Research Journal of Applied Sciences, Engineering and Technology 11(2): 221-228, 2015
DOI: 10.19026/rjaset.11.1710
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: May 16, 2015 Accepted: June 22, 2015 Published: September 15, 2015

Research Article

Optimizing Cash Management Model Using Computational Intelligence

[1, 2]A. Alli and [1]M.M. Ramya
[1]Department of Computer Applications, Hindustan University, Chennai, Tamilnadu,
[2]Department of Computer Applications, Presidency College, Bangalore, Karnataka, India

Abstract: In today's technical era, the financial organizations have great challenges to optimize the cash management process. Maintaining minimum cash leads to customer frustration. At the same time, upholding excess cash is a loss to the organization. Hence, soft computing based cash management solutions are required to maintain optimal cash balance. An Artificial Neural Network (ANN) is one such technique which plays a vital role in the fields of cognitive science and engineering. In this study, a novel ANN-based cash Forecasting Model (ANNCFM) has been proposed to identify the cash requirement on daily, weekly and monthly basis. The six cash requirement parameters: Reference Year (RY), Month of the Year (MOY), Working Day of the Month (WDOM), Working Day of the Week (WDOW), Salary Day Effect (SDE) and Holiday Effect (HDE) were fed as input to ANNCFM. Trials were carried out for the selection of ANNCFM network parameters. It was found that number of hidden neurons, learning rate and the momentum when set to 10, 0.3 and 0.95, respectively yielded better results. Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE) were used to evaluate the performance of the proposed model. MSE that was less than 0.01 proves the capability of the proposed ANNCFM in estimating the cash requirement.

Keywords: ANN, ANNCFM, back-propagation, cash forecasting

INTRODUCTION

Forecasting cash demand needs to be more accurate for any financial organization, including banks (Fraydoon *et al*., 2010; Kumar and Walia, 2006; Alli *et al*., 2013). If the forecast is flawed, in addition to making financial losses to the banks, it results in customer dissatisfaction. In the banking industry, an earlier cash requirement study was made using a feed forward neural network with back propagation for short term data of two months (Kumar and Walia, 2006). Subsequently, another comparative study was made in the cash anticipation using a classic time series model and artificial neural networks (Fraydoon *et al*., 2010). The daily cash requirement models for a bank were optimized with particle swarm and compared with the least square method for short term data (Alli *et al*., 2013). The main objective of the paper is to design, develop and test a unique supervised method to forecast the cash requirement for banks from their historical data.

ANN background: ANN is an efficient tool in understanding the complexities of real world problems in all fields of our daily life (Haykin, 1994). It is used as a function optimizer for linear as well as nonlinear problems in science, engineering, technology, management and finance (Nakamura, 2006; Refenes and White, 1998; Aminian *et al*., 2006; Hanna *et al*., 2007; Gorr *et al*., 1994). Artificial neural network learning methods provides the best approach for approximating discrete, real and vector valued target functions (Zhang *et al*., 1998; Geem and Roper, 2009; Yokoyama *et al*., 2009), for complex problems, which are not possible to solve by conventional mathematical methods like analytical and numerical technique. ANN are applied in the forex market prediction, portfolio optimization, decision making, meteorological parameters forecasting (Bishop, 1999; Taubenböck *et al*., 2010; Tenti, 1996; Leigh *et al*., 2005; Manfred and Hans-Georg, 1997; Mabel and Fernandez, 2008; Sharda and Delen, 2006) etc..

The various ANN based approaches applied by researchers in finance field as an alternative to the traditional time series model include Financial and economic forecasting, credit authorization screening, simulation of market behavior, mortgage risk assessment, risk rate of investments and detection of regularities in security price movements (Tenti, 1996; Leigh *et al*., 2005; Manfred and Hans-Georg, 1997; Mabel and Fernandez, 2008; Sharda and Delen, 2006).

