

Cash Forecasting: An Application of Artificial Neural Networks in Finance

PremChand Kumar

IT Services Department, State Bank of
India, Local Head Office, Sector 17-B,
Chandigarh, India
(premkchand@gmail.com)

Ekta Walia

Department of Computer Science,
National Institute of Technical Teachers
Training & Research
Sector 26, Chandigarh, India
(wekta@yahoo.com)

Abstract

Artificial Neural Networks are universal and highly flexible function approximators first used in the fields of cognitive science and engineering. In recent years, Neural Networks have become increasingly popular in finance for tasks such as pattern recognition, classification and time series forecasting. The ability to predict cash requirement within reasonable accuracy of actual demand provides target for supply optimization well in time. Every financial institution (large or small) faces the same daily challenge. While it would be devastating to run out of cash, it is important to keep cash at the right levels to meet customer demand. In such case, it becomes very necessary to have a forecasting system in order to get a clear picture of demand well in advance. This paper presents two neural network models for cash forecasting for a bank branch. One is daily model – taking the parameter values for a day as input to forecast cash requirement for the next day and the other is weekly model, which takes the withdrawal affecting input patterns of a week to predict cash requirement for the next week. The system performs better than other cash forecasting systems. This system can be scaled for all branches of a bank in an area by incorporating historical data from these branches.

1. Introduction

Estimating and forecasting future conditions govern many critical business activities, such as inventory control, procurement of supplies, labour cost estimation, prediction of product demand, and prediction of cash requirement of a bank branch or ATM. Such predictions have been difficult, if not impossible. Incomplete understanding leads us to develop models that have uncertainties. In such cases, we resort to system identification models based on historical data. Input/Output data relevant to the application at hand must be collected over a period of time and analyzed by automated model fitting procedures. The key to all these forecasting applications is to capture and process the historical data such that it provides insight into the future. The ability to predict the future demand estimate of currency within a reasonable accuracy is called cash forecasting. Cash forecasting is integral to the effective operation of an ATM/branch network. The primary objective of cash forecasting is to ensure that cash is used efficiently and effectively throughout the branch network. In this work, the problem of cash forecast has been studied in respect of a cash-balance branch and attempt has been made to:-

- Adapt this forecasting model to the specific conditions in the various regions.
- Carry out short-term forecasts on the basis of alternative scenarios of the economic development.
- Include a comprehensive environmental impact analysis for forecast.
- Ensure implementation of forecasting tool.

2. Existing Approaches to Cash Forecasting

Techniques used for cash forecasting can be broadly classified into three groups.

- a) Time-series Method
- b) Factor analysis Method
- c) Expert system approach

2.1 Time Series Method

This method predicts future cash requirement based on the past values of variable and/or past errors. The objective here is to discover the pattern in the historical data series & extrapolate that pattern into the future. The assumption in this method is that a cash requirement pattern is nothing more than a time series signal with known hourly, daily and seasonal periodicities. The difference between the prediction and the actual can be considered stochastic. By the analysis of the random signals, a more accurate prediction can be obtained. Problems inherent with the time series approach include the inaccuracy of prediction and numerical instability. Generally, techniques in time series approach work well unless there is an abrupt change in the environment or sociological variables that are believed to affect the cash pattern.

2.2 Factor Analysis Method

This method is based on the determination of various factors that influence the cash requirement pattern, and calculating their correlation with cash. The purpose is to determine the functional form of this influence (independent variables) and to use this to forecast future values of the dependent variables. The general approach consists of identifying the independent variables and then assuming a basic functional relationship between the dependent and independent variables, and finally determining the coefficients of the assumed functional relationship. The main problem lies in the selection of the functional relationship, this usually has to be an iterative process, i.e. one assumes a certain form to the data, tests its validity, then assumes another form and tests its validity, and continues until a close fitting valid functional form is obtained. Estimating the functional form between the dependent and independent variables is more difficult when they have a non-linear relationship.

2.3 Expert Systems Approach

Expert systems are heuristic models which are usually able to take both quantitative and qualitative factors into account. Many models of this type have been proposed. A typical approach is to try to imitate the reasoning of a human operator. The idea is then to reduce the analogical thinking behind the intuitive forecasting to formal steps of logic. Efforts to make expert systems general have run into a number of problems. As the complexity of the system increases, the system simply demands too much computing resources and becomes too slow. Expert systems have been found to be feasible only when narrowly confined.

2.4 Neural Networks Approach

Artificial neural networks offer a completely different approach to problem solving and they are sometimes called the sixth generation of computing. They try to provide a tool that programs itself and learns on its own. Neural networks are structured to provide the capability to solve problems without the benefits of an expert and without the need of programming. They are capable of seeking patterns in data.

Artificial neural networks (ANN) as they are often called refer to a class of models inspired by biological nervous systems. Neural networks forecasting has recently enjoyed considerable success in pattern recognition and prediction and as such has gained considerable research attention resulting in a plethora of articles on this subject. The concept is based on computing systems that are able to learn through experience by recognizing patterns existing within a data set. Neural systems require their implementer to meet a number of conditions. These conditions include:

- A data set which includes the information which can characterize the problem.
- An adequately sized data set to both train and test the network.
- An understanding of the basic nature of the problem to be solved so that basic first-cut decision on creating the network can be made.
- These decisions include the activation and transfer functions, and the learning methods.
- An understanding of the development tools.

- Adequate processing power (some applications demand real-time processing that exceeds what is available in the standard, sequential processing hardware. The development of hardware is the key to the future of neural networks).

Once the necessary inputs (factors) are identified, it is relatively simple to train neural network to form a non-linear model of the underlying system and then use this model to generalize to new cases that are not part of the training data.

3. Artificial Neural Network Model for Cash Forecasting

The general idea behind use of ANN in cash forecasting is that of allowing the network to map the relationships between various factors affecting the cash withdrawal and the actual cash withdrawal. Once this relationship between inputs and outputs is mapped, it gives the cash forecast after accepting the parameter values for various factors affecting the cash withdrawal as input.

The evaluation of these cash forecasts systems has been done on the basis of standard statistical measures like percentage errors. The forecast error for each pair of actual and forecasted cash withdrawal can be given by

$$\text{Error}_i = \frac{(\text{withdrowal}_{\text{actual}})_i - (\text{withdrowal}_{\text{predicted}})_i}{(\text{withdrowal}_{\text{actual}})_i} \times 100\%$$

for $i = 1, 2, \dots, N$, where N is the number of testing data points. After calculating the forecast error, forecasting accuracy can be calculated as:

$$\text{Forecasting Accuracy} = 100 - \text{\%age of error in forecasting}$$

Another most important criterion used for the evaluation of the systems is generalization. A network is said to be generalized, when the output is correct (or close enough) for an input which has not been included in the training set.

