An Efficient Hybrid Model to Load Forecasting

Md. Khairul Hasan†, **Mohammad Asif A Khan**†† , **Suman Ahmmed**††† and **Ahmed Yousuf Saber**††††

†Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
†† M.Sc Student, Department of Computer Science, University of Saskatchewan, Saskatoon, Canada.
††† Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh.
†††† Real Time Power and Intelligent Systems Laboratory, Missouri University of Science and Technology, USA.

Summary
This paper presents a new hybrid approach of Neural Network (NN) along with Particle Swarm Optimization (PSO), hereafter called NN–PSO, to resolve the short–term load forecasting (STLF) tasks with improved generalization technique for the Regional Power Control Center of Saudi Electricity Company, Western Operation Area (SEC-WOA), The strength of this powerful technique lies in its ability to forecast accurately on weekdays, as well as, on weekends and holidays. In this research paper, first prediction is made by NN and then PSO is applied to optimize the prediction and thus improve its accuracy. NN is an effective mathematical tool for mapping complex relationships. It is also successful for doing forecasting, categorization, classification, and so forth. On the other hand, PSO is a novel random optimization method based on swarm intelligence, which has been found to be powerful in solving nonlinear optimization problems. It has better balance of local and global searching abilities and can handle huge multidimensional optimization problems efficiently with hundreds of thousands of constraints. PSO is applied to optimize the weighting factors of NN. To combine with NN, Particle Swarm Optimization has been proposed here to attain better performances. To evaluate the proposed NN-PSO algorithm it is applied on the real data set of SEC-WOA for the year 2003. The test results show that the proposed NN-PSO performs better and consistent with respect to only NN for most of the cases in load forecasting.

Key words:
Back-propagation, heuristic, neural network, particle swarm optimization.

1. Introduction

Load forecasting have become one of the major areas of research in electrical engineering because they are practically used in power system for unit commitment, real-time dispatch, maintenance, optimization of power systems and so on. Load forecasting and its optimization are challenging tasks depending on previous real data. Depending on the time horizon, load forecasting can generally be divided into short-term, mid-term, and long-term. Short-term load forecasting [1] ranges from an hour to a week, on the other hand mid/long-term load forecasting covers a few weeks to several years. Unit commitment uses this forecast load demand as input and generates proper schedule for available units with minimum cost. In this research, a forecasting model has been proposed for the Western Area of Saudi Arabia, which is shown in Figure 1. There are about 110 thermal generating units; weather is uncertain; and load mainly depends on weather and Islamic events (Hajj, prayer, holidays, etc.). Research outline is shown in Figure 2. In the figure, load characteristics and events based load are indicated. Desert weather is also shown in the figure. Huge real data are then processed in the central part of the figure. Brain like NNs is applied in the method. Besides, PSO is applied for the optimization of weight and bias matrices of NN.

Many researchers have addressed load forecasting. Load forecasting is a difficult task because of many input attributes and mapping of huge practical data of complex nature. Firstly the load series is complex and exhibits several levels of seasonality. Secondly the load at a given hour is dependent not only on the load at the previous hour, but also on the load at the same hour on the previous day. There are also many important exogenous variables that must be considered, especially the weather related variables [2]. Load forecasting plays an important role in power system planning, operation, dispatch, maintenance and so on. Basic operating functions such as unit commitment, economic dispatch, fuel scheduling, and unit maintenance and so on can be performed efficiently with an accurate load forecast [3]. Various statistical forecasting techniques have been applied to short term load forecasting (STLF). Examples of such methods including, time series [4], similar day approach [5], regression methods [6], expert systems [7]. In general, these methods are basically linear models and the load pattern is usually a nonlinear function of the exogenous variables. On the other hand, artificial neural network (NN) has been proved as a powerful alternative for STLF that does not rely on human experience. NN is a data driven method, in the sense that it is not necessary for the researcher to postulate tentative models and then estimate

Manuscript received August 5, 2010
Manuscript revised August 20, 2010

their parameters. Given a sample of input and output vectors, NN simply maps the relationship. However, the researchers want to include optimization for practical input load data with corresponding errors and weight matrix of NN, as NN is not an optimization method.

