Modeling and Forecasting Financial Time Series, comparison of forecasting ability of neural networks, genetic programming and econometric methods

Mohammad Ali Farid	Faiza Ather	Omer Sheikh	Umair Siddiqui
2003-02-0111	2003-02-0057	2003-02-00129	2003-02-0206
LUMS.	LUMS.	LUMS.	LUMS.

Financial time series like stock prices and exchange rate are non linear and non trivial in nature. The series is stochastic which makes it difficulty for modeling and prediction. Traditionally statistical methods like statistical clustering and regression analysis have been used for modeling the series. However most of models are more suitable for linear processes and their application to nonlinear series have generally shown less satisfactory results. Since the later half of the last decade, developments in the field of artificial intelligence and soft computing have made possible the use of neural networks for financial forecasting. Neural Networks, inspired from human neural system have the ability to approximate non linear functions. A further development in the field of AI has been the evolutionary regression or genetic programming method. Designing neural networks and genetic programming for robust financial prediction is a subject of on going research. This paper employs neural networks, genetic programming and regression based methods for modeling exchange rate series. Experimentation has been attempted with the input output set and the design of neural networks to achieve accurate modeling. This paper discusses the experimentation methods and the modeling techniques, which have been used, and compares the results that have been obtained from them. The results show that radial basis networks are the most suitable for forecasting the series and this model leads to very accurate prediction.

Introduction

Exchange rate forecasts play a significant role in the decision-making process involving economic policies and financial investment. However the prediction is hampered by uncertainty of the financial time series. The series is stochastic and non linear in nature, this nonlinear limits the performance of statistical regression based methods like ARMA and ARIMA. These methods use bit-wise linear approximation and hence show less satisfactory results where uncertainty and nonlinearity is present.

During the nineties, advances in computational methods especially in the field of AI made the use of neural networks for financial prediction. Neural Networks are based on the design of the human neural system and are used for the tasks of pattern recognition, classification, and function approximation. Studies by Refenes et al. (1995), Steiner et al. (1995), Freisleben (1992) and Abu-Mostafa (1995), all show that neural networks can outperform the statistical methods in accuracy of prediction. However, perhaps due to the secrecy in the methods of financial prediction kept to preserve the 'predicting edge' by many researchers or perhaps due to the novelity of neural networks, the regression based methods are still the most widely used. Most

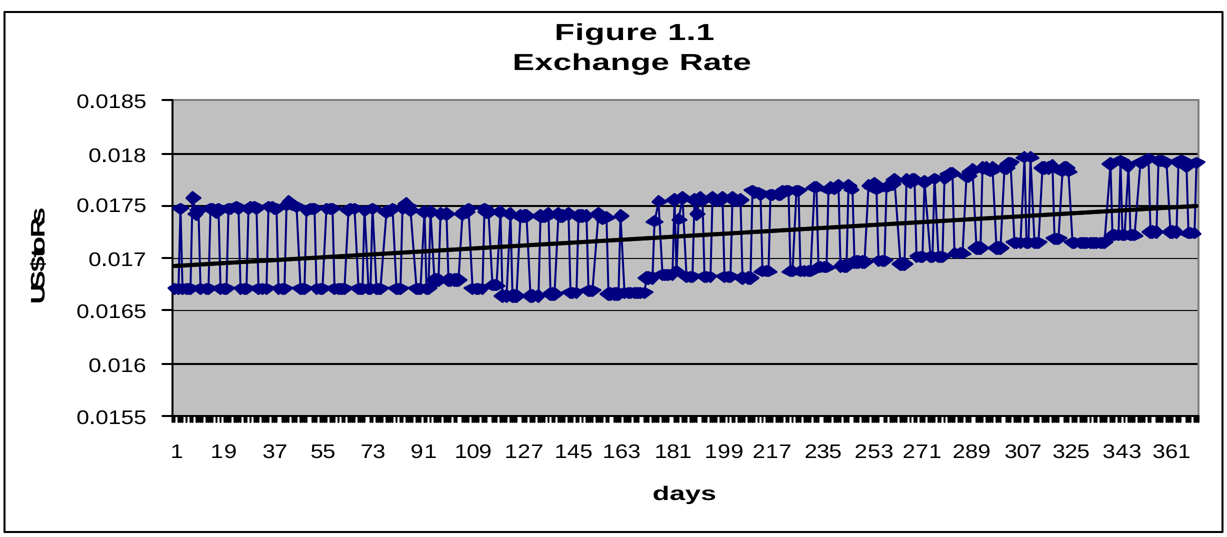
textbooks on time series modeling still focus on econometric methods. The results of this paper show that perhaps it is time to burry the tradition.
Developing neural networks is an art itself (Yao and Tan 1999), design of the network is a complex task. Various models of neural networks exist like the radial basis, feedforward networks, Jordan networks etc. In this paper we have used three primary network models, the feedforward networks, elman recurrent networks and radial basis networks. In these network models further experimentation has been done with the architecture, selection of activation functions and training algorithms. The results of prediction obtained after the experimentation has been used to reach the most optimal network structure.

