Research Journal of Applied Sciences, Engineering and Technology 11(6): 653-660, 2015
DOI: 10.19026/rjaset.11.2027
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: May 16, 2015 Accepted: June 22, 2015 Published: October 25, 2015

Research Article

Hybrid Swarm Optimization for Cash Management Using Evolutionary Computing

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

Abstract: Cash forecasting plays a vital role in any financial organization to maintain the optimal cash balance to satisfy the customer needs on a daily basis without any delay. In the traditional approach statistical methods were used for cash forecasting. Banks have great challenges to avoid the surplus cash as well as to keep adequate cash to meet the customer demand. An intelligent model is needed to identify the cash requirement using cognitive approach. Hence, an evolutionary computing using the hybrid swarm system was introduced for the cash management of a bank. In this study, cash prediction models were developed from the historic short term data and long term data. In order to find the daily cash requirement of financial organization an intelligent hybrid model composed of an Artificial Neural Network (ANN) and a Particle Swarm Optimization (PSO) was introduced. The proposed methodology was capable of training and adjusting the ANN parameters through PSO to improve the efficiency of the cash management model. In a PSO-based ANN model, PSO searches for a set of best weights and biases for an ANN to minimize the error were evaluated using Mean Square Error (MSE). The experimental analysis was made for the selected parameters to maintain the optimal cash. The proposed ANNPSO model has proven its accuracy with the best MSE of short term data was 0.0035 and for long term data was found at 0.0029.

Keywords: ANN, inertia, MSE, neurons, position, PSO, velocity

INTRODUCTION

Forecasting cash demand needs to be more accurate for any financial organization, including banks (Prem and Ekta, 2006). If the forecast is flawed, in addition to making financial losses to the banks, it results in customer dissatisfaction. The traditional approach such as a time-series method, the factor analysis method was used to estimate the cash demand earlier (Prem and Ekta, 2006). The problems associated with the existing approach was the system fails to predict the cash requirement, if there is a sudden change in the regular operations which affects the entire organization. The earlier research was made for financial organization such as ANN based cash forecasting using back propagation for short term data (Fraydoon *et al*., 2010), Comparative analysis was made between least square method and PSO (Alli *et al*., 2013), Optimization of cash management model using computational intelligence using back propagation leads to trap in local minima (Ramya and Alli, 2015) since there is a need to improve the performance of the system by minimizing the error for both long term and short term data a new hybrid approach was introduced in this study to forecast the cash requirement for a bank from the historic data using training feed forward neural network with PSO.

LITERATURE REVIEW

ANN methodology: Neural network mimics the human brain (Gaitonde and Karnik, 2012). It processes the data in a similar way, how the human brain performs the task. The heuristic model is essential for a particular problem to determine the optimal values (Gaitonde and Karnik, 2012). The architectures and training algorithms were proposed and used in various fields like simulations, finance, science, engineering and medical by researchers all over the world (Gaitonde and Karnik, 2012; Kennedy and Eberhart, 1995; Corchado *et al*., 2012).

The performance of ANN can be measured by calculating the correct weight and bias which minimizes the error rate. Feed Forward Networks (FNN) with input, hidden and an output layer which the signal flow in one direction from input to output without feedback loops is one of the most popular practices applied in NN technique (García *et al*., 2010; Corchado *et al*., 2010; Junliang and Xinping, 2008; Omran *et al*., 2006; Hagan and Menhaj, 1994). FNN with different layers are proved to be sufficient for continuous and discrete functions as well as classification linear and nonlinear patterns (Ho *et al*., 1990; Hagan and Menhaj, 1994). Most of the researches used FNN with back-propagation as the learning

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

method (Lee *et al*., 1992; Park *et al*., 1991; Rahman and Bhatnagar, 1988).

The parameter selection plays an important role to determine the optimal values. The different ANN parameters were used in our design to identify the cash requirement for both of iterations. The convergence of the back-propagation algorithm depends on the number of hidden neurons, weights, biases and other training parameters like learning rate, momentum. It has some drawbacks like trapping in local minima and slow convergence. To overcome the above limitations the proposed methodology uses learning based on evolutionary algorithms and their hybrids, including PSO algorithms (Leung *et al*., 2003; Park *et al*., 1991; Rahman and Bhatnagar, 1988). PSO is one of the most efficient and practical optimization algorithms in terms of reducing both of the aforementioned drawbacks.