3.1 Implementation

This cash forecasting method has been implemented using MATLAB, on a Pentium IV machine under Windows XP Professional platform.

3.2 Data Collection

One of the most important components in the success of neural network solution is the data collection. The quality, availability, reliability and relevance of the data used to develop and run the system are critical to its success.

For the implementation of Cash Forecasting for a bank branch, real data for three months regarding the cash withdrawal was collected from a Chandigarh (India) based branch of State Bank of India. The reason for selecting this particular branch for data collection was to gather data from such a branch which has accounts of different kinds, particularly – salary accounts in order to have more data patterns. The data collection period was from 2nd April 2004 to 30th June 2004. The values for other factor viz. holidays, Sundays, salary days etc were also included. This whole data set was then divided into two sets:-

Training Set	From 2 nd April 2004 to 14 th June 2004 for training the neural network
Validation Set	From 15 June 2004 to 30 th June 2004 for validating the network.

Table 1: Data Sets

Some of the test sets were taken initially from the training data itself to set all the parameters and finally these estimated parameters were validated with non-training data. Two models have been developed and tested viz.

1. Daily Model
2. Weekly Model

3.3 Architecture of the Model

Design of right architecture involves several important steps:

- a. Selecting the number of layers
- b. Basic decision about the amount of neurons to be used in each layer
- c. Choosing the appropriate neurons' transfer functions

If there are not enough neurons in each layer, the outputs will not be able to fit all the data points (under-fitting). On the other hand, if there are too many neurons in each layer, oscillations may occur between data points (over-fitting). Therefore, a topology study was conducted in order to find the most appropriate architecture neural network to fit Cash forecasting parameters. There are several combinations of neurons and layers but for practical purposes we tested the following:

3.3.1 Number of Layers

The model is a three layer feed-forward neural network and was trained using fast back propagation algorithm because it was found to be the most efficient and reliable means to be used for this study. Table 2 shows a comparison of the two algorithms.

Function	Technique	Time (s)
TRAINBP	Backpropagation	59
TRAINBPX	Fast Backpropagation	40

Table 2: Selection of Algorithms

3.3.2 Input Layer Size

Number of input neurons depends upon:

1. Number of cash withdrawal factors included in the model
2. Way these factors are encoded

Mainly calendar effects are included as parameters affecting cash withdrawal in this model, viz.:

- Working day
- Weekday
- Holiday effect
- Salary day effect

Total number of input neurons needed in this model hence is four, each representing the values of an individual variable at a particular instant of time.

3.3.3 Output Layer Size

In this model only one output unit is needed for indicating the value of forecasted cash.

3.3.4 Optimal Hidden Layer Size

There is no easy way to determine the optimal number of hidden units without training using number of hidden units and estimating the error of each. The best approach to find the optimal number of hidden units is trial and error. In practice, we can use either the forward selection or backward selection to determine the hidden layer size.

- **Forward selection:** Starts with choosing an appropriate criterion for evaluating the performance of the network. Then we select a small number of hidden neurons; record its performance i.e

forecast accuracy. Next we slightly increase the hidden neurons, train and test until the error is acceptably small or no significant improvement is noted.

- **Backward selection:** Starts with a large number of hidden neurons and the decreases the number gradually.

For this study, the forward selection approach was used to select the size of hidden layer and best result was with 9 neurons in hidden layer, as evident from the following table 3.

No. of Neurons in Hidden Layer	Mean Forecast Error for 60 data points	Sum Squared error at 5000 epochs
7	9.675	0.004343
8	9.85726	0.004153
9	8.51395	0.002881
10	9.52748	0.003768
11	9.39397	0.003565
12	8.56169	0.003645

Table 3: Optimal Neurons in Hidden Layer

3.3.5 Optimal Transfer Function

As shown in table 4, for best transfer function, **tansig** in hidden (at 9 neurons) and **logsig** in output layer were found to be optimal.

Transfer function	Mean Forecast Error for 60 data points	Sum Squared error at 5000 epochs
logsig-purelin	11.0786	0.007697
tansig-purelin	9.90157	0.004647
logsig-logsig	7.38031	0.002591
tansig-tansig	9.53892	0.004560
logsig-tansig	12.2373	0.011609
tansig-logsig	6.74556	0.002246

Table 4: Optimal Transfer Function

3.3.6 Other Network Parameters

The other parameters setting in an attempt to find the best model were as given in the following table 5.

Network parameters	Optimal value
Learning rate	0.2
Momentum	0.95
Training tolerance (Sum squared error Goal)	0.005
Maximum epoch (iteration)	5000

Table 5: Network Parameters

3.4 Results

The neural network forecasters can give slightly different results with training sessions, the training and forecasting was therefore performed multiple times with each model.

Weekly Model

The percentage forecasting errors in tabular form and in the form of graphs for the test data are shown in table 6, Fig 1 and table 7, Fig 2. Minimum forecast accuracy of about 88-92 % is observed per week and the average forecast accuracy per week is about 95-96%. These facts are verified by validating the model. In validation, the average generalization forecast accuracy observed per week was 95-97% as shown in table 8, Fig 3 and table 9, Fig 4.

Daily Model

Here forecasting accuracy was more than 96% as shown in the table 10, Fig 5 and 6. In this model generalization accuracy of about 94% was observed as shown in the table 11 and Fig 7.

Comparison with other Forecasting Techniques

Though, Neural Network forecast results were not free from errors, these were much better than other techniques - particularly Time series forecasting as detailed in table 12 and Fig 8 and 9. The forecast result was also compared with the result of one of the available Excel Add-in for forecasting "Alyuda Forecaster XL 2.3". This add-in is also based on Neural Network but its results were not better than the result obtained by the proposed method as shown in the Fig 10. Comparisons of predictions and their error (i.e. deviation from the actual) by different forecasting methods are shown in Fig 11 and 12.