The third technique used in this paper is the method of evolutionary regression or genetic programming. Genetic Programming is a variant of genetic algorithms and is a recent innovation in the field of Artificial Intelligence. This paper uses the Time Series Genetic Programming system developed by Mahmoud Kaubdan (Kaunbdan 1999) for forecasting the exchange rate.

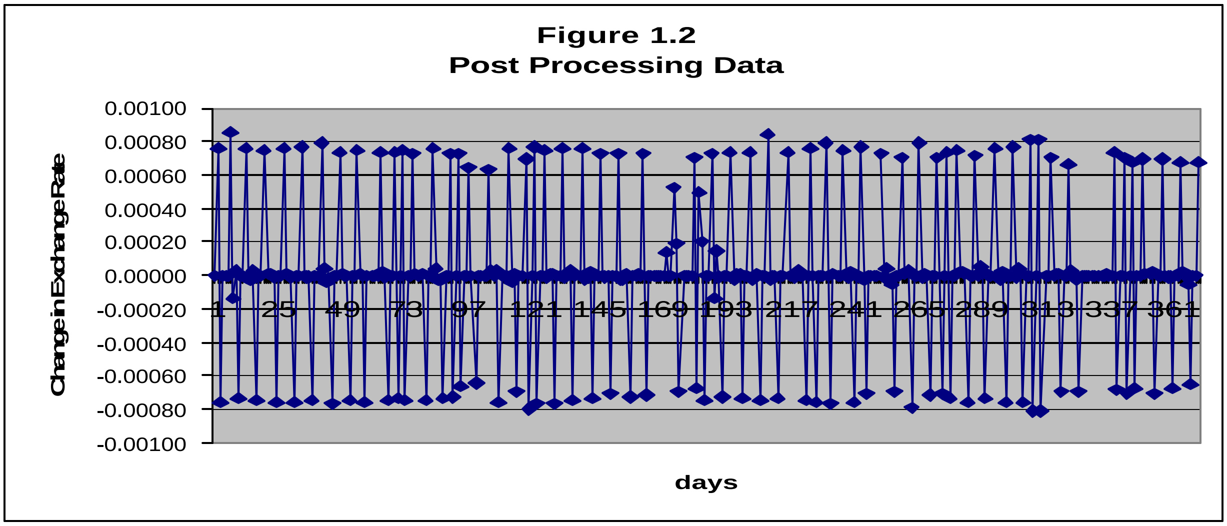
The financial time series used by the paper is the daily exchange rate between US Dollar and Pakistani Rupee. The data consists of 371 points and taken from the online portal of Onada Currency Changers (Onada 2003). In this paper we have also experimented with the data set, the methodology and results will be presented later. The next section analyzes the data and performs processing on the data.

Data Analysis

The financial time series chosen for the project is one year daily exchange rate between Pakistani rupee and US Dollar. For the current statistical study we have chose data from 30th Jan 2002 to 4th February 2003. The reason this period is used is because of the stability of the period. The data in this range was quite smooth and had no outliers. The political and economic stability during this period was one of the reasons for the bounded range of the data. Also during this time period the government has implemented the structural adjustment program with an aim of stabilization. Inflation has been stable and there have been no external shocks to the economy. Figure 1.1 shows the series under study.