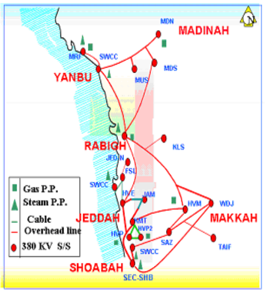

Fig 1: The Main Network of Western Area of Saudi Arabia

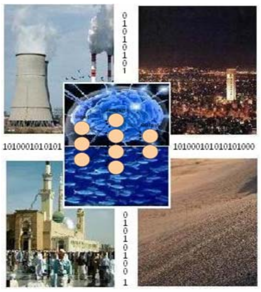

Fig 2: Visual Research Outline

In recent years, AI-based techniques such as neural networks [8] have been used to obtain promising results. Neural networks have the capability to approximate any continuous nonlinear function, and can adapt to a changing forecasting environment through self-learning. However, it is often difficult to decide whether the obtained neural network is the best due to the tedious and trial and error tuning process, and model overfitting that may occur and results in less satisfactory forecasting accuracy. Some variants of neural networks have been applied for the short term load forecasting such as fuzzy neural networks [9], adaptive neural networks [10], Bayesian neural network [11], and Back propagation neural networks [12]. Among them, the fuzzy neural network is one of the popular load forecasters. Some techniques have also been combined with neural network to find highest accuracy such as gradient methods [13], fuzzy expert system [14], and wavelet transformation [15]. Other techniques such as fuzzy logic [16], GA [17], and support vector machines (SVMs) [17] have also been applied to load forecasting.

The advantage of using particle swarm optimization (PSO) [18] algorithm over other techniques is that it is computationally inexpensive, easily implementable, and does not require gradient information of an objective function, but only its values. Therefore, the particle swarm optimization algorithm is applied to the neural network to obtain a set of weights that will minimize the error function in competitive time.

The main limitation of the NN in STLF is that it always depends on the back propagation algorithm to train the NN. It does not consider the randomness nature to escape from local optima. On the other hand PSO has the guided randomness property. That's why it can explore more search spaces and can avoid local optima gradually. So the authors have chosen NN for forecasting and then PSO for optimization.

This paper presents a hybrid approach of NN and PSO model for STLF, typically for next day, for the Western Area of Saudi Arabia. A significant improvement has found after applying PSO on NN. Weather, day type, special events and load related inputs are considered in this model. One year of historical dependent data are used. A special design for the forecasting system that copes with the features of Saudi Arabia (SECWOA) electrical load is presented. It is found that the error is in acceptable range for the proposed method. The proposed NN-PSO method performs better with respect to NN.

2. Problem Formulation

Given Day of the week (D_i), Special event indicators (E_i), Minimum and Maximum temperature of previous day (T_{i-1}), and pervious 7 days load ($Y_{i-j;\ j=1,\ldots7}$); where Y_i is the load of the day to be forecast (i), the predictor has the general form $Y_i = f\,(D_i, E_i, T_{i-1}, Y_{i-j})$. In other words one may use data from the past load history and temperature as well as day type and special event indicator of the forecasted day to forecast the load. This prediction will be used for scheduling the power generators to be activated in the following working day (i).

3. Proposed Model

3.1 Major Components of the Model

3.1.1 NN for Load Forecasting

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) among elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The network weights remain adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many input/target pairs are needed to train a network.

It is well known that a three layered NN can solve any kind of linear and nonlinear problems. Here the neural network consists of three layers: input layer, hidden layer and output layer. The layers are arranged to have a multilayer feed-forward design. A general network model consists of simple processing elements called neurons, which are connected by synaptic weights which are adjusted through a gradient learning process, Back-Propagation (BP) algorithm. These neurons are arranged in a distinct layered topology and perform a biased weighted sum of their inputs and pass this activation level through a transfer function such as sigmoid or tan-sigmoid function to generate their output.

A fully connected two layer feed-forward NN is used in this development. Other kinds of NNs such as recurrent NNs and radial basis function NNs will not discover any major advantage over the feed-forward Network for the load forecasting problem. The NN has 18 input neurons. The number of hidden neurons was parametrically optimized to 15 and the number of output was set at 1, comprising forecast of the load of the day concerned. The NN inputs and output are listed in Table 1.