Day	Actual Withdrawal (Rs in Lacs)	Predicted Withdrawal (Rs in Lacs)	Error %
02-04-04	24.34	24.31	0.130947

Training Period : 02/04/2004 to 14/06/2004
Forecast Period : 02/04/2004 to 10/04/2004
Average Forecast accuracy (Per week) : 95.72%
Minimum Forecast accuracy (Per week) : 91.47%

03-04-04	13.94	15.05	0.754095
05-04-04	11.21	10.25	8.53469
06-04-04	9.18	7.58	5.62811
07-04-04	7.24	6.99	3.51895
08-04-04	5.7	7.01	4.66032
10-04-04	9.04	7.50	6.74843

Table 6: Testing on Training Data

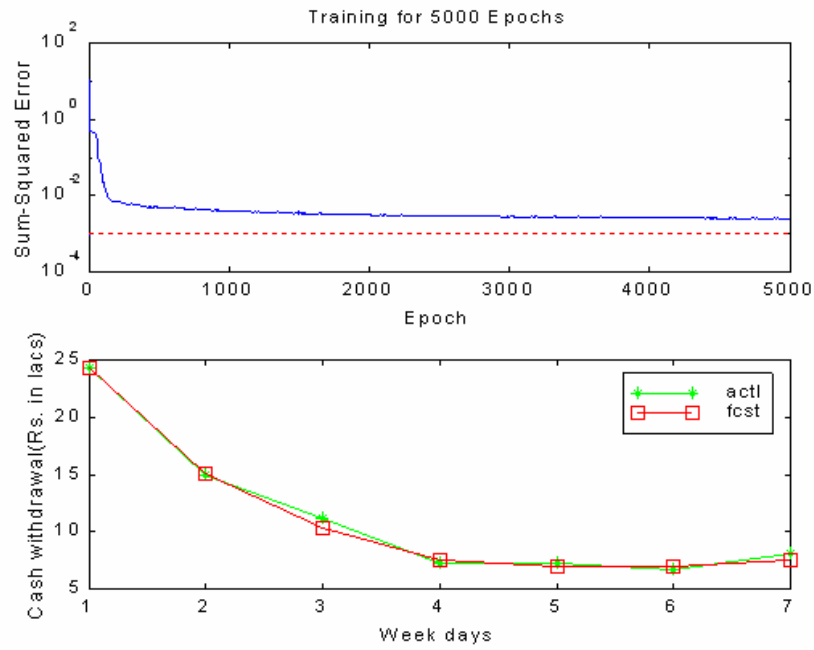


Figure 1: Graph Showing the Actual and Predicted Withdrawal for Training Data of a Week

Day	Actual Withdrawal (Rs in Lacs)	Predicted Withdrawal (Rs in Lacs)	Error %
27-05-04	6.15	6.14	0.117463
28-05-04	6.8	6.04	11.2095
29-05-04	6.37	5.91	7.1785
31-05-04	7.37	6.53	11.4154
01-06-04	27.22	27.37	0.547027
02-06-04	16.98	17.15	0.990085
03-06-04	12.21	11.71	4.13297

Training Period : 02/04/2004 to 14/06/2004
 Forecast Period : 27/05/2004 to 03/06/2004
 Average Forecast accuracy (Per week) : 94.92%
 Minimum Forecast accuracy (Per week) : 88.58%

Table 7: Testing on Training Data

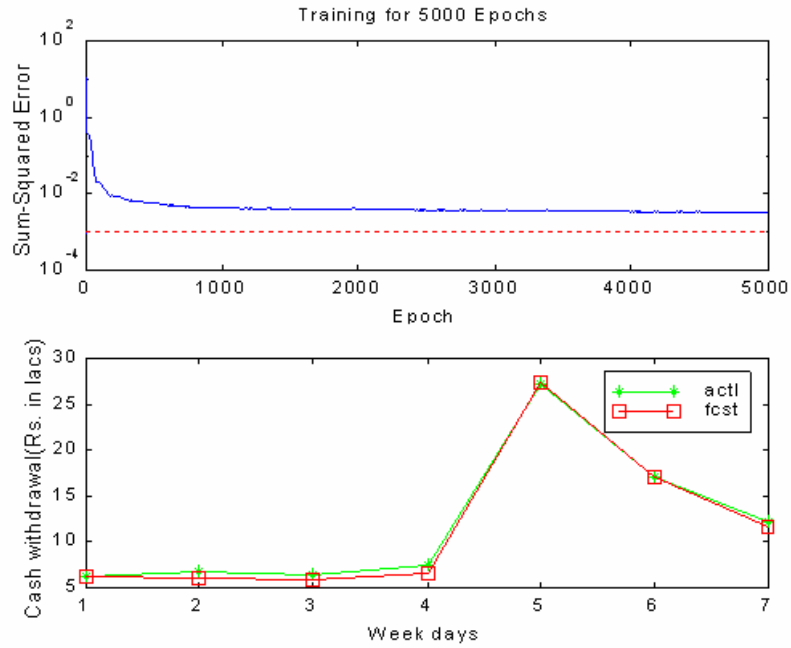


Figure 2: Graph Showing the Actual and Predicted Withdrawal for Training Data of a Week

Day	Actual Withdrawal (Rs in Lacs)	Predicted Withdrawal (Rs in Lacs)	Error %
15-06-04	5.47	5.55	1.54082
16-06-04	5.69	5.30	6.85248
17-06-04	5.47	5.64	3.06653
18-06-04	6.41	6.17	3.67808
19-06-04	5.75	5.94	3.27826
21-06-04	6.1	6.00	1.66546
22-06-04	6.07	5.91	2.67151

Training Period : 02/04/2004 to 14/06/2004
 Forecast Period : 15/06/2004 to 22/06/2004
 Average Generalization Forecast accuracy (Per week) : 96.75%
 Minimum Forecast accuracy (Per week) : 93.15%

Table 8: Validation of the Weekly Model

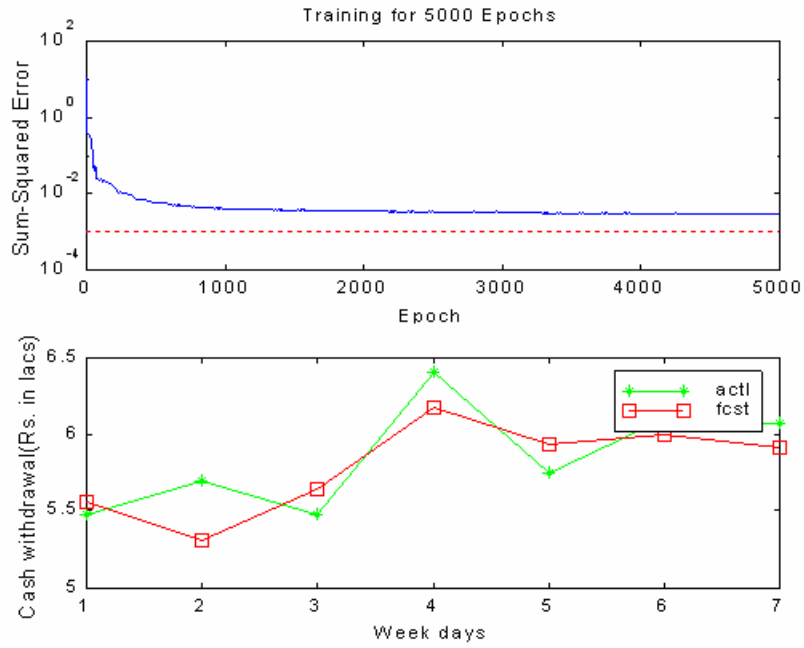


Figure 3: Graph Showing the Actual and Predicted Withdrawal for a Week

Day	Actual Withdrawal (Rs in Lacs)	Predicted Withdrawal (Rs in Lacs)	Error %
23-06-04	5.98	5.98	0.020989
24-06-04	5.8	6.28	8.29067
25-06-04	6.03	6.32	4.72955
26-06-04	7.34	6.61	9.96015
28-06-04	7.58	7.06	6.82313
29-06-04	6.56	6.49	1.08784
30-06-04	6.35	6.62	4.25926