The application of Dicky Fuller Unit Root test shows that the series is not stationary, this is also evident from the inspection of the rising black trend line in the data. The mean and variance of the data is changing with time. Stationarity of the series is a requirement of all modeling techniques and hence the trend in the series needed to be removed. The series was transformed to a stationary series by applying first order differencing. This preprocessing removed the trend and made the series stationary. Figure 1.2 shows the new series.



This series gives the daily change in exchange rate and is used for the modeling by the study. The preprocessing generates the series

$$x(t) = x(t) - x(t-1)$$

This differencing process is similar to the differentiation of continuous functions (Since the data is bounded so no other operators like the logarithmic operator etc. are required.

Input-Output Selection

The modeling technique used in this paper assumes that the past values of the data can be used to predict the future value. Hence it used an autoregressive model where the past values are used to forecast the future.

$$X(t)=F^{nn}\ (x(t-1), x(t-2), \dots x(t-n)) + u(t)$$

The selection of n the number of inputs and size of X(t) the output is based on statistical analysis of the data. This windowing size has been shown to be an important variable that affects the accuracy of predictability (Zekic (1997). This paper uses four data sets based on different sizes of windows and different data processing. The data sets used in this paper are:

1) Data Set A: Primary series, change in exchange rate, daily values with 7-1 window.
2) Data Set B: 14-1 window.
3) Data Set C: Moving averages, average of three days with 7-1 window.
4) Data Set D: 7-3 window

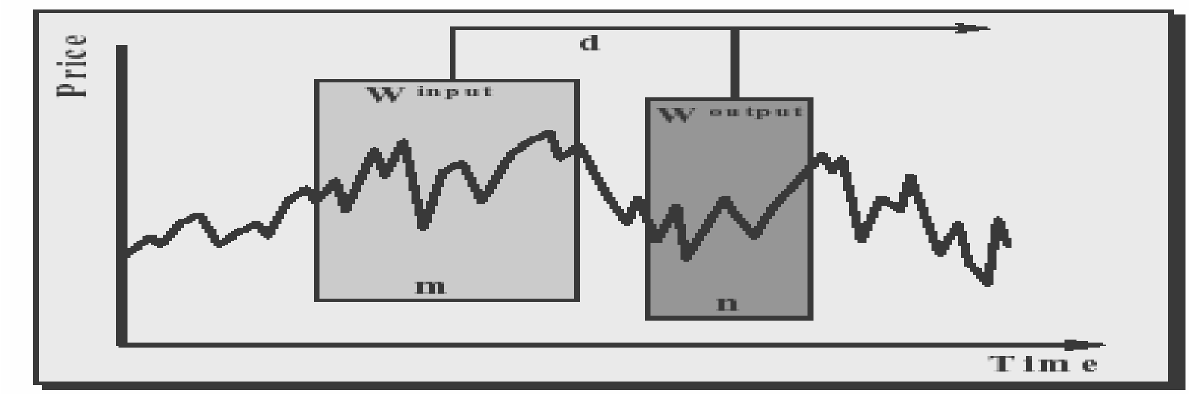
Data Set A, B and C are of the type:
$[X_1, X_2 \ldots\ldots X_n] \rightarrow [X_m]$
$[X_2, X_3 \ldots\ldots X_{n+1}] \rightarrow [X_{m+1}]$

while Data set C uses a higher level of processing, and is of the type:

$[\text{average}(X_1, X_2, X_3), \text{average}(X_2, X_3, X_4), \ldots\ldots \text{average}(X_{n-2}, X_{n-1}, X_n)] \rightarrow [\text{average}(X_{n+1}, X_{n+2}, X_{n+3})]$
$[\text{average}(X_2, X_3, X_4), \text{average}(X_3, X_4, X_5), \ldots\ldots \text{average}(X_{n-1}, X_n, X_{n+1})] \rightarrow [\text{average}(X_{n+2}, X_{n+3}, X_{n+4})]$

The series of data was divided into two sets, one was used for training the model and the next set was used for testing. First 300 points were included in testing and the rest 50 were for training.

Figure 1.3



Source: Schwaerzel(1996)

Data Set a is the primary data and is based on statistical analysis of the partial autocorrelation plots of the data.