Design of proposed ANNCFM architecture: The process of designing a neural network in many fields resulted in a satisfactory performance, but building a

Corresponding Author: A. Alli, Department of Computer Applications, Hindustan University, Chennai, Tamilnadu, India

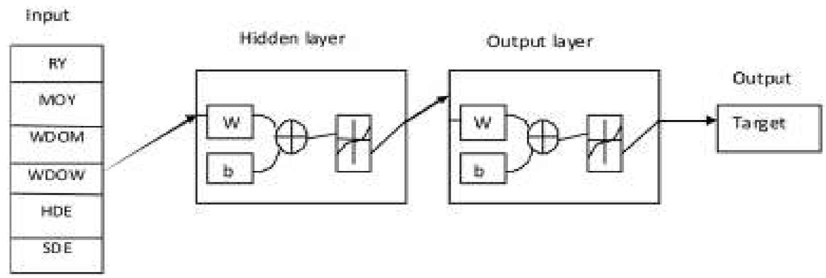

Fig. 1: Architecture of ANNCFM model

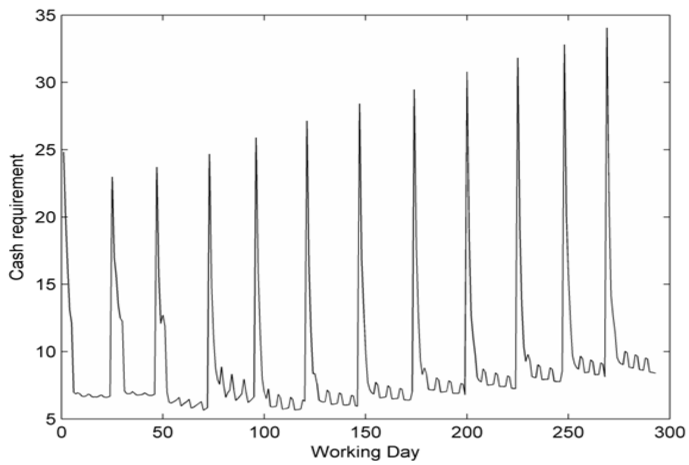

Fig. 2: Typical cash requirement ($\times 10^6$) for a year

neural network forecast for a particular problem is nontrivial task. The modeling issues that affect the performance of the neural network must be selected carefully.

Selection of ANN parameters: In general, the design of multilayer ANN can have many layers where a layer represents a set of distributed parallel processing nodes. The three layered ANN network with one input, one output and one intermediate hidden layer is sufficient to approximate any complex non-linear function. In the case of forecasting studies, many experimental results also confirm ANN with one hidden layer is enough to predict the required data (Refenes and White, 1998; Aminian *et al*., 2006; Hanna *et al*., 2007). The model architecture of ANNCFM is shown in the Fig. 1.

The important critical decision is to determine the architecture is:

- Number of layers
- Number of neurons in each layer
- Number of arcs which interconnect with nodes
- Activation function of hidden and output nodes
- Training algorithm
- Data transformation or normalization
- Training and test sets
- Performance measures

Design of proposed ANN models: The proposed ANNCFM model consists of one input, one hidden and an output layer as discussed in above section. In this study the data were collected from a semi-urban area bank located in India. The typical daily cash requirement for the bank for one year is shown in Fig. 2.

The collected data was for a period of three years (2010 to 2012) and was used for training and testing with the following input parameters:

- RY- Reference year: Ranges between 1 to 3 as three years
- MOY-Month of the year: Ranges from 1to 12,
- WDOM-Working day of the month: Ranging from 1 to 27
- WDOW-Working day of the week: Ranging from 1 to 6
- SDE- Salary day effect: Ranging from 1 to 3
- HDE- Holiday and the weekend effect: Either 0 or 1

The fore mentioned parameters were used as six input neurons. In the hidden layer, the number of neurons varied from 8 to 50. The output layer had one neuron that corresponds to the optimal cash requirement for a day.

Pseudocode- ANNCFM:

Main()
{
[W, V, $V_{oj,}$ W_{ok}] = ANNCFMtrain (x, nip, nh, op, α, μ, t)
y_k = ANNCFMtest(ts, W,V, $V_{oj,}W_{ok,}$t)
[Mserr,Mape] = ANNCFMevaluate()
}
FunctionANNCFMtrain (x, nip, nh, op, α, μ, t) returns network with modified weight
{
Repeat
{
For each training sample x(I,nip)
//Feed forward computation
//Determine the output neuron between input layer and hidden layer:

$Net_j = 0$

$$Net_{j=} \sum_{j=1}^{nh} \sum_{j=1}^{nip} + Net_j + V_{i,j} * X_{I,i}$$

$Net_j = Net_j + V_{oj}$

$$Z_j = \frac{1}{1 + e^{-Net_j}}$$

//Determine the output neuron between hidden layer and output layer:

$Net_k = 0$

$$Net_j = \sum_{k=1}^{op} \sum_{j=1}^{nh} +Net_j + V_{i,j} * X_{I,i}$$

$Net_k = Net_k + W_{j,k} * Z_j$

$Net_k = Net_k + W_{ok}$

$$y_k = \frac{1}{1 + e^{-Net_j}}$$

$e_k = t_k - y_k$

//Compute the error signal between the output and hidden layer:

$\delta_k = y_k(1 - y_k)(t_k - y_k)$

$$temp = \sum_{k=1}^{op} \sum_{j=1}^{nh} +temp + (\delta_k)w_{j,k}$$

$\Delta_j = Z_j\big(1 - Z_j\big)temp$

//Update the weights between the output (k) and Hidden (j) layer;
If itr = 1 then
{
$temp = \sum_{k=1}^{op} \sum_{j=1}^{nh} \propto \delta_k Z_j$
Else

$$Cw = w_{j,k} - Tw_{j,k}$$

$temp = \propto \delta_k Z_j + \mu\, Cw$
End if

$$Tw_{j,k} = w_{j,k}$$
$$w_{j,k} = w_{j,k} + temp$$

}

//Update bias between the output and hidden
If itr =1 then
{

$$temp = \propto \delta_k$$

Else

$$Cw = w_{ok} - Tw_{old}$$
$$temp = \propto \delta_k + \mu\, Cw$$

End if

$$Tw_{old} = w_{ok}$$
$$w_{ok} = w_{ok} + temp$$

}

//Update the weights between the input (i) and Hidden (j) layer;
If itr = 1 then
{

$$temp = \sum_{i=1}^{nip} \sum_{j=1}^{nh} \propto \delta_j X_i$$

Else

$$CV = V_{i,j} - TV_{i,j}$$
$$temp = \propto \delta_j X_i + \mu\, CV$$

End if

$$TV_{i,j} = V_{i,j}$$
$$V_{i,j} = V_{i,j} + temp$$

}
//Update bias between the hidden and input
If itr =1 then
{

$$temp = \propto \Delta_j$$

Else

$$Cw = V_{oj} - Tw_{old}$$
$$temp = \propto \Delta_j + \mu Cw$$

End if

$$Tw_{old} = V_{oj}$$
$$V_{oj} = V_{oj} + temp$$

}

$$mse = mse + (e_k)^2$$

}
Until mse<0.0001 or itr<6000
}

Function ANNCFMtest (ts, W,V, $V_{oj,}W_{ok,}$t) returns output (y)
{
For each testsample ts
//Feed forward computation
//Determine the output neuron between input layer and hidden layer

$$Net_j = 0$$

$$\mathrm{Net}_{j=}\sum_{j=1}^{nh}\sum_{j=1}^{nip} + \mathrm{Net}_j + V_{i,j} * X_{I,i}$$
$$\mathrm{Net}_j = \mathrm{Net}_j + V_{oj}$$
$$Z_j = \frac{1}{1+e^{-\mathrm{Net}_j}}$$

//Determine the output neuron between hidden layer and output layer:

$$\mathrm{Net}_k = 0$$
$$\mathrm{Net}_j = \sum_{k=1}^{op}\sum_{j=1}^{nh} + \mathrm{Net}_j + V_{i,j} * X_{I,i}$$

$$\mathrm{Net}_k = \mathrm{Net}_k + W_{j,k} * Z_j$$
$$\mathrm{Net}_k = \mathrm{Net}_k + W_{ok}$$
$$y_k = \frac{1}{1+e^{-\mathrm{Net}_j}}$$

}
ANNCFM evaluate (t_k, y_k, ts)
{

$$\mathrm{Mserr} = \sum_{k=1}^{ts} + \mathrm{mserr} + (t_k - y_k)^2$$

$$\mathrm{mape} = \mathrm{abs}(\frac{(t_k - y_k)}{t_k})$$

$$\mathrm{Mserr} = \frac{\mathrm{mserr}}{ts}$$
$$\mathrm{Mape} = \frac{\mathrm{mape}}{ts} * 100$$