Introduction to PSO: Particle Swarm Optimization (PSO) was first introduced by Kennedy and Eberhart (1995). The method was discovered through simulation of a simplified social model. Later on, it was developed as a general heuristic exploration technique, which performs effective exploration through memory and feedback. With the imitation of the behavior of bio-community, it enjoys a rapid calculation and a sound global exploration when applied in a large-scale optimization. Like evolutionary algorithms, PSO technique conducts a search using a population of particles, corresponding to individuals. The mathematical notation used in PSO was represented in (1) and (2) as follows:

$$v_i^{t+1} = wv_i^t + c_1 * r1 * (pbest_i - x_i^t) + c_2 * r2 * (gbest_i - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

where,
V_i^{t+1} = Velocity of the particle
x_i^{t+1} = A current position of the particle
c_1, c_2 = The learning factors, generally, $c_1 = c_2 = 2$
w = The weight scale operator
r1, r2 = The random values within the interval of (0, 1)
t = The number of iteration
n = The number of particles
m = The number of dimensions

PROPOSED DESIGN-CASH FORECASTING (CF)-FNN-PSO

A novel heuristic based design proposed for training CF-FNN-PSO for finding a combination of weights and biases which provide the minimum error for an FNN. In our previous study short term cash forecasting using simple PSO (Alli *et al*., 2013) and for long term cash forecasting using simple ANN (Ramya and Alli, 2015) was developed in which the weights and bias were updated in the direction of negative gradient. Hence, there is a possibility of convergence point getting trapped into local minima. Since, there is a need to improve and optimize the result by using PSO.

The collected data was for a period of 3 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 3 years
- MOY-Month of the year: Ranges from 1 to 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 was varied from 7 to 20. The output layer had one neuron that corresponds to the optimal cash requirement for a day.

The following ANN parameters were sufficient to reduce the error by assigning the appropriate values:

- Number of hidden nodes
- Number of training samples
- Maximum number of iterations

A training algorithm was to determine the optimal values by varying the number of hidden nodes. It is a three layered network architecture which has one input, one hidden and one output layer. The activation function to calculate the weights and biases between input layer and the hidden layer was denoted in (3), in a similar way the weights and biases can be calculated between the hidden layer and the output layer using (4):

$$n = \frac{2}{\left(1+exp\left(-\left(2*n\right)\right)\right)-1} \quad (3)$$

$$op = \frac{1}{(1+exp(-n))} \quad (4)$$

The learning error rate (E_k) can be calculated by finding the square of the difference between the actual and the predicted value using (5):

$$E_k = \sum_{i=1}^{m}\left(o_i^k - d_i^k\right)^2 \quad (5)$$

To improve the performance of our proposed method ANN uses PSO parameters to calculate the weight and bias are:

- Cognitive and learning factor
- Initial velocity and position of the particle
- Dimension of particles
- Number of particles
- Inertia weights