Literature Review

Econometric Model

If a time series is stationary, it can be modeled in a variety of ways. One method is the Auto Regressive (AR) model. The Auto-Regressive model comprises of a value at time *t* that is regressed on its own lagged values (hence the term auto-regressive). This means that a value in a time series at time *t* is dependent on its own values in previous time periods.

In our case the time series data is of the Exchange rate at time t, *E(t)*. This can be modeled as:

$$(E(t) - d) = a1 \; (E(t-1) - d \;) + \mu(t)$$

where *d* is the mean of *E(t)* and *μ(t)* is an uncorrelated random error term with zero mean and constant variance. We can say that *E(t)* follows a first order auto-regressive or AR(1) model.

Hence it depends on its own lag of one period and on a random error term. Here the exchange rate values are expressed as deviations from the mean value.

To allow for the assumption of stationarity in our model, we took the first difference of our time series data. Since we are using differenced data to keep inline with our models assumption of stationarity, our AR(1) model (that is Auto-Regressive model with one lag) can be expressed as:

$$(?E(t) - d) = a1\ (?E(t-1) - d) + \mu(t)$$

where *d* is the mean of *?E(t)* (the first differenced Exchange rate data) and *µ(t)* is again an uncorrelated random error term with zero mean and constant variance. We can express this model as follows: the change in Exchange rate in a particular time period depends on a proportion (that is, *a1*) of the change in Exchange rate in the previous period plus a random shock or disturbance term at time *t*.

This model may be extended to two lags as follows:

$$(?E(t) - d) = a1\ (?E(t-1) - d) + a2\ (?E(t-2) - d) + \mu(t)$$

In this model the change in exchange rate is second order auto-regressive or AR(2) process. This means that the value of the change in Exchange rate depends on its own value in the previous two time periods.

In general we can have:

$$(?E(t) - d) = a1\ (?E(t-1) - d) + a2\ (?E(t-2) - d) + \ldots\ldots\ldots\ldots + ap\ (?E(t-p) - d) + \mu(t)$$

In this case the change in exchange rate is auto-regressive of order *p*.

This regression model is estimated using the method of least square (OLS), where the sum of squared errors of the model is minimized. The coefficients of the model which minimize the error function are reached. This method assumes that the residuals or the errors are normally distributed and random.

Genetic Programming

Genetic Algorithms were invented to mimic some of the processes observed in natural evolution. Many people, biologists included, are astonished that life at the level of complexity that we observe could have evolved in the relatively short time suggested by the fossil record. The idea with GA is to use this power of evolution to solve problems.

Most symbolic AI systems are very static. Most of them can usually only solve one given specific problem, since their architecture was designed for whatever that specific problem was in the first place. Thus, if the given problem were somehow to be changed, these systems could have a hard time adapting to them, since the algorithm that would originally arrive to the solution may be either incorrect or less efficient. The architecture of systems that implement genetic algorithms (or GA) is more able to adapt to a wide range of problems.

In 1992 John Koza used genetic algorithms to evolve programs to perform certain tasks. He called his method *genetic programming* (GP).

Genetic programming is a branch of genetic algorithms. The main difference between genetic programming and genetic algorithms is the representation of the solution. Genetic programming creates computer programs in the LISP or Scheme computer languages as the solution whereas GA's create a string of numbers that represent the solution.

Genetic programming uses four steps to solve problems:

1. Generate an initial population of random compositions of the functions and terminals of the problem (computer programs).
2. Execute each program in the population and assign it a fitness value according to how well it solves the problem
3. Create a new population of computer programs
   - Copy the best existing programs
   - Create new computer programs by mutation
   - Create new computer programs by crossover (sexual reproduction)

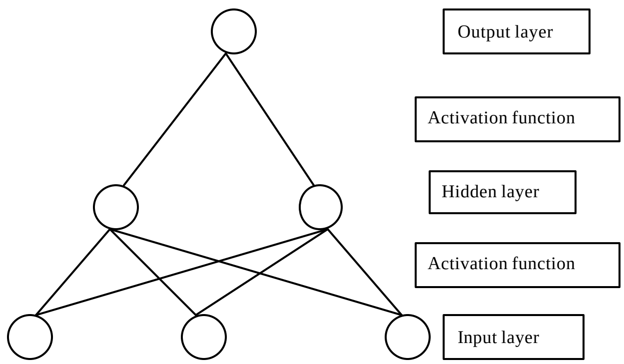
4. The best computer program that appeared in any generation, the best-so-far solution, is designated as the result of genetic programming [Koza 1992].

