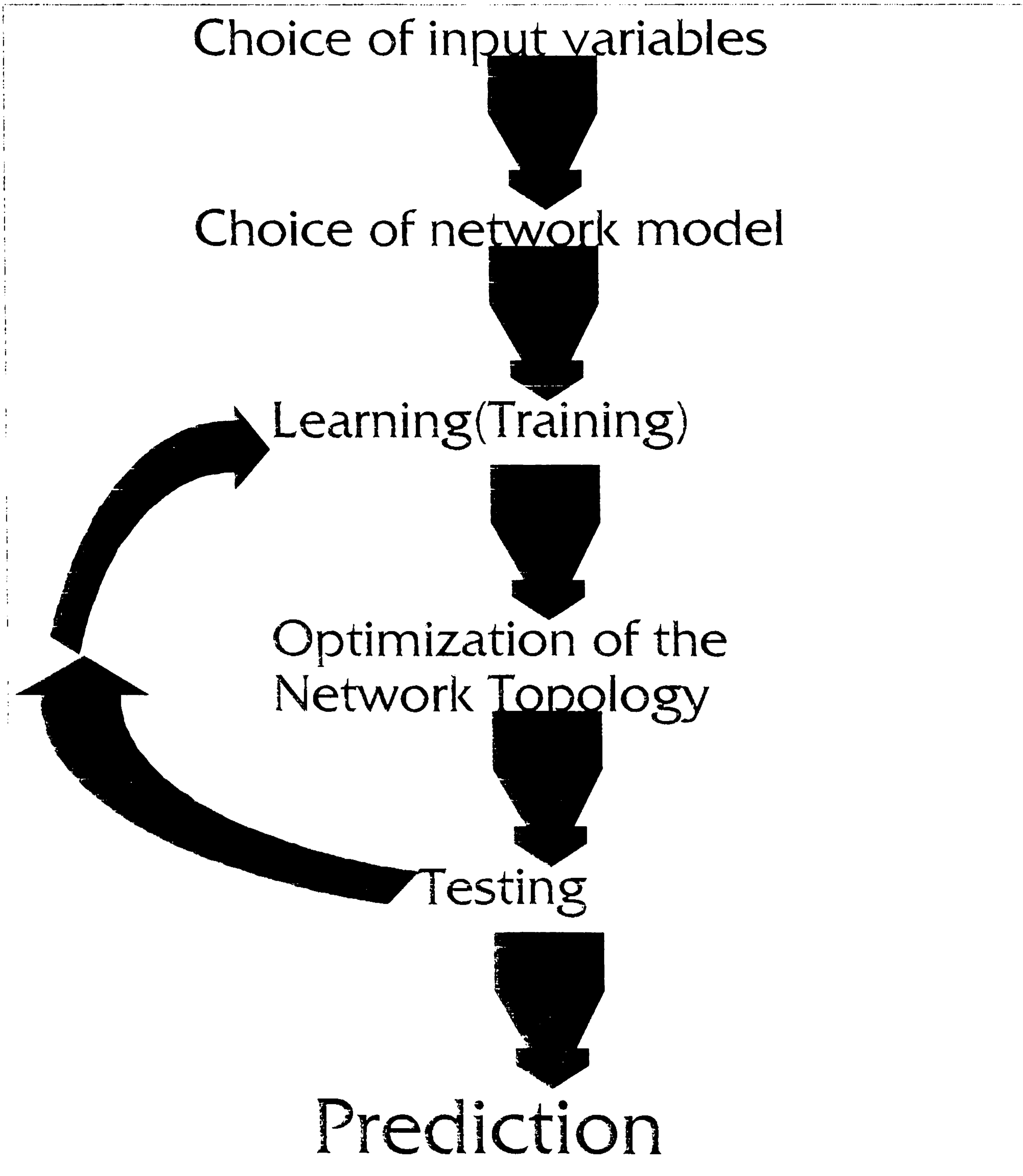


Figure 2: The Mechanism.

- include the last three days' price as inputs to the neural network. (The number three was not chosen because of the fact that it is the typical lag of the stock price, but that it is the standard time lag that is chosen in the literature). Therefore, the number of inputs is now two more than the relevant factors that were chosen in the previous step.
- Train the network with historical data in such a manner:

When the inputs i1 was x_1, i2 was y_1,...,i45 was z_1, the next period price of Stock S_1 was P_1.

When the inputs i1 was x_2, i2 was y_2,...,i45 was z_2, the next period price of Stock S_1 was P_2.

input	*I1*	*I2.......I45*	*Price of Stock*
values	x_1	y_1.........z_1	P_1
	x_2	y_2.........z_2	P_2
	.	.	.
	x_k	y_k........ z_k	P_k
	.	.	.
	x_n	y_n.........z_n	P_n

Figure 3

Periodically test the network as:

If $i1=x_k$, $i2=y_k$,...,$i45=z_k$, what will be the price of s_k (the prediction)? (The actual value is p_k, so the predicted output is compared with p_k).

The error value, defined as the difference between the actual and the predicted outputs is, *backpropagated*, that is, fed back to the network (the detailed mathematical background is given in Appendix A),

- Use the currently available data for this test; that is, the actual value of the price for this combination of input values (x_k, y_k, ..., z_k) is known (it is the value p_k in the Figure 3).

- The performance criteria in this test is the difference between the actual and the predicted outputs. Continue the training session until the error in the value of the prediction goes below some predetermined amount. (percentage).

- The training is complete then. Now, as inputs, offer the current values of the input variables and as output get the prediction result from the network. If the network inputs are close to the data that were previously used in training, the prediction is more likely to be a successful one.

4 . ANALYSIS

4.1 Pre-analysis

4.1.1 The Choice of Inputs

Having vital importance for the accuracy of the prediction, as many inputs as possible that might be related to percentage changes in stock prices must be picked. However, including all the relevant factors could never be realistic, there exists no negative evidence for the weather's not effecting the stock price, for example. Therefore, the inputs that carry more "information", that is that tells my network more about the structure underlying the price fluctuations was chosen.

At this stage, the choice of inputs was made via the knowledge and expertise presented in previous literature and interviews of experts in Turkish Stock Market. Therefore, in a way, the past expertise was transferred to the neural network.

The initial list of "intuitive" inputs that came into the scene after this stage were filtered by a process of calculating correlations. The correlation of each of the following *intuitive* factors with the price of the predicted stock were calculated and the ones that were below absolute 0.1 (the standard value used in statistics) were eliminated from the model. This was done considering the fact that using large number of input factors (and the reason it is generally not done in comparable statistical models) may inhibit performance. The list of factors that are assured to effect stock prices(inputs) are given below in Table 1.

THE VARIABLES THAT ARE INPUT TO THE PRICE PREDICTION NETWORK:

([illegible] delta values, that is, the percentage change from the last period)

	cumulative number of inputs
[illegible] Price Fluctuations(last [illegible] days)	3
[illegible] price Fluctuations	
[illegible]change rate TL $	5
Exchange rate TL DM	6
G[illegible]d Prices	7
[illegible]rest rate 10 Bank avera[illegible] [illegible]1 Month)	8
[illegible]rest rate 10 Bank avera[illegible] [illegible]3 Month)	9
Tr[illegible]sury Bill rates 3 month	10
G[illegible]ernment bonds	1[illegible]
C[illegible]porate Bonds	12

Table 1

The correlation of all these factors with the next day price of the four stocks that were tested are given in Table 2.

One of the main advantages of neural network is that after the training is finished, the inputs that have less significance could be indicated by examination of the input connection weights. The inputs which have lower connection weights contributes less to the prediction, so they could be eliminated. These all go into the modifying the topology of the network and discussed under the related topic.

Correlations of the Input Factors with the Prices of Stocks

	Dollar	Gold	Mark	1 Month Interest	3 Months Interest	1 Year Gvrmnt Bond	3 Months T-Bill	Corporate bond	IMKB index	Stock previous day	Stock 2 days ago	Stock 3 days ago
Arcelik	0.555	0.592	0.585	0.570	0.404	0.527	0.247	-0.585	0.677	0.988	0.975	0.964
Kepez	0.535	0.567	-0.540	0.160	-0.267	0.457	-0.570	-0.736	0.852	0.990	0.979	0.966
Sarkuysan	0.423	0.481	0.408	0.153	-0.183	0.423	-7.242	-0.657	0.953	0.992	0.983	0.975
Deva	-0.622	-0.596	-0.636	-0.524	-0.428	-0.481	-6.993	0.128	0.148	0.991	0.982	0.972

Table 2

4.1.1.1 Previous Price Fluctuations

One obvious input factor are the lagged returns for previous weeks. Fama in his famous paper Efficient Capital Markets II [1991] indicates that for the predictability of the short horizon returns of the stocks the past returns prove to be confidentially reliable, but not always sufficient. Fama [1991] states that at least for the individual stocks, variation in daily and weekly expected returns is a small part of the variance of returns) variables. The issue was also excavated by many researchers, like Lo and MacKinlay [1988] and also French and Roll [1988], and all the results proved high positive autocorrelations. With the light of the research made by experts, we can conclude confidently that percent price change in a stock's price for a previous days affect the change in price of that stock for the coming day or days.

The above discussion also suggests that, it is not only the immediately preceding day's price changes that are relevant to the upcoming day's changes. In this work, that is the lag was arbitrarily cut off at three days (but still kept this as a settable parameter), as this is the standard in the neural network literature [Hect and Nielsen, 1990]. Using three previous days' data as separate input factor increases the number of input factors and leads to a richer financial model and a more effective neural network.

4.1.1.2 *The Index*

The capital asset pricing model states that every stock has a marginal contribution to the risk and return of the market portfolio, therefore the index. Therefore, the index possesses information about the price of any individual stock. The trends of every stock (that is included in the index calculation) is averaged in the index so the inclusion of the index introduces the general trend to the forecast as an input. If we had included the stocks that contribute to the index one by one to the neural network would what we would have done would be performing a weighted average with the data similar to the index calculations. Therefore inclusion of the index decreases the number of inputs tremendously, as if it were not for the index, plenty of other stocks' trends would have been included as inputs. Consequently, it can be thought that the index will bring extra information to the topology, and it was included explicitly as an input neuron.

4.1.1.3 *Interest Rates, Exchange Rates, Gold Prices, Government Bonds and Corporate Bonds*

Other than the above factors, some macroeconomic factors may help in predicting these stock prices [Fama, 1991]. One commonly used indicator of stock price movements is the interest rate.

The interest rate has the well known effect on the stock prices. When the interest rate increases, ceteris paribus (the money supply is constant), the investors follow high return. An investor who invests on stocks sells redirects his investment towards the high return and the demand for stocks decreases due to the sell orders in the stock exchange. I used the interest rates (the average of the 10 major banks), and the treasury bill yields (3-month) as the interest rate indicators. Limiting the interest rate indicators to these two is due to the shallowness of the Turkish financial environment. Both the 3-month and the 1-month interest rates that the banks offer was included as the these two short term rates can have different effects on the price because of the fact that liquidity of these two choices are different and liquidity of the alternative investment vehicles effect the stock price [Fama, 1991].

Another possibly useful class of indicators of stock price movements are foreign exchange rates. The foreign currency, especially the Dollar and the Mark act as serious alternatives to the stock investments xx. In Turkey, this effect is even more apparent, as the financial instruments are much more limited than the financial contexts the developed countries enjoy. Therefore, I incorporated the percentage changes in the two major currencies, namely the TL/Mark and TL/Dollar parities as the exchange rate indicators to the model. Adding any more would be no more than a bare increase in the inputs because of the above mentioned shallowness of the Turkish financial environment.

To enrich the forecasting model, three other alternatives for the stock exchange were included as inputs, namely the corporate bonds, the government bonds and the gold prices. The inclusion of these three, especially the gold prices may not mean a lot for a developed country's framework, however, in the Turkish context, the limitedness of the financial investment tools once again makes it necessary to include such vehicles, as they constitute serious contestants for the stock investments. The "gold" mentioned above is the Cumhuriyet Altini.

The main problem was that for all the above variables, the data I collected is weekly. Even though I had scanned all the main archives, I found out that no daily data was collected for the above variables. What I did for daily price forecast was to linearly interpolate the weekly data and form the daily data. This should normally bring no complications other than the times when some extraordinary (daily) change takes place. This issue is one of the main flaws in my forecast methodology.

4.1.2 The Formatting of the Raw Data

The formatting process of the raw data is an important step in the analysis. In making forecasting with neural networks, so as not to lose the accuracy, the input data should be normalized in such a way that they are in the order of the output (the variable that is being forecasted). This substantially decreases the amount of computations that the network must perform, hence decreases the amount of time needed for convergence.

Also, the outputs being in the order of 10 maximum helps for the decrease of the computation time, also the accuracy.

Considering all the above factors, the normalization was made as follows:

I. The forecasted variable's values were divided by 1000, as well as Dollar and Mark prices, the index and the previous three day's prices.

II. The one month and three month interest rates, treasury bill rates, corporate bond yields were expressed as percentages (between zero and one).

The next step in formatting of data was the adjusting of the stock's price for the dates when stocks dividends were distributed, and preemptive rights were offered. Occurrence of such an issue normally appears as a jump in the price of the stock. Figure 4 exhibits the trends for the four stocks analyzed before the correction. For the period of analysis, January 1991 up to May 1993, the correction was conducted as follows:

Starting from the most recent occurrence, the standard correction formula was used for each occurrence:

Price(corrected)=[Po+(Stock Dividend*1000)]/[new # of shares/old # of Shares]

The Trends of the Four Stocks Before Correction

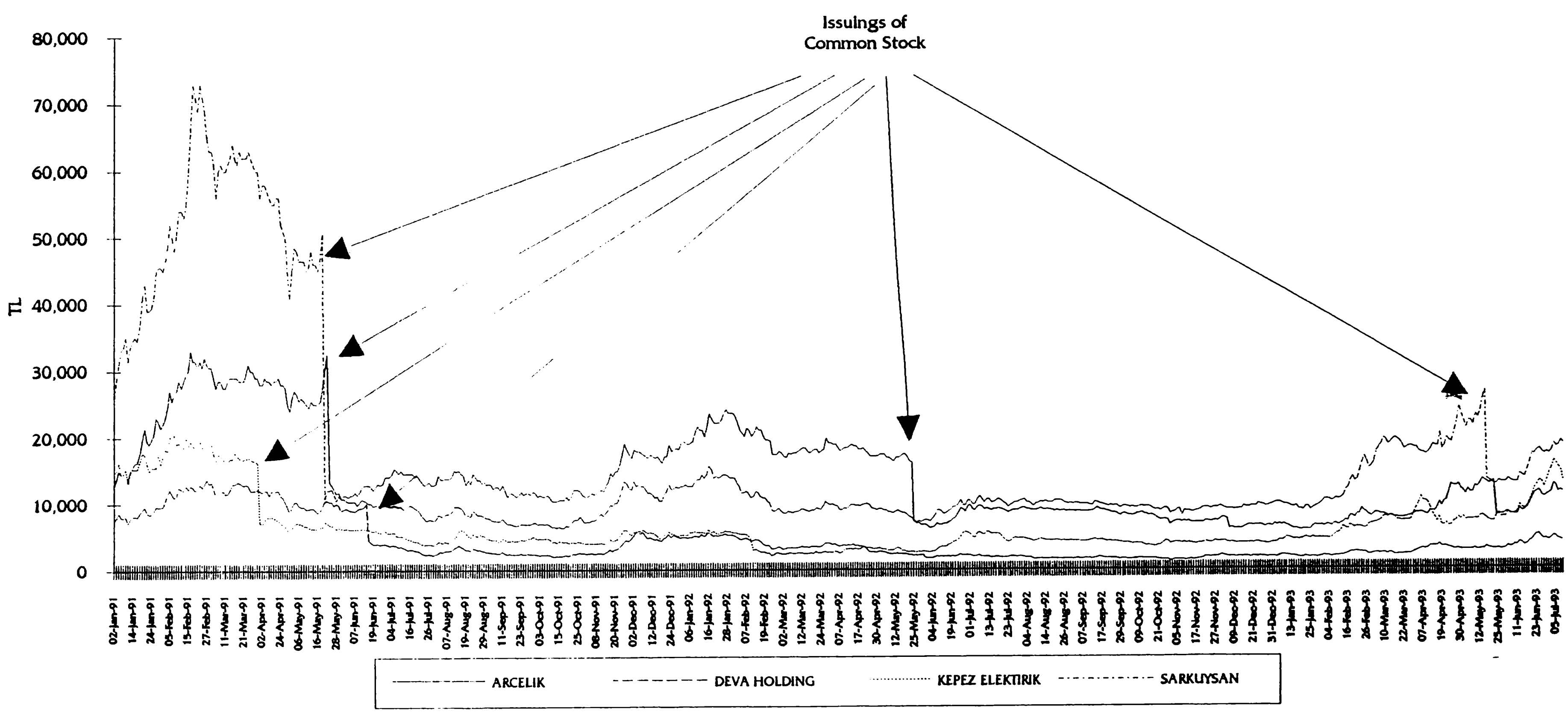


Figure 4

This trend was modified all the way back to January 1991 from the point of occurrence. Therefore the adjusted data has no discontinuities, other than the standard 10% (at maximum) changes daily. This adjusted data was also used for the previous (3) day's fluctuations, which are presented as inputs to the network.

4.1.3 Benchmarks for Comparison

So as to be able to make a comparison, in addition to the neural network forecasts, two other forecasts were made by using two different methods.

4.1.3.1 Linear Regression Model

With the same inputs as the neural network model, detailed linear regression models were constructed, with 12 independent variables, which are exactly the same as the inputs of the neural network model. For each stock, the same data points presented to the neural network in the training sessions were used for the linear regression model building.

After the model is constructed, the data points that are going to be forecasted (the latest 42 values for Arcelik, 39 for Sarkuysan,(96 for Sarkuysan 2), 56 for Kepez, and 54 for Deva) were regressed with the regression parameters that were found. The important point here is that these data points were not included in either the regression model construction or the training of the neural network. These data points for each stock respectively are the ones that were also forecasted by the neural network. Hence, the linear regression model can be utilized for full comparison with the neural network model, the resulting parameters and other detailed statistics for the regression models are presented in Tables 4,10,12,14,16 for Arcelik, Sarkuysan test 1, Sarkuysan test 2, Kepez, and Deva respectively.

4.1.3.2 Ten Day Moving Average

Other than the regression model, the 10 day moving average was calculated for the Arcelik trend (a total of 517+41=558 points).

The results of these two methods were compared with the results of two neural network predictions, one trained for 5130 steps, the other trained for 15000 steps. One step means the representation of one(out of 517) randomly chosen 12 input-1 output pair to the network. This random choice and representation is arranged by the software I had used.

4.2 Data

Four stocks were analyzed, which are Arcelik, Sarkuysan, Kepez, and Deva. The purpose underlying this choice is that these stocks exhibited different trends in the period of analysis. Arcelik trend is very near to the index, the Sarkuysan increases more than the index does, Deva decreases whereas the index increases, and finally Kepez has an extraordinary increase and decrease (totally irrelevant to the input factors because of the speculations on Kepez at that period). The comparative analysis were also conducted for each of the stocks. For each forecast, the all descriptive statistics of the error were calculated (mean, standard error, median, standard deviation, variance, kurtosis, skewness, range minimum, maximum and sum), for detailed analysis. The overall period used for analysis is January 1991- June 1993. The periods that were presented as input (for training of the neural networks) and the periods that were forecasted differs from test to test, so as to forecast the exact periods that each stock exhibited its characteristic trend (the above mentioned extraordinary increase, decrease etc.). The neural network predictions were performed with the network trained for 15,000 steps, except where indicated otherwise.

The number of data points used in the training sessions and the number of data points that are forecasted for each session governing the four stocks are given in Table 3.

	Arcelik 1	Arcelik 2	Arcelik 3	Sarkuysan 1	Sarkuysan 2	Kepez	Deva
# of Data Points Used for Training	517	500	524	519	432	538	538
# of Data Points Forecasted.	41	58	34	39	96	54	54
Time Period (For Training Data)	01/91-01/93	random	01/91-04/93	01/91-04/93	01/91-12/92	01/91-04/93	01/91-01/93

The forecasted points are the data points immediately following the data points used in training.

Table 3

The number of data points as well as the periods used for building up the linear regression model are exactly the same as the above table. Furthermore, the forecasted periods are also the same.

The reason that two different predictions were performed for February-April 1993 and April-May 1993 (for Arcelik and similar separation for Sarkuysan) was that for the last two months (April and May), the stock exchange prices increased abnormally due to some "external effects"(External effects here means the factors that have an effect on the output but were not included as inputs to the model. The stocks price increases\decreases without any significant change in the input variables.) Therefore, separating the forecasts for these months would be provide further information about the neural network forecasts for periods of extraordinary increase. Nevertheless, even for these months, the predictions are very admirable, as revealed in the coming pages.

4.3 *Findings*

The findings are probed one by one for each stock and each separate analysis. interpretation of the findings are made via the comparison between the neural network forecasts and the regression model.

4.3.1 Arcelik

For the period of analysis, Arcelik exhibited a trend that is very similar to the IMKB index (Figure 5). The daily percentage changes are in Figure 6.