}

Evaluation metrics: In order to derive and evaluate the performance of the most appropriate model that fulfills our objective of optimizing the cash management, few metrics were used. The accuracy of the proposed ANNCFM is evaluated using MAPE and MSE which are defined as follows:

$$\mathrm{MAPE} = \frac{\Sigma\left|\frac{X_t - F_t}{X_t}\right|}{n}(100) = \frac{\Sigma\left|\frac{e_t}{X_t}\right|}{n}(100) \quad (1)$$

$$\mathrm{MSE} = \frac{\Sigma|X_t - F_t|^2}{n} \quad (2)$$

where,
X_t = The actual data at period t
F_t = The forecast at period, t
e_t = The forecast error at period t
n = The number of observations

Table 1: ANNCFM performance for different number of hidden neurons

No. of hidden neurons	MSE (before parameter selection)	MSE-(after parameter selection)
8	0.0196	0.0086
9	0.0188	0.0085
10	0.0181	0.0079
11	0.0226	0.0099
12	0.0221	0.0101
13	0.0169	0.0085
15	0.0157	0.0099
20	0.0181	0.0079
25	0.0175	0.0084
30	0.0187	0.0080
35	0.0184	0.0088
40	0.0181	0.0079
45	0.0182	0.0079
50	0.0181	0.0079

RESULTS AND DISCUSSION

The data for a period of three years (2010-2012) was collected from a particular branch of City Union Bank to simulate the network using MATLAB. For the proposed study the total number of data for the three years is 879, in which the first two and half years, 737 data were used for training (80%) and the remaining six months 142 data sets (20%) were used for testing. Studies found that input data normalization with certain criteria, prior to the training process, is crucial to obtain good results, as well as to fasten significantly the calculations (Sola and Sevilla, 1997). Hence the input data was normalized before training.

In ANNCFM, 15 runs were made by varying the number of hidden neurons from 10 to 50 using gradient descent with momentum back-propagation (traingdm) for the default training parameters learning rate = 0.01, momentum = 0.95, Goal = 0 and number of iterations as 6000, are illustrated in Table 1-column2.

The convergence of ANNCFM is influenced by number of hidden neurons in which by varying the number of hidden neurons between 10 through 50. The error was minimal when the number of hidden neurons was set to 10, 20, 40, 45 and 50, by achieving an MSE of 0.0079 as observed from column 3 of Table 1. As the number of hidden neurons increase, there is a significant increase in the computational time. Hence the number of hidden neurons in the proposed study was fixed at 10. The pictorial representation for the optimal hidden neuron against its MSE are shown in Fig. 3.

The learning rate 'lr' arrives at a local optimum for the higher learning rate and global optimum for slow learning process. Different trials were made to identify the optimal learning rate to avoid the unstable condition and fluctuations in the results. The learning rate varied between 0.1 through 0.5 in which 0.3 yielded an optimal learning rate for the given data set, as shown in Fig. 4.

The momentum plays a vital role in identifying the convergence point. Momentum, when set too low, it