Table 1: Pseudocode CFNN-PSO

```
//Initialize Parameters for PSO and FNN
Tin = 737 (longterm) /48 (shortterm); noP = 30;
maxiter = 500 (longterm) /100 (shortterm); maxrun = 30; r1 = r2 = rnd ();
gbest = 0; pbest = 0; pbestscore = 0; gbestscore = inf; fitness = 0
wmax = 0.9; wmin = 0.4; C1= C2 = 2; dt = 0.8
CF-FNN-PSO (tn, dim, hn, nop, wt, vel, pos)
{
For ts = 1 : Tin
   pos (i, j) = rnd ()
   vel (i, j) = 0.3 * rnd ()
for Iter = 1: maxiter
   for i = 1: nop
      for ww = 1: 7 * hno
          wt (ww) = pos (i, ww)
      for bb = 7: hno
          Biases (bb − (5 * hno)) = pos (i, bb)
     for pp = 1 to tin
av (pp) =
MY_FNN3y (Ino, Hno, Ono, W (i), B (i), RY, MOY, WDOM, WDOW, SDE, HDE)
        fitness = fitness + (t (pp) − av (pp))^2
    fitness = fitness/Tin
if iter = 1
{
  pbestscore (i) = fitness
  else
        if pbestscore (i) > fitness
{
      pbestscore (i) = fitness
                 pbest (i, : ) = pos (i, : )
   }
 if gbestscore > fitness
{
      gbestscore = fitness
                 gbest = pos (i, : )
}
    if gbestscore == 1
   Break;
   {
         wt = wmax − iter * (wmax − wmin) /maxiter
vel (i, j) = Σ_{i=1}^{nop} Σ_{j=1}^{dim} w * vel (i, j) + c1 * r1 * (pbest (i, j) −
pos i, j+c2*r*(gbest j−pos i,j)
         pos (i, j) = Σ_{i=1}^{nop} Σ_{j=1}^{dim} pos (i, j) + dt * vel (i, j)
                }
   }
av = My_FNN3y (Ino, Hno, Ono, W (i), B (i), RY, MOY, WDOM, WDOW, SDE, HDE)
{
 h1 = Σ_{i=1}^{Hno} (RY * w (i) + MOY * w (hno + i) + WDOM *
w 2*hno+i+WDOW*w 3*hno+i+SDE*w
4*hno+i+HDE*w 5*hno+i+b i
Compute av as per Eq. (3), (4).
 }
```

The pseudocode for the proposed model was illustrated in Table 1 defines the functionality of ANN using PSO.

Evaluation metrics: The error between the actual and forecast data is calculated using:

- Mean Absolute Percentage Error (MAPE)
- Mean Squared Error (MSE)
- Mean Absolute Error (MAE) which are defined as follows:

$$\mathrm{MAPE} = \frac{\Sigma\left|\frac{X_t - F_t}{X_t}\right|}{n}(100) = \frac{\Sigma\left|\frac{e_t}{X_t}\right|}{n}(100) \tag{6}$$

$$\mathrm{MSE} = \frac{\Sigma|X_t - F_t|^2}{n} \tag{7}$$

$$\mathrm{MAE} = \frac{\Sigma\left|\frac{X_t - F_t}{X_t}\right|}{n} \tag{8}$$

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

RESULTS AND DISCUSSION

The CF-FNN-PSO was developed to maintain the right amount of cash to provide effective customer support services using evolutionary computing technique were implemented for two different bank branches, short term data of a particular branch of the State Bank of India (Prem and Ekta, 2006) and for long term data of City Union Bank data set for three years has been collected and simulated using MATLAB.

In our proposed methodology total data set used for long term was 879 and for the short term was 60 in which 80% of data (737 for long term, 48 for short term) were used for training and the remaining 20% were used for testing (142 for long term, 12 for short term). The CF-FNN-PSO was simulated using MATLAB in which all the inputs and target values were normalized (Sola and Sevilla, 1997).

The parameter selection varies for short term and long term cash forecasting in which four input parameters were used for short term forecasting which was <1 year, such as WDOM, WDOW, SDE, HDE, since the prediction has been made for 60 days. In case of long term cash forecasting with the fore mentioned input parameters we had used two additional input parameters which are RY, MOY. The neural network parameters were alone will not be sufficient, hence,

