International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:4, 2009

STLF Based on Optimized Neural Network Using PSO

H. Shayeghi, H. A. Shayanfar, G. Azimi

Abstract**—**The quality of short term load forecasting can improve the efficiency of planning and operation of electric utilities. Artificial Neural Networks (ANNs) are employed for nonlinear short term load forecasting owing to their powerful nonlinear mapping capabilities. At present, there is no systematic methodology for optimal design and training of an artificial neural network. One has often to resort to the trial and error approach. This paper describes the process of developing three layer feed-forward large neural networks for short-term load forecasting and then presents a heuristic search algorithm for performing an important task of this process, i.e. optimal networks structure design. Particle Swarm Optimization (PSO) is used to develop the optimum large neural network structure and connecting weights for one-day ahead electric load forecasting problem. PSO is a novel random optimization method based on swarm intelligence, which has more powerful ability of global optimization. Employing PSO algorithms on the design and training of ANNs allows the ANN architecture and parameters to be easily optimized. The proposed method is applied to STLF of the local utility. Data are clustered due to the differences in their characteristics. Special days are extracted from the normal training sets and handled separately. In this way, a solution is provided for all load types, including working days and weekends and special days. The experimental results show that the proposed method optimized by PSO can quicken the learning speed of the network and improve the forecasting precision compared with the conventional Back Propagation (BP) method. Moreover, it is not only simple to calculate, but also practical and effective. Also, it provides a greater degree of accuracy in many cases and gives lower percent errors all the time for STLF problem compared to BP method. Thus, it can be applied to automatically design an optimal load forecaster based on historical data.

Keywords**—**Large Neural Network, Short-Term Load Forecasting, Particle Swarm Optimization.

I. INTRODUCTION

THE quality of the short-term hourly load forecasting with lead times ranging from one hour to several days ahead has a significant impact on the efficiency of operation of any electrical utility. Many operating decisions are based on short-term load forecasts, such as economic allocation of generation, energy transaction and system security analysis, optimal energy interchange between utilities, unit commitment, control of spinning reserve and maintenance scheduling [1]-[2].

Short-Term Load Forecasting (STLF) is not an easy task. Load series are generally complex and the load at a certain hour depends on the loads from undetermined number of past hours. Moreover, weather variables such as temperature, daylight time, winds, humidity, etc. affect the consumption considerably [3]. Various forecasting techniques or methods, mainly including the time series model [4]-[5] multiple linear regression Auto Regressive Moving Average (ARMA) [6]-[7], and the like, have been introduced in the past few decades. However, these techniques can not properly represent the complex nonlinear relationships between the consumed load and a series of stochastic factors such as time of day, special events and correlation properties of weather condition which are highly affecting power load curve.

For the past several years, Artificial Neural Networks (ANNs) methods received a great deal of attention and are now being proposed as powerful computational tools to solve the load forecasting problem and able to give better performance in dealing with the nonlinearity and other difficulties in modeling time series forecasting [1]-[2], [8]-[12]. ANNs is able to extract an implicit nonlinear relationship among input variables (higher-order combination input) by learning from training data. Using trained supervised networks requires a measure of the discrepancy between the networks output value and desired value. The difference between these values results in an error function. It should be noted that one neural network will not be capable of handling all load types, several data clusters are formed. As resemblance measure, correlation analysis is selected. Since the past loads, temperature and time (hour, day, season, etc.) play the greatest roles in next day's load; they are used as the input variables to the proposed model [3].

For the solution of the STLF problem, a large artificial neural network intelligence approach based on day-type cluster is chosen in this paper. As the three-layer perceptron is the most common architecture. Thus, the ANN architecture for STLF is feed-forward three-layer perceptron (an input layer, a hidden layer and an output layer). Neural network forecasts are sufficiently good for weekdays and weekends; but, they have to be revised and modified for holidays. Thus, a new approach based on the shape of the daily load curves and correlation analysis on the available data is proposed for such cases. It should be noted that optimization of neural network architecture design, including selecting the number of input variables, input nodes and the number of hidden neurons, to improve forecasting performance is becoming more and more important and desirable. At present, there is no best method in determining the number of neurons for hidden layer. This number is usually found by using trial and error approach. For this reason, Particle Swarm Optimization (PSO) is employed to evolve the optimum large neural networks structure and connecting weights for one-day ahead electric load forecasting problem in this paper.