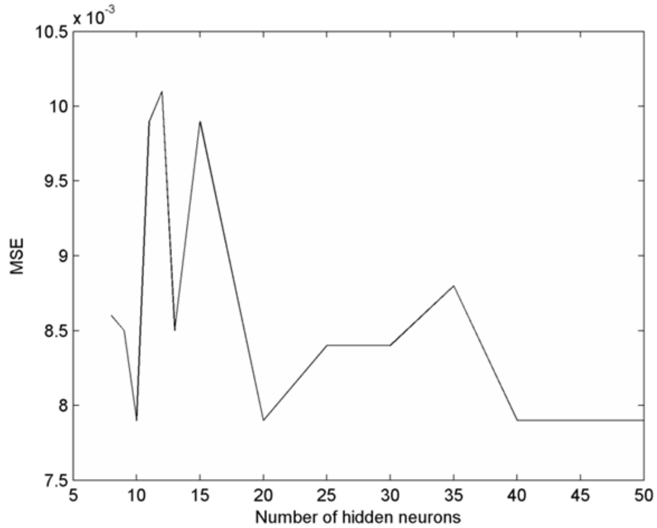

Fig. 3: Optimal number of hidden neurons

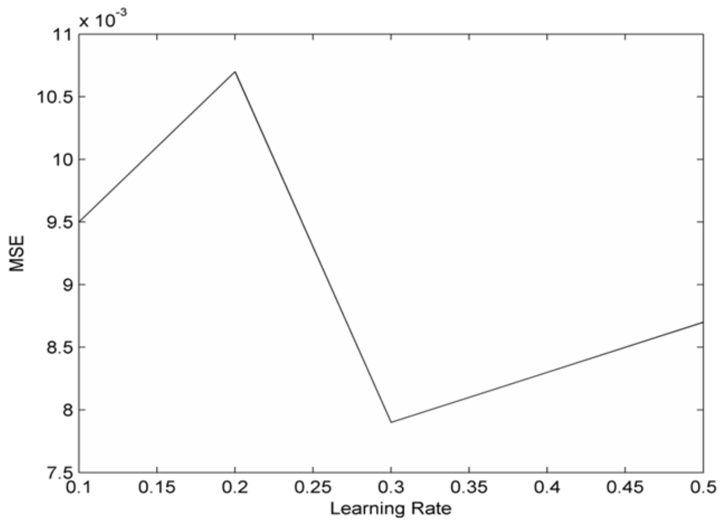

Fig. 4: Optimal learning rate

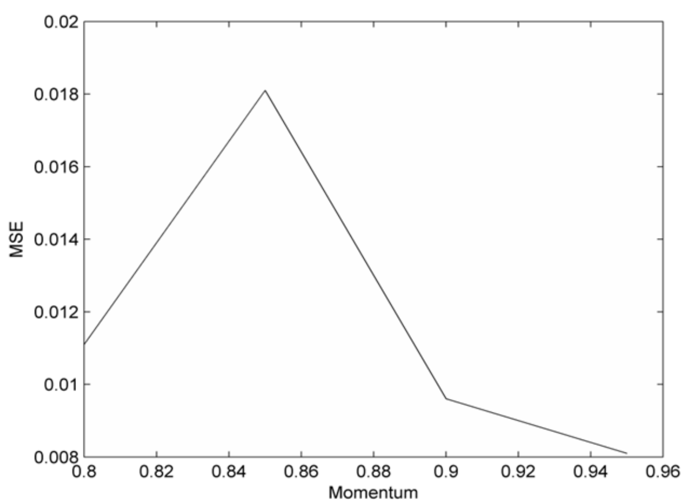

Fig. 5: Optimal momentum rate

may get stuck in local minima and if it is too high, the network will become unstable. So there is a need to identify the optimal momentum for ANNCFM, various momentum values were tested between 0.8 and 1.0, the trained results shows that the optimal momentum value was 0.95 are shown in the Fig. 5.

In the ANNCFM model to train and test the cash requirement for a day, week, month the following parameter values are selected based on their performance from the different number of runs made above:

- The number of input neurons = 6
- Maximum number of iteration = 6000
- Learning rate = 0.3
- Momentum = 0.95
- Transfer function = Sigmoidal (hidden and output layer)

The optimal selection of the above parameters helped in improving the performance, by minimizing the error rate. This is evident from Table 1, that shows the MSE achieved before and after parameter selection.

The ANNCFM was used to estimate daily, weekly and monthly cash requirement. The estimated values were compared with the actual values for the testing period are shown in Fig. 6 to 8, for the daily, weekly and monthly prediction. The obtained results show the ANNCFM was found to perform reasonably well for all the three models. The weights calculated by our ANNCFM were found to be sufficient for cash prediction in which RY, MOY, WDOM, WDOW are essential parameters and SDE, HDE is additional parameters. The connection weight approach was used to quantify the importance of input variable (Julian *et al*., 2004). The preference of the input parameters