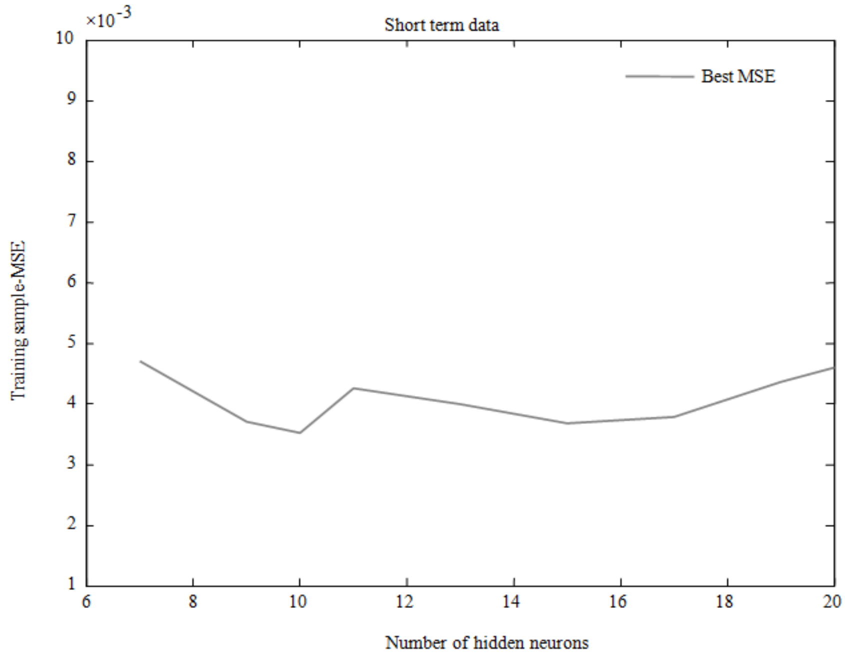

Fig. 1: Performance of short term data

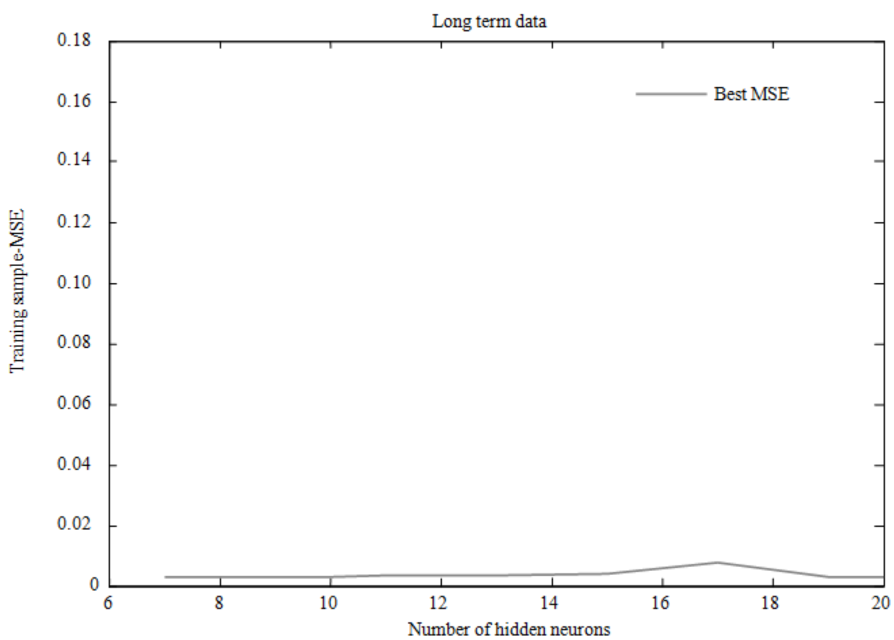

Fig. 2: Performance of long term data

we had used PSO parameters to minimize the MSE using appropriate weights and bias.

The process of choosing the number of hidden neurons in our CF-FNN-PSO is very crucial to evaluate the performance of the network, since it influences our results in over fitting or under fitting problems. Therefore, in our proposed study, the experimental analysis was made by varying the number of hidden neurons between 7 and 20 out of which 10 gives us the best results with the minimal MSE 0.0035 for short term data was plotted in Fig. 1, in case of long term data the number of hidden neurons was 9 with minimal MSE 0.0029 was the evidence from Fig. 2

The acceleration coefficients C1 and C2, which regulates the flow of particles to reach the global best by sharing the knowledge to determine the optimal values in the problem space (Olden *et al*., 2004; Kennedy and Eberhart, 1995). In our CF-FNN-PSO the

Table 2: Experimental analysis of C1 and C2 to find the optimal values

C1	C2	MSE
0.0	4.0	2.7010
0.5	3.5	1.5624
1.0	3.0	2.3144
1.5	2.5	0.8030
2.0	2.0	0.7803
2.5	1.5	2.7835
3.0	1.0	1.4831
3.5	0.5	4.6530
4.0	0.0	2.7010

experimental analysis was made to identify the suitable values for a social and cognitive parameter was tabulated in Table 2, column-3. The minimum MSE for the combination C1 = C2 = 2 was found to be the best for our model with an MSE = 0.7803.