Index and Arcelik Trend

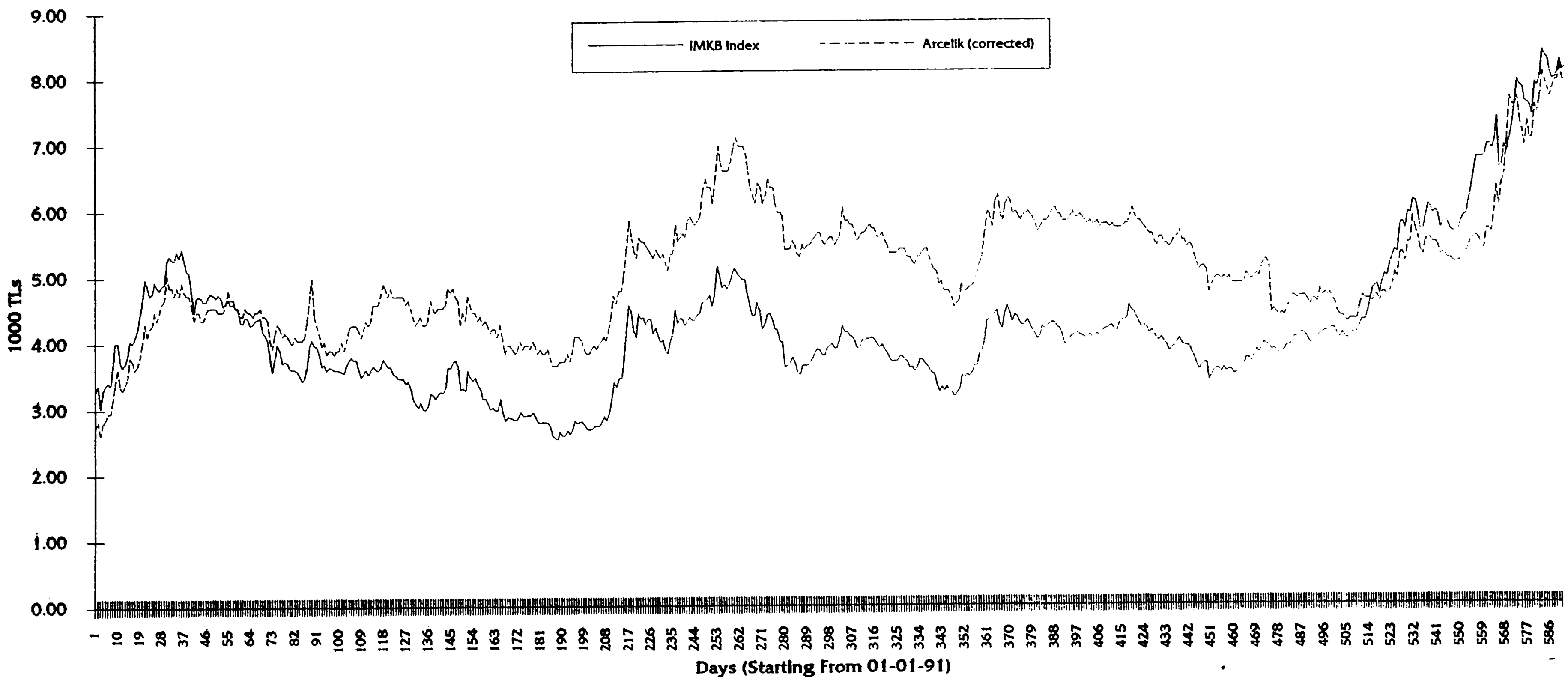


Figure 5

Arcelik Daily Percentage Changes For the Period January 1991-June 1993

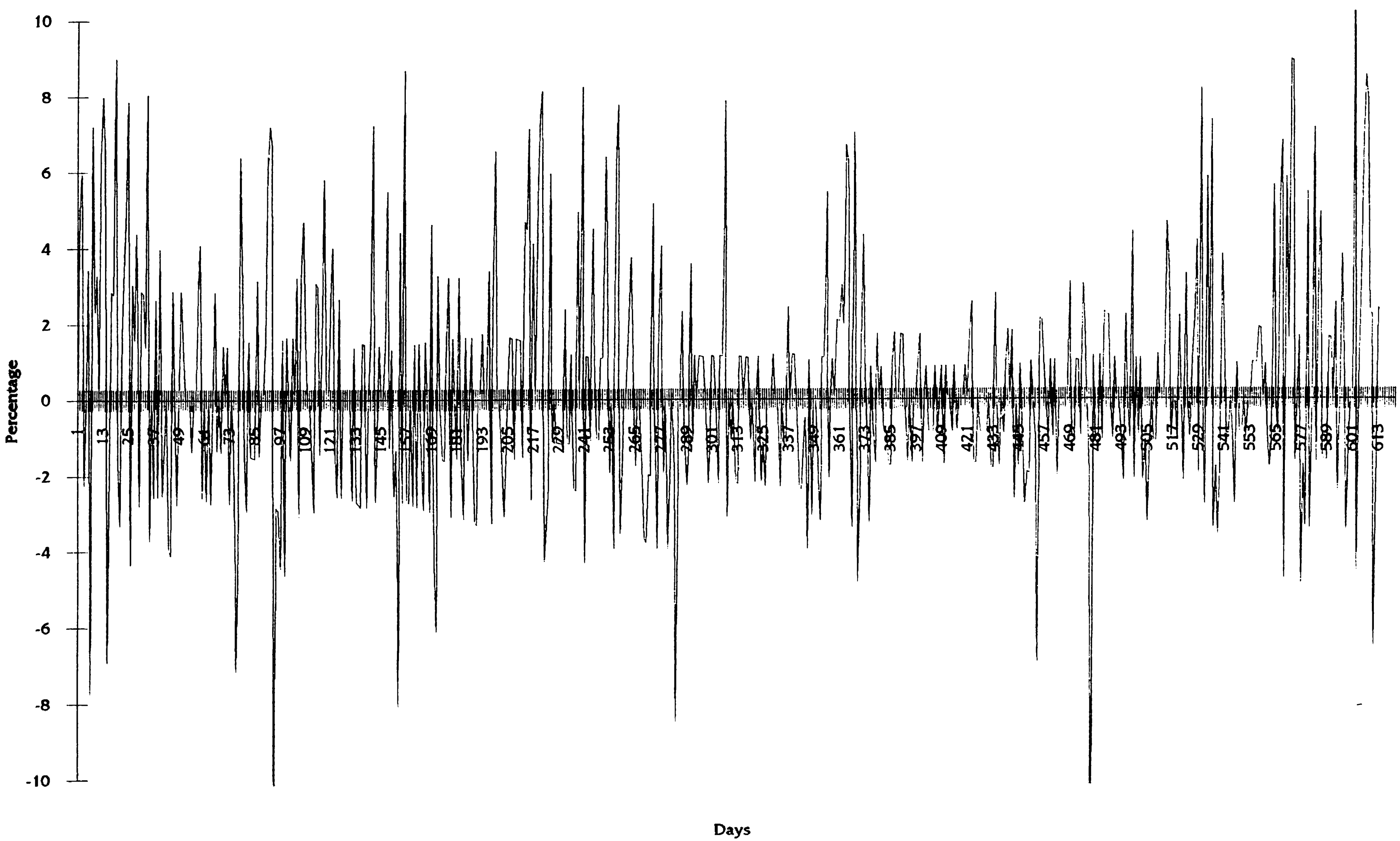


Figure 6

The Corrected Trends of the Stocks

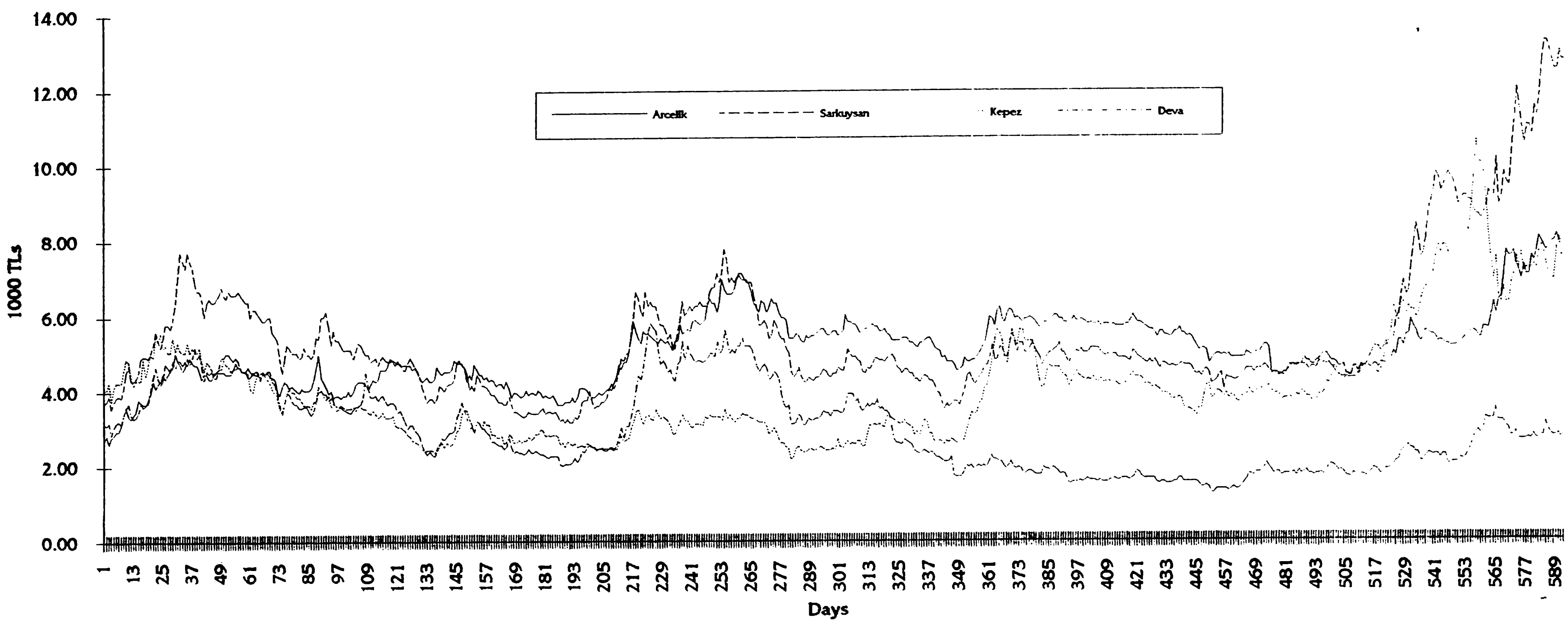


Figure 4-b

4.3.1.1 Forecasting the February 1993- April 1993 Period

Two neural network predictions were made, one with the network trained for 5130 steps, and the other trained for 15,000 steps (Figure 7). The regression model statistics are presented in Table 4. The results of predictions were compared with the linear regression model and the 10-day moving average and are exhibited in Table 5 and figure 8. The corresponding percentage errors for each are in Table 6.

The descriptive statistics of the error made in each prediction lays in Table 7. The results show that the main competition is between the regression model and the neural network trained for 15,000 steps. Furthermore, it is clearly seen that the neural network (trained for 15000 steps) is the most successful among all, with a mean of 0.573287 percent and a standard deviation of 0.8410. This means that more than 97% of the instances lay in the vicinity of 0.59±2.52 (2.52 is 3 times the standard deviation) which is a significantly successful result. The nearest accuracy was captured by the linear regression model, which has a mean of -0.72891 and a standard deviation of 2.52 (this is substantially higher than the standard deviation of the neural network prediction). Also, the neural net predictions after 15,000 steps training has a standard error much more smaller than

Regression Model Statistics for Arcelik

Regression Statistics	
Multiple R	0.98859
R Square	0.97732
Adjusted R Square	0.97678
Standard Error	0.12342
Observations	517

Analysis of Variance

	df	Sum of Squares	Mean Square	F	Significance F
Regression	12	330.7749258	27.5645771	1809.59	0
Residual	504	7.677182388	0.0152325		
Total	516	338.4521082			

	Coefficients	Standard Error	t Statistic	P-value	Lower 95%	Upper 95%
Intercept	-1.13131	0.208487799	-5.4262856	8.9E-08	-1.5409264	-0.7217
x1	-0.09421	0.063136637	-1.4921317	0.13628	-0.2182516	0.02984
x2	-0.04786	0.10702639	-0.4472058	0.65491	-0.2581356	0.16241
x3	0.14734	0.049547176	2.9737304	0.00308	0.0499955	0.24468
x4	1.15506	0.727914727	1.58680314	0.11317	-0.2750631	2.58518
x5	2.39492	0.690509575	3.46833132	0.00057	1.0382847	3.75155
x6	-2.08976	0.338523807	-6.1731605	1.4E-09	-2.7548532	-1.42467
x7	0.81595	0.367786009	2.21853363	0.02695	0.0933633	1.53853
x8	0.06422	0.282759985	0.22711581	0.82042	-0.491314	0.61975
x9	0.17722	0.018825612	9.41374059	1.6E-19	0.1402331	0.21421
x10	0.81825	0.044275539	18.4808938	6.7E-59	0.7312642	0.90524
x11	-0.07312	0.057186157	-1.278701	0.20158	-0.1854766	0.03923
x12	0.08699	0.040528033	2.14634086	0.03231	0.0073623	0.16661

Regression Model:

Prediction= -1.131 + -0.094*x1 + -0.047*x2 + 0.147*x3 + 1.155*x4 + 2.395*x5 + -2.090*x6 + 0.816*x7 + 0.064*x8 + 0.177*x9 + 0.818*x10 + (-0.073)*x11 + 0.086*x12

Table 5: The Forecats Results of the Neural Network Predictions, Regression Model, 10 day Moving Average

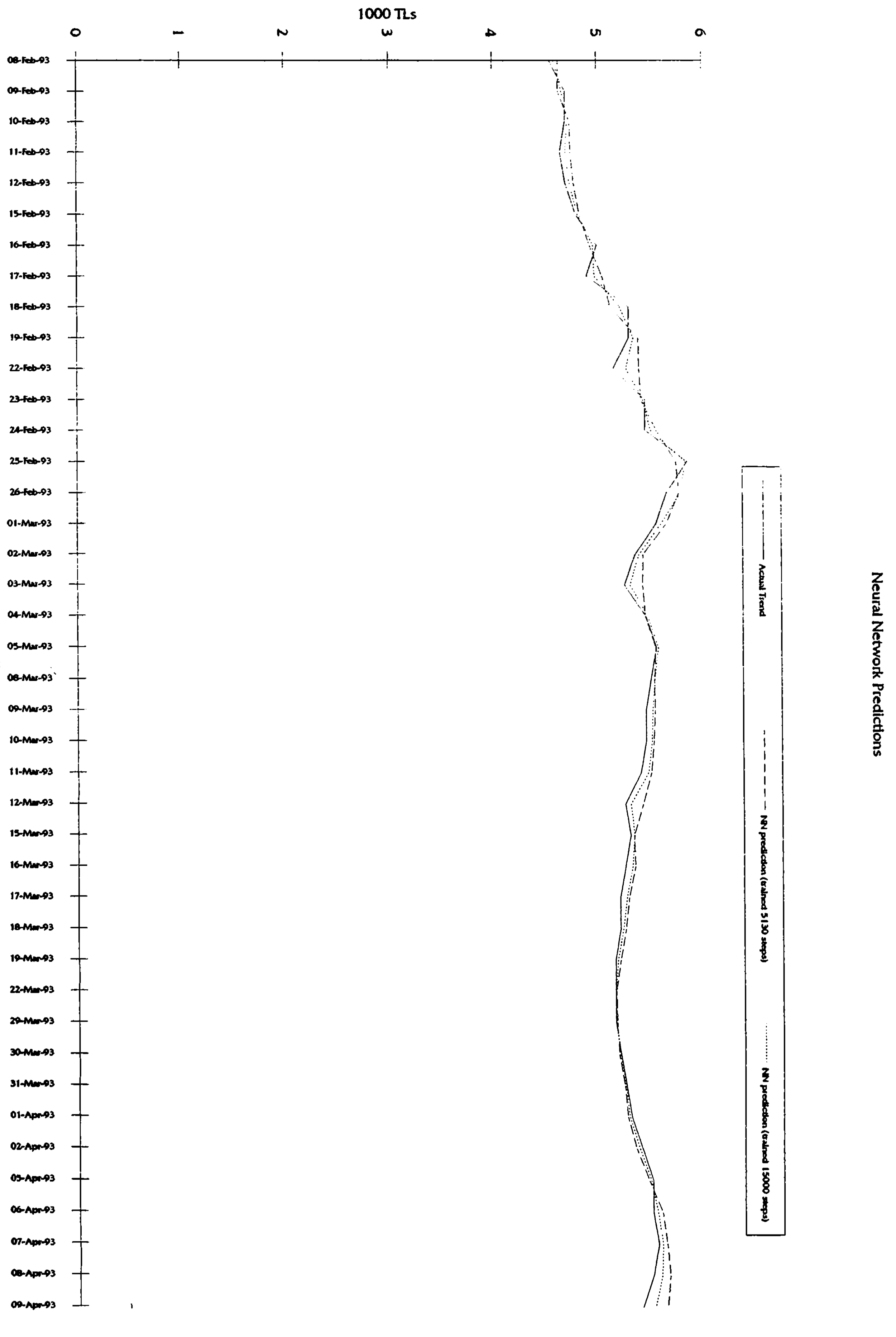
ALL 1000s	Actual Trend	NN prediction trained 5130 steps	NN prediction trained 15000 steps	10-day moving average	Regression model extracted from the training data
08-Feb-93	4.55	4.6346	4.592302	4.54	4.73095
09-Feb-93	4.7	4.63419	4.667096	4.565	4.68028
10-Feb-93	4.7	4.74327	4.721636	4.605	4.81823
11-Feb-93	4.65	4.75323	4.701616	4.625	4.78196
12-Feb-93	4.7	4.77977	4.739883	4.625	4.7755
15-Feb-93	4.8	4.83488	4.817439	4.635	4.82918
16-Feb-93	5	4.92841	4.964206	4.655	4.90866
17-Feb-93	4.9	5.04595	4.972976	4.695	5.05093
18-Feb-93	5.3	5.12608	5.213042	4.73	5.03223
19-Feb-93	5.3	5.38635	5.343174	4.795	5.37342
22-Feb-93	5.15	5.39642	5.273209	4.86	5.30553
23-Feb-93	5.45	5.41248	5.431238	4.92	5.25337
24-Feb-93	5.45	5.55756	5.503781	4.995	5.4902
25-Feb-93	5.85	5.74099	5.83197	5.07	5.48191
26-Feb-93	5.65	5.7672	5.784406	5.19	5.82148
01-Mar-93	5.55	5.64532	5.597662	5.285	5.58972
02-Mar-93	5.35	5.42345	5.386723	5.36	5.49955
03-Mar-93	5.25	5.41959	5.301309	5.395	5.31424
04-Mar-93	5.45	5.44812	5.469061	5.43	5.26036
05-Mar-93	5.55	5.5471	5.568551	5.445	5.44241
08-Mar-93	5.5	5.53101	5.535503	5.47	5.48359
09-Mar-93	5.45	5.53511	5.512555	5.505	5.426
10-Mar-93	5.45	5.52568	5.507839	5.505	5.3969
11-Mar-93	5.4	5.4974	5.468702	5.505	5.37473
12-Mar-93	5.25	5.41587	5.302937	5.46	5.28304
15-Mar-93	5.3	5.33328	5.336639	5.42	5.17282
16-Mar-93	5.25	5.34213	5.316067	5.395	5.21729
17-Mar-93	5.2	5.28588	5.262939	5.385	5.13701
18-Mar-93	5.2	5.2515	5.225749	5.38	5.07771
19-Mar-93	5.15	5.19931	5.174653	5.355	5.06606
22-Mar-93	5.15	5.15706	5.15353	5.315	5.00744
29-Mar-93	5.15	5.16499	5.157495	5.28	5.01391
30-Mar-93	5.2	5.18075	5.190375	5.25	5.01949
31-Mar-93	5.25	5.23355	5.241775	5.225	5.06827
01-Apr-93	5.3	5.26042	5.280212	5.21	5.09665
02-Apr-93	5.4	5.34311	5.371557	5.215	5.1772
05-Apr-93	5.5	5.46075	5.480375	5.225	5.30389
06-Apr-93	5.5	5.58815	5.544075	5.25	5.42708
07-Apr-93	5.55	5.62855	5.589276	5.28	5.46561
08-Apr-93	5.5	5.6582	5.5791	5.315	5.51627
09-Apr-93	5.4	5.63143	5.515716	5.35	5.47473

Table 6: The Percent Forecast Errors of the Neural Network Predictions, Regression Model, 10 day Moving Average

ALL PERCENTAGE ERRORS	NN prediction (trained 5130 steps)	NN prediction (trained 15000 steps)	10-day moving average	Regression model extracted from the training data
08-Feb-93	1.85943	0.929714	-0.21978	3.977
09-Feb-93	-1.40017	-0.70009	-2.87234	-0.41964
10-Feb-93	0.92066	0.46033	-2.02128	2.51547
11-Feb-93	2.22002	1.110011	-0.53763	2.83788
12-Feb-93	1.69713	0.848564	-1.59574	1.60648
15-Feb-93	0.7266	0.363302	-3.4375	0.60785
16-Feb-93	-1.43176	-0.71588	-6.9	-1.82677
17-Feb-93	2.97861	1.489306	-4.18367	3.08025
18-Feb-93	-3.28143	-1.64072	-10.7547	-5.05219
19-Feb-93	1.62919	0.814594	-9.5283	1.38527
22-Feb-93	4.78482	2.392408	-5.63107	3.01999
23-Feb-93	-0.68851	-0.34426	-9.72477	-3.60781
24-Feb-93	1.97361	0.986807	-8.34862	0.73767
25-Feb-93	-1.8635	-0.30821	-13.3333	-6.29216
26-Feb-93	2.07439	2.378867	-8.14159	3.03508
01-Mar-93	1.71755	0.858775	-4.77477	0.71573
02-Mar-93	1.3728	0.686402	0.18692	2.79524
03-Mar-93	3.23023	0.977314	2.7619	1.22354
04-Mar-93	-0.03446	0.349743	-0.36697	-3.4797
05-Mar-93	-0.05222	0.334252	-1.89189	-1.9386
08-Mar-93	0.56375	0.645509	-0.54545	-0.2984
09-Mar-93	1.56163	1.147789	1.00917	-0.44043
10-Mar-93	1.38857	1.061257	1.00917	-0.97431
11-Mar-93	1.80376	1.27225	1.94444	-0.4679
12-Mar-93	3.15949	1.008314	4	0.6294
15-Mar-93	0.62787	0.691292	2.26415	-2.39967
16-Mar-93	1.75493	1.258419	2.7619	-0.62297
17-Mar-93	1.6515	1.210365	3.55769	-1.21133
18-Mar-93	0.99035	0.495173	3.46154	-2.35169
19-Mar-93	0.95738	0.478689	3.98058	-1.62991
22-Mar-93	0.13707	0.068534	3.20388	-2.76814
29-Mar-93	0.29105	0.145524	2.52427	-2.64257
30-Mar-93	-0.37019	-0.1851	0.96154	-3.47136
31-Mar-93	-0.31335	-0.15668	-0.47619	-3.46152
01-Apr-93	-0.74672	-0.37336	-1.69811	-3.83686
02-Apr-93	-1.05346	-0.52673	-3.42593	-4.12594
05-Apr-93	-0.71364	-0.35682	-5	-3.56571
06-Apr-93	1.60271	0.801355	-4.54545	-1.32574
07-Apr-93	1.41533	0.707667	-4.86486	-1.5206
08-Apr-93	2.87635	1.438173	-3.36364	0.29589
09-Apr-93	4.28578	2.142889	-0.92593	1.38389

Table 7: Descriptive Statistics of the Neural Net Predictions, Regression Model, 10 day Moving Average

Descriptive Statistics	*NN prediction (trained 5130 steps)*	*NN prediction (trained 15000 steps)*	*10-day moving average*	*Regression model extracted from the training data*
Mean	0.983003	0.59136	-2.08494	-0.72891
Standard Error	0.258264	0.131351	0.69135	0.3949685
Median	1.372804	0.691292	-1.59574	-0.622973
Standard Deviation	1.653696	0.841056	4.42679	2.5290322
Variance	2.734709	0.707375	19.5964	6.396004
Kurtosis	0.357918	0.508784	-0.21722	-0.743635
Skewness	-0.140098	-0.14108	-0.63944	-0.031976
Range	8.066249	4.033125	17.3333	10.269158
Minimum	-3.281434	-1.64072	-13.3333	-6.292158
Maximum	4.784816	2.392408	4	3.977
Sum	40.30313	24.24576	-85.4824	-29.88529
Count	41	41	41	41



Neural Network Predictions
1000 TLs
0
1
2
3
4
5
6
08-Feb-93
09-Feb-93
10-Feb-93
11-Feb-93
12-Feb-93
15-Feb-93
16-Feb-93
17-Feb-93
18-Feb-93
19-Feb-93
22-Feb-93
23-Feb-93
24-Feb-93
25-Feb-93
26-Feb-93
01-Mar-93
02-Mar-93
03-Mar-93
04-Mar-93
05-Mar-93
08-Mar-93
09-Mar-93
10-Mar-93
11-Mar-93
12-Mar-93
15-Mar-93
16-Mar-93
17-Mar-93
18-Mar-93
19-Mar-93
22-Mar-93
29-Mar-93
30-Mar-93
31-Mar-93
01-Apr-93
02-Apr-93
05-Apr-93
06-Apr-93
07-Apr-93
08-Apr-93
09-Apr-93
Days
Actual Trend
NN prediction (trained 5130 steps)
NN prediction (trained 15000 steps)