H. Shayeghi is with the Department of Technical Eng., University of Mohaghegh Ardabili, Ardabil, Iran (corresponding author to provide phone: 98-551-2910; fax: 98-551-2904; e-mail: hshayeghi@ gmail.com).
H. A. Shayanfar and G. Azimi are with the Center of Excellence for Power Automation and Operation, Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran.

In this study, the optimized large neural networks modeling approach using the PSO algorithm for short-term load forecasting based on only multiple delayed historical power load and temperature data is proposed. Unlike traditional neural network short-term load forecasting modeling approach, the interrelationship among the input variables and outputs of the neural network is considered. The input variables of the proposed neural network based model are historical load data and temperature. The outputs produced by this model are peak and hourly load values. The performance of the proposed method is also compared with the forecasting results of the back propagation based neural network model.

Currently, there have been many algorithms used to train the Feed-forward Neural Networks (FNNs), such as Back Propagation (BP) algorithm, Genetic Algorithm (GA) [13]-14], Simulating Annealing (SA) [3], Particle swarm optimization(PSO) algorithm [1], [8], [15].

Regarding the feed-forward neural networks (FNNs) training, the mostly used training algorithm is the back propagation (BP) algorithm, which is a gradient-based method. Although the BP algorithm has solved a number of practical problems, firstly, the BP algorithm will easily get trapped in local minima, so that back propagation may lead to failure in finding a global optimal solution. Second, the convergent speed of the BP algorithm is too slow even if the learning goal, a given termination error, can be achieved. The important problem to be stressed is that the convergent behavior of the BP algorithm depends very much on the choices of initial values of the network connection weights as well as the parameters in the algorithm such as the learning rate and momentum. To improve the performance of the original BP algorithm, researchers have concentrated on the following two factors: (1) selection of better energy function; (2) selection of dynamic learning rate and momentum. However, these improvements haven't removed the disadvantages of the BP algorithm getting trapped into local optima in essence. In particular, with FNN's structure becoming more complex, its convergent speed will be even slower [15]. In order to overcome some of these drawbacks, the evolutionary algorithms such as genetic algorithm [14, 16] and particle swarm optimization [15] based design of neural networks method have been proposed. Since they are heuristic and stochastic, they are less likely to get stuck in local minimum, and they are based on populations made up of individuals with a specified behavior similar to biological phenomenon [16].

Most of evolutionary techniques have the following procedure:

1. Random generation of an initial population
2. Reckoning of a fitness value for each subject. It will directly depend on the distance to the optimum.
3. Reproduction of the population based on fitness values.
4. If requirements are met, then stop. Otherwise go back to 2.

In our pervious paper [3] a Continues Genetic Algorithm (CGA) technique based on the ANN method have been proposed for STLF problem solution. GA is more likely to find the global solution of a given problem. In addition, they use only a simple scalar performance measure that does not require or use derivative information, so they are general-purpose optimization methods for solving search problems. Two major primary areas in which GAs have been employed are optimization and machine learning. In machine learning, GAs have been successfully applied to the learning of neural networks [12], [14]. Although, CGA seems to be good methods to solve optimization problems, when applied to problems consisting of more number of local optima, the solution from CGA are just near global optimum areas. Also, it takes long simulation time to obtain the solution. Moreover, when the number of parameter is more, optimization problem is complex and coding chromosomes with more gens for increasing algorithm accuracy is caused CGA convergent speed will become very slow, so that convergent accuracy may be influenced by the slow convergent speed. Here, to overcome these drawbacks, a PSO based design of neural networks is proposed for STLF problem. PSO is a novel population based metaheuristic, which utilize the swarm intelligence generated by the cooperation and competition between the particle in a swarm and has emerged as a useful tool for engineering optimization [17]. Unlike the GA, the PSO algorithm has no complicated evolutionary operators such as crossover and mutation. PSO algorithm can be used to solve many of the same kinds of problems as GA and does not suffer from of GA's difficulties [17]. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases. Successful applications of PSO to several optimization problems, like function minimization [18] and feed forward neural network design [19]-[20], have demonstrated its potential. The PSO algorithm has a strong ability to find the most optimistic result. However, because the PSO algorithm has several parameters to be adjusted by empirical approach, if these parameters are not appropriately set, the search will be come very slow near the global optimum [15]. In this study, PSO is used to evolve the optimum neural network structure and weights for one-day ahead electric load forecasting problem. This method is proposed to improve optimization synthesis such that the global optima are guaranteed and the speed of algorithms convergence is extremely improved, too. According to the performance criteria, particle swarm optimization algorithm is used to globally optimization the large neural networks architecture. The optimization process determines the number of neurons in the hidden layer. The proposed method is applied to STLF of the local utility. The simulation results show that the proposed PSO based method provides a greater degree of accuracy in many cases for STLF problem, compared to the BP method. Moreover, it gives lower Mean Absolute Percentage Error (MAPE). Thus, it can be applied to automatically design an optimal load forecaster based on historical data.