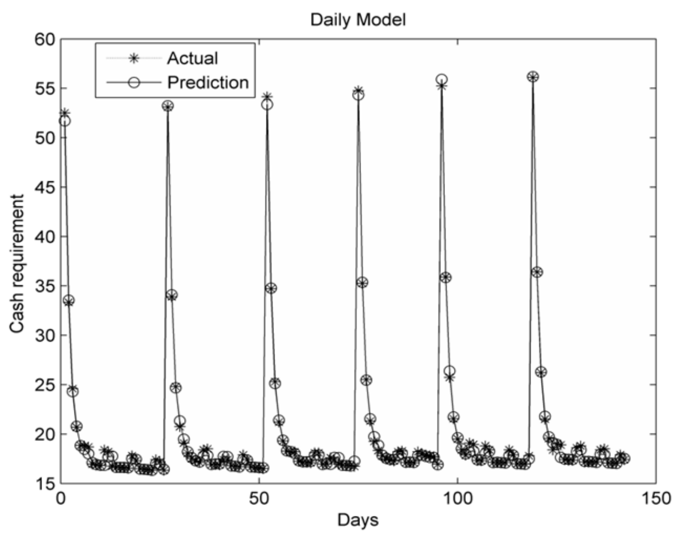

Fig. 6: Daily model

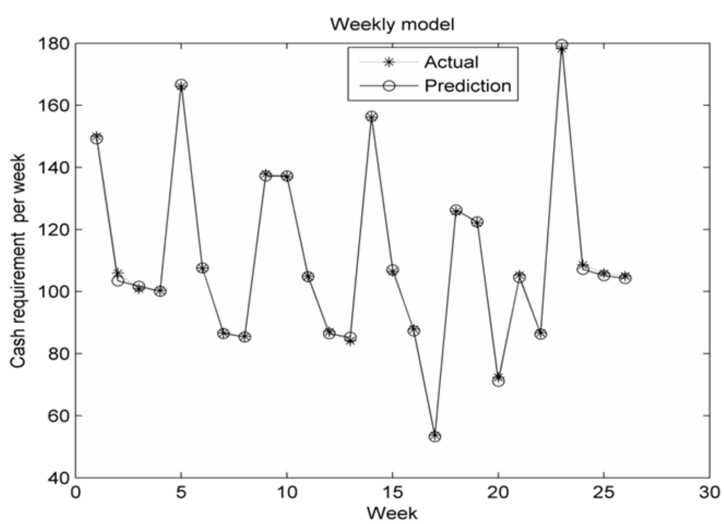

Fig. 7: Weekly model

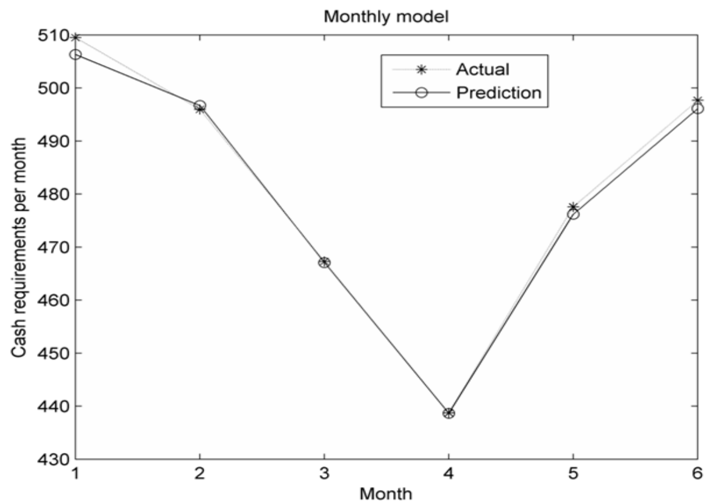

Fig. 8: Monthly model

Table 2: ANNCFM weights-preferences

S.NO	Input parameters	Weights
1	RY	0.542063
2	MOY	0.118675
3	WDOM	-3.75447
4	WDOW	0.034225
5	SDE	1.243722
6	HDE	0.017973

Table 3: MAPE and MSE errors for ANNCFM

Model	Average MAPE	Average MSE
Daily	8.2400	0.1206
Weekly	8.4900	0.7312
Monthly	9.1400	2.5017

was founded based on the weights obtained was evident from Table 2, column-3.

The input parameters SDE and HDE play a vital role in the daily and weekly model as it was observed from the above table it effectively takes care the need of peak cash requirement at the beginning of every month and during holiday periods. The role of SDE in the weekly cash prediction could be easily understood for the weeks like 1, 5, 14, where the cash requirement is maximum since the beginning of the month lies within the week. However, for the 9th and 10th as well as for the 18th and 19th week cash requirement shows the new month starts between the weeks. The monthly model was plotted for six months as shown in Fig. 8 in which the experimental results shows that the estimated values were most influenced by WDOM. The cash required and predicted to be minimized for the fourth month in which WDOM was minimum. The MAPE and MSE for ANNCFM are shown in Table 3

The comparison made between the actual and forecast data shown from the figures indicates that the six input variables selected in our model are sufficient to identify the cash need which is changing from time to time.