Table 1: NN Inputs and Output

Inputs	Description
1-3	Day of the week (bit encoded, 001-111)
4	School Index
5	Examination
6	Ramadan
7	Eid
8	Hajj
9	Holiday
10	Minimum temperature of the previous day
11	Maximum temperature of the previous day
12-18	Previous 7 days actual load
Output	**Description**
1	Forecasted Load of the day

Figure 3 reports the schematic of the architecture of the NN, as implemented. A simplified form of the proposed NN model can thus be summarized as follows:

$$Y_i = f\,(D_i, E_i, T_{i-1}, Y_{i-j}) \quad \dots \quad \dots. \quad \dots. \quad \dots. \quad \dots. \quad (1)$$

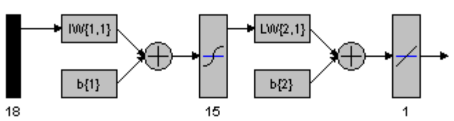

Fig 3: NN load forecasting

The NN is trained on training sets composed by the daily load and temperature patterns as well as the day of the week and special events of 11 months. Therefore a weight set is obtained from the training stage for the forecasting task of 11 month. For instance, in order to forecast the daily load for December 3 2003, the weight set obtained from the NN training stage up to November 2003 is used. The input used in the forecasting activity will include the daily load and temperature data of December 2 2003 and the day of the week of December 3 2003. No further specialization is required concerning the day type, since the integer code included in the patterns permits the network to achieve a sufficient accuracy for all the day types included in periods with normal load conditions. But sometimes NN cannot able to provide high accuracy. So the authors propose an evolutionary algorithm named Particle Swarm Optimization (PSO) for weight set optimization of NN.

3.1.2 Particle Swarm Optimization for Weight Optimization of NN

Particle Swarm Optimization is a heuristic approach first proposed by Kennedy and Eberhart in 1995 as an evolutionary computational method developed for dealing with the optimization of continuous and discontinuous function decision making. The PSO algorithm is based on the biological and sociological behavior of animals such as schools of fish and flocks of birds searching for their food. PSO imitates this behavior by creating a population with random search solution and each potential solution is represented as a particle in a population (called swarm). The social sharing of information among the particles of a population may provide an evolutionary advantage. Each particle is flown through the multi dimension search space with random and adaptable velocity in order to find the lower function values (global minimum).

In standard PSO algorithm, particles are manipulated according to (2) and (3) where each particle tries to adjust its velocity according to best positions ever visited that is stored in its memory called personal best *pbest* and according to the best previous position attained by any particle in its neighborhood called global best *gbest* trying to search for a better position. Thus, particles communicate with each other and distribute their information among each other during their search.

$$v_i(t+1) = wv_i(t) + r_1c_1\left[pbest - x_i(t)\right] + r_2c_2\left[gbest - x_i(t)\right], \; i = 1, 2, \; \dots. \; \dots. \; N, \; \dots \; \dots \quad (2)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad \dots. \quad \dots. \quad (3)$$

Here w is an inertia weight, which provides a balance between the local and global exploration. V_i and V_i (t+1) are current and modified velocity for each iteration, respectively. C_1 and C_2 are positive numbers, used to control the particle's movement at each iteration. They represent cognitive and social components, respectively. r_1 and r_2 are uniform distribution numbers in the range [0, 1]. $X_i(t)$ and $X_i(t+1)$ are the current and modified position for each iteration, respectively. N is number of particles.

In this paper the connection weights between input to hidden and hidden to output layers are represented in two matrices W_1 of size ($m_1 \; X \; n_1$) and W_2 of size ($n_1 \; X \; m_2$) respectively. The current position $X_i(t)$ for each particle is represented by $W_i(t) = \{W_1, W_2\}$. Therefore, the position of each particle represents a set of weights for the current iteration. Thus, using equations (2) and (3), updated velocity and then position for each particle is determined to minimize the error of the network. The dimension of the search space for each particle is the number of weights associated with the network. The fitness value for each particle is determined based on the new position value of each weight matrix.