Training Period : 02/04/2004 to 14/06/2004
 Forecast Period : 23/06/2004 to 30/06/2004
 Average Generalization Forecast accuracy (Per week) : 94.97%
 Minimum Forecast accuracy (Per week) : 91.71%

Table 9: Validation of the Weekly Model

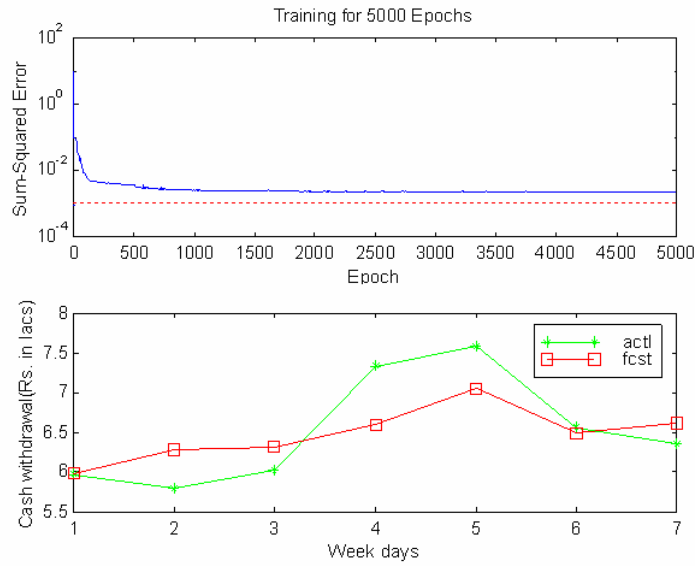


Figure 4: Graph Showing the Actual and Predicted Withdrawal for a Week

Day	Actual Withdrawal (Rs in Lacs)	Test Set	Predicted Withdrawal (Rs in Lacs)	Error %	Average Accuracy percent
01-05-04	25.16	1	24.7623	1.58085	99.19
		2	25.3584	0.788589	
		3	25.1191	0.162754	
		4	24.8134	1.37748	
		5	25.1938	0.134501	
03-05-04	13.91	1	14.0277	0.846243	98.77
		2	13.7383	1.2344	
		3	14.084	1.25099	
		4	13.7198	1.36719	
		5	14.1106	1.44198	
04-05-04	10.21	1	10.7176	4.97149	97.82
		2	10.2479	0.37158	
		3	10.5007	2.84705	
		4	10	2.0564	
		5	10.2747	0.63325	
05-05-04	7.32	1	7.6861	5.00121	96.53
		2	7.4719	2.07499	
		3	7.5497	3.13744	
		4	7.0372	3.8638	
		5	7.5587	3.26095	
02-06-04	16.98	1	17.2366	1.51146	99.42
		2	16.943	0.218099	
		3	17.042	0.365197	
		4	17.0641	0.495084	
		5	17.033	0.31229	
03-06-04	12.21	1	11.8116	3.2631	96.97
		2	11.9082	2.47209	
		3	11.6001	4.99519	

		4	11.7088	4.10509	
		5	12.1694	0.332758	
04-06-04	8.8	1	9.0216	2.51811	97.00
		2	9.2589	5.21514	
		3	9.1982	4.52482	
		4	8.8116	0.1319	
		5	9.0294	2.60629	

Table 10: Testing on Training Data for Daily Model (Five forecast test sets taken for each date)

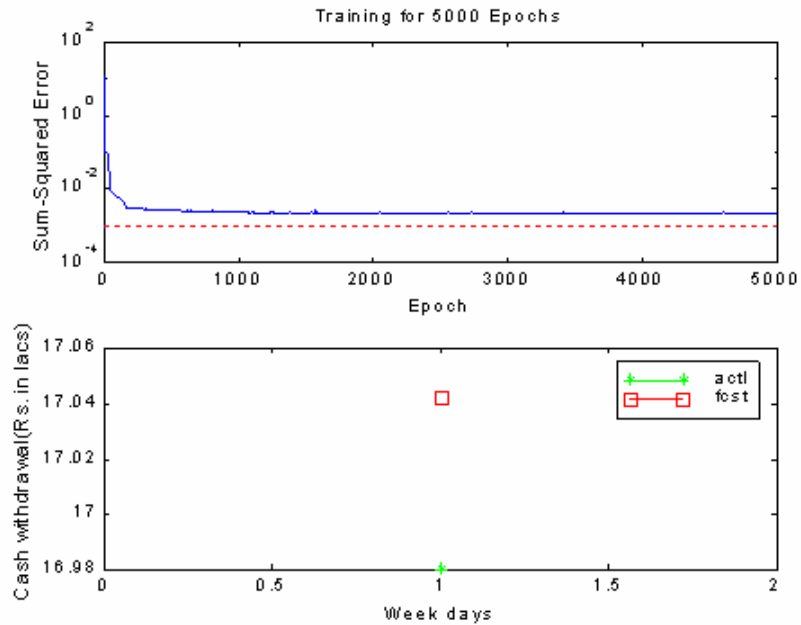


Figure 5: Actual withdrawal for 02/06/2004 = Rs. 16.98 (in lacs), Predicted withdrawal for this day= Rs. 17.04(in lacs)

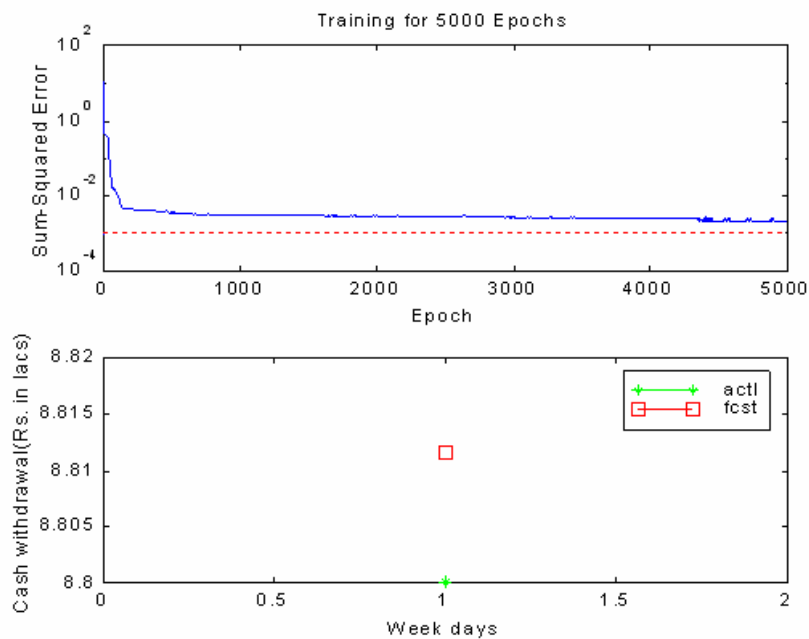


Figure 6: Actual withdrawal for 04/06/2004 = Rs. 8.8 (in lacs), Predicted withdrawal for this day= Rs. 8.81(in lacs)