II. Data Analysis and Preprocessing

The available data for this research are total hourly actual loads of a local utility in Iran for the years 2000 to 2003. In order to use these data in a meaningful and logical manner, first of all they should be closely analyzed and their dynamics should be clearly understood. Then, they can be clustered into smaller sets according to some common characteristics and separate models can be built for each cluster. This is necessary because it has always been emphasized in the literature that it is impossible to reflect every different type of load behavior

with a single model. The load profile is a dynamic process. Temporal variations, abrupt increases in demand, outages or other random disturbances all affect the load level. It is noteworthy that load shapes for the same weekday for different weeks are quite similar; and that load shapes for different weekdays for the same week are roughly similar. Fig 1 shows hourly load curves for a sample week. This graph gives an idea about how the electric load varies from hour to hour and day to day. It is seen that four working days (Sunday-Wednesday) have very similar patterns and Saturday (first working day of the week in Iranian calendar) is slightly different from the other workings days. Also, Weekend days, i.e. Thursday and Friday are different from the other days [3].

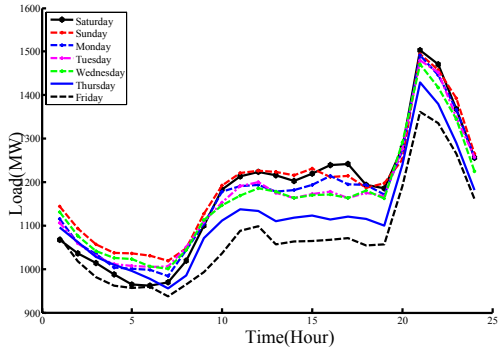


Fig. 1. Daily load curves for a typical week, Saturday, Sunday, to Wednesday, Thursday and Friday

A. Correlation analysis

If the training set of a neural network contains patterns that have characteristics close to each other and the output carries the same kind of information as the inputs then this model gives successful result. In order to evaluate the validity of this hypothesis, a measure of the resemblance between daily load sequences is thought to be established. For this reason, the correlation function is taken into consideration. Cross correlation coefficients are computed for each data pair as follows:

$$C_{xy} = \frac{S_{xy}}{\sqrt{S_{xx} S_{yy}}} \quad (1)$$

Where,

$$S_{xy} = \sum_{i=1}^{n} |x_i - \bar{x}||y_i - \bar{y}|, \qquad S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

$$S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

and x and y represent the data pairs, $\bar{x}$ and $\bar{y}$ are the mean values calculated over the samples and n is number of samples.

Table I summarizes the correlations of the daily electric load consumptions in year 2003. Column ($D+i$) represents the ith day after the day D, given row-wise. It can be seen that, weekdays highly correlated with each other; but Thursday and Friday have lower correlations with each other and with weekdays. Saturday is the day which has the lowest correlations with the other weekdays.