Figure 7

Comparative Forecast Results For Arcelik

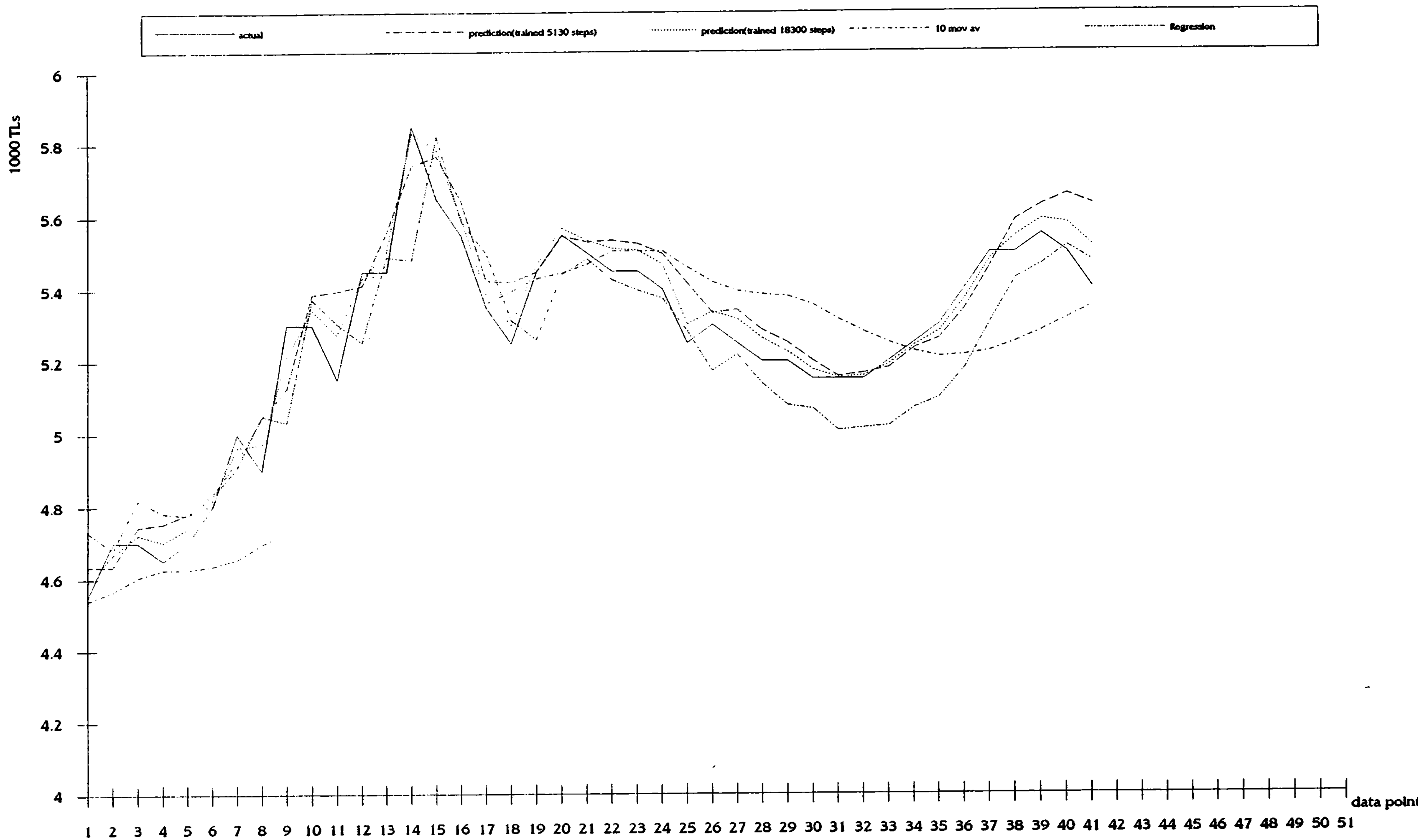


Figure 8

even the regression model (0.13 versus 0.39 percent). The kurtosis of the two are very near to each other, the neural network model's being slightly better(0.50 vs. -0.74), indicating that the neural network prediction is a little bit more peaked, but does not bring any complication, as the values are near to zero. The same is valid for the skewness values. Both are skewed to the left with negative values very close to zero. The frequency distribution graph of the error for the neural network and the regression model forecasts are represented in figures 9 and 10 respectively. For both the neural network prediction and the regression model, the mean and the median are very close to each other, 0.1% difference in both cases, and in each case, the median is greater, justifying the leftward skewness of the forecast errors.

The maximum and minimum values of the forecast errors too are in favor of the 15,000 steps trained neural network prediction, having a maximum error of just 2.39 percent and a minimum of -1.64. which means a maximum absolute deviation of 2.39 among all predictions(versus 6.29 in the regression model forecast error). This means that the return of any daily investment made by the guide of neural network predictions can be forecasted with an error of at most 2.39 percent of error. This is a very important result as, the range of the daily return is 20%.

Figure 9: The Frequency Distribution of the Forecast Error (Neural Network)

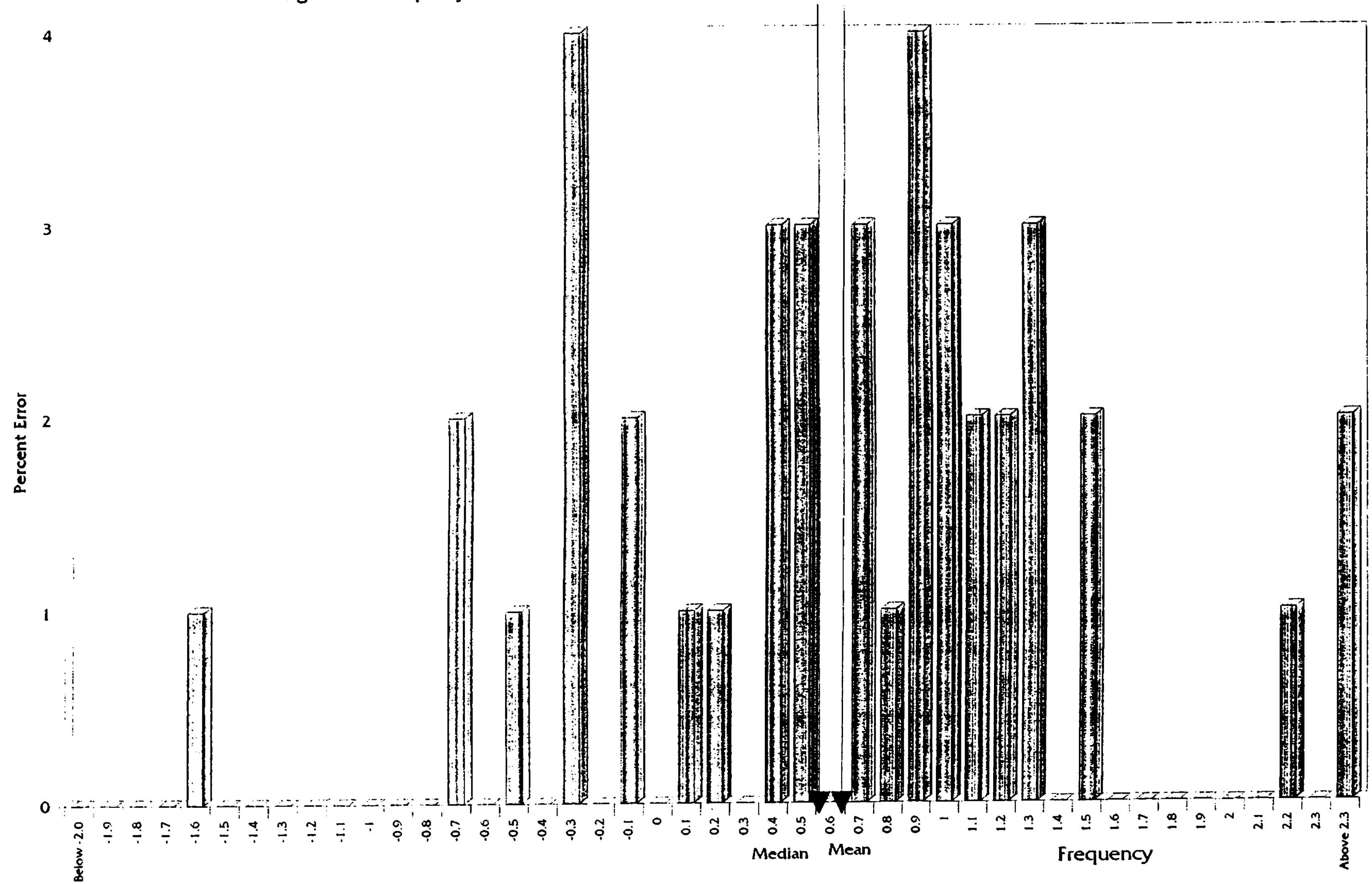
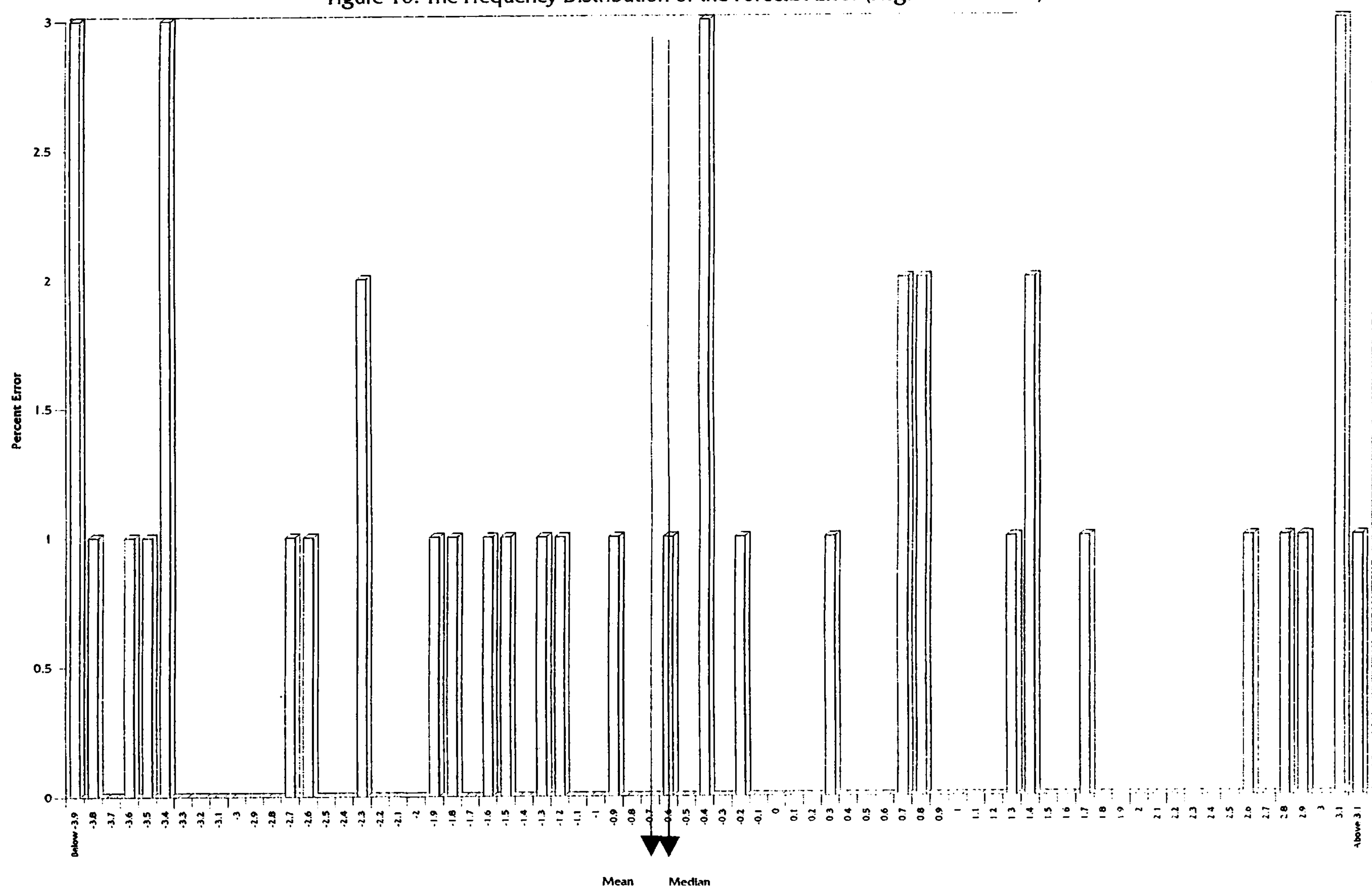


Figure 10: The Frequency Distribution of the Forecast Error (Regression Model)



4.3.1.2 Forecasting the April 1993-May 1993 Period

The Arcelik trend exhibits an extraordinary increase for these months (compared to the IMKB index), so my expectations before the test was that the forecast would be worse than that of the previous, furthermore, that the predicted values could not follow the real trend. However, the forecast results were a lot better than my expectations. Table 8 lists the results of the forecast together with the percent error made and all the descriptive statistics. Figure 11 is the graphical presentation.

Although the prediction is worse than the February-April prediction, the mean of the error made is just -2.1, which makes the forecast still a very remarkable one. This negative value also justifies the discussion above, the neural network prediction lags the real trend.

This time, the difference between the mean and the median is also higher (nearly four times the previous forecast), so the error distribution is also more skewed (skewness=0.54).

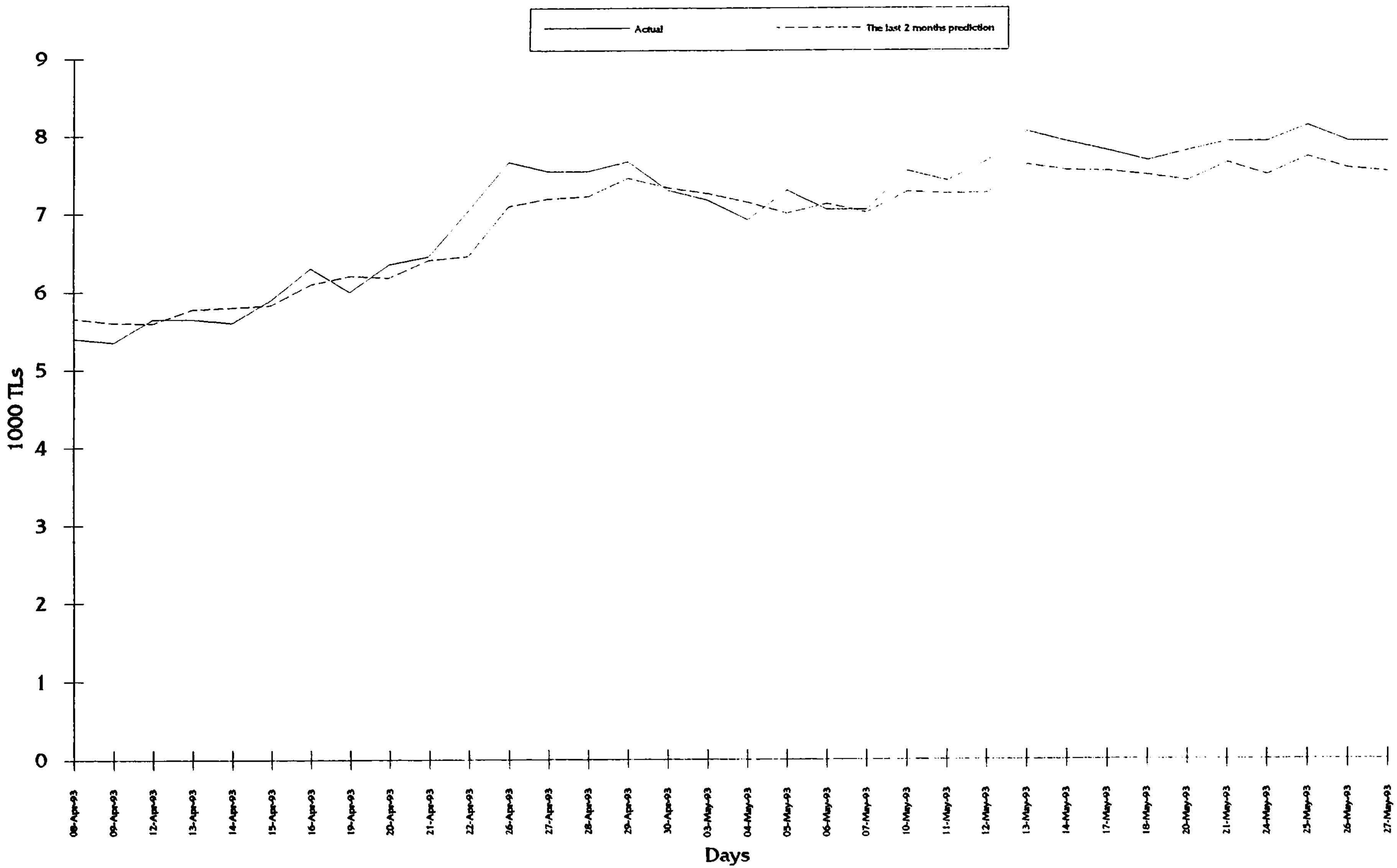
The maximum absolute deviation is 8.27 (for April 22nd). Furthermore, the following session's error is -7.43 which is the second largest. Although these values look high, a close investigation of the actual trend (Figure 5) reveals the reason. For these two consecutive sessions, the first

Table 8: Arcelik-May 93-June 93 Prediction.

	Actual	The last 2 months prediction	Error
08-Apr-93	5.4	5.66147	4.84196
09-Apr-93	5.35	5.60319	4.73245
12-Apr-93	5.65	5.59754	-0.92851
13-Apr-93	5.65	5.77531	2.21782
14-Apr-93	5.6	5.79893	3.55223
15-Apr-93	5.9	5.82814	-1.21805
16-Apr-93	6.3	6.09158	-3.30822
19-Apr-93	6	6.20196	3.36607
20-Apr-93	6.35	6.17682	-2.72721
21-Apr-93	6.45	6.39828	-0.80183
22-Apr-93	7.03	6.44839	-8.27321
26-Apr-93	7.65	7.08118	-7.43552
27-Apr-93	7.53	7.17946	-4.65531
28-Apr-93	7.53	7.20598	-4.30303
29-Apr-93	7.65	7.43572	-2.80101
30-Apr-93	7.28	7.31041	0.41777
03-May-93	7.15	7.23456	1.18262
04-May-93	6.9	7.12785	3.3022
05-May-93	7.28	6.981	-4.10716
06-May-93	7.03	7.10388	1.05085
07-May-93	7.03	6.99217	-0.53819
10-May-93	7.53	7.25638	-3.63368
11-May-93	7.4	7.2347	-2.23381
12-May-93	7.65	7.24242	-5.32791
13-May-93	8.03	7.60335	-5.31315
14-May-93	7.9	7.52744	-4.71596
17-May-93	7.78	7.51752	-3.37373
18-May-93	7.65	7.46565	-2.40982
20-May-93	7.78	7.39366	-4.96582
21-May-93	7.9	7.62216	-3.51694
24-May-93	7.9	7.47245	-5.41205
25-May-93	8.1	7.69838	-4.95827
26-May-93	7.9	7.55109	-4.41653
27-May-93	7.9	7.50628	-4.98385

Error Statistics	
Mean	-2.10867
Standard Error	0.59476
Median	-3.05461
Standard Deviation	3.468
Variance	12.027
Kurtosis	-0.55944
Skewness	0.53951
Range	13.1152
Minimum	-8.27321
Maximum	4.84196
Sum	-71.6948
Count	34

Figure 11: Arcelik April-May 93 Prediction



being a Friday and the second a Monday, the Arcelik price skyrocketed with no particular reason.

Another factor that is clearly observed from the trend and the forecast is that for the May sessions, although the neural network made and average error of -3.7, the growth and the decline days of the trend are predicted almost with full exactitude; only four out of seventeen days were predicted wrongly.

Therefore a very important result follows for the April-May forecast of Arcelik: Even though the mean, standard deviation, and other statistics are worse than the February-April forecast, when thought as a binary buy\sell game, the accuracy of April-May prediction is almost as good as the previous forecast. Moreover, any player in the stock market who invests depending on the buy-sell commands outputted from the neural network could make large profits given that they invest daily. This is especially important as there exists and offset error of -2.1 so when an investor waits for three days for example, the neural network could output sell even if the overall return could be negative. Nevertheless, a neural network that could perform cumulative forecasts is easy to construct, yet is beyond the scope of this thesis.