Day	Actual Withdrawal (Rs in Lacs)	Test Set	Predicted Withdrawal (Rs in Lacs)	Error %	Average Accuracy percent
16-06-04	5.69	1	5.6979	0.138178	98.50
		2	5.7076	0.309794	
		3	5.6344	0.977613	
		4	5.7816	1.60979	
		5	5.4362	4.46014	
25-06-04	6.03	1	6.4241	6.53581	94.28
		2	6.3764	5.74439	
		3	6.3926	6.01293	
		4	6.3998	6.13299	
		5	6.2813	4.16681	

Table 11: Testing for Validation of the Daily Model (Five forecast test sets taken for each date)

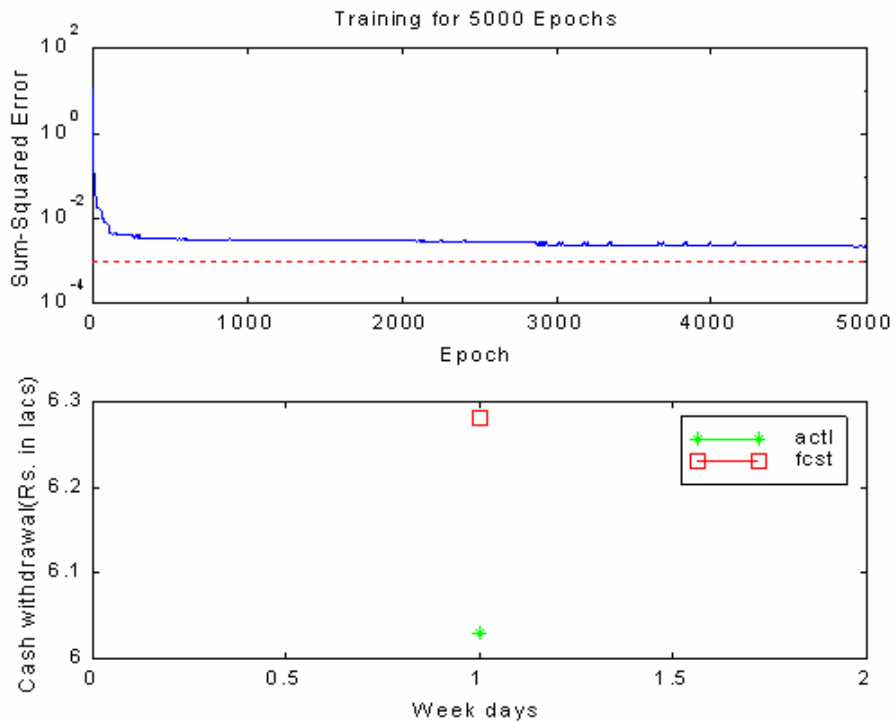


Figure 7: Actual withdrawal for 25/06/2004 = Rs. 6.03, Predicted withdrawal for this day= Rs. 6.28

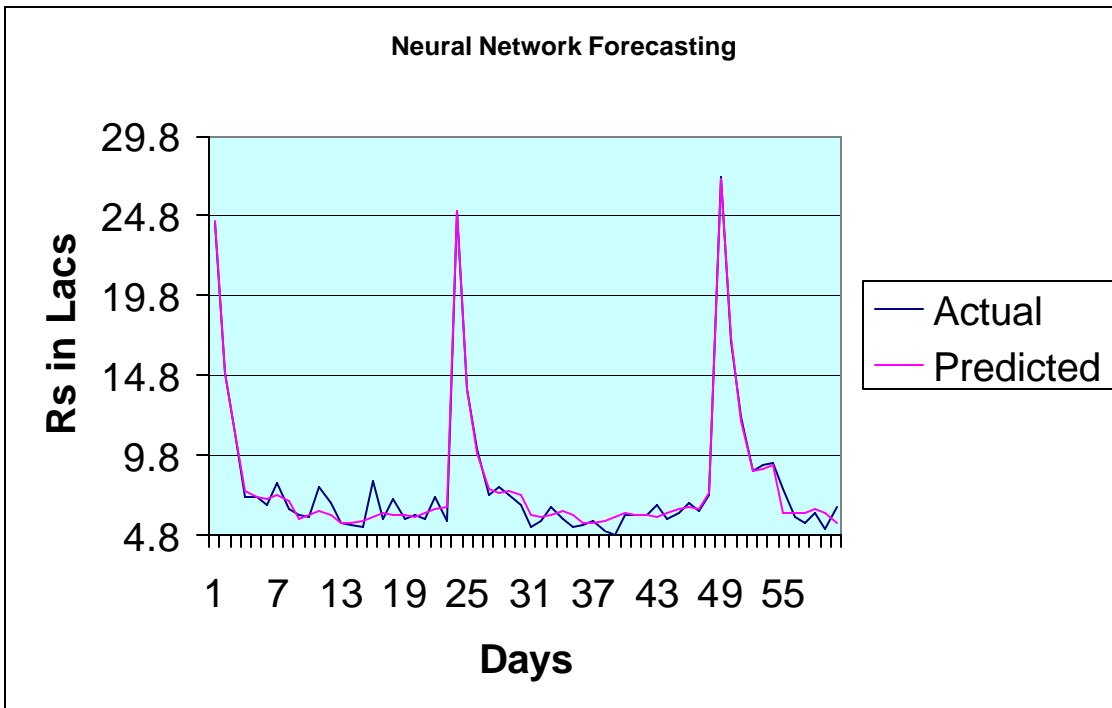


Figure 8: Graph Showing the Actual and Predicted Withdrawal by Proposed Artificial Neural Network Method