B. Data clustering

Based on the shape of the daily load curves and correlation analysis on the available data, an efficient clustering can be done. First of all, religious and national holidays should be excluded from the regular day data and handled separately since their characteristics are completely different. Thus, four weekdays (Sunday-Wednesday) can be examined in the same group. It does not seem necessary to create a distinct group for each of these weekdays as they are highly correlated. Moreover, a separate group should be formed for the first working day (Saturday), because they come just after the weekend and do not resemble the other weekdays. For weekends, two groups should be formed as Thursdays and Fridays, since they have unique characteristics [3].

TABLE I
DAILY LOAD CORRELATIONS IN YEAR 2003

Day	D+1	D+2	D+3	D+4	D+5	D+6	D+7	D+14
Saturday	0.9879	0.9904	0.9821	0.9750	0.9636	0.9411	0.9905	0.9908
Sunday	0.9937	0.9895	0.9858	0.9802	0.9633	0.9879	0.9937	-
Monday	0.9953	0.9930	0.9845	0.9612	0.9904	0.9937	0.9954	-
Tuesday	0.9950	0.9876	0.9620	0.9821	0.9895	0.9953	0.9902	-
Wednesday	0.9852	0.9652	0.9750	0.9858	0.9930	0.9950	0.9900	-
Thursday	0.9793	0.9636	0.9802	0.9845	0.9876	0.9852	0.9893	0.9872
Friday	0.9411	0.9633	0.9612	0.9620	0.9652	0.9793	0.9773	0.9575

III. ARCHITECTURE OF NEURAL NETWORK BASED STLF

In a supervised leaning ANN, a feed-forward multi-layered perceptron neural networks is widely used, and many enhancements have been explored. The partitioning method is one of the enhancements. It was developed because of differences in the load shape for every season (see Fig. 2) and every day (vide Fig. 1). The partitioning method divides the network into several sub-networks. In this study, the network is divided into the following groups: Sunday through Wednesday, Thursday, Friday and Saturday. Moreover, for partitioning a year, is divided into four seasons (spring, summer, fall and winter), and every season is divided into three different kinds of day (Saturday, weekday and weekend). Fig. 2 depicts the hourly loads for each season. The highest load would occur in summer. The time of peek load in each season is difference. [3].

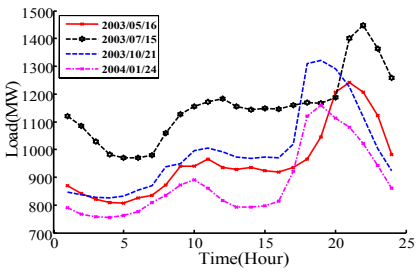


Fig. 2. Daily load curves for a typical day for each season

It should be noted that would be better to distinguish between the seasons by using different ANN modules.

Accordingly, the training would be easier and there is a chance to have better results. Thus, four ANN modules for summer, winter, spring and fall is used in STLF problem. In the training process, every network is only supplied by data on that particular season.

Four seasonal networks have the same architecture, which is a three-layer feed-forward neural network. For every season, the number of neurons in the input or the output layer is already fixed, based on the input and output data chosen. But the number of neurons in the hidden layer is different and here, is obtained using the PSO based method.

A. Proposed ANN Architecture

There are several nontrivial tasks associated with the design of a neural network based load forecaster. One such task is the selection of input variables and the network structure that would secure an acceptable forecast accuracy and network training time. For example, a network with too many hidden neurons will memorize the training data instead of learning general relationships and will perform poorly while applied to new data. Training data extraction and design of an efficient and reliable learning algorithm are also of great importance. At present, there is no systematic methodology for optimal design and training of an artificial neural network. One has often to resort to the trial and error approach. This paper summarizes the optimal design of a neural network based load forecaster. A systematic approach to solving these problems is proposed. It can be applied to the automatic design of an optimal forecaster based on the available historical data [11].

The process of developing an artificial neural network based on load forecasting can be divided into 5 steps [3]:

1. Selection of input variables.
2. Design of neural network structure.
3. Extraction of training, test and validation data.
4. Training of the designed neural networks.
5. Validation of the trained neural networks.