4.3.1.3 Prediction With Randomly Extracted Block of Data

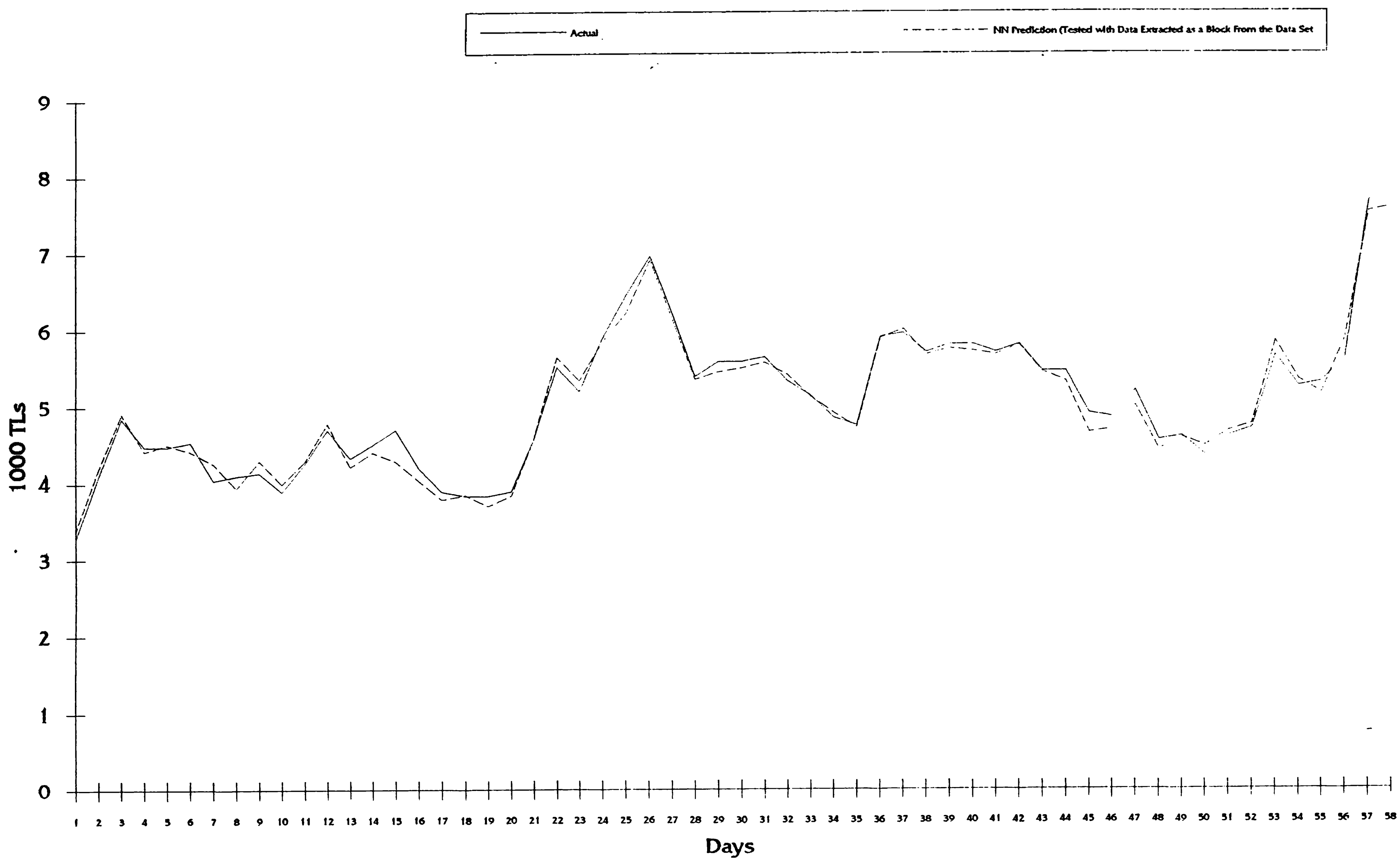
This is a standard test (reference) for the neural network for testing its ability of interpolation. For this test the

Table 9: Arcelik-Test with data extracted from the middle of the data

Actual	NN Prediction (Tested with Data Extracted as a Block From the Data Set	Error
3.29	3.39545	3.20508
4.1	4.19459	2.30702
4.85	4.91019	1.24097
4.48	4.42018	-1.33529
4.48	4.50709	0.60478
4.54	4.41994	-2.64452
4.04	4.25977	5.43993
4.1	3.93945	-3.91576
4.14	4.30086	3.88551
3.89	3.99434	2.68234
4.26	4.291	0.72777
4.7	4.78368	1.78047
4.33	4.22017	-2.53661
4.51	4.40509	-2.32616
4.7	4.28647	-8.79851
4.2	4.03603	-3.90414
3.89	3.78704	-2.64679
3.83	3.84093	0.28535
3.83	3.70444	-3.27836
3.89	3.83776	-1.34293
4.58	4.5842	0.09164
5.51	5.64227	2.40053
5.2	5.33823	2.65821
5.89	5.84531	-0.75871
6.45	6.21436	-3.65333
6.95	6.90156	-0.69701
6.2	6.12622	-1.19002
5.39	5.35233	-0.69898
5.58	5.44883	-2.3507
5.58	5.49515	-1.52057
5.64	5.56835	-1.27032
5.33	5.4083	1.46912
5.14	5.12398	-0.31167
4.85	4.9155	1.35056
4.75	4.72253	-0.57842
5.9	5.88684	-0.22307
5.95	6.00084	0.85442
5.7	5.66845	-0.55349
5.8	5.74635	-0.92493
5.8	5.71485	-1.46812
5.7	5.66594	-0.59761
5.8	5.78775	-0.21122
5.45	5.44365	-0.11661
5.45	5.32035	-2.3789
4.9	4.64657	-5.1721
4.85	4.69073	-3.28384
5.2	5.00192	-3.80933
4.55	4.42466	-2.75464
4.6	4.58319	-0.36541
4.35	4.47072	2.77506
4.6	4.66098	1.32574
4.7	4.75626	1.19698
5.65	5.83563	3.28547
5.25	5.33606	1.63928
5.3	5.15838	-2.67206
5.6	5.86743	4.77557
7.65	7.51734	-1.73408
8.03	7.58151	-5.58516

Error Statistics	
Mean	-0.5453
Standard Error	0.35025
Median	-0.58802
Standard Deviation	2.66742
Variance	7.11511
Kurtosis	0.66119
Skewness	-0.25377
Range	14.2384
Minimum	-8.79851
Maximum	5.43993
Sum	-31.6276
Count	58

Figure 12: Arcelik-Test with Randomly Extracted Block of Data



neural network was trained with the same data set, excluding a block that was extracted randomly. The forecast is then performed for this randomly extracted data.

The results of this test are given in Table 9 and Figure 12. The standard error is as low as the February-April forecast. The skewness and the kurtosis of the data is also very near to that of the previous forecast. The main difference is in the standard deviation; it is 2.5 which is three times as much as the February-April forecast. This is due to the reason that the extracted data belongs to the middle of the year 1992; however, the forecast carries effects of the large increase experienced in 1993, together with the steady trend in 1991 and the first months of 1992. This is the fact that forces the error to be distributed over a frequency range larger than the February-April forecast.

4.3.2 Sarkuysan

Two forecasts were carried out for this stock one being the April-May 1993 and the other being the December 92- April 93. The reason for such a choice again is in the fact that like the Arcelik trend, "external effects" are observed on this trend (explained in Arcelik Section). An observation of the trend for April-May period (figure 13) clearly reveals the issue. The regression model statistics are represented in Tables 10 and 12 for the first and the second test respectively.

Figure 13: Index and Sarkuysan Trend

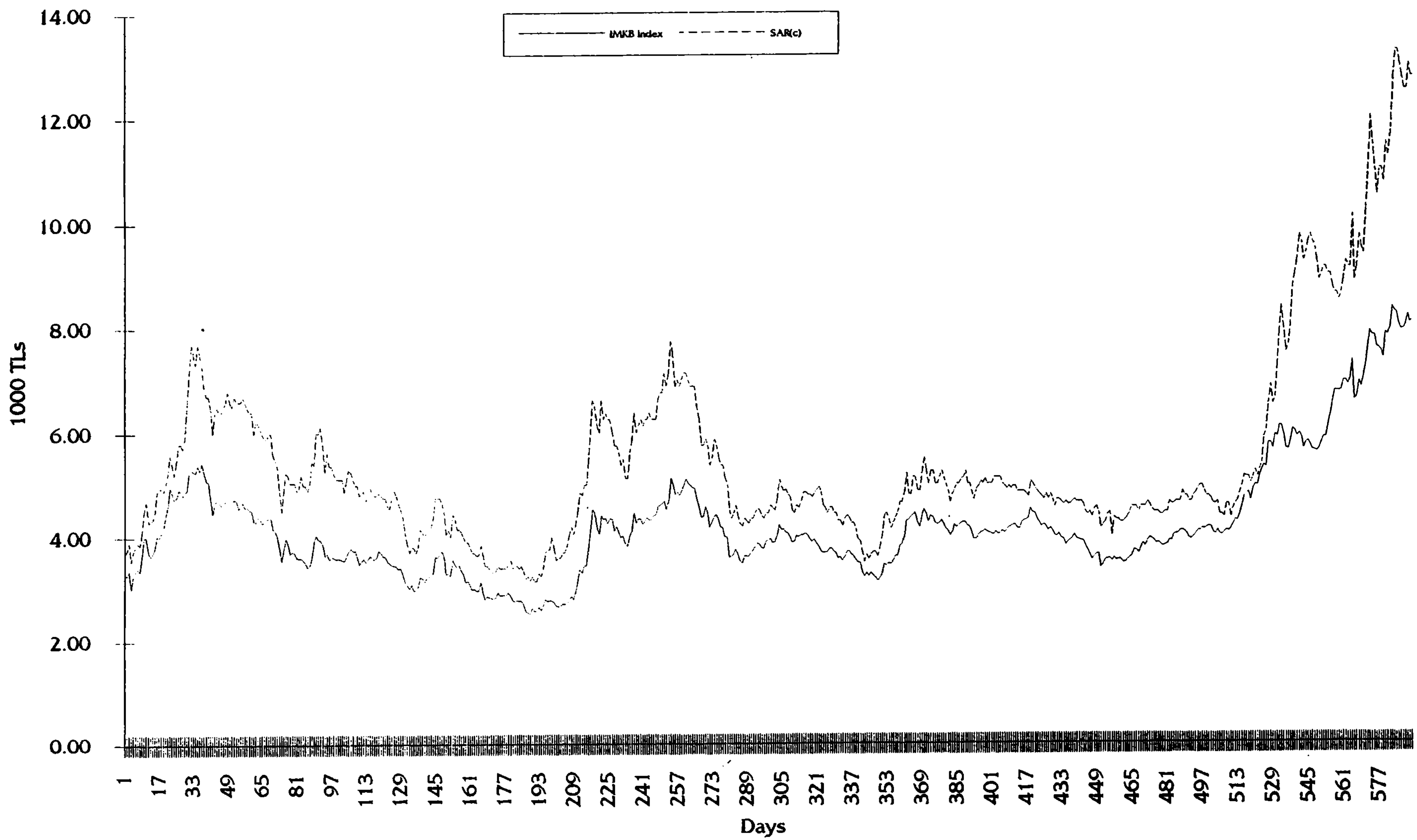


Table 10: Linear Regression Model For Sarkuysan (test 1)

Regression Statistics

Multiple R	0.985536334
R Square	0.971281865
Adjusted R Square	0.970600802
Standard Error	0.159001416
Observations	519

Analysis of Variance

	df	*Sum of Squares*	*Mean Square*	*F*	*Significance F*
Regression	12	432.6548212	36.05456844	1426.13	0
Residual	506	12.79241384	0.02528145		
Total	518	445.4472351			

	Coefficients	*Standard Error*	*t Statistic*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-2.05034039	0.276466779	-7.41622699	5E-13	-2.593504	-1.507176
x1	-0.01850453	0.080595301	-0.229598081	0.81849	-0.176847	0.1398381
x2	0.038677796	0.136594389	0.283158019	0.77717	-0.229684	0.3070398
x3	-0.19897301	0.059951155	-3.318918753	0.00097	-0.316757	-0.081189
x4	1.696860285	0.925498042	1.833456376	0.06731	-0.121432	3.5151522
x5	-0.73348913	0.756869119	-0.969109594	0.33294	-2.220482	0.7535038
x6	0.169567904	0.395210005	0.429057721	0.66806	-0.606887	0.9460224
x7	1.242827154	0.473259045	2.626103329	0.00889	0.3130325	2.1726218
x8	0.653759876	0.339333467	1.926600056	0.05458	-0.012916	1.3204358
x9	0.513528085	0.039774435	12.91100895	2.9E-33	0.4353847	0.5916715
x10	0.796139172	0.044105762	18.05068414	7.3E-57	0.7094862	0.8827921
x11	-0.16841583	0.056043917	-3.005068906	0.00278	-0.278523	-0.058308
x12	0.064847222	0.038023586	1.705447299	0.08871	-0.009856	0.1395508

Regression Model:
Prediction = -2.053 + -0.0185*x1 + 0.0386*x2 + -0.199*x3 + 1.696*x4 + -0.733*x5 + 0.169*x6 + 1.243*x7 + 0.654*x8 + 0.514*x9 + 0.796*x10 + -0.168*x11 + 0.065*x12

Table 12: Linear Regression Model for Sarkuysan (test 2)

Regression Statistics	
Multiple R	0.98843
R Square	0.97699
Adjusted R Square	0.97633
Standard Error	0.15371
Observations	432

Analysis of Variance

	df	*Sum of Squares*	*Mean Square*	*F*	*Significance F*
Regression	12	420.2367	35.01973	1482.27	0
Residual	419	9.899186	0.023626		
Total	431	430.1359			

	Coefficients	*Standard Error*	*t Statistic*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-3.56487	0.326932	-10.904	1.3E-24	-4.20751	-2.92224
x1	-0.06844	0.079524	-0.86063	0.38992	-0.22476	0.08788
x2	-0.27768	0.145388	-1.90993	0.05681	-0.56346	0.0081
x3	-0.04163	0.069494	-0.59907	0.54944	-0.17823	0.09497
x4	3.71457	0.954601	3.891232	0.00012	1.83817	5.59098
x5	0.30389	0.781867	0.388667	0.69771	-1.23298	1.84076
x6	-1.60885	0.464361	-3.46465	0.00058	-2.52162	-0.69608
x7	2.87051	0.572679	5.012427	7.9E-07	1.74483	3.99619
x8	1.08206	0.496458	2.179558	0.02983	0.1062	2.05792
x9	0.71745	0.046186	15.53408	1.6E-43	0.62667	0.80824
x10	0.67197	0.046636	14.40869	1.1E-38	0.5803	0.76364
x11	-0.1638	0.056493	-2.89945	0.00393	-0.27484	-0.05275
x12	0.07332	0.038239	1.917534	0.05583	-0.00184	0.14849

Regression Model:
Prediction= -3.564 + -0.068*x1 + -0.277*x2 + -0.041*x3 + 3.714*x4 + 0.303*x5 + -1.608*x6 + 2.870*x7 + 1.082*x8 + 0.717*x9 + 0.671*x10 + -0.163*x11 + 0.073*x12

There are drastic differences between the two forecasts. The actual data, neural network predictions, regression model predictions and the corresponding percent errors together with detailed descriptive statistics are presented in Table 11 for the first test and in Table 13 for the second.

4.3.2.1 April-May 1993 Forecast

The mean of the error in the first forecast is a shocking -9.07, together with a median of -9.88. These, with no need for further investigation, suggests that the forecast is of no good. For the stock exchange, where the absolute maximum change is 10% daily, a standard deviation of 5.39 is surely a very poor result. However, an important fact observed here Is that the regression model outputs are extremely close to the neural network predictions, suggesting that the neural network forecast's being not good must not be considered as a flaw of the method.

The reason underlying this result can be understood if the trend of Sarkuysan, together with the index is examined. For April and May, the price goes up eminently, even faster (in percentage) than the Arcelik's. The neural network simply could not *catch up with* such an increase, for the reason that this increase can not be observed from the input variables (the external effects discussion again). The forecasted values(both neural network and regression model) are (figure 14) almost

Table 13: Comparative Prediction Results for Sarkuysan Test 1

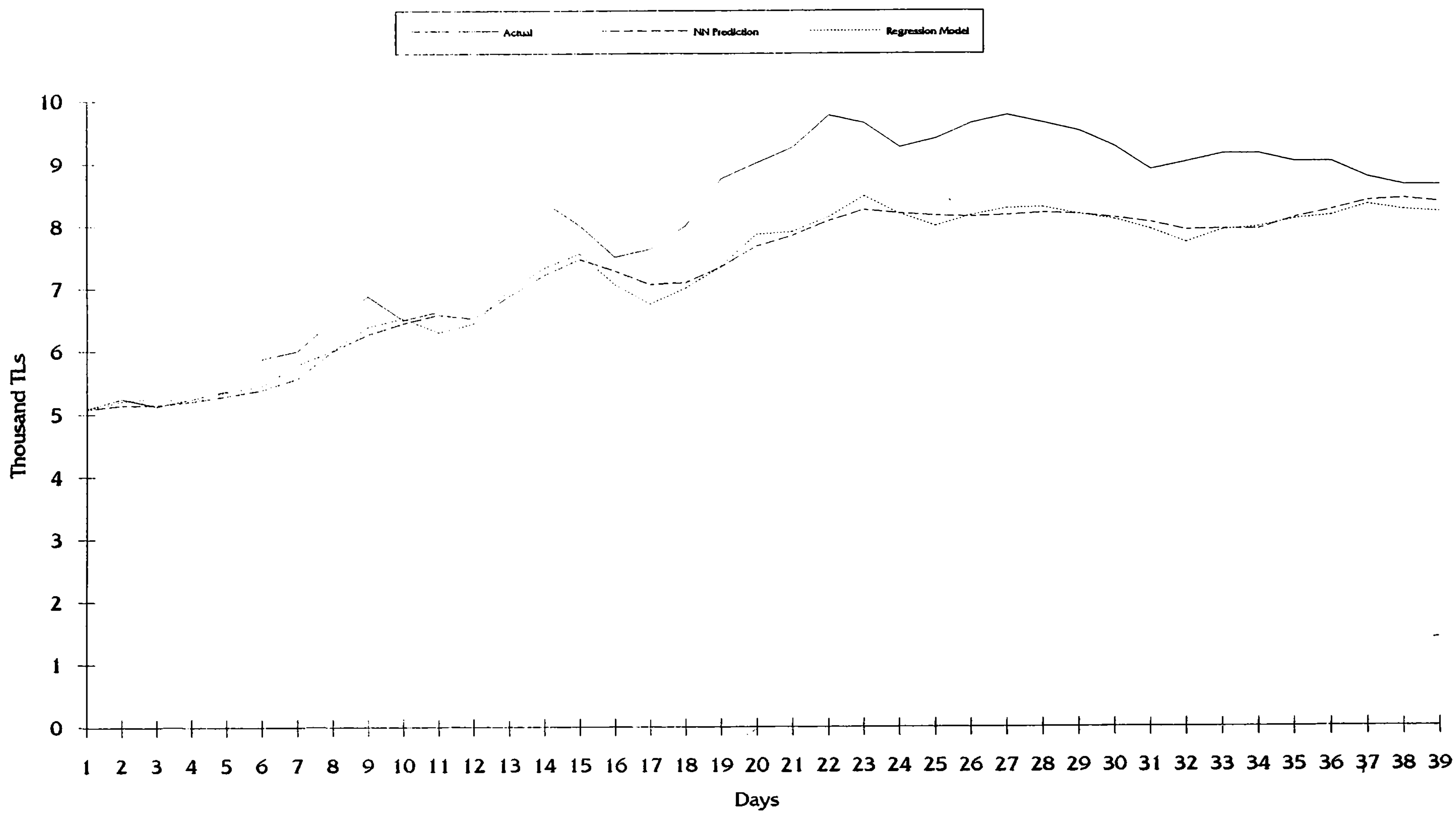
Sarkuysan Prediction 1

Actual	NN Prediction	Regression Model	Error NN	Error Regression
5.1	5.08734	5.08622	0.2482	-0.27012
5.25	5.14226	5.21135	2.05217	0.73623
5.13	5.15079	5.25929	0.40522	2.5203
5.25	5.20906	5.20479	0.77981	0.8611
5.38	5.29656	5.36306	1.55097	0.31494
5.88	5.38953	5.45147	-8.34141	7.28786
6	5.56566	5.78479	7.239	3.58677
6.5	6.00312	6.01031	7.64435	7.53371
6.88	6.26783	6.39002	-8.89783	-7.12176
6.5	6.4414	6.52742	-0.90157	0.42181
6.63	6.5775	6.30382	-0.79189	-4.91969
7.25	6.51972	6.43542	10.0728	-11.2356
7.88	6.8703	6.96657	-12.8134	-11.5918
8.38	7.21329	7.32871	-13.9226	-12.5452
8	7.46188	7.55124	-6.72648	5.60948
7.5	7.27921	7.06536	2.94381	-5.79514
7.63	7.06655	6.75285	-7.38469	-11.496
8	7.09259	6.9994	-11.3426	-12.5075
8.75	7.34563	7.35058	-16.05	-15.9934
9	7.67718	7.86602	-14.698	-12.5998
9.25	7.84634	7.90933	15.1747	14.4937
9.75	8.07936	8.13707	-17.1348	-16.5428
9.63	8.2567	8.4706	-14.2606	-12.0395
9.25	8.19971	8.19225	-11.3544	-11.4351
9.38	8.15562	7.99649	-13.0531	-14.7496
9.63	8.14583	8.16556	-15.4119	-15.207
9.75	8.16373	8.277	-16.2694	-15.1077
9.63	8.19659	8.28964	-14.8848	-13.9186
9.5	8.17868	8.17716	13.9087	-13.9246
9.25	8.12006	8.0916	-12.2156	-12.5232
8.88	8.04473	7.93719	-9.4062	-10.6172
9	7.92324	7.72493	-11.9641	-14.1675
9.13	7.93707	7.91755	-13.0661	-13.2798
9.13	7.93716	7.96948	-13.065	-12.7111
9	8.1107	8.08933	-9.88112	-10.1186
9	8.23216	8.14582	-8.5316	-9.49086
8.75	8.37422	8.31518	-4.29469	-4.96939
8.63	8.40894	8.23552	-2.56154	-4.57097
8.63	8.35703	8.20085	-3.16304	-4.97275

Error Statistics

	Neural Net	Regression
Mean	9.06661	9.07446
Standard Error	0.86319	0.85186
Median	9.88112	11.2356
Standard Deviation	5.3906	5.31983
Variance	29.0586	28.3006
Kurtosis	1.20718	0.82573
Skewness	0.36128	0.58772
Range	17.54	19.0631
Minimum	17.1348	16.5428
Maximum	0.40522	2.5203
Sum	353.598	-353.904
Count	39	39

Figure 14: Sarkuysan Test 1 Forecast Results



always lower than the actual, with very high error percentages. Although there exists an offset error of nearly -10 percent, the rise and fall times of the trend can be said to be predicted with high accuracy. This is the main difference between the neural network predictions and the regression model outputs. The neural network eliminates the lag that exists in the linear regression outputs.

4.3.2.2 December 1992- April 1993 Forecast

In the second test things changed extremely. The actual price trend for Sarkuysan for the December 92-April 93 period were very *steady*, and the rises and falls of the price were very near to that of the index. Therefore, as depicted, the neural network forecast is very admirable (Table 13 and Figure 15). The mean is only -0.57, even better than the best Arcelik trend forecast. The regression model was rebuild for test two so as to maintain compatibility with the neural network forecasts(for the two tests, two different linear regression models were built, first one with 518 data points, the second one with 432 data points).