3.2 Proposed Algorithm

The major steps of the algorithm are summarized as follows:

Step 1: To select the NN structure initially a fully connected two layered feed-forward NN is taken. The numbers of input and output neurons are determined by the input factors and the number of targeted outputs. To determine the number of hidden neurons various experiments of STLF are performed and finally fixed the number which shows the best result.

Step 2: Train the NN by BP algorithm for selected dataset. The dataset is divided into three different sets like training, validation and testing datasets. Training dataset is used to train the NN. Validation dataset is used for determining the stopping criteria of the NN and the Testing dataset is used for determining STLF performance. After training completion we store the weight matrix for further updates.

Step 3: STLF is done by using the present NN found in Step 2.

Step 4: A number of matrix sets that means candidate particles are generated from the stored weight matrices in Step 2. This step explores the opportunity of randomness and generates more search spaces.

Step5: Initialize local best (*pbest*), global best (*gbest*), and other PSO parameters w, c1 and c2 using standard PSO rules.

Step 6: STLF is done using the NN having the new weight matrices.

Step 7: If the performance of step 6 is better than the previous one, then go to Step 8, otherwise go to Step 9.

Step 8: Update *pbest* and *gbest* parameters based on the performances of current solutions.

Step 9: In this step the present solutions (particles) are updated based on equations (2–3). Thus a new set of solutions is obtained and the new search spaces are explored.

Step 10: Check whether the stopping criterion is met or not. If met then the global best weight matrix set is taken as the solution of the proposed method, and go to next step, otherwise go to step 6 for further exploration.

Step 11: The updated weighted matrix is put in the NN structure. Thus the new NN is constructed applying PSO. Therefore the STLF is done by this new NN and the performance is treated as the performance of the proposed method. In most of the cases the proposed method performs better with respect to only NN approach.

Step 12: Compare the performance of Step 3 and Step 11.

Step 13: Print the results.

For a simplified representation of the proposed algorithm, a Block diagram of it has furnished in Figure. 4. In Figure. 4, it can be seen that initially a NN (Block 1) has trained for the given dataset. Thus the performance of the NN has determined (Block 3 lower part). The weight matrix set of NN is sent to Block 2 for applying PSO on it to explore more solution spaces or particles. In Block 2, the interim performance of the different candidate particles is measured for STLF. An iterative process in Block 2 is continued until it is reached at its stopping criteria.

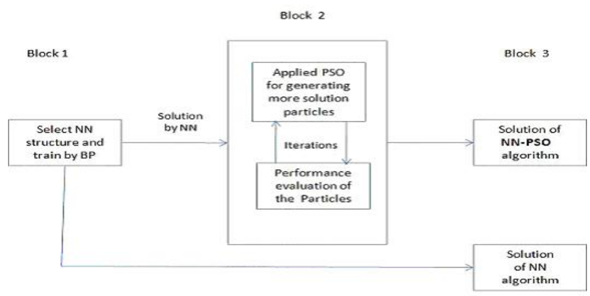

Fig 4: Block diagram of the proposed method

Once it reaches at its stopping criteria it provides the solution particle, *gbest*, of the proposed NN-PSO algorithm. Block 3 shows the two separate solutions delivered by NN-PSO and NN algorithm.

4. Simulation and Result

All calculations have been run on Intel(R) Core(TM)2 Duo 2.66GHz CPU, 2.96 GB RAM, Microsoft Windows XP OS and MATLAB Version 7.6 compiler for coding. NN delivers four weight matrices. PSO is applied to tune these weight matrices to achieve more accuracy in forecasting. n (20) number of matrix sets that means candidate particles are generated from the main matrix set by multiplying each element with a different random number ranged from 0-1. Initialize the values of PSO parameters, w is decreasing in each iteration from 1.0 to 0.4 uniformly, and c1 = 1.5 and c2 = 0.5. The maximum number of iterations, m = 500. The performance is measured in terms of mean absolute percentage error (MAPE) defined as:

$$MAPE = 100 / N \sum_{i=1}^{N} (|Actual\ (i) - Forecast\ (i)|) / Actual\ (i) \quad .... \quad .... \quad (4)$$

where N is the number of test data.