CONCLUSION

The observations of the experimental results of this study show that ANNCFM is a useful tool to predict the cash requirement in emerging banking sector. ANNCFM using feed forward neural network training with back-propagation algorithm optimize the needs of cash on a daily, weekly and monthly basis. During the implementation process the data set used for the years between 2010 and 2012 were trained and tested to measure the performance. The input parameters were initialized and different runs were made to the proposed model to find out the optimal number of hidden neurons as 10, momentum as 0.95 and learning rate as 0.3 to train and test the network using a sigmoid transfer function. The estimated results were with minimal error for better performance with an accuracy of 91.23%.

REFERENCES

Alli, A., M. Ramya, M. Srinivasa and V. Kumar, 2013. Cash management using particle swarm optimization. Proceeding of the International Conference on Data Mining and Soft Computing. SASTRA University, Thanjavur, India.

Aminian, F., E.D. Suarez, M. Aminian and D.T. Walz, 2006. Forecasting economic data with neural networks. Comput. Econ., 28: 71-88.

Bishop, C.B., 1999. Neural Networks for Pattern Recognition. Oxford University Press, New York.

Fraydoon, R.R., H. Farshad and M. Seyed Reza, 2010. A comparison of classic time series models and artificial neural networks in anticipation of cash requirements of banks: A case study in Iran. Proceeding of the International Conference on Academic and Business Research Institute, Orlando, USA.

Geem, Z.W. and W.E. Roper, 2009. Energy demand estimation of South Korea using artificial neural network. Energ. Policy, 37(10): 4049-4054.

Gorr, W., D. Nagin and J. Szczypula, 1994. Comparative study of artificial neural network and statistical models for predicting student point averages. Int. J. Forecasting, 10(1): 17-34.

Hanna, A.M., D. Ural and G. Saygili, 2007. Evaluation of liquefaction potential of soil deposits using artificial neural networks. Eng. Computation, 24: 5-16.

Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Co., Inc., New York.

Julian, D.O., K.J. Michael and G.D. Russell, 2004. An accurate comparison of methods for quantifying variable importance in artificial neural network using simulated data. Ecol. Modell., 178(3-4): 389-397.

Kumar, P.C. and E. Walia, 2006. Cash forecasting: An application of artificial neural networks in finance. Int. J. Comput. Sci. Appl., 3: 61-77.

Leigh, W., R. Hightower and N. Modani, 2005. Forecasting the New York stock exchange composite index with past price and interest rate on condition of volume spike. Expert Syst. Appl., 28(1): 1-8.

Mabel, M.C. and E. Fernandez, 2008. Analysis of wind power generation and prediction using ANN: A case study. Renew. Energ., 33(5): 986-992.

Manfred, S. and W. Hans-Georg, 1997. Portfolio optimization with a neural network implementation of the coherent market hypothesis. Eur. J. Oper. Res., 100(1): 27-40.

Nakamura, E., 2006. Inflation forecasting using a neural network. Econ. Lett., 86: 373-378.

Refenes, A.P. and H. White, 1998. Neural networks and financial economics. Int. J. Forecasting, 6(17).

Sharda, R. and D. Delen, 2006. Predicting box-office success of motion pictures with neural networks. Expert Syst. Appl., 30(2): 243-254.

Sola, J. and J. Sevilla, 1997. Importance of data normalization for the application of neural networks to complex industrial problems. IEEE T. Nucl. Sci., 44(3): 1464-1468.

Taubenböck, H., T. Esch, M. Wurm, A. Roth and S. Dech, 2010. Object-based feature extraction using high spatial resolution satellite data of urban areas. J. Spat. Sci., 55(1): 117-132.

Tenti, P., 1996. Forecasting foreign exchange rates using recurrent neural networks. Appl. Artif. Intell., 10: 567-581.

Yokoyama, R., T. Wakui and R. Satake, 2009. Prediction of energy demands using neural network with model identification by global optimization. Energ. Convers. Manage., 50(2): 319-327.

Zhang, G., B.E. Patuwo and M.Y. Hu, 1998. Forecasting with artificial neural networks: The state of the art. Int. J. Forecasting, 14: 35-62.