Even though the other statistics are comparable, the mean of the regression model outputs are -8.77, which is much more worse than the neural network's. Yet, the standard deviation and all the other statistics are at least as powerful as the neural network's. And also the rise

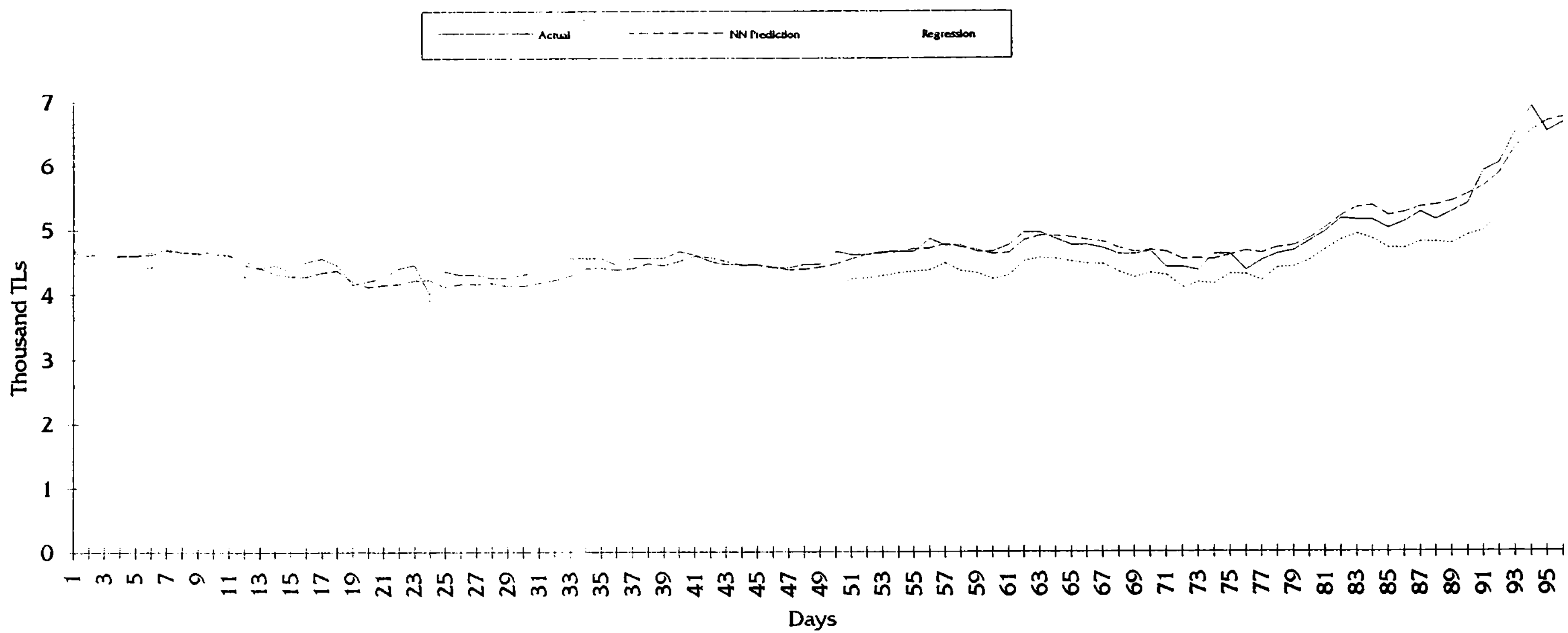
Table 13: Comparative Prediction Results for Sarkuysan Test 2

Actual	NN Prediction	Regression	Error NN(%)	Regression Error(%)
4.65	4.67427	4.48847	0.52196	-3.4737
4.6	4.63928	4.4364	0.85389	-3.5566
4.65	4.54389	4.33188	-2.2818	-6.8412
4.6	4.59058	4.40692	-0.2047	-4.1973
4.6	4.58895	4.38138	-0.2402	-4.7525
4.65	4.6114	4.41573	-0.83	-5.0381
4.7	4.6778	4.50026	-0.4724	-4.2497
4.65	4.64662	4.4579	-0.0726	-4.1312
4.65	4.62804	4.40486	-0.4722	-5.2717
4.65	4.61312	4.40204	-0.7932	-5.3325
4.6	4.60192	4.38953	0.0418	-4.5755
4.45	4.51908	4.27742	1.55231	-3.8783
4.4	4.40531	4.12515	0.12064	-6.2467
4.45	4.31471	4.04376	-3.0402	-9.129
4.35	4.27299	4.01121	-1.7705	-7.7884
4.5	4.26969	3.97435	-5.118	-11.681
4.55	4.33548	4.11352	-4.7148	-9.5929
4.45	4.36494	4.11815	-1.9114	-7.4574
4.15	4.19812	3.85758	1.15957	-7.0463
4.2	4.11382	3.71349	-2.052	-11.584
4.25	4.142	3.83959	-2.5412	-9.6566
4.4	4.15235	3.85115	-5.6283	-12.474
4.45	4.20863	3.94211	-5.4241	-11.413
4	4.21697	3.91092	5.4243	-2.2271
4.35	4.11005	3.64949	-5.5161	-16.104
4.3	4.15588	3.91176	-3.3517	-9.0288
4.3	4.15555	3.80021	-3.3593	-11.623
4.25	4.16878	3.82507	-1.9112	-9.9984
4.25	4.1282	3.73465	-2.8659	-12.126
4.3	4.12792	3.73999	-4.0019	-13.024
4.35	4.16721	3.79828	-4.202	-12.683
4.45	4.20877	3.83732	-5.4209	-13.768
4.55	4.27447	3.91908	-6.0556	-13.866
4.55	4.38723	4.04524	-3.5773	-11.094
4.55	4.39998	4.02755	-3.2972	-11.482
4.45	4.37096	3.98613	-1.7761	-10.424
4.55	4.38916	3.98478	-3.535	-12.422
4.55	4.46527	4.12341	-1.8622	-9.3755
4.55	4.43658	4.05102	-2.4927	-10.967
4.65	4.49685	4.12919	-3.2935	-11.2
4.6	4.57673	4.24093	-0.506	-7.8059
4.5	4.55437	4.1768	1.20813	-7.1822
4.45	4.50117	4.10473	1.14991	-7.7588
4.45	4.4298	4.04329	-0.454	-9.1396
4.45	4.43836	4.06209	-0.2617	-8.717
4.4	4.40923	4.03091	0.20968	-8.3883
4.4	4.36542	3.96945	-0.786	-9.7852
4.45	4.37553	3.99784	-1.6735	-10.161
4.45	4.40103	4.04467	-1.1004	-9.1085
4.65	4.45686	4.09413	-4.1536	-11.954
4.6	4.53282	4.22503	-1.4605	-8.1515
4.6	4.60421	4.24221	0.09146	-7.778
4.65	4.62684	4.27885	-0.4981	-7.9818
4.65	4.6482	4.31615	-0.0386	-7.1436
4.65	4.68931	4.34148	0.84544	-7.4175
4.85	4.69762	4.35967	-3.1419	-7.1941
4.75	4.76694	4.46984	0.35655	-6.2325
4.75	4.72041	4.3459023	-0.6229	-7.9338
4.65	4.67813	4.31787	0.6049	-7.7008
4.65	4.614	4.22878	-0.7742	-8.3489
4.75	4.62689	4.27431	-2.5918	-7.6202
4.95	4.82677	4.49989	-2.4895	-6.7722
4.95	4.89566	4.55054	-1.0979	-7.0494
4.85	4.89512	4.53252	0.93039	-7.4075
4.75	4.87325	4.48856	2.59482	-7.8939
4.75	4.8292	4.45198	1.66745	-7.8113
4.7	4.80147	4.44478	2.15902	-7.4288
4.6	4.6994	4.32781	2.16091	-7.9072
4.6	4.63142	4.24136	0.68311	-8.422
4.65	4.66457	4.304	0.3134	-7.73
4.4	4.63199	4.27741	5.27243	-7.6549
4.4	4.52173	4.08291	2.76648	-9.7046
4.35	4.5275	4.16015	4.08034	-8.1135
4.6	4.51616	4.13938	-1.8226	-8.3428
4.6	4.58645	4.288	-0.2947	-6.5071
4.35	4.64509	4.2798	6.78372	-7.864
4.5	4.61577	4.1908	2.57256	-9.2068
4.6	4.69989	4.38587	2.17146	-6.6813
4.65	4.73033	4.40066	1.72759	-6.9693
4.8	4.83877	4.50001	0.80779	-7.0009
4.95	5.00699	4.66083	1.15129	-6.9134
5.15	5.19202	4.82205	0.81583	-7.1256
5.13	5.32319	4.91109	3.76585	-7.7416
5.13	5.34711	4.8338	4.23211	-9.5997
5	5.19768	4.68921	3.95352	-9.7825
5.1	5.24214	4.681	2.78696	-10.704
5.25	5.32789	4.7871	1.48358	-10.15
5.13	5.35695	4.78183	4.42405	-10.736
5.25	5.41174	4.75408	3.08084	-12.153
5.38	5.52547	4.89277	2.70387	-11.451
5.88	5.64083	4.95672	-4.0676	-12.128
6	5.842	5.19312	-2.6333	-11.107
6.5	6.25216	5.47066	-3.813	-12.5
6.88	6.51003	5.76515	-5.3775	-11.442
6.5	6.65237	5.80992	2.3442	-12.664
6.63	6.71887	5.66512	1.34048	-15.683

Error Statistics

	Neural Net	Regresion
Mean	-0.570683397	-8.777417108
Standard Error	0.279747864	0.280657528
Median	-0.472288149	-8.247142611
Standard Deviation	2.769362561	2.749870947
Variance	7.669368997	7.561790226
Kurtosis	-0.330168984	-0.068873313
Skewness	0.164768102	-0.156228067
Range	12.83932853	13.8765942
Minimum	-6.055604396	-16.1036967
Maximum	6.783724138	-2.2271025
Sum	-55.92697289	-842.6320423
Count	96	96

Figure 15: Sarkuysan Test 2



and the fall times of the stock are outputted as good as the neural network predictions. The only problem is that the regression model has an offset value of approximately -9.

The skewness of the of the error distribution of the neural network predictions suggest that the error is distributed with an asymmetric tail towards more positive values. However this effect is not serious at all, as the value is 0.16. The kurtosis is also lower than the Arcelik forecasts, and the distribution is flatter this time, which adds value to the forecast results. The low skewness is reflected to the maximum and the minimum values of the error, the values are very close to each other, 6.78 and -6.06.

The relatively higher standard deviation (compared to Arcelik forecasts) justifies the above maximum and minimum values, which yield a range of 12.83. The total number of forecasted data is 98 this time(December 92 - April 93).

4.3.3 Kepez

The speculations made over Kepez for this time period caused its price to rise sharply and then fall (nearly from TL 7000 to 11000 in a week), causing and unusual peak in its price (Figure 16). Even though the predictions tried to keep up with the rise, the fall was nearly impossible for the network to follow up, so as observed from Figure 17. Before this peak, the mean of the error

Figure 16: Index and Kepez Trend for January1991 -June 1993

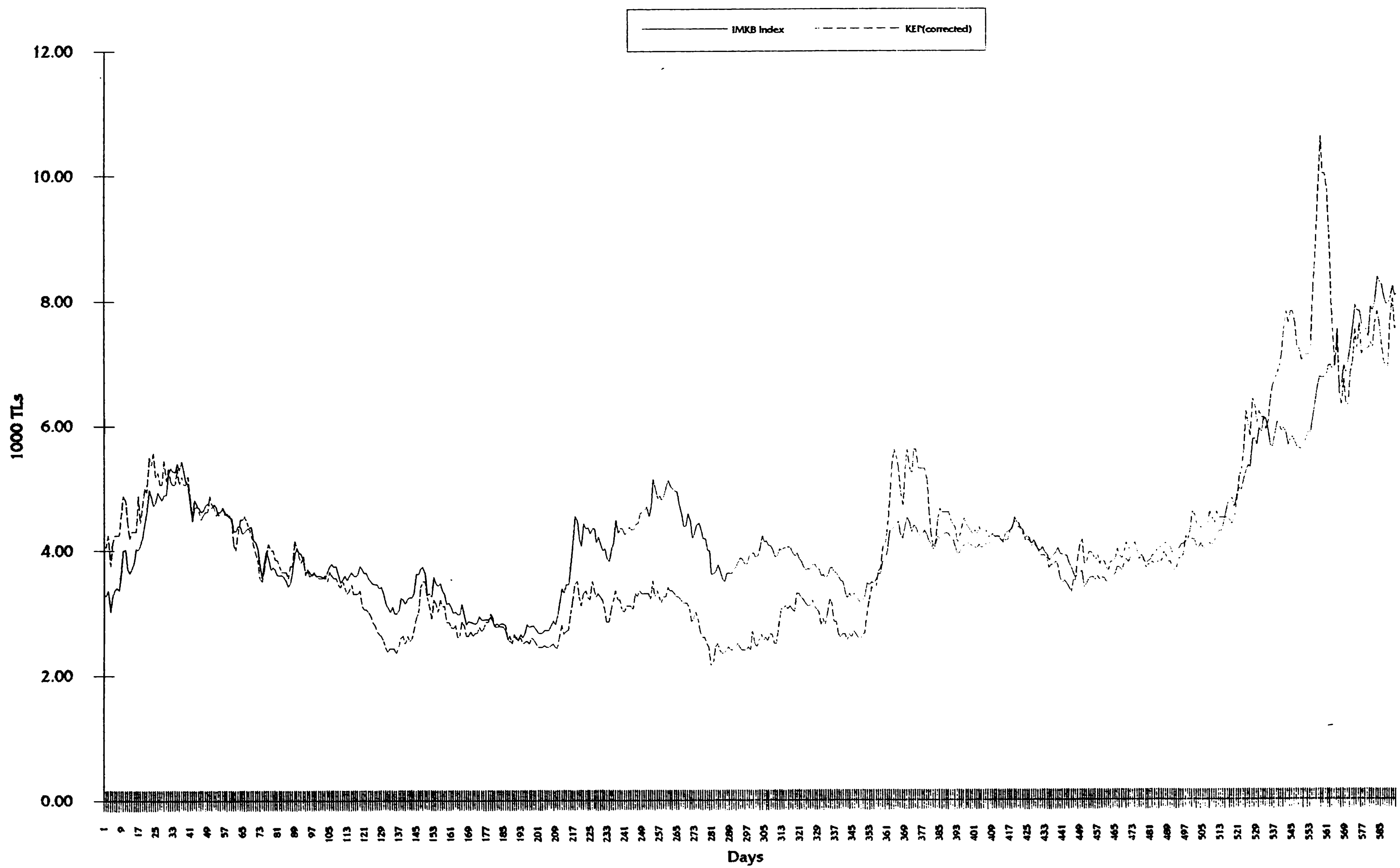


Table 14: Linear Regression Model For Kepez

Regression Statistics	
Multiple R	0.986759
R Square	0.973693
Adjusted R Square	0.973092
Standard Error	0.152539
Observations	538

Analysis of Variance

	df	*Sum of Squares*	*Mean Square*	*F*	*Significance F*
Regression	12	452.15032	37.67919	1619.336	0
Residual	525	12.215854	0.023268		
Total	537	464.36617			

	Coefficients	*Standard Error*	*t Statistic*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	0.334092	0.2420774	1.380102	0.168129	-0.14147	0.80965
x1	0.089391	0.0768204	1.163636	0.245088	-0.06152	0.240304
x2	-0.16726	0.1283586	-1.30306	0.193113	-0.41942	0.0849
x3	0.046882	0.0581956	0.805591	0.420835	-0.06744	0.161207
x4	1.147547	0.8639141	1.328311	0.18464	-0.5496	2.844698
x5	-3.06423	0.7859486	-3.89877	0.000109	-4.60822	-1.52024
x6	0.12744	0.3880798	0.328386	0.742748	-0.63494	0.889819
x7	1.583746	0.4485989	3.530426	0.000451	0.702477	2.465014
x8	-0.24644	0.0874528	-2.81797	0.005011	-0.41824	-0.07464
x9	0.091396	0.0178844	5.110394	4.48E-07	0.056263	0.12653
x10	0.94286	0.0435557	21.64724	3.53E-75	0.857295	1.028425
x11	-0.08314	0.0597278	-1.39192	0.164522	-0.20047	0.034198
x12	0.019699	0.0427037	0.461298	0.644772	-0.06419	0.10359

Regression Model:
Prediction= 0.334092 + 0.089*x1 + -0.167*x2 + 0.046*x3 + 1.147*x4 + -3.064*x5 +
0.127*x6 + 1.583*x7 + -0.246*x8 + 0.091*x9 + 0.942*x10 + -0.083*x11 + 0.020*x12

Table 15: Comparative Prediction Results for Kepez

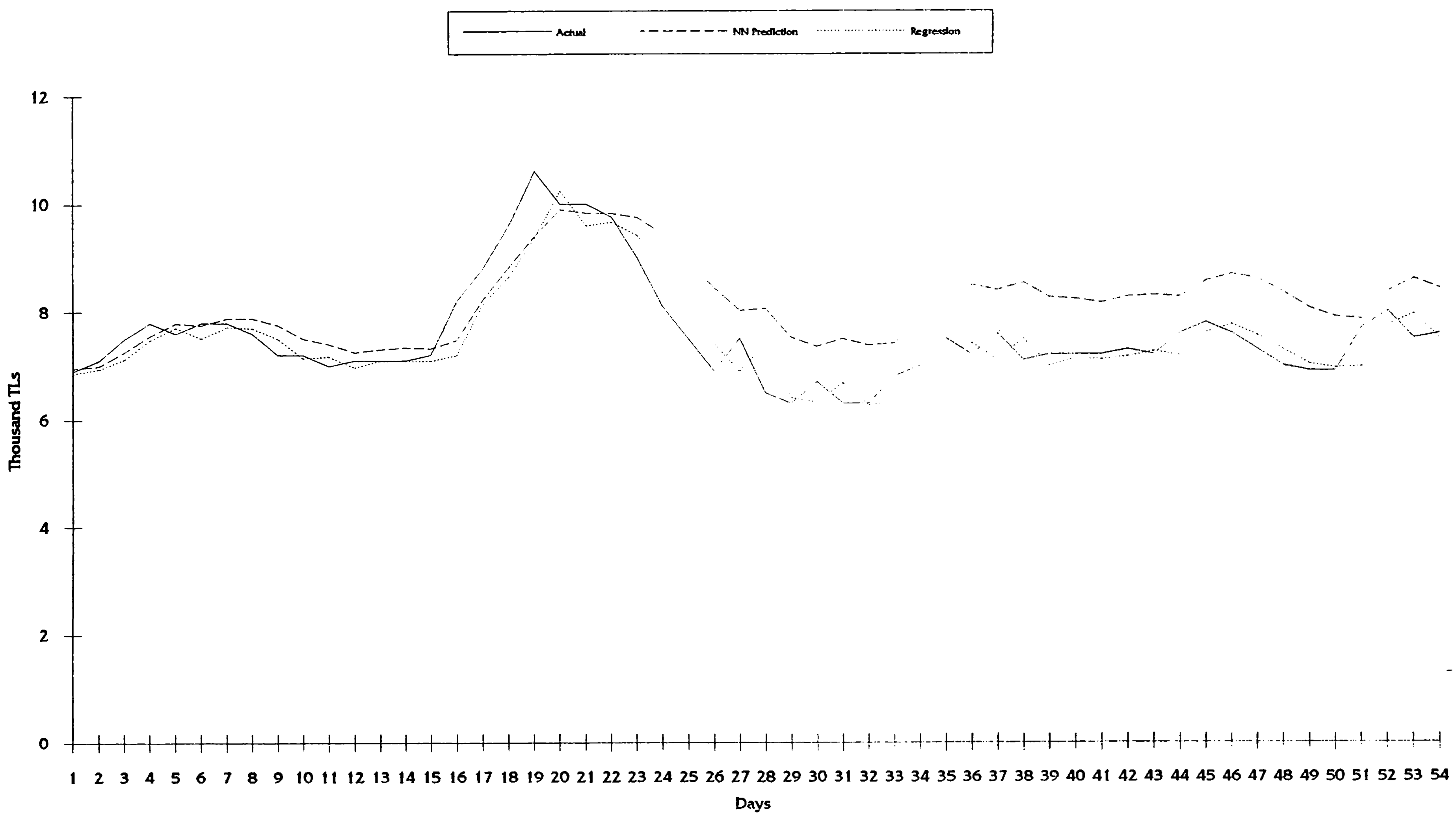
Kepez Forecast Results

Actual	NN Prediction	Regression	NN Error (%)	Regression Error (%)
6.9	6.95429	6.860321	0.78674	2.42494
7.1	6.99952	6.938029	-1.41521	0.71871
7.5	7.26084	7.123402	-3.18876	-2.0213
7.8	7.56053	7.478126	-3.07009	-1.12659
7.6	7.79193	7.711236	2.52534	4.46363
7.8	7.75981	7.514034	-0.51522	-0.66624
7.8	7.88019	7.727015	1.0281	2.0643
7.6	7.88297	7.698326	3.7233	4.29377
7.2	7.75484	7.503925	7.70615	7.22118
7.2	7.50276	7.141648	4.20504	2.18955
7	7.39791	7.16754	5.68447	5.39342
7.1	7.24828	6.976133	2.08842	1.2554
7.1	7.30416	7.095583	2.87549	2.93779
7.1	7.33674	7.090874	3.33442	2.87146
7.2	7.32098	7.091864	1.68024	1.49811
8.2	7.46932	7.193681	-8.91068	9.27218
8.8	8.21537	8.13768	-6.6435	-4.52637
9.6	8.81929	8.632265	-8.13245	-7.08058
10.6	9.39847	9.362752	-11.3352	-8.67215
10	9.89601	10.23851	-1.03989	5.38513
10	9.8287	9.595485	-1.713	-1.04515
9.75	9.82449	9.655609	0.76403	2.03189
9	9.74746	9.412482	8.30507	7.58313
8.1	9.43775	8.716276	16.5154	10.6083
7.5	8.90712	7.9079	18.7616	8.43867
6.9	8.43772	7.40093	22.2857	10.2599
7.5	8.01985	6.890393	6.93136	-5.12809
6.5	8.05531	7.412384	23.9279	17.0367
6.3	7.52885	6.400588	19.5055	4.59663
6.7	7.35227	6.32441	9.73543	-2.60582
6.3	7.49201	6.676857	18.9208	8.98186
6.3	7.37638	6.269757	17.0854	2.51995
6.8	7.4073	6.31943	8.93093	-4.06721
7	7.82239	6.802348	11.7484	0.1764
7.5	8.14055	6.967718	8.54068	-4.09709
7.2	8.49389	7.412793	17.9707	5.95546
7.6	8.39505	7.098567	10.4611	-3.5978
7.1	8.53256	7.495909	20.1769	8.57618
7.2	8.26352	6.989586	14.7711	0.07758
7.2	8.22898	7.134715	14.2913	2.09326
7.2	8.15928	7.111383	13.3234	1.76921
7.3	8.27476	7.163242	13.3529	1.1266
7.2	8.29771	7.258765	15.2459	3.81618
7.6	8.27115	7.175121	8.83091	-2.59051
7.8	8.56327	7.604389	9.78554	0.49217
7.6	8.68909	7.756996	14.3301	5.06574
7.3	8.60364	7.561878	17.8581	6.58737
7	8.34744	7.287278	19.2492	7.10397
6.9	8.05575	7.022094	16.7499	4.76947
6.9	7.89705	6.953512	14.4501	3.77553
7.7	7.85767	6.969363	2.0476	-6.4888
8	8.366	7.744551	4.575	-0.19311
7.5	8.6058	7.952627	14.744	9.03502
7.6	8.42302	7.48344	10.8292	1.46631

Error Statistics

	NN	Regression
Mean	8.04954	2.10152
Standard Error	1.18195	0.70813
Median	8.68579	2.07878
Standard Deviatior	8.68555	5.2037
Variance	75.4388	27.0785
Kurtosis	-0.76053	0.38599
Skewness	-0.23301	0.07729
Range	35.2631	26.3089
Minimum	-11.3352	-9.27218
Maximum	23.9279	17.0367
Sum	434.675	113.482
Count	54	54

Figure 17: Kepez Forecast Results



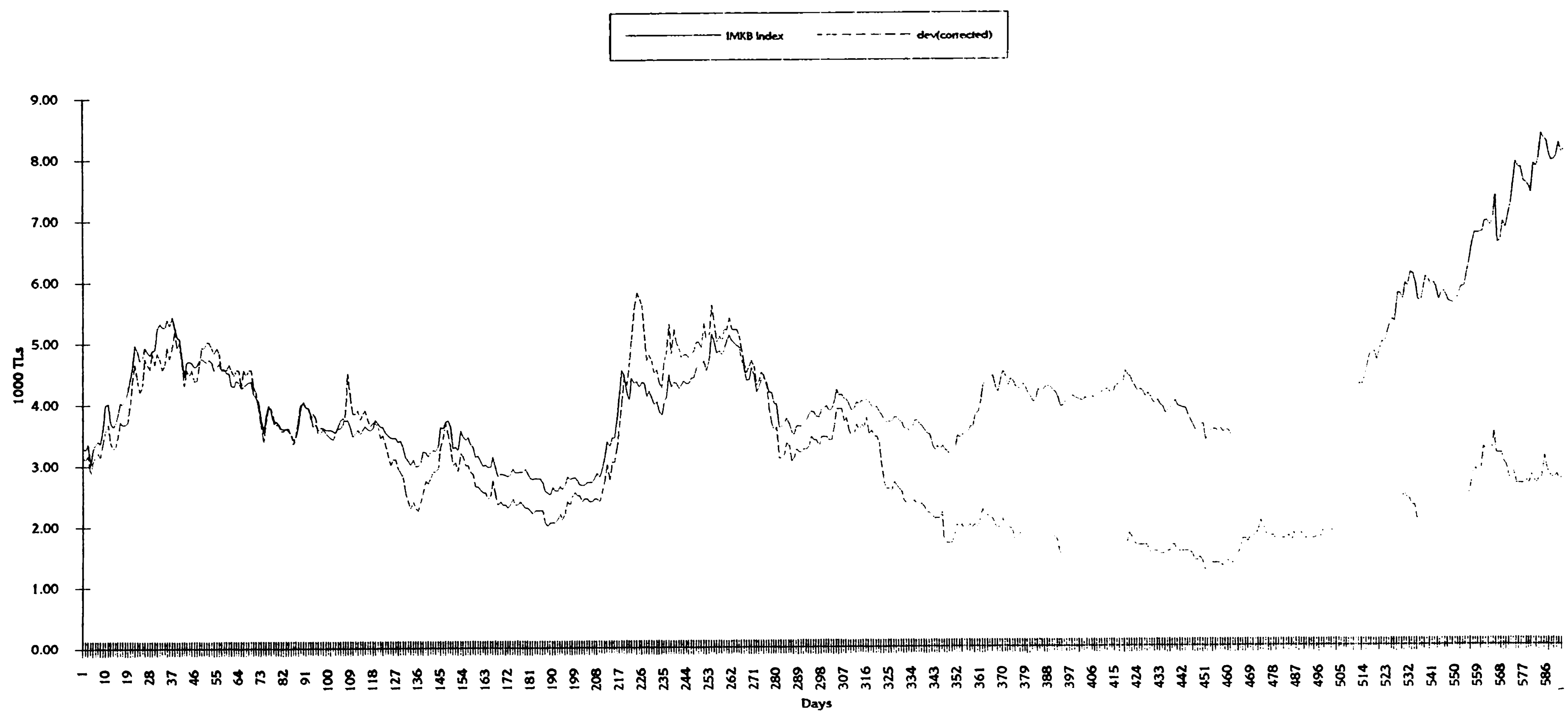
is just 2.6, whereas after the peak it skyrockets to 11.12%. Table 14 represents the linear regression model for Kepez. The comparative results for the forecasts are given in Table 15. The comparison of the neural network predictions with the regression model show that this after-the-peak symptom was not a problem for the regression model, and the mean of the error is as low as 2.1. The regression model outputs follow up the actual price, whereas after the peak an offset was is observed for the Neural Network predictions.