In CF-FNN-PSO, on experimental based 30 independent runs were made with the following parameters to reduce the effect of random initialization. The number of swarms was 30. The number of hidden

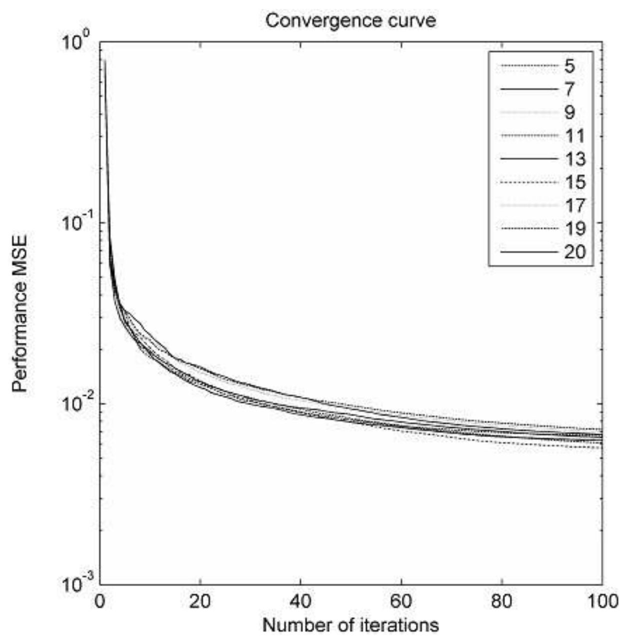

Fig. 3: Convergence curve for short term data

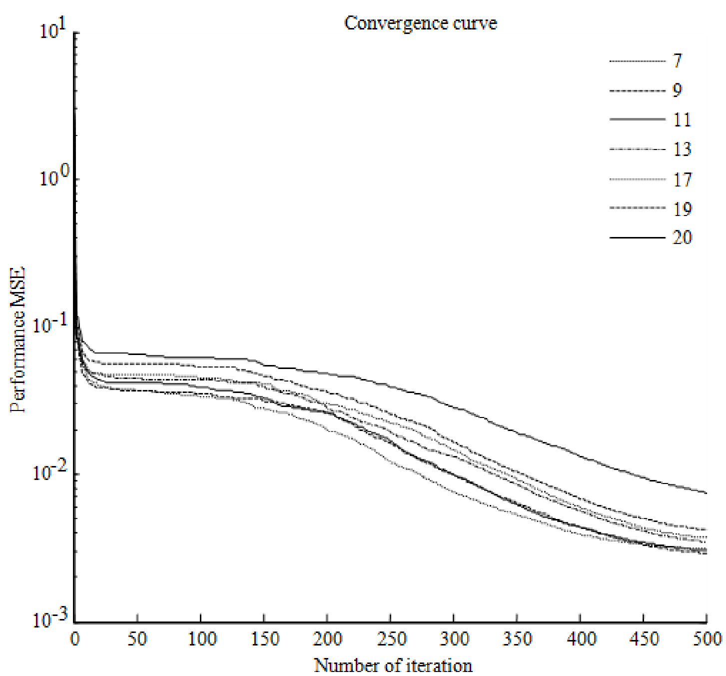

Fig. 4: Convergence curve for long term data

layer neurons in the neural network was set to 9 for long term data. The different sigmoidal functions were analyzed and used to determine the best MSE using tansig-logsig. The C1, C2 values were taken as 2 which has minimal MSE The r1, r2 values were random values between (0-1) respectively. The Δt (dt) value was set to 0.8 was the time interval of the movement of particles in the solution space. The inertia weight was used to improve the convergence rate of PSO algorithm. The inertia 'w' influenced the results through local and global explorations. The initial and the final inertia value of previous velocity was set at 0.9 and gradually decreases to 0.4 during execution, which specifies the range of the search space as well as it regulates the speed of the particles to reach the convergence point.