4.1 Selection of Dataset

The data set used for this study consists of historical load, weather information, day type and special events for a dataset area of Saudi Arabia. In order to avoid convergence problems during the training process, the network input data and the corresponding target vector for the forecasting models are normalized such that they fall within the range [0, 1]. The data are divided into two data sets: the training data set and the testing data set. For each test case January, 2003 to November, 2003 data are used for training and data of December 2003 are used for testing.

4.2 NN Training

Firstly, the network uses the Levenberg-Marquardt algorithm for training. Input vectors and target vectors are randomly divided into two sets so that 75% are used for training and 25% are used to validate that the network is generalizing and to stop training before overfitting. The tan-sigmoid function is used as transfer function for hidden layer and linear function “purelin” is used for output layer. The training stops when the performance goal that is Mean Square Error (MSE = 0.001) is met or when the training error decreases but the validation error increases or when the upper limit of iterations is reached.

4.3 Determination of NN Structure

After a lot of investigation the number of input nodes is determined by the number of inputs that are responsible for determining forecasting. Weather conditions, day type, special events and historical load data are important factors playing important roles in short term load forecasting. In this paper these factors are taken into account. Table I presents eighteen input characteristics which are key factors for load forecasting. So the number of input is determine at eighteen (18). As the target is to forecast the load for next day’s average load, so the number of output nodes becomes one (1). The number of hidden neurons is determined by a wide variety of the number of hidden neurons from 10 to 30 and simulates the performances and thus based on performance 15 hidden neurons are found optimal. Therefore the BP NN whose topological structure is 18x15x1 has been set up for the STLF of Jeddah, KSA data.

4.4 PSO Optimization

The NN topology is 18 – 15 – 1 with 18 neuron representing inputs, 15 neurons in the hidden layer and 1 neuron in the out-put layer with total number of weights equal to 301 including 19 weights linking bias nodes. For the above mentioned network structure of 18 – 15 – 1 the dimensionality of a particle in the PSO system is (18 x 15 + 15 x 1 + 15 + 1) = 301. This means that each particle fly in 301 dimensional spaces in search of optimal set of weights that minimizes the MSE for the given training.

4.5 Results

Table 2 presents a comparison of the percentage error, minimum error, maximum error, and standard deviation based on NN and NN – PSO models for 10 test cases. From the table it is evident that NN – PSO performs better than the NN technique. For training, NN uses eleven months data (January 2003 – November 2003) and for testing, it uses December, 2003 data. Table 2 shows that the average MAPE using NN is 4.0928%. However, the average MAPE is 1.8532% for proposed method. Accuracy is improved by 2.2396%.

Table 2: Performance of NN and NN–PSO and achieved improvements

		Tst_01	Tst_02	Tst_03	Tst_04	Tst_05	Tst_06	Tst_07	Tst_08	Tst_09	Tst_10	Average
	NN	4.1214	4.5302	3.4836	4.6341	4.1777	4.3137	4.1898	3.0223	4.2653	4.1898	4.0928
MAPE	NN-PSO	2.0286	1.2493	1.7835	1.4621	1.7887	1.5373	2.315	1.5598	2.493	2.315	1.8532
	Imp.	2.0928	3.2809	1.7001	3.1719	2.389	2.7764	1.8748	1.4625	1.7723	1.8748	2.2396
	NN	0.1499	0.2854	0.0644	0.2642	0.5078	0.4902	0.0615	0.0209	0.0337	0.0615	0.194
MIN	NN-PSO	0	0	0	0.0015	0	0	0	0.0001	0	0	0.0002
	Imp.	0.1499	0.2854	0.0644	0.2627	0.5078	0.4902	0.0615	0.0209	0.0337	0.0615	0.1938
	NN	13.946	13.096	7.9547	14.814	13.235	10.543	15.776	8.7301	12.676	15.776	12.655
MAX	NN-PSO	8.1955	6.4771	5.8215	5.7115	7.7957	10.16	8.4268	6.1855	7.4062	8.4268	7.4607
	Imp.	5.7506	6.6183	2.1333	9.1027	5.4395	0.3828	7.3497	2.5447	5.27	7.3497	5.1941
	NN	3.1824	3.7001	2.3652	3.1839	2.9021	2.7289	3.6441	2.314	2.7516	3.6441	3.0416
STD	NN-PSO	2.5515	1.5417	1.8193	1.5578	2.0008	2.2352	2.2936	1.7761	2.3702	2.2936	2.044
	Imp.	0.631	2.1584	0.5458	1.6261	0.9013	0.4937	1.3505	0.5379	0.3814	1.3505	0.9977

Note: Tst_01 means test no. 1; MAPE means mean absolute percentage error; STD means standard deviation; Imp. means Improvement.