Having such high values for errors encouraged me to train the network further, up to 21,000 steps. The results improved, yet were still far away from what we can call as successful. There is a slight improvement, which brought the error down to 8.04 and the standard deviation to 8.68. Therefore, the result of the comparison of the predictions made on the price of Kepez reveal out an important fact: Even the 21,000 steps-trained neural network could not make any healthy forecast in the existence of an instant rise and fall.

4.3.4 Deva

The last stock that was used in my forecast sessions was Deva. The reason for this choice lies in the fact that in the period of concern, contradicting the increase in the index and in the prices of Arcelik, Sarkuysan and Kepez, the price of Deva exhibited a non-increasing trend(figure

Figure 18: Index and Deva Trend for January 1991- June 1993



18). The regression model statistics for Deva are in Table 16.

In presence of such a trend, neural network prediction results are difficult to foretell. The results were in general encouraging. The error has a nearly symmetric distribution(skewness=0.24), with mean -1.21 and standard deviation 4.44 (Table 17). The high standard deviation suggests that the small value of the mean is just a coincidence, as the range of error is significantly high.

The decline of the trend for the forecasted period can be said to be captured by the neural network forecasts, whereas the regression model can not at all catch up with the decline, as it was not expected.

Therefore an analysis follows: Even though the standard deviation is high, the forecast results are successful, because the neural network managed to sense the decline in the price, whereas the regression model could not at all.

Yet, at any single point, the neural network is not expected to make accurate forecasts, as the standard deviation is high; but when the overall error is considered, the positive errors cancel the negative values and the low mean value follows(figure 19). The steady price for Deva is not supported by the variables

Table 16: Linear Regression Model For Deva

Regression Statistics

Multiple R	0.9933049
R Square	0.9866546
Adjusted R Square	0.9863495
Standard Error	0.1378584
Observations	538

Analysis of Variance

	df	Sum of Squares	Mean Square	F	Significance F
Regression	12	737.66284	61.471903	3234.5231	0
Residual	525	9.977591	0.0190049		
Total	537	747.64043			

	Coefficients	Standard Error	t Statistic	P-value	Lower 95%	Upper 95%
Intercept	-1.390768	0.2242421	-6.202082	1.113E-09	-1.83129	-0.950246
x1	-0.017144	0.0694104	-0.247002	0.8050012	-0.153501	0.1192115
x2	0.2305408	0.1162579	1.9830124	0.0478751	0.0021533	0.4589282
x3	-0.413	0.0619518	-6.666466	6.517E-11	-0.534703	-0.291296
x4	2.3071452	0.7931235	2.9089357	0.0037768	0.7490618	3.8652286
x5	0.3292499	0.6320822	0.5208972	0.6026531	-0.91247	1.5709693
x6	-0.138504	0.3390505	-0.408505	0.6830658	-0.804565	0.5275576
x7	-0.951809	0.3951777	-2.408559	0.0163515	-1.728131	-0.175486
x8	0.5061724	0.088551	5.7161709	1.809E-08	0.3322149	0.6801299
x9	0.2625593	0.0272109	9.649031	2.011E-20	0.2091036	0.3160149
x10	0.8889056	0.0435205	20.425008	4.815E-69	0.8034101	0.9744011
x11	-0.043059	0.0582612	-0.739065	0.4601905	-0.157512	0.0713947
x12	-0.018542	0.0400377	-0.463112	0.6434715	-0.097196	0.0601117

Regression Model:
Prediction= -1.39 + -0.017*x1 + 0.230*x2 + -0.413*x3 + 2.307*x4 + 0.329*x5 + -0.138*x6 + -0.951*x7 + 0.506*x8 + 0.262*x9 + 0.888*x10 + -0.043*x11 + -0.018*x12

Table 17: Comparative Prediction Results for Deva

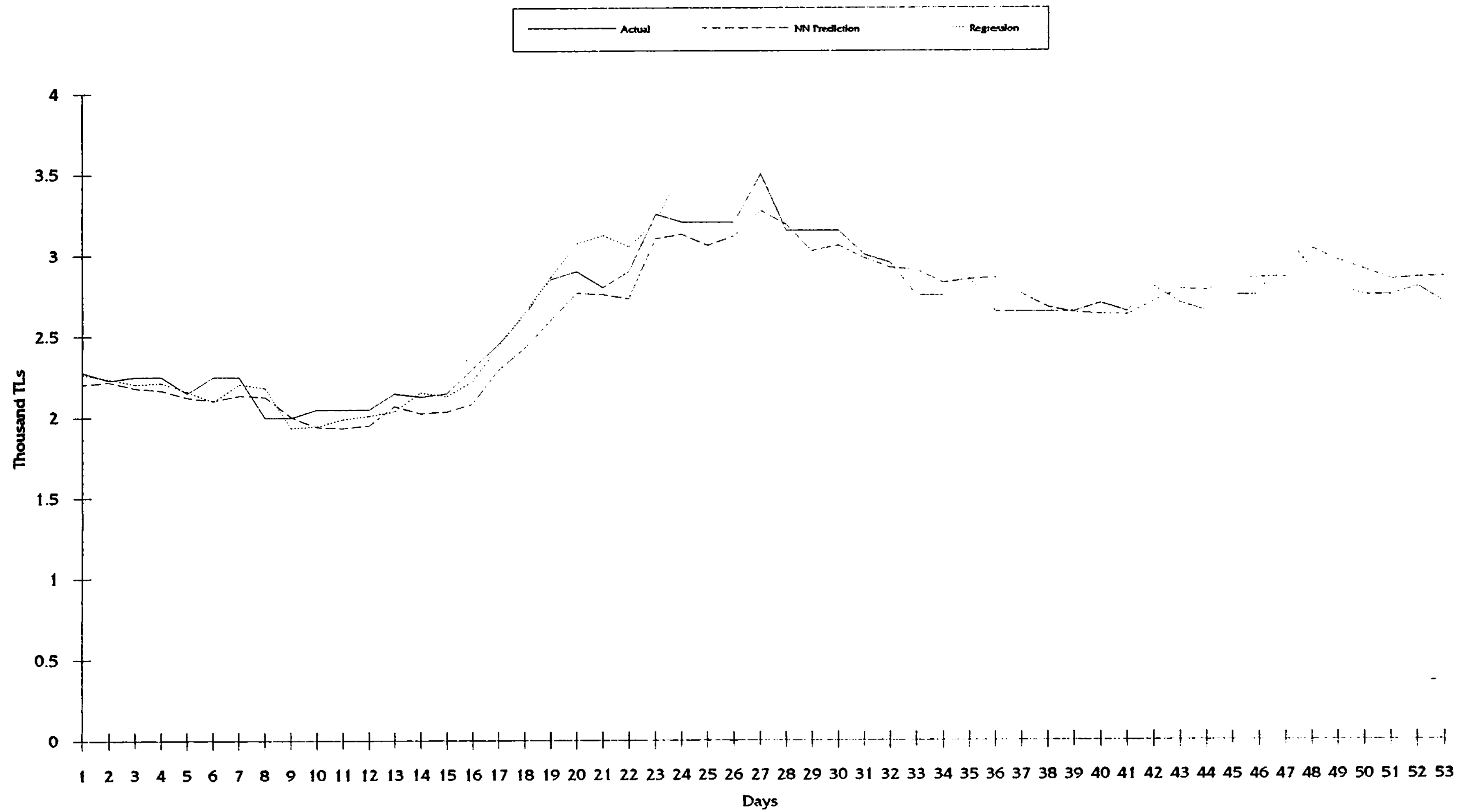
Deva Prediction

Actual	NN Prediction	Regression	NN Error(%)	Regression Error(%)
2.28	2.20604	2.267463	-3.2439	-0.549869
2.23	2.2188	2.237382	-0.5022	0.331022
2.25	2.18184	2.205971	-3.0292	-1.956828
2.25	2.16733	2.214161	-3.6743	-1.59285
2.15	2.12455	2.160933	-1.1838	0.508519
2.25	2.10523	2.100361	-6.4343	-6.650607
2.25	2.13789	2.204531	-4.9829	-2.020866
2	2.12913	2.184107	6.4566	9.205338
2	2.00388	1.936876	0.1941	-3.156212
2.05	1.94256	1.94701	-5.2408	-5.023915
2.05	1.93594	1.991749	-5.5638	-2.84152
2.05	1.95263	2.00934	-4.7496	-1.983404
2.15	2.06969	2.038709	-3.7353	-5.176338
2.13	2.02691	2.154728	-4.8401	1.16092
2.15	2.03788	2.130685	-5.2147	-0.89838
2.3	2.08526	2.225327	-9.3365	-3.246647
2.45	2.29381	2.438788	-6.375	-0.457635
2.63	2.42804	2.646223	-7.6792	0.616832
2.85	2.59699	2.862962	-8.8776	0.454819
2.9	2.76583	3.069686	-4.6264	5.851256
2.8	2.75658	3.121321	-1.5506	11.47576
2.9	2.73212	3.049701	-5.7889	5.162096
3.25	3.10153	3.205725	-4.5682	-1.362302
3.2	3.12727	3.535157	-2.2727	10.47365
3.2	3.05937	3.473821	-4.3947	8.556918
3.2	3.11654	3.516673	-2.608	9.896024
3.5	3.27389	3.634636	-6.4603	3.84673
3.15	3.18698	3.719036	1.17403	18.06463
3.15	3.02461	3.423549	-3.9807	8.6841
3.15	3.0575	3.534021	-2.9366	12.19113
3	2.97823	3.53123	-0.7258	17.70768
2.95	2.92005	3.46913	-1.0154	17.59764
2.75	2.90502	3.506811	5.63716	27.52039
2.75	2.82624	3.442642	2.77225	25.187
2.85	2.85084	3.556556	0.02954	24.79143
2.65	2.85639	3.644553	7.78842	37.5303
2.65	2.75644	3.445073	4.01657	30.00275
2.65	2.67475	3.37604	0.93385	27.39773
2.65	2.64475	3.356309	-0.198	26.65318
2.7	2.63244	3.325931	-2.5023	23.18261
2.65	2.62857	3.321353	-0.8088	25.33406
2.8	2.71097	3.383858	-3.1797	20.85206
2.7	2.78313	3.487625	3.0787	29.1713
2.65	2.77874	3.413072	4.85819	28.79518
2.75	2.84038	3.457186	3.28658	25.71585
2.75	2.85759	3.513736	3.91244	27.7722
3.1	2.85933	3.481262	-7.7637	12.29877
2.9	3.0337	3.719491	4.61038	28.2583
2.8	2.95961	3.482317	5.70036	24.36845
2.75	2.90434	3.380665	5.61225	22.93329
2.75	2.8435	3.348688	3.39993	21.77047
2.8	2.85809	3.386059	2.07464	20.93067
2.7	2.8635	3.373476	6.05537	24.94354
2.8	2.92454	3.278876	4.44779	17.10272

Error Statistics

	Neurai Net	Regression
Mean	-1.21744	12.154863
Standard Error	0.60608	1.7007686
Median	-1.91165	10.974704
Deviation	4.45376	12.498046
Variance	19.83598	156.20115
Kurtosis	-0.95843	-1.438414
Skewness	0.242671	0.1177729
Range	17.12489	44.180902
Minimum	-9.33648	-6.650607
Maximum	7.788415	37.530295
Sum	-65.742	656.36261
Count	54	54

Figure 19: Deva Prediction Results



that I formed as inputs for my network. For the second half of the forecast period, the input variables favor an increase, so the forecast stays a little bit above the actual value of the stock.

5. SUMMARY AND CONCLUSIONS

As fully analyzed in the Analysis part, the forecast results of the neural network model are very successful and outperforms the benchmarks chosen for comparison. The neural networks have proven to be very good vehicles for forecasting in the Turkish financial framework.

The accuracy of the forecast results vary depending on various variables (other than the inputs): The stock being predicted, the forecasted period, the number of training steps of the neural network. From the forecast results, a generalization can be made: For the periods where no "extraordinary" increase or decrease takes place, the neural network performs superbly, with an error that has a little mean and standard deviation; a very insignificant asymmetry and peakedness. However, when an unexpected price movement occurs (here, by unexpected, I mean that a price increase or decrease that related to some "unknown" variables that were not included as inputs to the neural network model, expressed as 'external effects in this work'), the accuracy of the forecasts made decreases significantly.

Such unexpected cases must be divided to three:

(1) A sudden increase, (2) A sudden decrease, (3) A sudden increase followed by a sudden decrease (a peak).

For the first case of above, the Sarkuysan tests could not provide enough information to conclude that neural network was unsuccessful or not (but it sure that it was not *extremely* successful). The neural network forecasts were not very accurate, nevertheless they were better (very slightly) than the regression model's. This makes it impossible to conclude that for a sudden increase in the trend the neural network "can not" catch up with the increase and forecast accurately, as maybe for the case of Sarkuysan, for the period that the forecast were performed, no method could perform any better.

For the last two cases this thesis provides robust evidence for discussing the success of the neural network predictions. For the second one, i.e. the sudden decline case in Deva, the neural network proves very successful as its prediction is very good compared to the regression model output. For the third situation, as in the Kepez case, the neural network forecast was not as successful as the regression model.

As a last point, for certain situations where everything looks stable (no abnormal change in the inputs or the trend itself), as in the case of Kepez and Sarkuysan, the model has difficulty in forecasting the price changes where the price increases (or decreases) 10% steadily for more than 3 consecutive days. When such a situation takes place, the model's forecasts tend to be "smoother" than the actual trend, for example, a 10% increase (or decrease) for the first day, a 9% for the second, an 8.5% for the third day, and likewise for the following days.

Although the forecasts based on the neural network model are very successful, some certain point could be altered and/or modified for the enhancement of the forecasts, especially in forecasting extraordinary prices moves. Some of the issues that can be probed further are as follows:

The inputs that I have included in my model seem to be enough when the number is considered: Twelve. Yet, no method, including the neural network modeling is perfect for a stock exchange like IMKB. There are some days when even the weather may contribute to the price determination. In this context, the number of inputs could be extended to infinity.

There are some variables that I did not include in my model, like the volume and the volatility of the index and the stock that is being forecasted. The reason for this was that in IMKB, there exist many transactions, which artificially increase the volume but has no contribution to the price determination, like the buy and sell orders coming from the same source (dealer) for speculative purposes. As a further study, these orders can be secluded and the volume and volatility of the stocks can be included as inputs.

Another main variable for inclusion as input can be seasonality. This issue can be investigated in several ways. The well known daily effects could be included as an input, that is, a new variable could be defined, taking values from 1 to 5 indicating the days of the week. Also, the times when stock dividend is distributed could be inputted with a dummy variable into the model. The variable could take value 1 beginning (for example) 2 weeks before the distribution and zero for other times. In this way the dividend distribution times would be introduced to the model.

The current model incorporates the price of the last three days. This could be increased to a week for example. Also time lag for the other variables, like the index could be introduced.

Although the index is included as input, distinguishing some stocks -which are though to be closely related to the one that is forecasted- could be separately included as inputs; because that there exists some stocks that move closely related to each other, like the stocks of the companies that belong to the same group.

The usage of interpolated weekly data instead of daily for some of the input variables were because of the contingency of the current situation; that is because no daily data were available. However, if from a source that I could not reach, daily data were supplied, and the network was trained with it, I believe that the accuracy of the forecast will increase even further.

As in the case of Kepez, increasing the number of training steps for the training sessions can improve the forecasts, and eliminate the "smoothing out" effect that I explained in the Summary and Conclusions. It did for the Kepez case (Figure 17). However, a training of 15,000 steps is a satisfactory one for any measure, and training for more than this value takes enormous times (like a day). Moreover, this has no guarantee for improving the results.

APPENDICES

Appendix 1: The Mathematical Basis of the Artificial Neural Networks

An artificial neural network is a computer-based, highly interconnected computational model with many simple individual processing elements or nodes arranged in layers[Grudnitski and Osburn 1993]. Nodes in one layer are fully connected to all nodes in an adjacent layer.

In general, a node contains a summation function, which totals, through the application of weights, incoming signals from other connected nodes. A node also has a transfer function, which determines the strength with which the summed signal will be transmitted to other nodes in the next connected layer.

A simple three layer neural network was illustrated in Figure 1. Each node is represented by a circle and each interconnection, with its associated weight, by an arrow. The nodes labeled b are bias nodes, which produce constant values, and enable the construction of intercept weights.

Nodes in the input layer receive input, and nodes in the output layer provide output. Nodes in the middle layer receive signals from the input layer nodes and pass signals to output layer nodes. Because the output of these nodes is not directly observable, their layer can be thought of as hidden.

Let the subscripts *i*, *j*, and *l* refer to the input, hidden, and output layers, respectively. Further, let *i* represent input, *o* represent output, and *w* stand for a connection weight. Signals in the network feed forward (i.e., flow from left to right). The signal presented to a hidden layer node m the network is the output value of the input node times the value of the hidden layer node's connection weight. The net input to the hidden node is equal to the sum of all connections into it. This value is described in equation. (A-l).

$$i_j = \sum_i w_{ij} o_i \qquad \text{A-1}$$

The output of a hidden node is transformed according to a function which is usually a sigmoid so that the output of each neuron is normalized between 0 and 1. This function can also be a hyperbolic tangent, or just a linear mapping. Equation A-2 shows this transformation:

$$o_j = \frac{1}{1 + \exp(-i_j)} \qquad \text{A-2}$$

Once the inputs of all hidden layer nodes are calculated, the net input to the output layer nodes is calculated in an analogous manner, as described in equation (A-3). Similarly , the output of each output layer node is transformed according to equation(A-4)

$$i_l = \sum_j w_{jl} o_j \qquad \text{A-3}$$

$$o_l = \frac{1}{1+exp(-i_l)} \qquad \text{A-4}$$

To summarize, the feedforward process necessitates performing two operations per hidden and output layer nodes. The first operation sums the output of the previous layer nodes times the interconnecting weights, and the second operation functionally transforms the output.

Instances of the input called patterns, *p*, usually are divided into training and testing sets. Generally, training a neural network involves presenting the inputs and correct outputs or target values, t_l, of each training pattern over and over again to the network. The network adapts its weights so that its outputs mirror the target values of the training patterns. The neural-network term for this adaptation process is learning and is discussed next.

An error term measures how well a network is trained. The term represents the difference between an output target value and the value a node actually has as a result of feedforward calculations. The error term is calculated for a given pattern and summed over all output nodes for that pattern. Then the grand total is divided by the number of patterns to give the following average sum-squared value:

$$E_p = \sum \}t_{pl} - o_{pl}\{^2 \qquad \text{(A-5)}$$

The goal of training is to minimize the average sum-squared error over all training patterns. The process of-adapting the node weights to achieve this objective is called backpropagation.

Recall that by equation (A-4) the output of a node in the output layer is a function of its input, or $o_l = f(i_l)$. The first derivative of the function $f'(i_l)$ is called the error signal δ_l and defined by the equation (A-6) for the output layer nodes.

$$\delta_l = f'(i_j)(t_l - o_l) \qquad \text{(A-6)}$$

The first derivative of the sigmoid transfer function of equation. (A-4) is $o_l(1- o_l)$. Equation. (A-7) represents the output layer error signal, calculated for each node.

$$\delta_l = (t_l - o_l)o_l(1 - o_l) \qquad \text{(A-7)}$$

This error value must be propagated back through the output layer nodes and the network must perform appropriate adjustments to weights of the nodes. Specifically, using a learning coefficient η and a momentum factor α equation. (A-8) describes how the weights feeding the output layer nodes are updated.

$$w_{lj}(t-1) = w_{lj}(t) + \eta\delta_l o_j + \alpha|\Delta w_{lj}(t-1)| \qquad \text{(A-8)}$$

where w(old) represents the weight change on the previous training iteration.

What is being applied in equation. (A-8) is weight adaptation according to the gradient, descent method. The idea of gradient descent is to make a change in the weight that Is proportional to the negative of the derivative of the error, as measured on the current pattern, with respect to each weight.