The convergence of CF-FNN-PSO was influenced by number of hidden neurons as well as the fore mentioned training parameter by varying the number of hidden neurons between 7 through 20 with the maximum of 100 iterations was illustrated in Fig. 3 for the short term data using 6-10-1 pattern, similarly for long term data using 6-9-1 pattern with the maximum number of 500 iterations, it was shown in Fig. 4 using which the performance can be measured by calculating the weights and the bias value.

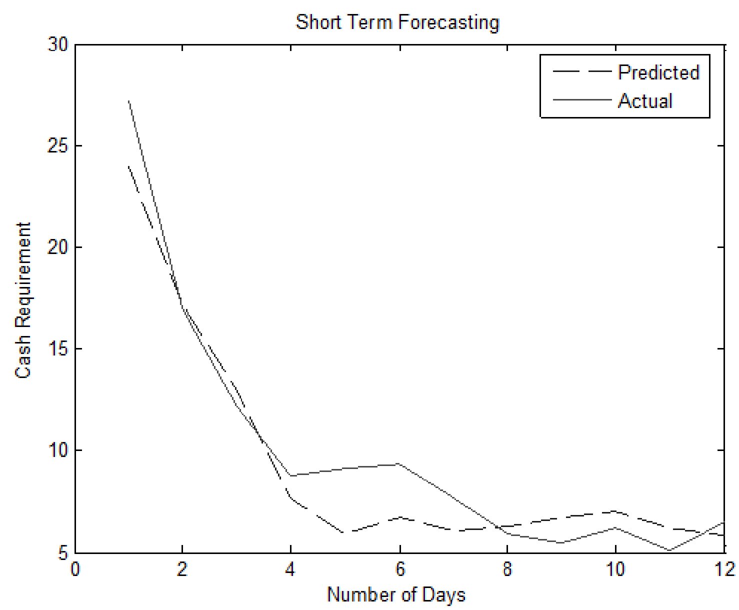

Fig. 5: Short term cash forecasting

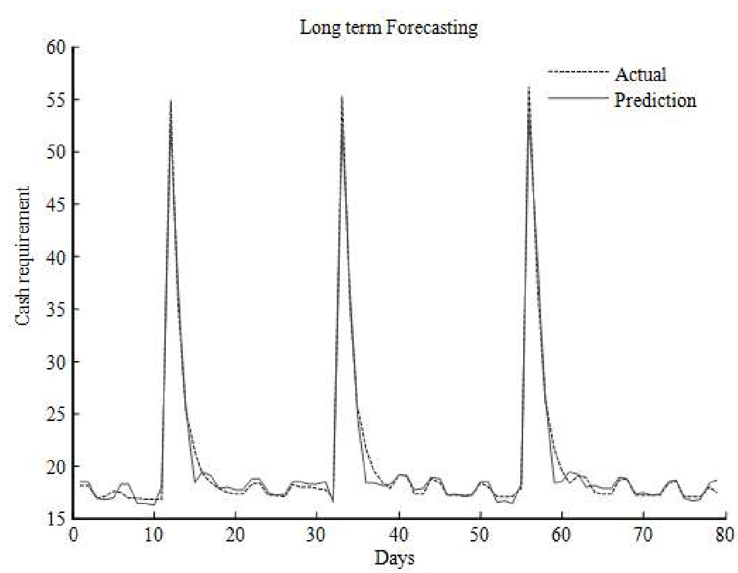

Fig. 6: Long term cash forecasting

Table 3: MAE, MAPE, MSE for the test data set

Type of data	MAE	MAPE	MSE
Short term	1.6161	6.1392	5.1824
Long term	1.0767	4.8031	3.3161

The particle velocity and position was updated using local best and global best values. This will avoid the weights and biases trapped into a local minimum. The performance of CF-FNN-PSO was shown in Fig. 5 and 6 guaranteed that the results of the statistical variable have the best ability to avoid local minima.