A.1. Selection of input variables

Neural network input variables are selected from load affecting factors. One of the keys for designing a good architecture in ANN is choosing appropriate input variables from load affecting factors. Those factors may vary from one utility to another based on the load characteristics. However, there are several factors that are commonly used. In the case of short-term load forecasting problem, these inputs can be divided into time, electrical load and weather information. The time information may include the type of season, days of a week, and hours of a day. The load information may include previous loads. The weather information may include previous and future temperatures, cloud cover, thunderstorm, humidity, and rain. As shown in Fig. 1, load changes during the day from one hour to another and from one day to another during the week. On the other hand, 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 and on the load at the same hour on the day with the same denomination in the previous week.

Until now, there have been no general regulations on input types in designing the ANN for STLF problem. However, as a matter of principle, historical load and temperature represent the most important inputs. For a normal climate area, these two inputs and other related inputs (e.g., time) would be sufficient to make a good short-time load forecasting model. However, for extreme weather conditions in humid areas or in areas with many thunderstorms, additional weather factors should be included for forecasting.

In the proposed architecture, ANN is designed based on previous loads, type of season, type of day, hours of a day, previous day's temperature and temperature forecast. Only two weather factors are used in this architecture, since the area of the forecasted load is a normal climate area.

A block diagram for the proposed ANN architecture is shown in Fig. 3.

The numbers of neurons in the input and output layers are determined by the number of input and output variables respectively. The nodes in the input layer are used for the distribution of the inputs to the hidden-layer neurons and are not actual neurons. There are a total of 120 neurons in the input layer. The details of ANN input variables for groups Sunday through Wednesday, Thursday, Friday and Saturday are given as follows, and the variables are selected according to the discussions as mentioned in Sec. 2 and by trial and error.

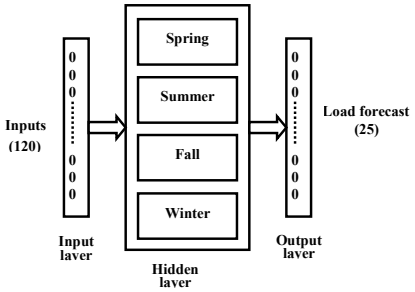


Fig. 3. Block diagram of the proposed ANN Architecture

A. The ANN input variables for group Sunday through Wednesday:
   1. The first 24 input neurons represent hourly scaled load values of the previous day.
   2. The next 48 neurons are used to capture the effect of temperature: hourly temperature data from previous day and hourly temperature data for the day of forecast.
   3. The next 24 input neurons represent hourly scaled load values of the two previous day.
   4. The next 24 input neurons represent hourly scaled load values of the same day in the previous week.

B. The ANN input variables for groups Thursday, Friday and Saturday:
   1. The first 24 input neurons represent hourly scaled load values of the same day in the previous week.
   2. The next 48 neurons are used to define the effect of temperature: The first 24 neurons represent hourly scaled temperature data of the same day in the previous week and another 24 neurons represent hourly load temperature data of the next day (the day of forecast).

3. The next 48 neurons represent hourly scaled load values of the same day in the two and three previous week.

A. 2 Design of neural network structure

To design a multi-layer feed-forward network, one needs to select the number of hidden layers, type of connection between the layers, number of neurons in each layer, and a neuron's activation function. In practice, a fully connected network with one hidden layer is a reasonable choice. Thus, three-layered perceptron, that has proved its good performance, is used in this application. According to the discussions as mentioned in Sec. 2, a separate ANN model is designed for each of the four-day classes. Each network has 120 neurons in input layer and its output layer consists of 25 neurons; the first 24 neurons, each represent the predicted hourly load covering 24 hours of day and 25th neuron represent the predicted maximum load of day. To design a three-layered perceptron network, one needs to select the number of neurons in hidden layer and neuron's activation function. Good candidates for an activation function are sigmoid (S-shaped) functions. The exact shape of the sigmoid function has little effect on the network performance. It may have a significant impact on the training speed. In this work, the output and hidden layers have sigmoid activation function in order to eliminate additional errors for extreme forecasts due to the saturation of the activation function.