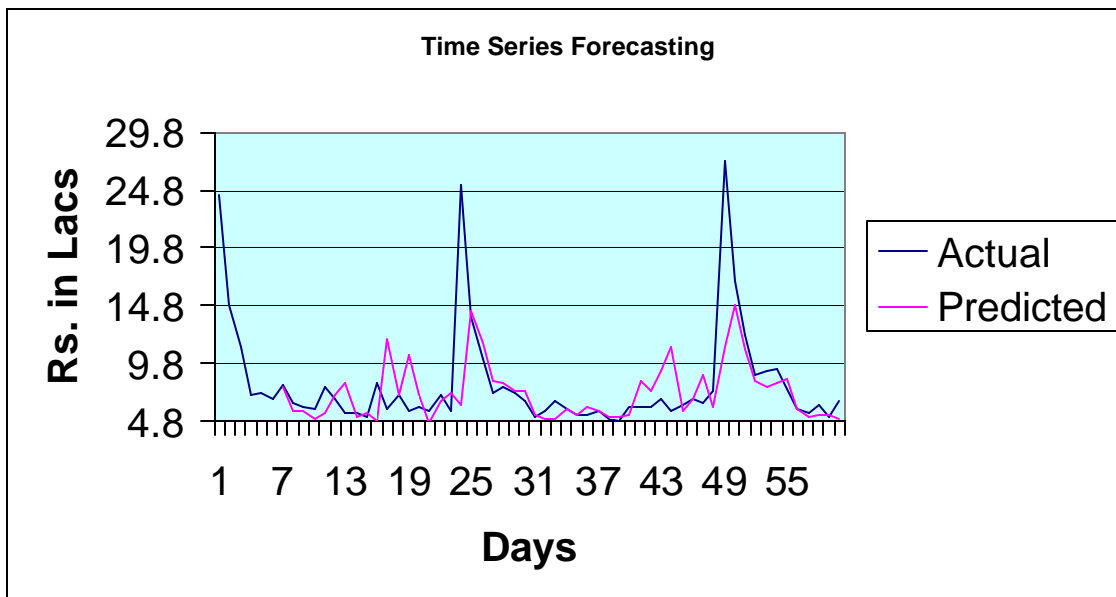


Figure 9: Graph Showing the Actual and Predicted Withdrawal by Time Series

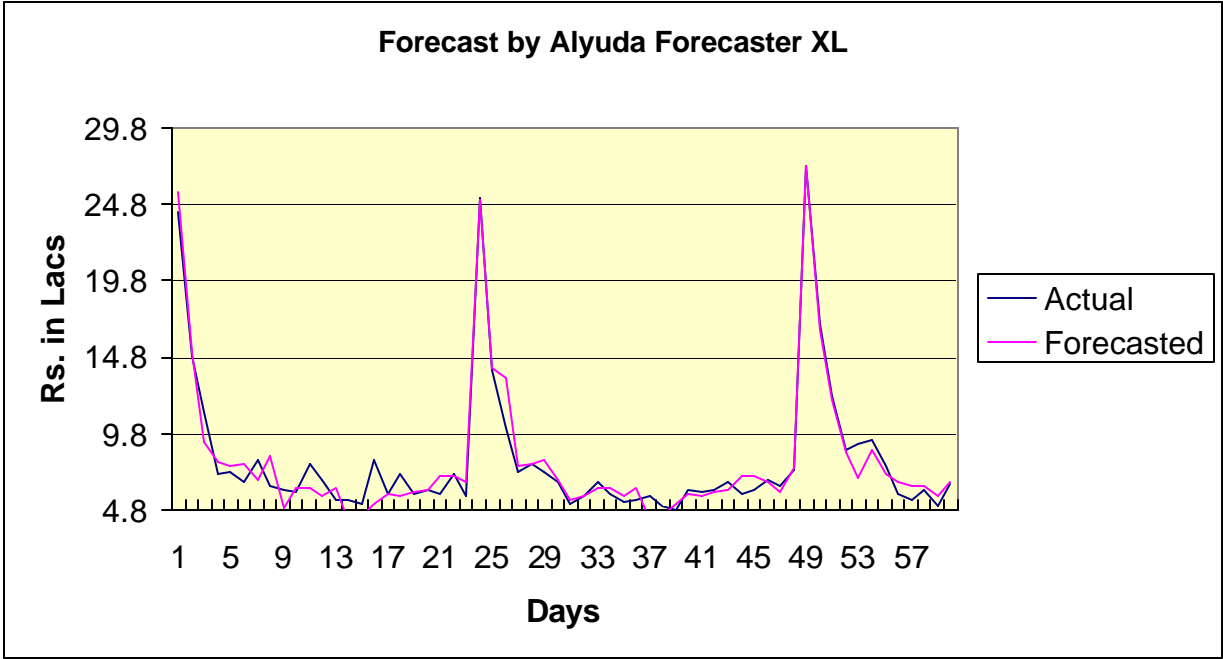


Figure 10: Graph Showing the Actual and Predicted Withdrawal by Alyuda Forecaster XL