For the purpose of this project we used software called TSGP written by Mahmoud Kaboudan (School of Business, University of Redlands). (M. Kaboudan, (2002), TSGP: Time Series Genetic Programming Software, www.compumetrica.com.)

Neural Networks

Feed forward systems

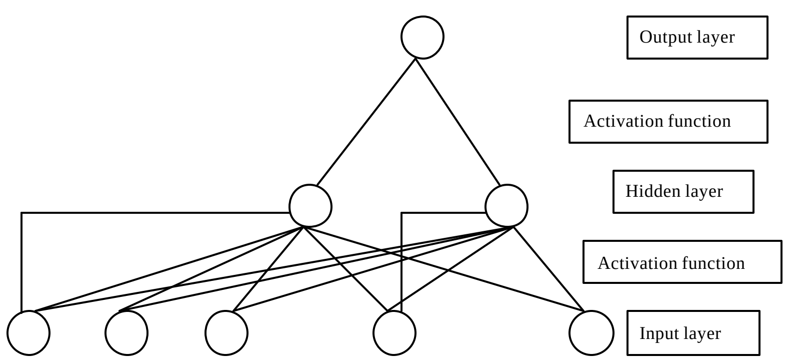
Feed forward neural networks are the basic kind of neural networks, which are used for modeling of financial time series. The network consists of neurons, which are placed, in layers. The data is provided to the input layers, which pass it to the hidden layers. The neurons at the hidden layers apply some function at the inputs and pass it to the output layer. The output is determined by the application of the transfer function on the summation of the inputs weighted by the weights of each neuron. The function activation function applied can be linear or non linear type like sigmoid and hyperbolic. The figure below shows the representation of a simple feedforward neural network.



A simple feed forward network with one output unit, two hidden units and three input units

Recurrent Networks

Recurrent neural networks like Elman and Jordon network also have back linkages from the output to the inputs. This gives them a dynamic nature. Also they can capture the autoregressive kind of relationships where the output is not only dependent on previous inputs but also on the output itself. The figure below shows a kind of recurrent network called the Elman network which has been extensively used for prediction of stochastic systems.



A simple Elman (1990) network with hidden unit activation feedback

Radial Basis

Radial Basis Function Networks are based on alternate view of the neural network design, that is, a curve fitting (approximation) problem in a high dimensional space. According to this viewpoint learning is similar to finding a surface in a multi-dimensional space that provides a best fit to the training data, with the criterion for best fit being measured in some statistical sense. Therefore, any generalization is equivalent to the use of this multi-dimensional surface to interpolate the test data. This viewpoint is the motivation behind the method of radial basis functions. In the context of a neural network the hidden units provide set of “functions” that constitute an arbitrary “basis” for the input patterns (vectors) when they are expanded into the hidden-unit space; these functions are called *radial-basis functions*. These networks are used for function approximation.

Experimentation Methodology

As mentioned earlier the design of an optimal forecasting model requires lot of experimentation with the structure of the model and other related factors like the size of the input output windows. The windowing variation has been discussed earlier in this paper. In this section we shall describe the experimentation attempted with the designing of the neural networks, the specification of genetic programming and the econometric model.

Neural Networks

For this research we trained and tested more than one hundred neural networks. These neural networks have been implemented using Matlab 6.1 Neural Networks toolbox. Three models of neural networks were used. These were:

1) Feedforward.Network
2) Recurrent Elman Network
3) Radial Basis

Within these networks, the architecture, activation function and learning algorithms were varied to reach the most optimal prediction.

With the feedforward and recurrent networks, the number of layers in the network were varied from one to four, logsigmoid, tansigmoid and linear activation functions were used in various layers, and two learning algorithms, gradient descent with momentum and gradient descent without momentum were compared. In radial basis the spread of the basis function was the only factor which was experimented with.