The following five diagrams, Figures 5–9, are drawn based on the load forecasting results of test case 01 for December 2003.

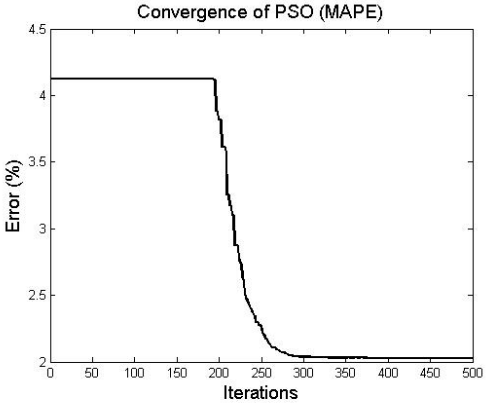

Fig 5: Convergence of PSO (MAPE)

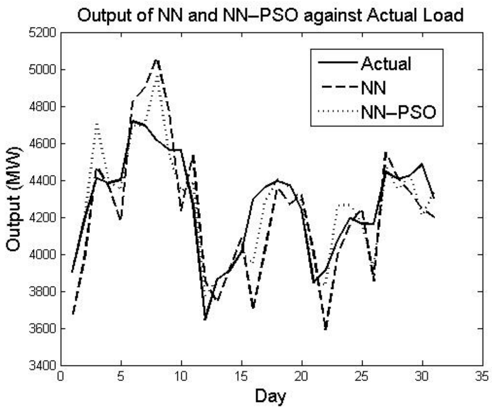

Fig 6: Load comparison

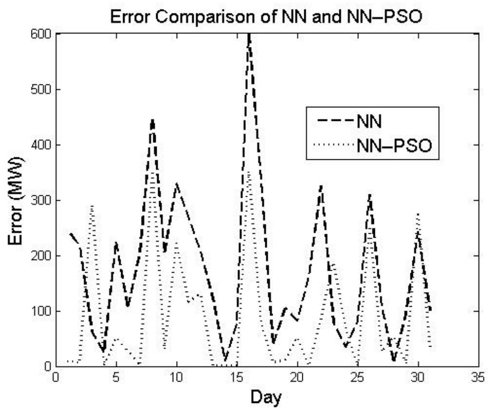

Fig 7: Actual error comparison

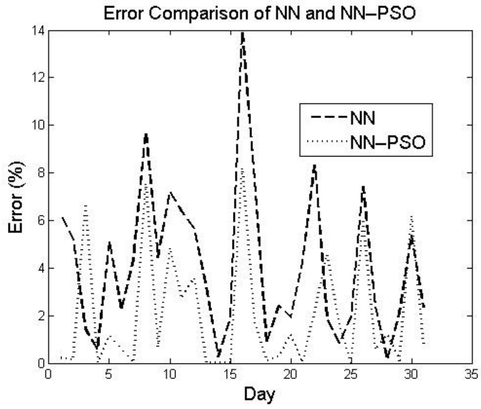

Fig 8: Percent error comparison

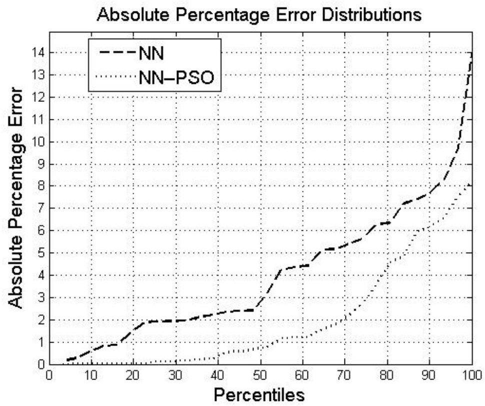

Fig 9: Absolute Percentage error distributions

In Figure 5, the curve shows how the error decreases in each iteration of NN–PSO exploration. It is observed that the major performance improvement occurs during 200 – 300 iterations. Why convergence does not start at the beginning, but at this stage will be a subject to research in future. In Figure 6, dashed and the doted curves represent forecasted load by NN and NN–PSO respectively. It is clearly visible that the actual load and NN–PSO method's forecasted loads are very close, whereas the only NN based forecasted load is apart from the actual load. NN depends on only BP training algorithm and it has been proven that gradient techniques are slow to train and are sensitive to the initial guess, which could possibly be trapped in a local minimum, thus the performance of NN is lower. On the other hand, when applying PSO, there exposes the randomness nature and also applies a guided search mechanism for converging into the best solution. As a result NN–PSO performs better. So it can be concluded that the overall performance of NN–PSO method is significantly better with respect to NN approach. In Figure 7, the dashed and the doted curves represent the actual amount of errors occurred by applying NN and NN–PSO methods respectively, which are depicted in percentage in Figure 8. The error of NN is 4.1214% with a maximum error of 13.946%, and the error of the proposed NN–PSO is 2.0286% with a maximum error of 8.1955%. In Figure 9, the dashed and the doted curves represent the absolute percentage error distributions occurred by applying NN and NN–PSO methods respectively. The NN-PSO curve shows that 70% of the time the error is within 2% whereas only 30% of the time the error is within 2% in case of only NN.

5. Conclusion