For the hidden layer nodes, the error term is calculated as

$$\delta_j = f'(i_j)\sum w_{lj}\delta_l \tag{A-9}$$

and equations. (A-10) and (A-11) are equivalent to equations (A-7) and (A-8):

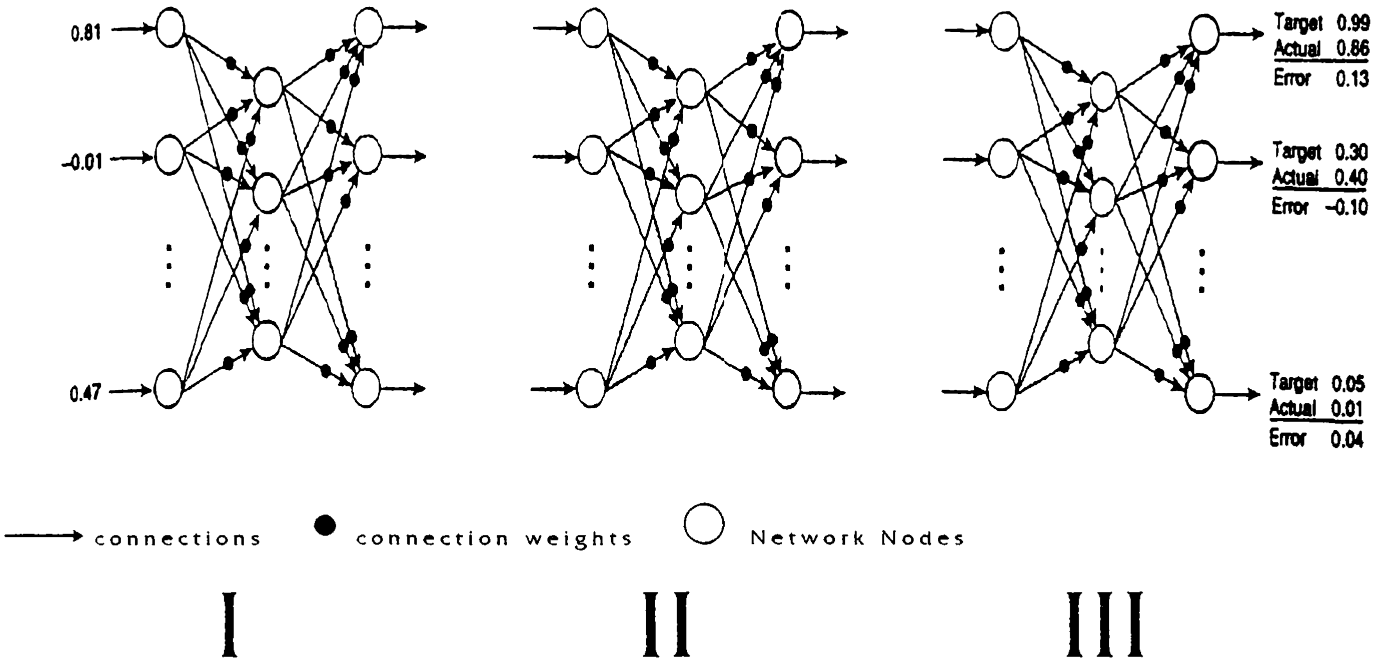
$$\delta_j = o_j(1 - o_j)\sum w_{lj}\delta_l \tag{A-10}$$

$$w_{ji}(t) = w_{ji}(t-1) + \eta\delta_j o_i + \alpha\{\Delta w_{ji}(t-1)\} \tag{A-11}$$

In summary, for each pattern in the training set the network first calculates the error terms for each output layer node using equation. (A-7) and then for each hidden layer node using equation. (A-10). It then sums the error terms, and, after all patterns have been presented once, adjusts the weights of the nodes according to equations. (A-8) and (A-11). Learning is complete and the weights in the neural network are no longer modified once the sum of the error terms reaches zero.

Appendix 2:A Training Step

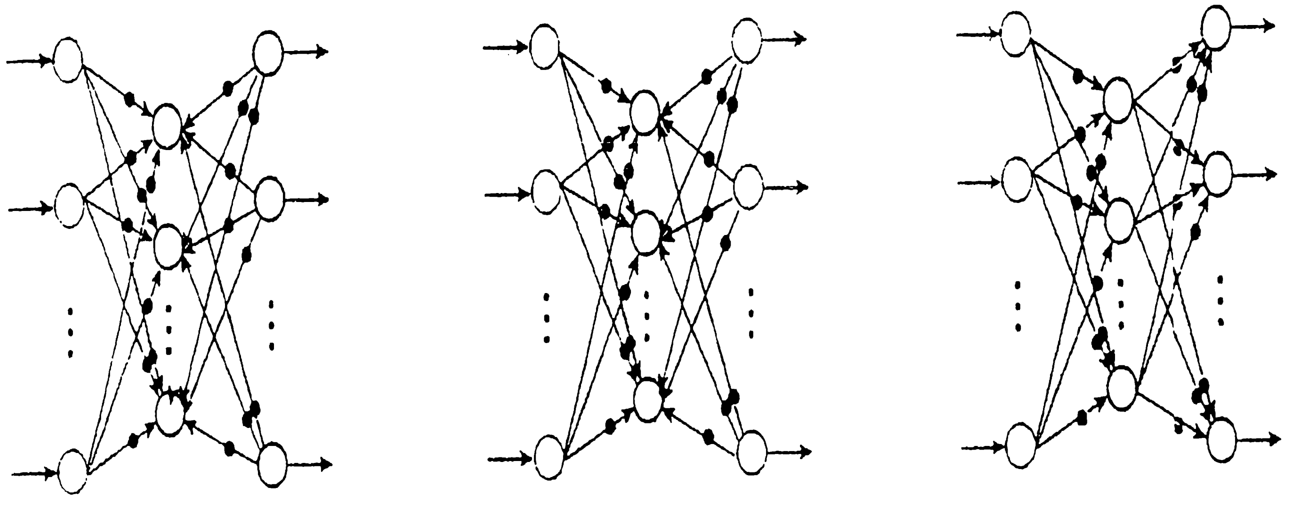
A standard training step can be summarized as follows:



I. During training, the input layer transmits a pattern(the input vector) to the hidden layer nodes. These calculate a weighted sum of inputs (A-1) as passed through a transfer function (A-2).

II. The hidden nodes broadcast their results to all output nodes. Each output node then calculates a weighted sum(A-3) and passes it through the same transfer function to generate an actual result(A-4).

III. Each output node subsequently subtracts its result (the output) from the real (target) result. This yields the output error (A-5).



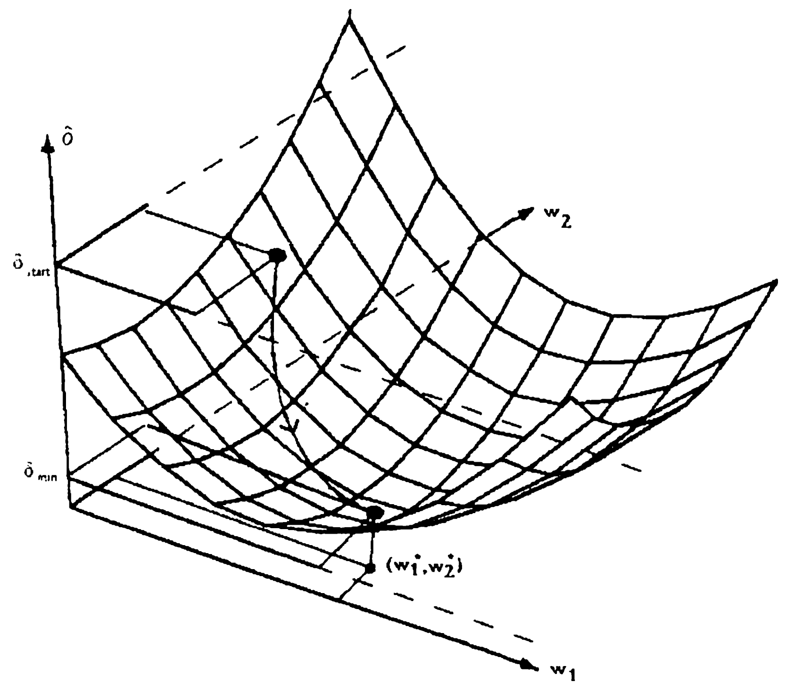
IV V VI

IV. The output nodes calculate the derivatives of the error vector components with respect to the weights, subsequently passing these derivatives back to the hidden layer (A-6).

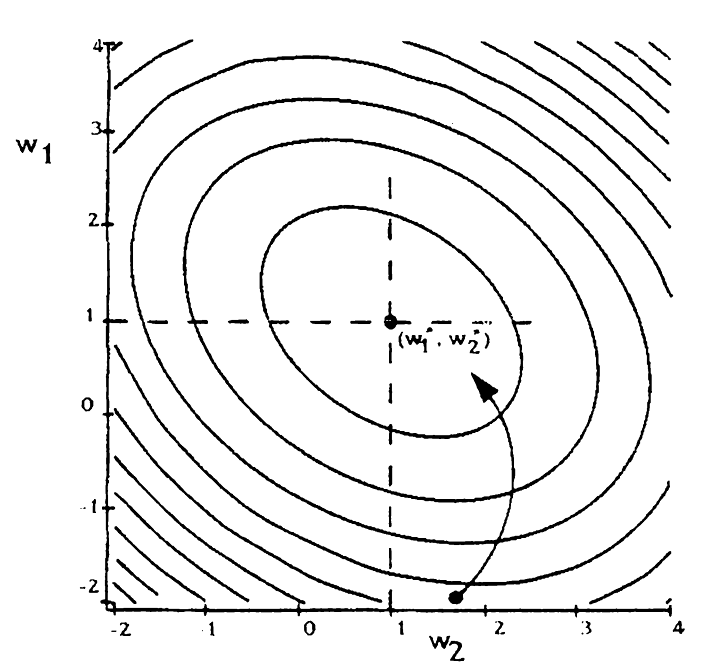
V. Each hidden node calculates the weighted sum of the error derivatives to find its own contribution to the output error(A-9).

VI. Each output layer node and subsequently each output layer node changes its weights according to the mathematical criterion (e.g. least squares) to reduce its error (A-7 and A-8 for output layer, A-10 and A-11 for hidden layer).

Appendix 3: Convergence In The Error Space



Projection on the Weight Space



Appendix 4:

The Full Configuration and Topology of The Neural Network After 15,000 Training Steps

```
itle: IMKB PRICE PREDICTION:ARCELIK
     Display Mode: Network                              Type: Hetero-Associative
    Display Style: default
 Control Strategy: backprop                  L/R Schedule: backprop
     15144 Learn                  0 Recall                  0 Layer
        16 Aux 1                  0 Aux 2                   0 Aux 3
/R Schedule: backprop
 Recall Step                 1           0           0           0           0
   Input Clamp          0.0000      0.0000      0.0000      0.0000      0.0000
   Firing Density     100.0000      0.0000      0.0000      0.0000      0.0000
   Temperature          0.0000      0.0000      0.0000      0.0000      0.0000
   Gain                 1.0000      0.0000      0.0000      0.0000      0.0000
   Gain                 1.0000      0.0000      0.0000      0.0000      0.0000
 Learn Step               5000           0           0           0           0
   Coefficient 1        0.9000      0.0000      0.0000      0.0000      0.0000
   Coefficient 2        0.0000      0.0000      0.0000      0.0000      0.0000
   Coefficient 3        0.0000      0.0000      0.0000      0.0000      0.0000
   Temperature          0.0000      0.0000      0.0000      0.0000      0.0000
0 Parameters
      Learn Data: File Randomize (serifp3) Binary
     Recall Data: File Sequential (serifpt3)
     Result File: Desired Output, Output
  UserIO Program: userio
      I/P Ranges:     -1.0000,      1.0000
      O/P Ranges:     -0.8000,      0.8000
   I/P Start Col:           1                  MinMax Table: serifp3
   O/P Start Col:          13              Number of Entries: 13

inMax Table <serifp3>:
ol:            1           2           3           4           5           6
in:       2.9500      2.4400      1.9600      0.5900      0.5100      0.5900
ax:          9.5        7.08        5.89        0.75         0.7        0.83

ol:            7           8           9          10          11          12
in:       0.5400      0.0000      2.5200      2.6000      2.6000      2.6000
ax:         0.77        0.88        6.74        7.08        7.08        7.08

ol:           13
in:       2.6000
ax:         7.08

ayer: 1
      PEs: 1        Wgt Fields: 2                       Sum: Sum
  Spacing: 5         F' offset: 0.00               Transfer: Linear
    Shape: Square                                    Output: Direct
    Scale: 1.00      Low Limit: -9999.00         Error Func: standard
   Offset: 0.00     High Limit: 9999.00               Learn: --None--
 Init Low: -0.100    Init High: 0.100          L/R Schedule: (Network)
 Winner 1: None                                    Winner 2: None
 PE: Bias
     1.000 Err Factor        0.000 Desired
     0.000 Sum               1.000 Transfer           1.000 Output
         0 Weights          -2.562 Error              0.000 Current Error
ayer: In
      PEs: 12       Wgt Fields: 1                       Sum: Sum
  Spacing: 5         F' offset: 0.00               Transfer: Linear
    Shape: Square                                    Output: Direct
    Scale: 1.00      Low Limit: -9999.00         Error Func: standard
   Offset: 0.00     High Limit: 9999.00               Learn: --None--
 Init Low: -0.100    Init High: 0.100          L/R Schedule: (Network)
 Winner 1: None                                    Winner 2: None
 PE: $$
     1.000 Err Factor       -0.982 Desired
    -0.982 Sum              -0.982 Transfer          -0.982 Output
         0 Weights          -0.001 Error              0.000 Current Error
 PE: Gold
     1.000 Err Factor       -0.983 Desired
    -0.983 Sum              -0.983 Transfer          -0.983 Output
         0 Weights           0.000 Error              0.000 Current Error
 PE: Dm
     1.000 Err Factor       -0.975 Desired
    -0.975 Sum              -0.975 Transfer          -0.975 Output
         0 Weights           0.000 Error              0.000 Current Error
 PE: int1m
     1.000 Err Factor       -0.875 Desired
    -0.875 Sum              -0.875 Transfer          -0.875 Output
         0 Weights           0.000 Error              0.000 Current Error
 PE: int3
     1.000 Err Factor       -0.895 Desired
    -0.895 Sum              -0.895 Transfer          -0.895 Output
         0 Weights           0.002 Error              0.000 Current Error
 PE: tb3
     1.000 Err Factor       -0.917 Desired
```

```
     -0.913 Sum                 -0.913 Transfer             -0.913 Output
          0 Weights              0.001 Error                 0.000 Current Error
 PE: c-bd
      1.000 Err Factor           0.523 Desired
      0.523 Sum                  0.523 Transfer              0.523 Output
          0 Weights             -0.001 Error                 0.000 Current Error
 PE: indx
      1.000 Err Factor          -0.436 Desired
     -0.436 Sum                 -0.436 Transfer             -0.436 Output
          0 Weights              0.003 Error                 0.000 Current Error
 PE: prevd
      1.000 Err Factor          -0.692 Desired
     -0.692 Sum                 -0.692 Transfer             -0.692 Output
          0 Weights              0.008 Error                 0.000 Current Error
 PE: 2dys
      1.000 Err Factor          -0.665 Desired
     -0.665 Sum                 -0.665 Transfer             -0.665 Output
          0 Weights              0.001 Error                 0.000 Current Error
 PE: 3dys
      1.000 Err Factor          -0.554 Desired
     -0.554 Sum                 -0.554 Transfer             -0.554 Output
          0 Weights              0.001 Error                 0.000 Current Error
ayer: Hidden1
      PEs: 20       Wgt Fields: 3                        Sum: Sum
  Spacing: 5         F' offset: 0.00                Transfer: TanH
    Shape: Square                                     Output: Direct
    Scale: 1.00      Low Limit: -9999.00          Error Func: standard
   Offset: 0.00     High Limit: 9999.00                Learn: Norm-Cum-Delta
 Init Low: -0.100    Init High: 0.100           L/R Schedule: hidden1
 Winner 1: None                                     Winner 2: None
/R Schedule: hidden1
 Recall Step                   1           0           0           0           0
   Input Clamp            0.0000      0.0000      0.0000      0.0000      0.0000
   Firing Density       100.0000      0.0000      0.0000      0.0000      0.0000
   Temperature            0.0000      0.0000      0.0000      0.0000      0.0000
   Gain                   1.0000      0.0000      0.0000      0.0000      0.0000
   Gain                   1.0000      0.0000      0.0000      0.0000      0.0000
 Learn Step                10000       30000       70000      150000      310000
   Coefficient 1          0.3000      0.1500      0.0375      0.0023      0.0000
   Coefficient 2          0.4000      0.2000      0.0500      0.0031      0.0000
   Coefficient 3          0.1000      0.1000      0.1000      0.1000      0.1000
   Temperature            0.0000      0.0000      0.0000      0.0000      0.0000
 PE: 14
      1.000 Err Factor           0.000 Desired
      0.570 Sum                  0.515 Transfer              0.515 Output
         13 Weights             -0.003 Error                -0.005 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias   +1.0000   +0.1190 V-r   -0.0014   +0.0011
      $$   -0.9820   +0.0268 V-r   -0.0001   -0.0012
    Gold   -0.9830   -0.0445 V-r   -0.0000   -0.0012
      Dm   -0.9750   -0.0909 V-r   -0.0002   -0.0014
   int1m   -0.8750   -0.0082 V-r   -0.0005   -0.0015
    int3   -0.8950   -0.1135 V-r   -0.0005   -0.0010
     tb3   -0.9170   +0.1256 V-r   -0.0001   -0.0007
    gvbd   -0.9130   +0.0777 V-r   -0.0005   -0.0002
    c-bd   +0.5230   +0.0707 V-r   -0.0012   +0.0009
    indx   -0.4360   -0.0830 V-r   +0.0004   +0.0005
   prevd   -0.6920   -0.4699 V-r   +0.0002   +0.0003
    2dys   -0.6650   +0.0038 V-r   +0.0003   +0.0002
    3dys   -0.5540   -0.0476 V-r   +0.0003   -0.0001
 PE: 15
      1.000 Err Factor           0.000 Desired
     -0.165 Sum                 -0.164 Transfer             -0.164 Output
         13 Weights              0.000 Error                 0.000 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias   +1.0000   -0.0220 V-r   +0.0000   -0.0000
      $$   -0.9820   -0.0005 V-r   -0.0000   +0.0000
    Gold   -0.9830   +0.0171 V-r   -0.0000   +0.0000
      Dm   -0.9750   +0.0792 V-r   +0.0000   +0.0000
   int1m   -0.8750   +0.0016 V-r   +0.0000   +0.0000
    int3   -0.8950   -0.0564 V-r   +0.0000   +0.0000
     tb3   -0.9170   +0.0245 V-r   +0.0000   +0.0000
    gvbd   -0.9130   +0.0247 V-r   +0.0000   +0.0000
    c-bd   +0.5230   +0.0424 V-r   +0.0000   -0.0000
    indx   -0.4360   +0.1063 V-r   -0.0000   -0.0000
   prevd   -0.6920   -0.0300 V-r   -0.0000   -0.0000
    2dys   -0.6650   +0.0729 V-r   -0.0000   -0.0000
    3dys   -0.5540   +0.0033 V-r   -0.0000   +0.0000
```

```
   1.000 Err Factor        0.000 Desired
   0.283 Sum               0.276 Transfer          0.276 Output
      13 Weights          -0.001 Error            -0.001 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias   +1.0000   +0.0130 V-r   -0.0004   +0.0003
      $$   -0.9820   +0.1231 V-r   +0.0000   -0.0003
    Gold   -0.9830   -0.0203 V-r   +0.0000   -0.0003
      Dm   -0.9750   -0.0652 V-r   -0.0000   -0.0004
   int1m   -0.8750   -0.0681 V-r   -0.0001   -0.0005
    int3   -0.8950   -0.0617 V-r   -0.0001   -0.0004
     tb3   -0.9170   -0.0369 V-r   -0.0000   -0.0002
    gvbd   -0.9130   +0.0077 V-r   -0.0001   -0.0001
    c-bd   +0.5230   -0.0040 V-r   -0.0004   +0.0002
    indx   -0.4360   -0.0340 V-r   +0.0001   +0.0002
   prevd   -0.6920   -0.0981 V-r   +0.0001   +0.0001
    2dys   -0.6650   -0.0638 V-r   +0.0001   +0.0001
    3dys   -0.5540   -0.0768 V-r   +0.0001   -0.0000
 PE: 17
   1.000 Err Factor        0.000 Desired
   0.159 Sum               0.157 Transfer          0.157 Output
      13 Weights          -0.002 Error            -0.002 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias   +1.0000   -0.0525 V-r   -0.0007   +0.0005
      $$   -0.9820   +0.0072 V-r   +0.0000   -0.0006
    Gold   -0.9830   -0.0037 V-r   +0.0000   -0.0006
      Dm   -0.9750   +0.0061 V-r   -0.0000   -0.0007
   int1m   -0.8750   +0.0502 V-r   -0.0002   -0.0008
    int3   -0.8950   -0.0306 V-r   -0.0002   -0.0006
     tb3   -0.9170   +0.0093 V-r   -0.0000   -0.0004
    gvbd   -0.9130   -0.0207 V-r   -0.0002   -0.0001
    c-bd   +0.5230   +0.0345 V-r   -0.0006   +0.0004
    indx   -0.4360   -0.1353 V-r   +0.0002   +0.0003
   prevd   -0.6920   -0.2597 V-r   +0.0001   +0.0002
    2dys   -0.6650   +0.0194 V-r   +0.0002   +0.0001
    3dys   -0.5540   +0.0311 V-r   +0.0002   -0.0000
 PE: 18
   1.000 Err Factor        0.000 Desired
  -0.250 Sum              -0.244 Transfer         -0.244 Output
      13 Weights           0.002 Error             0.002 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias   +1.0000   +0.0295 V-r   +0.0008   -0.0006
      $$   -0.9820   -0.0874 V-r   -0.0000   +0.0006
    Gold   -0.9830   -0.0279 V-r   -0.0000   +0.0007
      Dm   -0.9750   -0.0316 V-r   +0.0000   +0.0007
   int1m   -0.8750   -0.0172 V-r   +0.0002   +0.0009
    int3   -0.8950   +0.0993 V-r   +0.0002   +0.0007
     tb3   -0.9170   +0.0307 V-r   +0.0000   +0.0005
    gvbd   -0.9130   +0.0646 V-r   +0.0002   +0.0002
    c-bd   +0.5230   -0.0564 V-r   +0.0007   -0.0004
    indx   -0.4360   +0.1602 V-r   -0.0003   -0.0004
   prevd   -0.6920   +0.2329 V-r   -0.0002   -0.0002
    2dys   -0.6650   +0.0104 V-r   -0.0002   -0.0002
    3dys   -0.5540   -0.0095 V-r   -0.0002   +0.0000
 PE: 19
   1.000 Err Factor        0.000 Desired
  -0.071 Sum              -0.071 Transfer         -0.071 Output
      13 Weights           0.002 Error             0.002 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias   +1.0000   +0.0369 V-r   +0.0005   -0.0004
      $$   -0.9820   -0.0500 V-r   -0.0000   +0.0004
    Gold   -0.9830   -0.0180 V-r   -0.0000   +0.0004
      Dm   -0.9750   +0.0341 V-r   +0.0000   +0.0005
   int1m   -0.8750   -0.0709 V-r   +0.0001   +0.0006
    int3   -0.8950   -0.0243 V-r   +0.0001   +0.0004
     tb3   -0.9170   -0.0834 V-r   +0.0000   +0.0003
    gvbd   -0.9130   +0.0088 V-r   +0.0001   +0.0001
    c-bd   +0.5230   -0.0719 V-r   +0.0004   -0.0003
    indx   -0.4360   +0.0791 V-r   -0.0002   -0.0002
   prevd   -0.6920   +0.1809 V-r   -0.0001   -0.0001
    2dys   -0.6650   +0.1139 V-r   -0.0001   -0.0001
    3dys   -0.5540   +0.0372 V-r   -0.0001   +0.0000
 PE: 20
   1.000 Err Factor        0.000 Desired
   0.056 Sum               0.056 Transfer          0.056 Output
      13 Weights           0.000 Error             0.000 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias   +1.0000   +0.0286 V-r   +0.0000   -0.0000
      $$   -0.9820   -0.0129 V-r   -0.0000   +0.0000
    Gold   -0.9830   -0.0852 V-r   -0.0000   +0.0000
      Dm   -0.9750   +0.0724 V-r   +0.0000   +0.0000
   int1m   -0.8750   -0.0786 V-r   +0.0000   +0.0000
    int3   -0.8950   +0.0270 V-r   +0.0000   +0.0000
```

```
      tb3    -0.9170    +0.0870 V-r    +0.0000    +0.0000
     gvbd    -0.9130    -0.0457 V-r    +0.0000    +0.0000
     c-bd    +0.5230    +0.0755 V-r    +0.0000    -0.0000
     indx    -0.4360    +0.0998 V-r    -0.0000    -0.0000
    prevd    -0.6920    -0.0295 V-r    -0.0000    -0.0000
     2dys    -0.6650    +0.0696 V-r    -0.0000    -0.0000
     3dys    -0.5540    -0.0457 V-r    -0.0000    +0.0000
 PE: 21
     1.000 Err Factor        0.000 Desired
    -0.119 Sum              -0.118 Transfer           -0.118 Output
        13 Weights           0.002 Error               0.002 Current Error
Input PE  Input Value Weight Type Delta Weight
     Bias    +1.0000    -0.0271 V-r    +0.0006    -0.0004
       $$    -0.9820    +0.0687 V-r    -0.0000    +0.0004
     Gold    -0.9830    -0.0446 V-r    -0.0000    +0.0004
       Dm    -0.9750    -0.0353 V-r    +0.0000    +0.0005
    int1m    -0.8750    -0.0129 V-r    +0.0001    +0.0006
     int3    -0.8950    -0.0111 V-r    +0.0002    +0.0004
      tb3    -0.9170    +0.0050 V-r    +0.0000    +0.0003
     gvbd    -0.9130    -0.0476 V-r    +0.0002    +0.0001
     c-bd    +0.5230    +0.0809 V-r    +0.0005    -0.0003
     indx    -0.4360    +0.0495 V-r    -0.0002    -0.0002
    prevd    -0.6920    +0.2334 V-r    -0.0001    -0.0001
     2dys    -0.6650    -0.0281 V-r    -0.0001    -0.0001
     3dys    -0.5540    +0.0728 V-r    -0.0001    +0.0000
 PE: 22
     1.000 Err Factor        0.000 Desired
     0.274 Sum               0.268 Transfer            0.268 Output
        13 Weights          -0.003 Error              -0.003 Current Error
Input PE  Input Value Weight Type Delta Weight
     Bias    +1.0000    -0.0532 V-r    -0.0010    +0.0007
       $$    -0.9820    -0.0395 V-r    +0.0000    -0.0008
     Gold    -0.9830    +0.0652 V-r    +0.0001    -0.0008
       Dm    -0.9750    +0.1105 V-r    -0.0000    -0.0009
    int1m    -0.8750    -0.0494 V-r    -0.0002    -0.0011
     int3    -0.8950    -0.1450 V-r    -0.0003    -0.0008
      tb3    -0.9170    +0.0321 V-r    -0.0000    -0.0005
     gvbd    -0.9130    -0.0840 V-r    -0.0003    -0.0002
     c-bd    +0.5230    -0.0193 V-r    -0.0008    +0.0005
     indx    -0.4360    +0.0306 V-r    +0.0003    +0.0004
    prevd    -0.6920    -0.2917 V-r    +0.0002    +0.0002
     2dys    -0.6650    +0.0025 V-r    +0.0002    +0.0002
     3dys    -0.5540    -0.1146 V-r    +0.0002    -0.0000
 PE: 23
     1.000 Err Factor        0.000 Desired
    -0.207 Sum              -0.204 Transfer           -0.204 Output
        13 Weights          -0.000 Error              -0.000 Current Error
Input PE  Input Value Weight Type Delta Weight
     Bias    +1.0000    -0.0482 V-r    -0.0001    +0.0001
       $$    -0.9820    +0.0656 V-r    +0.0000    -0.0001
     Gold    -0.9830    +0.0678 V-r    +0.0000    -0.0001
       Dm    -0.9750    -0.0468 V-r    -0.0000    -0.0001
    int1m    -0.8750    +0.0121 V-r    -0.0000    -0.0001
     int3    -0.8950    -0.0515 V-r    -0.0000    -0.0001
      tb3    -0.9170    +0.0397 V-r    -0.0000    -0.0000
     gvbd    -0.9130    +0.0200 V-r    -0.0000    -0.0000
     c-bd    +0.5230    -0.0381 V-r    -0.0001    +0.0000
     indx    -0.4360    -0.0836 V-r    +0.0000    +0.0000
    prevd    -0.6920    +0.0040 V-r    +0.0000    +0.0000
     2dys    -0.6650    +0.0741 V-r    +0.0000    +0.0000
     3dys    -0.5540    +0.0340 V-r    +0.0000    -0.0000
 PE: 24
     1.000 Err Factor        0.000 Desired
     0.136 Sum               0.135 Transfer            0.135 Output
        13 Weights          -0.004 Error              -0.004 Current Error
Input PE  Input Value Weight Type Delta Weight
     Bias    +1.0000    -0.0223 V-r    -0.0013    +0.0009
       $$    -0.9820    +0.1682 V-r    +0.0000    -0.0010
     Gold    -0.9830    +0.0411 V-r    +0.0001    -0.0010
       Dm    -0.9750    +0.0117 V-r    -0.0000    -0.0012
    int1m    -0.8750    -0.0305 V-r    -0.0003    -0.0014
     int3    -0.8950    +0.0128 V-r    -0.0004    -0.0010
      tb3    -0.9170    +0.1203 V-r    -0.0000    -0.0007
     gvbd    -0.9130    -0.1058 V-r    -0.0003    -0.0003
     c-bd    +0.5230    -0.0023 V-r    -0.0011    +0.0007
     indx    -0.4360    -0.1824 V-r    +0.0004    +0.0005
    prevd    -0.6920    -0.3715 V-r    +0.0003    +0.0003
     2dys    -0.6650    -0.0056 V-r    +0.0003    +0.0003
     3dys    -0.5540    -0.0625 V-r    +0.0003    -0.0000
 PE: 25
     1.000 Err Factor        0.000 Desired
    -0.177 Sum              -0.175 Transfer           -0.175 Output
        13 Weights           0.001 Error               0.001 Current Error
```

```
Input PE  Input Value Weight Type Delta Weight
    Bias    +1.0000    -0.0117 V-r    +0.0004   -0.0003
      $$    -0.9820    -0.0224 V-r    -0.0000   +0.0003
    Gold    -0.9830    +0.0181 V-r    -0.0000   +0.0003
      Dm    -0.9750    -0.0799 V-r    +0.0000   +0.0004
   int1m    -0.8750    -0.0489 V-r    +0.0001   +0.0005
    int3    -0.8950    +0.0832 V-r    +0.0001   +0.0003
     tb3    -0.9170    +0.0631 V-r    +0.0000   +0.0002
    gvbd    -0.9130    -0.0530 V-r    +0.0001   +0.0001
    c-bd    +0.5230    -0.0065 V-r    +0.0004   -0.0002
    indx    -0.4360    +0.0789 V-r    -0.0001   -0.0002
   prevd    -0.6920    +0.1199 V-r    -0.0001   -0.0001
    2dys    -0.6650    +0.1085 V-r    -0.0001   -0.0001
    3dys    -0.5540    +0.0237 V-r    -0.0001   +0.0000
 PE: 26
     1.000 Err Factor       0.000 Desired
     0.269 Sum              0.263 Transfer             0.263 Output
        13 Weights         -0.002 Error               -0.002 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias    +1.0000    -0.0407 V-r    -0.0006   +0.0004
      $$    -0.9820    +0.0095 V-r    +0.0000   -0.0005
    Gold    -0.9830    +0.0139 V-r    +0.0000   -0.0005
      Dm    -0.9750    -0.0640 V-r    -0.0000   -0.0005
   int1m    -0.8750    -0.0279 V-r    -0.0002   -0.0007
    int3    -0.8950    -0.0473 V-r    -0.0002   -0.0005
     tb3    -0.9170    +0.0638 V-r    -0.0000   -0.0003
    gvbd    -0.9130    -0.0843 V-r    -0.0002   -0.0001
    c-bd    +0.5230    +0.0748 V-r    -0.0005   +0.0003
    indx    -0.4360    -0.0273 V-r    +0.0002   +0.0003
   prevd    -0.6920    -0.2056 V-r    +0.0001   +0.0001
    2dys    -0.6650    +0.0287 V-r    +0.0001   +0.0001
    3dys    -0.5540    -0.0192 V-r    +0.0001   -0.0000
 PE: 27
     1.000 Err Factor       0.000 Desired

    -0.430 Sum             -0.405 Transfer            -0.405 Output
        13 Weights          0.002 Error                0.003 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias    +1.0000    -0.0398 V-r    +0.0009   -0.0007
      $$    -0.9820    +0.0048 V-r    +0.0000   +0.0008
    Gold    -0.9830    +0.0421 V-r    -0.0000   +0.0008
      Dm    -0.9750    +0.0142 V-r    +0.0001   +0.0009
   int1m    -0.8750    +0.0380 V-r    +0.0003   +0.0010
    int3    -0.8950    +0.0657 V-r    +0.0003   +0.0007
     tb3    -0.9170    -0.0333 V-r    +0.0001   +0.0005
    gvbd    -0.9130    -0.1076 V-r    +0.0003   +0.0002
    c-bd    +0.5230    +0.0173 V-r    +0.0008   -0.0006
    indx    -0.4360    +0.1970 V-r    -0.0003   -0.0003
   prevd    -0.6920    +0.2190 V-r    -0.0002   -0.0002
    2dys    -0.6650    +0.1634 V-r    -0.0002   -0.0001
    3dys    -0.5540    +0.0533 V-r    -0.0002   +0.0001
 PE: 28
     1.000 Err Factor       0.000 Desired
     0.032 Sum              0.032 Transfer             0.032 Output
        13 Weights         -0.001 Error               -0.001 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias    +1.0000    -0.0486 V-r    -0.0003   +0.0002
      $$    -0.9820    +0.0659 V-r    +0.0000   -0.0002
    Gold    -0.9830    -0.0226 V-r    +0.0000   -0.0002
      Dm    -0.9750    +0.0256 V-r    -0.0000   -0.0002
   int1m    -0.8750    -0.0485 V-r    -0.0001   -0.0003
    int3    -0.8950    -0.0518 V-r    -0.0001   -0.0002
     tb3    -0.9170    +0.0627 V-r    -0.0000   -0.0001
    gvbd    -0.9130    -0.0021 V-r    -0.0001   -0.0001
    c-bd    +0.5230    +0.0350 V-r    -0.0002   +0.0001
    indx    -0.4360    +0.0110 V-r    +0.0001   +0.0001
   prevd    -0.6920    -0.0635 V-r    +0.0001   +0.0001
    2dys    -0.6650    -0.0271 V-r    +0.0001   +0.0001
    3dys    -0.5540    -0.0706 V-r    +0.0001   -0.0000
 PE: 29
     1.000 Err Factor       0.000 Desired
     0.506 Sum              0.467 Transfer             0.467 Output
        13 Weights         -0.002 Error               -0.003 Current Error
Input PE  Input Value Weight Type Delta Weight
    Bias    +1.0000    +0.0041 V-r    -0.0009   +0.0006
      $$    -0.9820    -0.0216 V-r    -0.0000   -0.0007
    Gold    -0.9830    -0.0570 V-r    -0.0000   -0.0007
      Dm    -0.9750    +0.0066 V-r    -0.0001   -0.0008
   int1m    -0.8750    -0.0095 V-r    -0.0003   -0.0011
    int3    -0.8950    -0.0439 V-r    -0.0003   -0.0007
     tb3    -0.9170    -0.0135 V-r    -0.0001   -0.0005
    gvbd    -0.9130    -0.0797 V-r    -0.0003   -0.0002
    c-bd    +0.5230    +0.0176 V-r    -0.0007   +0.0005
```

```
   indx    -0.4360    +0.0060 V-r    +0.0003    +0.0004
  prevd    -0.6920    -0.3020 V-r    +0.0002    +0.0002
   2dys    -0.6650    -0.1452 V-r    +0.0002    +0.0002
   3dys    -0.5540    +0.0251 V-r    +0.0002    -0.0000
PE: 30
    1.000 Err Factor       0.000 Desired
    0.259 Sum              0.253 Transfer              0.253 Output
       13 Weights         -0.001 Error                -0.001 Current Error
Input PE  Input Value Weight Type Delta Weight
   Bias    +1.0000    +0.0601 V-r    -0.0002    +0.0001
     $$    -0.9820    -0.0708 V-r    +0.0000    -0.0002
   Gold    -0.9830    -0.0634 V-r    +0.0000    -0.0002
     Dm    -0.9750    +0.0699 V-r    -0.0000    -0.0002
  int1m    -0.8750    +0.0012 V-r    -0.0001    -0.0002
   int3    -0.8950    +0.0278 V-r    -0.0001    -0.0002
    tb3    -0.9170    +0.0317 V-r    -0.0000    -0.0001
   gvbd    -0.9130    +0.0019 V-r    -0.0001    -0.0000
   c-bd    +0.5230    +0.0299 V-r    -0.0002    +0.0001
   indx    -0.4360    -0.1097 V-r    +0.0001    +0.0001
  prevd    -0.6920    -0.0543 V-r    +0.0000    +0.0000
   2dys    -0.6650    -0.0792 V-r    +0.0000    +0.0000
   3dys    -0.5540    -0.0687 V-r    +0.0000    -0.0000
PE: 31
    1.000 Err Factor       0.000 Desired
    0.433 Sum              0.408 Transfer              0.408 Output
       13 Weights         -0.002 Error                -0.003 Current Error
Input PE  Input Value Weight Type Delta Weight
   Bias    +1.0000    -0.0570 V-r    -0.0009    +0.0006
     $$    -0.9820    +0.0024 V-r    -0.0000    -0.0007
   Gold    -0.9830    -0.0694 V-r    +0.0000    -0.0007
     Dm    -0.9750    -0.0651 V-r    -0.0001    -0.0008
  int1m    -0.8750    +0.0611 V-r    -0.0002    -0.0010
   int3    -0.8950    -0.0041 V-r    -0.0003    -0.0007
    tb3    -0.9170    -0.0185 V-r    -0.0000    -0.0005
   gvbd    -0.9130    -0.1043 V-r    -0.0003    -0.0002
   c-bd    +0.5230    +0.0166 V-r    -0.0007    +0.0005
   indx    -0.4360    -0.0252 V-r    +0.0003    +0.0004
  prevd    -0.6920    -0.2936 V-r    +0.0002    +0.0002
   2dys    -0.6650    -0.1619 V-r    +0.0002    +0.0002
   3dys    -0.5540    +0.0573 V-r    +0.0002    -0.0000
PE: 32
    1.000 Err Factor       0.000 Desired
    0.122 Sum              0.121 Transfer              0.121 Output
       13 Weights         -0.001 Error                -0.001 Current Error
Input PE  Input Value Weight Type Delta Weight
   Bias    +1.0000    -0.0898 V-r    -0.0002    +0.0002
     $$    -0.9820    -0.0617 V-r    +0.0000    -0.0002
   Gold    -0.9830    -0.0760 V-r    +0.0000    -0.0002
     Dm    -0.9750    -0.0365 V-r    -0.0000    -0.0002
  int1m    -0.8750    -0.0354 V-r    -0.0001    -0.0003
   int3    -0.8950    +0.0486 V-r    -0.0001    -0.0002
    tb3    -0.9170    +0.0864 V-r    -0.0000    -0.0001
   gvbd    -0.9130    -0.0637 V-r    -0.0001    -0.0001
   c-bd    +0.5230    +0.0076 V-r    -0.0002    +0.0001
   indx    -0.4360    -0.0490 V-r    +0.0001    +0.0001
  prevd    -0.6920    -0.0979 V-r    +0.0000    +0.0001
   2dys    -0.6650    +0.0181 V-r    +0.0001    +0.0000
   3dys    -0.5540    +0.0118 V-r    +0.0001    -0.0000
PE: 33
    1.000 Err Factor       0.000 Desired
    0.095 Sum              0.094 Transfer              0.094 Output
       13 Weights         -0.000 Error                -0.000 Current Error
Input PE  Input Value Weight Type Delta Weight
   Bias    +1.0000    +0.0341 V-r    -0.0001    +0.0001
     $$    -0.9820    -0.0142 V-r    +0.0000    -0.0001
   Gold    -0.9830    +0.0086 V-r    +0.0000    -0.0001
     Dm    -0.9750    +0.0503 V-r    -0.0000    -0.0001
  int1m    -0.8750    -0.0637 V-r    -0.0000    -0.0001
   int3    -0.8950    -0.0661 V-r    -0.0000    -0.0001
    tb3    -0.9170    -0.0803 V-r    -0.0000    -0.0000
   gvbd    -0.9130    +0.0610 V-r    -0.0000    -0.0000
   c-bd    +0.5230    -0.0149 V-r    -0.0001    +0.0000
   indx    -0.4360    -0.0744 V-r    +0.0000    +0.0000
  prevd    -0.6920    -0.0161 V-r    +0.0000    +0.0000
   2dys    -0.6650    +0.0162 V-r    +0.0000    +0.0000
   3dys    -0.5540    +0.0970 V-r    +0.0000    -0.0000
yer: Out
      PEs: 1        Wgt Fields: 3                   Sum: Sum
  Spacing: 5         F' offset: 0.00           Transfer: TanH
    Shape: Square '                              Output: Direct
    Scale: 1.00      Low Limit: -9999.00    Error Func: standard
   Offset: 0.00     High Limit: 9999.00           Learn: Norm-Cum-Delta
 Init Low: -0.100    Init High: 0.100       L/R Schedule: out
```

Winner 1: None   Winner 2: None

/P Schedule: out

Recall Step	1	0	0	0	0
Input Clamp	0.0000	0.0000	0.0000	0.0000	0.0000
Firing Density	100.0000	0.0000	0.0000	0.0000	0.0000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000
Gain	1.0000	0.0000	0.0000	0.0000	0.0000
Gain	1.0000	0.0000	0.0000	0.0000	0.0000
Learn Step	10000	30000	70000	150000	310000
Coefficient 1	0.1500	0.0750	0.0188	0.0012	0.0000
Coefficient 2	0.4000	0.2000	0.0500	0.0031	0.0000
Coefficient 3	0.1000	0.1000	0.1000	0.1000	0.1000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000

PE: Price

1.000 Err Factor   -0.521 Desired

-0.610 Sum   -0.544 Transfer   -0.544 Output

21 Weights   0.016 Error   0.023 Current Error

Input PE	Input Value	Weight	Type	Delta	Weight
Bias	+1.0000	-0.0009	V-r	+0.0028	-0.0018
14	+0.5151	-0.2847	V-r	+0.0008	-0.0003
15	-0.1637	+0.0017	V-r	-0.0001	-0.0000
16	+0.2755	-0.0814	V-r	+0.0001	-0.0002
17	+0.1574	-0.1337	V-r	+0.0002	+0.0005
18	-0.2445	+0.1538	V-r	-0.0002	-0.0004
19	-0.0709	+0.0994	V-r	-0.0004	-0.0007
20	+0.0564	+0.0055	V-r	+0.0001	-0.0004
21	-0.1180	+0.1026	V-r	-0.0001	-0.0004
22	+0.2676	-0.1921	V-r	-0.0002	+0.0001
23	-0.2043	-0.0160	V-r	-0.0003	+0.0004
24	+0.1353	-0.2425	V-r	+0.0002	+0.0010
25	-0.1749	+0.0799	V-r	-0.0003	-0.0003
26	+0.2625	-0.1129	V-r	+0.0001	-0.0001
27	-0.4049	+0.1741	V-r	-0.0005	-0.0001
28	+0.0318	-0.0501	V-r	-0.0000	+0.0001
29	+0.4667	-0.1751	V-r	+0.0002	-0.0001
30	+0.2531	-0.0385	V-r	+0.0005	+0.0000
31	+0.4082	-0.1656	V-r	+0.0001	+0.0002
32	+0.1214	-0.0442	V-r	-0.0002	-0.0000
33	+0.0944	-0.0143	V-r	+0.0000	-0.0003

BIBLIOGRAPHY

Bulkley, G. And Tonks, I., 1992. "Tradding Rules and Excess Volatility",*Journal of Financial Quantitative Analysis*, 27:365-382.

De Groot, C. and Wurtz, D., 1991." Analysis of Univariate Time Series With Connectionist Nets: A Case Study of Two Classical Examples",*Neurocomputing,* 3: 177-192.

Dowen, R.J., and Bauman, W.S., 1988. "Growth Projections and Comman Stock Returns", *Financial Analysts Journal*,4:34-47.

Eberhart, R. C., Dobbins, R.W., and Hutton, L.V., 1990."Performance Metrics", in *Neural Networks PC Tools*, Eberhart, R. C., and Dobbins, R.W.(eds). San Diego, CA:Academic Press.

Fama, Eugene F., 1991."Efficient Capital Markets II", *Journal Of Finance,* 5:1515-1619.

____, 1970. "Efficient Capital Markets:A Review Of Theory And Empirical Work", *Journal Of Finance,25:383-417.*

____, Kenneth R. F. , 1988. "Permanent And Temporary Components Of Stock Prices", *Journal Of Political Economy*,96:246-273.

Finhoff, W., Hergert, F. And Zimmermann,H.G., 1993, "Neural Networks:The Future of Forecasting in Finance?", *Siemens Journal*, 1:117-120.

French K.R., and Roll, R.,1986,"Stock Return Variances", *Journal of Financial Economics*, 17:5-26.

Hecht N., Nielsen R.,1990. *Neurocomputing,* Reading, MA: Addison-Wesley.

Hoptroff, R. G., 1993. "The Principles And Practice Of Time Series Forecasting and Business modelling Using Neural Nets". *Neural Computing & Applications*,1: 59-66.

Kryzanowski, L., Galler, M., and Wright, W.,1993. "Using Artificial Neural Networks to Pick Stocks", *Financial Analysts Journal*. 4:21-27.

Landi L., Barucci, E., 1993. "Neural Nets for Financial Time Series Prediction", unpublished.

Lo, A.W., and Mackinlay, A.C.,1988.""Stock Mkets do not Follow Random Walks:Evidence From A Simple Specification Test", *Review of Financial Studies*, 1:41-66.

Lukac, L.P., And Brorsen, B.W.,1989."The Usefulnessof Historical Data In Selecting Parameters For Technical Trading Systems",*The Journal of Futures Markets*,9:55-65.

Poterba, J. And Lawrence S, 1988. "Mean Reversion In Stock Prices: Evidence And Applications", *Journal Of Financial Economics*,22:27-59

Rawani, A.M., and Mophatra, D.K.,1993. "Forecasting Strategy For the Foreign Exchange Market", Information and Decision Technologies, 19: 55-62.

Refenes, A. N., 1993. "Currency Exchange Rate Prediction And Neural Netwok Design", Neural Computing And Applications, 1:46-58.

Sharda, R., and Patil, R.B.,1990."Neural Networks As Forecasting Experts:An Empirical Test", *International Joint Conference on Neural Networks*, II:491-494.

Skapura, D. M., 1991. Neural Networks And Applications, New York:Addison-Wesleypublishers.

Summers, Lawrence H., 1986. "Does The Stock Market Rationally Reflect Fundamental Values?", *Journal Of Finance,* 41:347-368.

The Economist 1993."A Survey Of The Frontiers Of Finance: Mathematics Of Markets", *The Economist,* 7832:17-19.

White, H., 1988."Economic Prediction Using Neural Networks", *IEEE International Conference On Neural Networks*, II:451-458.

Wong, F.S., 1992. "Fuzzy Neural Systems for Stock Selection", *Financial Analysts Journal*, 1:21-29.