Econometric Model

The autoregressive model employed uses a linear autoregessive form

?E(t) = a + ß1 ?E(t-1) + ß2 ?E(t-2) + ß3 ?E(t-3)
+ ß4 ?E(t-4) + ß5 ?E(t-5) + ß6 ?E(t-6)
+ ß7 ?E(t-7) + ß8 ?E(t-8) + ß9 ?E(t-9)
+ ß10 ?E(t-10) + ß11 ?E(t-11) + ß12 ?E(t-12)
+ ß13 ? E(t-13) + ß14 ?E(t-14) + ß15 ?E(t-15)
+ ß16 ?E(t-16) + U(t)

Staring from an AR process of 16 lags i.e. AR(16), we use Hendry’s General to specific approach to iteratively build the model. Those variables whose coefficients fail the T test for significance are removed from the model. Monte Carlo simulations are done to find the confidence intervals for the tests. The final model estimated for the prediction is:

?E(t) = a + ß1 ?E(t-1) + ß2 ?E(t-2) + ß3 ?E(t-3)
+ ß4 ?E(t-4) + ß5 ?E(t-5) + ß6 ?E(t-6)
+ ß7 ?E(t-7) + ß8 ?E(t-8) + ß9 ?E(t-9) + u(t)

Microsoft Excel with regression has been used for the econometrical modeling.

Genetic Programming

The software TSGP used for genetic programming prompts for the following information before it starts execution:

- number of data points in Historical (Training) set: **T**
- total number of data points to Forecast: **k**
- number of data points for ex post Forecast: **f**
- population size: **p**

  Selecting p is important and is dependent on the relative complexity of the variable to evolve models for. Selecting a small population size may easily provide less than optimal models. It is therefore prudent to use a population size of 1000 or more especially when dealing with data whose data generating process is unknown.
- number of generations: **g**

  The number of generations' g is also dependent on the relative complexity of the dependent variable. For simple processes perhaps g = 100 is sufficient.
- number of explanatory variables: **n**
- number of searches desired: **s**

  Because GP gets trapped in local minima, it is necessary to evolve a large number of equations or models

Since a fitness function is to be evaluated in GP, the program is designed to look for solutions that minimize the sum of squared errors in each generation i.e. the fitness function for this program is the sum of squared errors.

The variables that we manipulated to obtain our results were

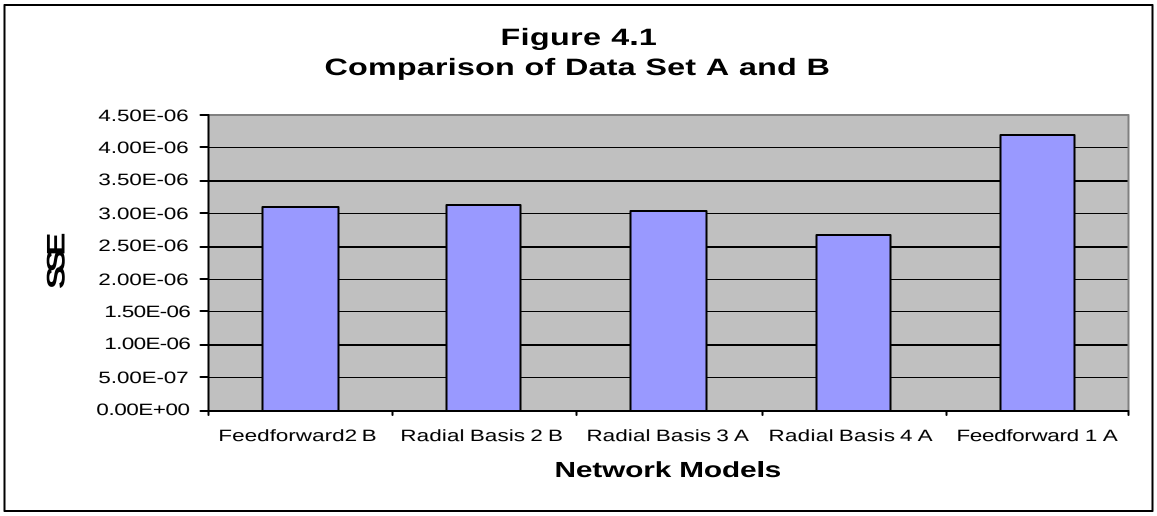
- Data in Historical set (T) (Increase T ➔ search time increases exponentially)
- Total points to forecast (k)
- Population size (p) (interesting observations)

Number of generations (g)

Results

The models constructed using the above mentioned methodology were used to predict the next value for 50 observations. The forecast obtained was the then compared with the actual values and the sum of squared error was calculated. SSE is measure of goodness of forecast used by the paper.