The proper load forecasting model for a practical case is critically important decision for operating the electric power system securely and economically. Accurate load forecasting can save a huge amount of money as well as do efficient generation and distribution planning. In this paper, a hybrid approach of neural network with particle swarm optimization learning algorithm for sort term load forecasting is presented. A significant improvement has found after applying PSO on NN. Average MAPE is 1.8532% where maximum, minimum and standard deviation of results are also tolerable. Robustness of the proposed method is shown by testing one month data. The NN–PSO performs better with respect to NN for that case as well. In this research, the authors' contributions are:

- Simplification has made by identifying the most important attributes needed in STLF, which is a real challenge for a practical dataset handling and as a consequence it makes NN much faster
- Proposing appropriate NN structure for a practical dataset
- The appropriate application of PSO to optimize and enhance the performance of NN.

The load of the Western Area of Saudi Arabia depends on uncertain desert weather and also some religious events like Eid, Hajj, Ramadan etc., and thus in future the authors will include fuzzy approach in this model to make it more optimized especially in some special cases. Moreover, considering the factual circumstance of STLF, how to introduce automated parameter selection mechanism for making it more automated will be an important issue in future research.

References

[1] S. Rahman and I. Drezga, "Identification of a standard for comparing short-term load forecasting techniques," Electric Power Systems Research, vol. 25, no. 3, pp. 149-158, Dec 1992.

[2] H. S. Hippert, C. E. Pedreira, and R. C. Souza. "Neural networks for short-term load forecasting: A review and evaluation," IEEE Trans. Power Syst., vol. 16, no. 1, pp. 44-55, Feb 2001.

[3] O. A. Alsayegh, "Short-term load forecasting using seasonal artificial neural networks," International Journal of Power and Energy Syst., vol. 23, no. 3, pp. 137-142, 2003.

[4] K. Lru, S. Subbarayan, R. R. Shoults, M. T. Manry, C. Kwan, F. L. lewis, and J. Naccarino, "Comparison of very short term load forecasting techniques," IEEE Trans. Power Syst., vol. 11, no. 2, pp. 877-882, May 1996.

[5] I. Drezga and S. Rahman,, "Short-term load forecastingwith local ANN predictors," IEEE Trans. Power Syst., vol. 14, no. 3, pp. 844–850, Aug 1999.

[6] D. Alex and C. Timothy, "A regression-based approach to short term system load forecasting," IEEE Trans. Power Syst., vol. 5, no. 4, pp. 1535–1550, Nov 1990.

[7] S. Rahman and O. Hazim, "A generalized knowledge-based short-term load forecasting technique", IEEE Trans. Power Syst., vol. 8, no. 2, pp. 508-514, May 1993.