The difference between the actual and the predicted values were plotted in the above graph which gives the guarantee that the estimated values were sufficient to maintain the optimal cash for both short term and long term forecasting. The computational time for the short term forecasting was 79.8335 sec and for the long term cash forecasting 392.3120 sec. The average MAE, MAPE, MSE estimated for the test data set of short term and long term data was tabulated in Table 3.

CONCLUSION

In this study, CF-FNN-PSO was successfully implemented for cash forecasting of two different banks for short term and long term data. The proposed model used a hybrid approach to avoid the slow convergence and the trap in local minima. The training samples for short term and long term data were trained using ANN through PSO to determine the optimized weight and bias. The values were assigned to parameters based on the experimental analysis. PSO-ANN are represented by particles position. These particle velocities and position are updated, which search for personal best and global best values. This will avoid the weights and biases being trapped in local minima. Therefore, the obtained results guaranteed that the proposed model is sufficient to maintain the optimal cash balance to satisfy the customer needs on a daily basis without any shortage of funds. This approach may be extended to apply on the other dataset to show its effectiveness. In the future, we will try to implement this approach for different data sets using different optimization technique.

REFERENCES

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

Corchado, E., A. Abraham and A. de Carvalho, 2010. Hybrid intelligent algorithms and applications. Inform. Sciences, 180(14): 2633-2634.

Corchado, E., M. Graña and M. Wozniak, 2012. New trends and applications on hybrid Artificial intelligence systems. Neurocomputing, 75(1): 61-63.

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

Gaitonde, V.N. and S.R. Karnik, 2012. Minimizing burr size in drilling using Artificial Neural Network (ANN)-Particle Swarm Optimization (PSO) approach. J. Intell. Manuf., 23: 1783-1793.

García, S., A. Fernández, J. Luengo and F. Herrera, 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inform. Sciences, 180(10): 2044-2064.

Hagan, M. and M. Menhaj, 1994. Training feed-forward networks with the Marquardt algorithm. IEEE T. Neural Networ., 5(6): 989-993.

Ho, K.L., Y.Y. Hsu, C.F. Chen, T.E. Lee, C.C. Liang, T.S. Lai and K.K. Chen, 1990. Short term load forecasting of Taiwan power system using a knowledge based expert system. IEEE T. Power Syst., 5(4): 1214-1221.

Junliang, L. and X. Xinping, 2008. Multi-swarm and multi-best particle swarm optimization algorithm. Proceeding of the IEEE 7th World Congress on Intelligent Control and Automation (WCICA, 2008). Chongqing, pp: 6281-6286.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proceeding of the IEEE International Conference on Neural Networks. Perth, Australia, Piscataway NJ, pp: 1942-1948.

Lee, K.Y, Y.T. Cha and J.H. Park, 1992. Short-term load forecasting using an artificial neural network. IEEE T. Power Syst., 7(1): 124-132.

Leung, F.H.F., H.K. Lam, S.H. Ling and P.K.S. Tam, 2003. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. IEEE T. Neural Networ., 14(1): 79-88.

Olden, J.D., M.K. Joy and R.G. Death, 2004. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. Ecol. Model., 178(3): 389-397.

Omran, M.G.H., A.P. Engelbrecht and A. Salman, 2006. Particle swarm optimization for pattern recognition and image processing. Stud. Comp. Intell., 34: 125-151.

Park, D.C., M.A. El-Sharkawi, R.J. Marks, L.E. Atlas and M.J. Damborg, 1991. Electric load forecasting using an artificial neural network. IEEE T. Power Syst., 6(3): 442-449.

Prem, C.K. and W. Ekta, 2006. Cash forecasting: An application of artificial neural networks in finance. Int. J. Comput. Sci. Appl., 3(1): 61-77.

Rahman, S. and R. Bhatnagar, 1988. An expert system based algorithm for short term load forecast. IEEE T. Power Syst., 3(2): 392-399.

Ramya, M.M. and A. Alli, 2015. Optimizing cash management model using computational Intelligence. Res. J. Appl. Sci. Eng. Technol., Vol. 06.

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