The results obtained showed that data set A was the most optimal for prediction. This is deduced from the finding that the best prediction achieved with data set A is better than the best achieved by the other data sets. This can also be seen from the figure below



Data Set A was the primary data set, based on statistical analysis and apriori it was expected to perform better. With Data Set A, the neural networks, especially radial basis with spread equal to 0.3, and 2 layer feedfoward network gave the best predictions. Genetic programming was better than recurrent network and the econometric model had the highest error.

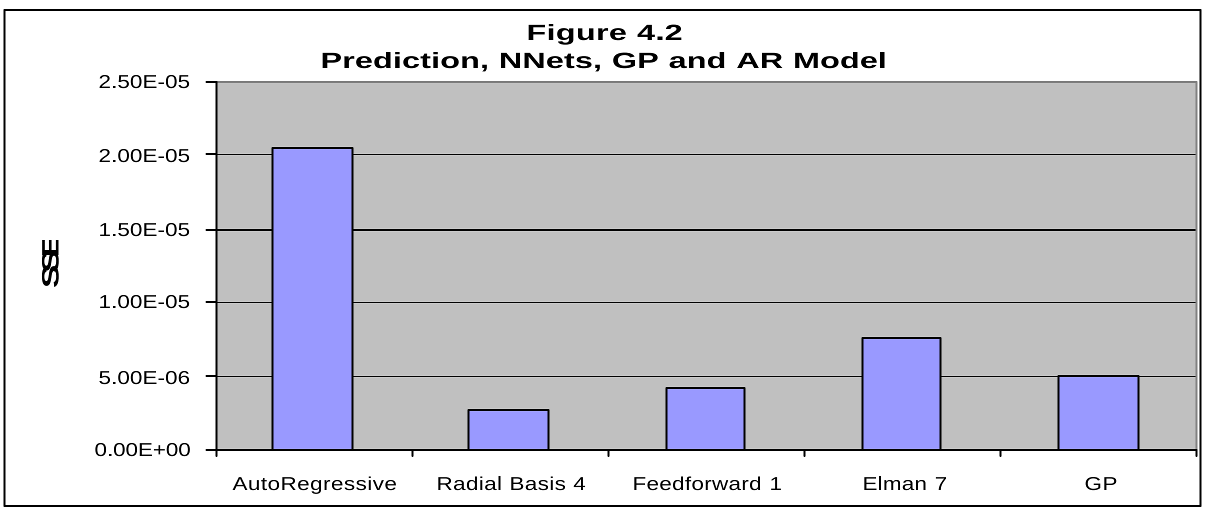


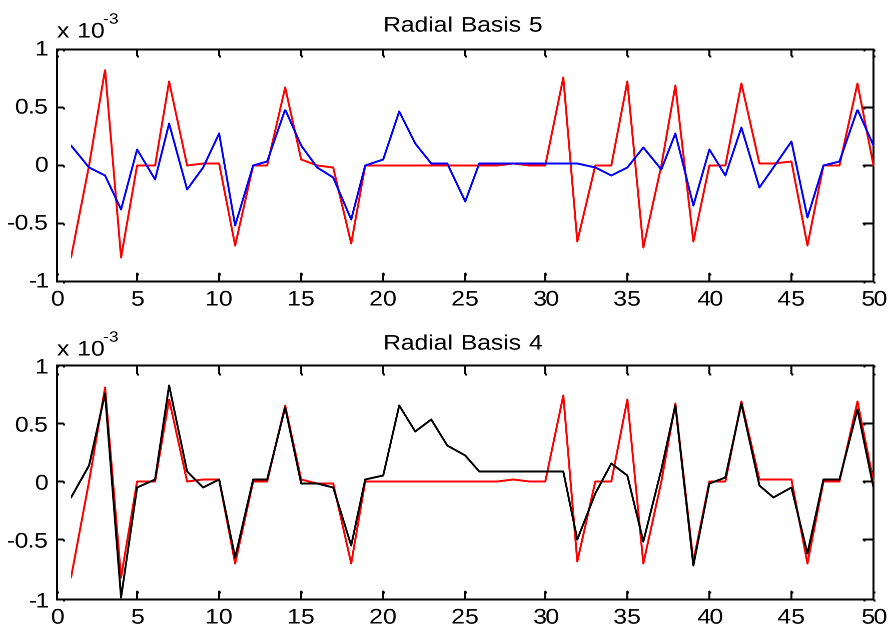
Figure above shows the bar chart of the errors of the various networks.