[8] M. N. Kandil, R. Wamkeue, M. Saad, and S. Georges, "An efficient approach for short term load forecasting using artificial neural networks," International Journal of Electrical Power & Energy Systems, vol. 28, no. 8, pp. 525-530, Oct 2006.

[9] H. Y. Yang, H. Ye, G. Wang, J. Khan, and T. Hu, "Fuzzy neural very-short-term load forecasting based on chaotic dynamics reconstruction," Chaos, Solitons & Fractals, vol. 29, no. 2, pp. 462-469, Jul 2006.

[10] T. S. Dillon, S. Sestito, and S. Leung, "Short term load forecasting using an adaptive neural network," International Journal of Electrical Power & Energy Systems, vol. 13, no. 4, pp. 186-192, Aug 1991.

[11] P. Lauret, E. Fock, R. N. Randrianarivony, and J. Manicom-Ramsamy, "Bayesian neural network approach to short time load forecasting," Energy Conversion and Management, vol. 49, no. 5, pp. 1156-1166, May 2008.

[12] Z. Xiao, S. Ye, Bo Zhong, C. Sun, "BP neural network with rough set for short term load forecasting," Expert Systems with Applications, vol. 36, no. 1, pp. 273-279, Jan 2009.

[13] L. Mohan Saini, M. Kumar Soni, "Artificial neural network-based peak load forecasting using conjugate gradient methods," IEEE Trans. Power Syst., vol. 17, no. 3, pp. 907 - 912, Aug 2002.

[14] D. Srinivasan, S. S. Tan, C. S. Chang, and E. K. Chan, "Parallel neural network-fuzzy expert system for short-term load forecasting: System implementation and performance evaluation," IEEE Trans. Power Syst., vol. 14, no. 3, pp. 1100–1106, Aug 1999.

[15] C. Kim, I. Yu, and Y. H. Song, "Kohonen neural network and wavelet transform based approach to short-term load forecasting," Elect. Power Syst. Res., vol. 63, no. 3, pp. 169-176, 2002.

[16] S. C. Pandian, K. Duraiswamy, C. Christober, Asir Rajan, and N. Kanagaraj, "Fuzzy approach for short term load forecasting," Electric Power Systems Research, vol. 76, no. 6-7, pp. 541-548, Apr 2006.

[17] P. Pai and W. Hong, "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms," Electric Power Systems Research, vol. 74, no. 3, pp. 417-425, Jun 2005.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, vol. 4, pp. 1942–1948, 1995.

Md. Khairul Hasan was born in Rajshahi City, Bangladesh, on December 27, 1976. He received the B.Sc. degree in Computer Science and Engineering from Ahsanullah University of Science and Technology, Dhaka, Bangladesh, in 2000 and the M.Sc. degree in Computer Science and Engineering from United International University, Dhaka, Bangladesh in 2010. He is currently working as a faculty of the Computer Science and Engineering Department, Ahsanullah University of Science and Technology. His research interests include computational intelligence, neural networks, optimization and their applications.

Mohammad Asif A Khan received the B.Sc. degree in Computer Science and Engineering from Ahsanullah University of Science and Technology, Dhaka, Bangladesh, in 2001 and the M.Sc degree in Computer Science and Engineering from the United International University in May 2010. He served as a faculty in the department of Computer Science and Engineering at Ahsnaullah University of Science and Technology from April 2002 to April 2010. He is currently pursuing the M.Sc. degree in Computer Science at University of Saskatchewan, Saskatoon, Canada. His research interests include Software Engineering and their applications, optimization, Neural networks and Data Mining.

Suman Ahmmed received the B.Sc. and M.Sc. degrees from the Computer Science and Engineering Department at Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1997 and 2006, respectively. He is currently an Assistant Professor of Computer Science and Engineering and the Director of Student Affairs at United International University, Dhaka, Bangladesh. His research interests are load forecasting, optimization, neural networks, and optimization.

Ahmed Yousuf Saber received the B.Sc. and M.Sc. degrees from the Computer Science and Engineering Department at Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1999 and 2002, respectively. He obtained his Ph.D. degree in Electrical Engineering (Power Systems) from University of the Ryukyus, Okinawa, Japan in 2007. His research interests are computational intelligence in power and energy systems, v-to grid (V2G), and renewable energy.