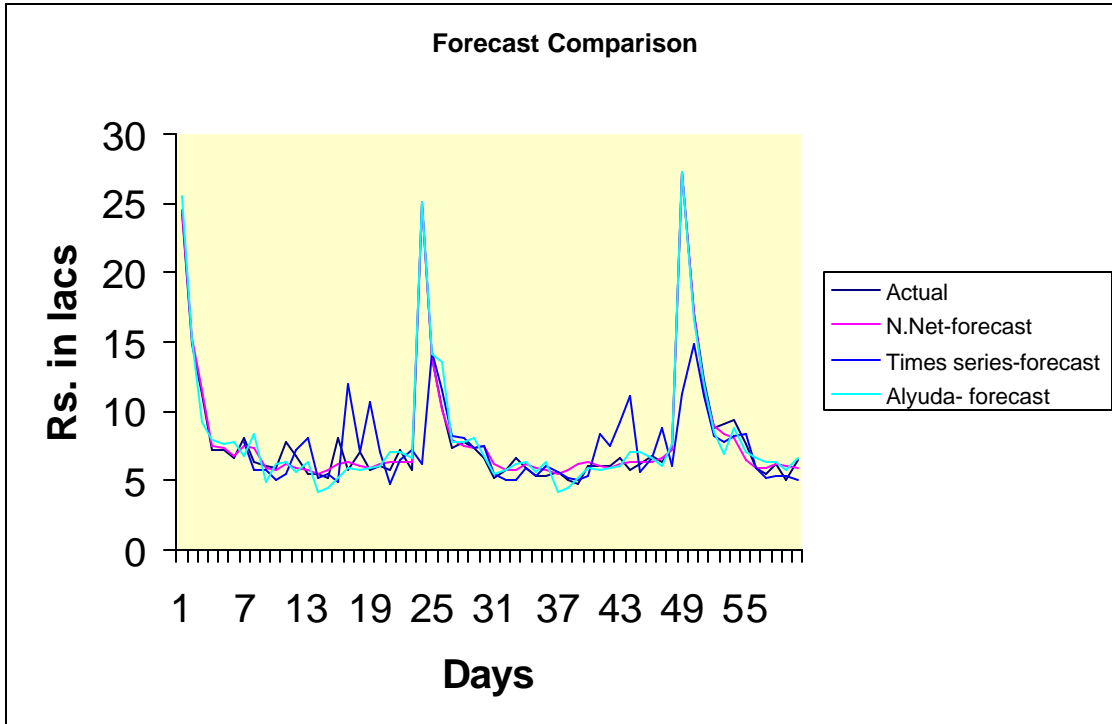


Figure 11: Comparison of Predictions by Different Forecasting Methods

Date	INPUT				Target	Neural Net		Time Series		Alyuda Forecaster	
	Working Day	Week Day	Weekend Effect	Salary Effect		Forecast	% Error	Forecast	% Error	Forecast	% Error
02-04-04	1	5	1	3	24.34	24.49	-0.6273624			25.60	5.16
03-04-04	2	6	1	2	14.94	15.28	-2.2429719			15.10	1.09
05-04-04	3	1	1	1	11.21	11.49	-2.514719			9.24	17.59
06-04-04	4	2	0	0	7.18	7.56	-5.318663			7.93	10.44
07-04-04	5	3	0	0	7.24	7.33	-1.1957182			7.71	6.46
08-04-04	6	4	0	0	6.7	6.86	-2.4567164			7.79	16.31
10-04-04	7	6	1	0	8.04	7.55	6.07301	7.80	3.0125217	6.83	15.02
12-04-04	8	1	1	0	6.39	7.34	-14.853365	5.80	9.2957155	8.31	30.01
13-04-04	9	2	0	0	6.07	5.95	1.8960461	5.79	4.6015421	4.89	19.48
15-04-04	10	4	0	0	5.93	5.82	1.8715008	5.03	15.245245	6.23	5.00
16-04-04	11	5	0	0	7.82	6.16	21.212916	5.53	29.268161	6.29	19.58
17-04-04	12	6	1	0	6.82	5.95	12.765543	7.16	-4.9630357	5.69	16.51
19-04-04	13	1	1	0	5.51	5.72	-3.8150635	8.11	-47.165043	6.29	14.15
20-04-04	14	2	0	0	5.45	5.53	-1.4488073	5.15	5.4960666	4.16	23.76
21-04-04	15	3	0	0	5.27	5.82	-10.51556	5.48	-3.95538	4.46	15.34
22-04-04	16	4	0	0	8.13	6.15	24.331734	4.92	39.506735	5.18	36.26
23-04-04	17	5	0	0	5.84	6.33	-8.3469178	11.96	-104.87926	5.87	0.47
24-04-04	18	6	1	0	7.1	6.07	14.569296	7.06	0.5424021	5.78	18.58
26-04-04	19	1	1	0	5.8	5.95	-2.6282759	10.62	-83.030121	5.96	2.71
27-04-04	20	2	0	0	6.11	6.18	-1.2252046	7.14	-16.899657	6.05	0.94
28-04-04	21	3	0	0	5.8	6.33	-9.1825862	4.73	18.3673	7.01	20.92
29-04-04	22	4	0	0	7.15	6.42	10.185455	6.52	8.8327036	7.06	1.29
30-04-04	23	5	0	0	5.74	6.39	-11.273868	7.27	-26.635737	6.68	16.32
01-05-04	1	6	1	3	25.16	25.00	0.6291733	6.22	75.295489	25.08	0.31
03-05-04	2	1	1	2	13.91	13.87	0.2544932	14.29	-2.6982799	14.09	1.28
04-05-04	3	2	0	1	10.21	10.14	0.6993144	11.55	-13.168476	13.50	32.26
05-05-04	4	3	0	0	7.32	7.98	-9.0394809	8.23	-12.423015	7.75	5.83
06-05-04	5	4	0	0	7.85	7.56	3.6779618	8.05	-2.5805856	7.84	0.06
07-05-04	6	5	0	0	7.35	7.35	-0.0631293	7.39	-0.533642	8.03	9.28
08-05-04	7	6	1	0	6.64	7.55	-13.730873	7.47	-12.538521	6.83	2.89
11-05-04	8	2	0	0	5.27	6.23	-18.206831	5.43	-3.0185347	5.51	4.56
12-05-04	9	3	0	0	5.72	5.74	-0.401049	5.10	10.788906	5.77	0.83
13-05-04	10	4	0	0	6.6	5.82	11.83303	5.05	23.476974	6.23	5.66
14-05-04	11	5	0	0	5.86	6.16	-5.1390785	5.94	-1.422345	6.29	7.32
15-05-04	12	6	1	0	5.29	5.95	-12.464839	5.41	-2.1821477	5.69	7.63
17-05-04	13	1	1	0	5.41	5.72	-5.7340111	6.10	-12.719516	6.29	16.26
18-05-04	14	2	0	0	5.68	5.53	2.6591549	5.67	0.2258637	4.16	26.84
19-05-04	15	3	0	0	5	5.82	-16.4834	5.27	-5.3291678	4.46	10.77
20-05-04	16	4	0	0	4.8	6.15	-28.163125	5.13	-6.8604949	5.18	7.96
21-05-04	17	5	0	0	6.05	6.33	-4.5861157	5.33	11.882948	5.87	3.01
22-05-04	18	6	1	0	6.03	6.07	-0.5900498	8.36	-38.617537	5.78	4.13
24-05-04	19	1	1	0	6.07	5.95	1.9367381	7.46	-22.979299	5.96	1.86
25-05-04	20	2	0	0	6.67	6.18	7.2734633	9.22	-38.185537	6.05	9.26
26-05-04	21	3	0	0	5.8	6.33	-9.1825862	11.17	-92.605431	7.01	20.92
27-05-04	22	4	0	0	6.15	6.42	-4.4185366	5.68	7.5666777	7.06	14.76
28-05-04	23	5	0	0	6.8	6.39	6.0717647	6.68	1.7419884	6.68	1.81
29-05-04	24	6	1	0	6.37	6.67	-4.6824176	8.85	-39.009857	6.04	5.18
31-05-04	25	1	1	0	7.37	7.24	1.768521	5.99	18.699597	7.61	3.27
01-06-04	1	2	0	3	27.22	27.21	0.0279206	11.21	58.80994	27.30	0.29
02-06-04	2	3	0	2	16.98	17.10	-0.729682	14.80	12.86626	16.58	2.34
03-06-04	3	4	0	1	12.21	11.51	5.7461097	11.14	8.7809188	11.96	2.05
04-06-04	4	5	0	0	8.8	9.00	-2.3069318	8.29	5.8390007	8.53	3.04
05-06-04	5	6	1	0	9.16	8.43	8.0050218	7.72	15.761046	6.96	24.05
07-06-04	6	1	1	0	9.37	8.15	13.017503	8.21	12.419273	8.78	6.32
08-06-04	7	2	0	0	7.64	6.53	14.531414	8.39	-9.8812276	7.13	6.65
09-06-04	8	3	0	0	5.9	5.98	-1.2781356	5.86	0.7348474	6.66	12.81
10-06-04	9	4	0	0	5.5	5.89	-7.066	5.25	4.5096828	6.32	14.93
11-06-04	10	5	0	0	6.18	6.20	-0.3108414	5.31	14.015922	6.32	2.33
12-06-04	11	6	1	0	5.13	6.10	-18.953801	5.36	-4.4006403	5.78	12.73
14-06-04	12	1	1	0	6.5	5.93	8.7995385	4.98	23.425887	6.61	1.64
Mean Forecast Error ->							6.5669724		19.43874		10.10915