Table Data Set A: SSE	
Econometric Model	**2.06E-05**
Radial Basis 4	**2.68E-06**
Feedforward 1	**4.19E-06**
Elman 7	**7.66E-06**

The results show that the AR model has the highest error rate which is almost ten times as high as radial basis. Changing the spread of the radial basis affected the prediction and best was achieved where spread = 1/6 * (Range of data). In the feedforward category, a 2 layer structure, with 25 neurons in the hidden layer, logsigmoid in the hidden and linear activation function in the output layer and gradient descent with momentum gave the best prediction of 4.19E-06, however this was more than that achieved with the radial basis. Recurrent network was not a good frecasting among the neural networks, however its predictions were still better than those achieved by AR model.

Table FeedForward: Single hidden Layer	
FF1a	4.24E-06
FF1b	4.40E-06
FF1c	4.12E-06

FF1a: single layer feedforward with 10 neurons and logsigmoid in the hidden layer.
FF1b: tansigmoid in the hidden layer
FF1c: 25 neurons in the hidden layer



Conclusion

The results show that the exchange series under study can be predicted and radial basis give the bets prediction. Radial basis and feedforward are universal approximators and the results confirms the theory on neural network which suggests that feedforward network should be able to approximate any function which radial basis networks can. These findings add to findings listed earlier which show that neural networks outperform statistical methods. Also we have shown than even genetic programming outperform econometric methods and give good prediction of exchange rates. This study also shows that the EMH does not hold for this financial market.

Some of the questions for further research can be developing trading rules based on these findings. Other methods of forecasting could try building some hybrid model combining neural networks with genetic programming. Using Expert systems with neural networks cab also be another area to work on.

Bibliography

1) Dorffner, Georg (?), Neural Networks for Time Series Processing, University of Vienna.
2) Gilesm L, Lawrence, S. and Tsoi, A.C. (1997), Rule Inference for Financial Prediction Using Neural Networks, IEEE.
3) Liu, T, and Ming C. (1995), Forecasting Exchange Rates Using Feedforward and Recurrent Neural Networks, Journal of Applied Econometrics.
4) McClusky, P. (1993), Feedforward and Recurrent Neural Networks and Genetic Programs for Stock Market and Time Series Forecasting, Master Thesis, Brown University.
5) Mahfoud, S. and Mani, G. (1996), Financial Forecasting Using Genetic Algorithms, Taylor and Francis.
6) Herbrich, R., Keilbach, M., Graepel T., and Obermayer, K. (1999), Neural Networks in Economics.
7) Tan, C. (1999), Soft Computing Applications in Finance, Queenland Finance Conference.

8) Refenes, A. N, Azema-Barac, M., Chen, L., and Karaussos, S.A., (1993), Currency exchange rate prediction and neural network design strategies, Neural Computing and Applications.
9) Pham, D. T. and Liu, X, (1995), Neural Networks for Identification, Prediction and Control, Springer-Verlag London.
10) Mathworks Inc (2002), Neural Networks ToolBox, Mathworks.
11) M. Kaboudan, (2002), TSGP: Time Series Genetic Programming Software, www.compumetrica.com
12) Hornik, K., Stinchcombe, M and H. White (1989), 'Multi-layer feedforwared networks are universal approximators' , *Neural Networks*, **2,** 359-366.
13) Rosenblatt, M., (1962), *Principles of neurodyanmic: Perceptron amd Theory of Brain Mechanisms.* Washington DC., Spartan Books.
14) Zekic, Marijan (1997), Neural Network Aqpplications in Stock Market Predictions, Croatia.