Table 12: Comparison with Other Existing Methods

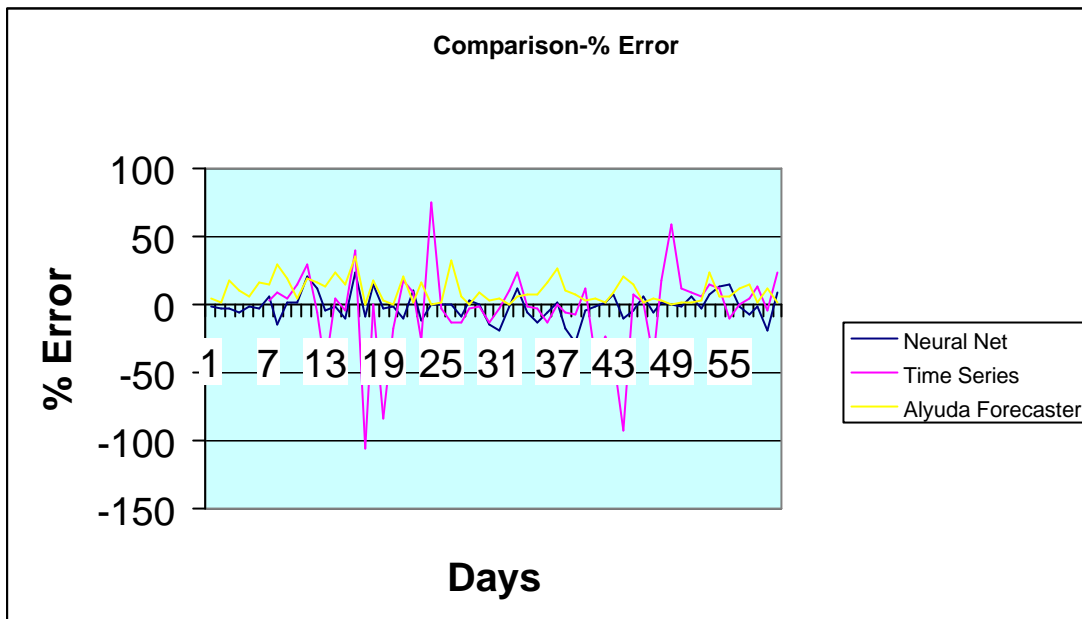


Figure 12: Comparison of Error in Prediction by Different Forecasting Methods

4. Conclusion and Future Scope

Neural Network can learn to approximate any function just by using example data that is representative of the desired task. They are model free estimators, which are capable of solving complex problem based on the presentation of a large number of training data. Neural Networks estimate a function without mathematical description of how the outputs functionally depend on the inputs. They represent a good approach that is potentially robust and fault tolerant.

In this work a cash forecasting system for a bank branch was implemented in MATLAB. The system performs better than other systems based on time series. Its performance was also better than one of the available Excel Add-in for forecasting “Alyuda Forecaster XL 2.3”. This system can be scaled for all branches of a bank in an area by incorporating historical data from these branches. Such a system will help the bank for proper and efficient cash management.

Only those effects that can be quantified and hence are possible candidates to be used as inputs for neural net training and forecasting have been incorporated as influencing variables. However, there are other factors like ‘Event in City’, weather effects, festival season and market behaviour - that definitely influence cash demand and if these are quantified and included as influencing variables, the result will improve with lesser error.

Further, since the Neural Networks simply interpolate among the training data, it will give high errors with the test data that is not close enough to any one of the training data. This accuracy can further be improved if we take more qualitative data as input, which is large enough to incorporate all the effects which can be quantified.

As cash withdrawal from a bank branch has not been dependent only on withdrawal on counter because of other delivery channels like ATMs, Internet Banking, Debit cards etc., the forecasting of cash will need to adapt to handle the new behaviour patterns. With latest concepts like Fuzzy systems and Genetic algorithms, the forecasting will become more accurate and reliable.

5. References

- [1] Philip. D. Wasserman, ANZA Research Inc., “Neural Computing theory and practice” Van Nostrand Reinhold, New York, 1989
- [2] Kosko Bart, “*Neural Network and Fuzzy systems: A Dynamical Systems Approach to Machine Intelligence*”, Prentice Hall of India, 1997.
- [3] Iebling Kaastra and Milton Boyd, “*Designing a neural network for forecasting financial and economic time series*”, Neurocomputing, (Elsevier) 10 (1996), pp 215-236.
- [4] Ray Pranik and Vani Vina, “*What moves Indian Stock Market: A study on the Linkage with Real Economy in the Post-Reform era*” Proceedings of the 6th International Conference of Asia-Pacific Operation Research Societies, New Delhi, 2003.
- [5] Haykin Simon, “*Neural Networks*”, Macmillan College Publishing Co. Inc., 1994.
- [6] Y. Yoon and G. Swales, “*Predicting stock price performance: A neural network approach*”, Proceedings of 24th Annual International Conference of System Sciences, Chicago, 1991.
- [7] Erkki Oja, Lecture material for the course “*Principles of Neural Computing*”, held at Helsinki University of technology, 1997.
- [8] Jason E. Kutsurelis, “*Forecasting Financial Markets using Neural Networks: An analysis of Methods and Accuracy*”, MS Thesis (1998), US Naval Postgraduate School.
- [9] Andrew Blais and David Mertz, “*An introduction to Neural Networks*”, IBM Developers, <http://www-106.ibm.com/developerworks/library/l-neural/>
- [10] Pauli Murto, “*Neural network models for short term load forecasting*”, Masters Thesis, Helsinki University of Technology, Department. of Engineering Physics and Mathematics, 1998.
- [11] Dave Anderson and George Mc Neil, “*Artificial neural networks technology*”, Data and Analysis Centre for Software report, 1992.
- [12] Zhanshou Yu “*Feed-Forward neural networks and their applications in forecasting*”, M.Sc. thesis, University of Houston.
- [13] Rudra Pratap “*Getting Started with MATLAB 5, A Quick introduction for Scientists and Engineers*”, PaperBack, 2001.
- [14] “*MATLAB Tutorial*”, <http://www.mathworks.com>
- [15] “*Neural Network Tool Box for use with MATLAB*”, Manual by the Mathworks Inc.