The number of neurons in the hidden layer determines the network's learning capabilities and its selection is the key issue in optimal network structure design. If the number is too small, the network cannot find the complex relationship between input and output and may have difficulty in convergence during training. If the number is too large, the training process would take longer and could harm the capability of ANN. It would vary for different applications and could usually depend on the size of the training set and the number of input variables. The hidden layer size and its neurons number are selected either arbitrarily or based on the trial and error approach. In this study, a particle swarm optimization based method is proposed for find the optimal number of the hidden layer neurons. The flowchart of the proposed method is shown in Fig. 4. The simulation results for the optimal number of neurons in the hidden layer based on the mean absolute percentage error (MAPE) performance index for each of the four-day classes in the Spring is shown in Fig. 5.

A.3 Extraction of training, test and validation data

Collecting training data is very important to achieve the desired level of ANN performance to STLF problem. It should be noted that for the network updating a few patterns is required if the numbers of training data are much. Moreover, to assure a good network performance, the training data should be representative and it is normalized. A normalization step helps in preventing the simulated neurons from being driven too far into saturation. Through transformation the data of the input and the output of neural network are limited to the interval $[\alpha,\beta]$, $\alpha \neq \beta$, where $0 \leq \alpha,\beta \leq 1$. In this study the values of α and β are 0.1 and 0.8, respectively. In this way, the training convergence speed can be increased and the overflow of calculation can be avoided. Data normalization can be calculated by the following equations:

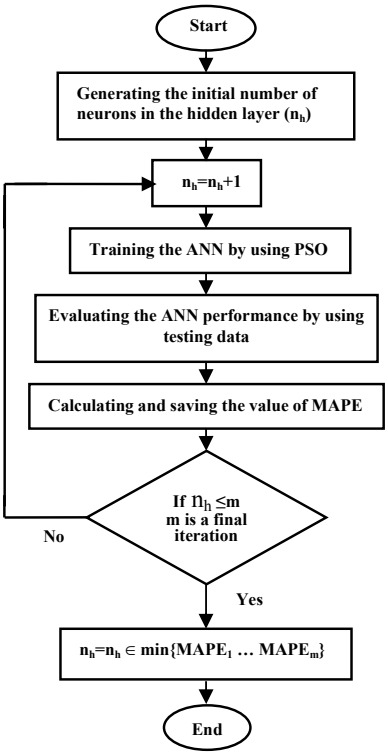


Fig. 4. The flowchart of the proposed method for selection of the number of hidden layer neurons

$$a_i=(\beta-\alpha)/(\max(x_i)-\min(x_i)) \text{ , } b_i=\beta-a_i\times \max(x_i) \quad (2)$$

$$X^N_{i,j}= a_i\times x_{i,j} +b_i \quad (3)$$

Where, $x_{i,j}$ and $X^N_{i,j}$ are the actual hourly temperature/load and the normalized value of the ith day at the jth hour, respectively; also $X^N_{i,j}$ is the input data of input nodes of neural network; $\max(x_i)$, $\min(x_i)$ refer to the maximum temperature/load and minimum temperature/load of the ith day, respectively. The training output values are also normalized in the same manner.

The data that were used for training, testing and validating of the ANN was total hourly actual loads and weather data of a local utility in Iran for the years 2000 to 2003. All data are divided into four parts based on the type of season. The data for each season are divided again into four parts based on day cluster. The data for each cluster are divided into three parts as training, test and validate sets. Test and validate sets will not be used for training; their purpose is only to examine errors produced by ANN after training [3].

A.4 Training of the designed neural networks

Training of a neural network is a process of determining the network parameters (weights) in order to achieve the desired objective based on the set of examples called the training set. The use of ANN for solution of the STLF problem can be broken down into two groups based on learning strategies:

supervised and unsupervised learning. Supervised learning is based on direct comparison between the inputs and outputs. This is usually formulated as the minimization of an error function such as the total mean square error between the actual output and the desired output summed over all available data. The unsupervised learning is solely based on the correlations among input data. No information on "correct output" is available for learning. Most applications use a supervised learning ANN.

The most commonly used training method is known as the back propagation algorithm [10]-[12]. This algorithm is an iterative procedure. It has some drawbacks. Some of them are the slowness of learning speed, possibility of falling into local minimum and the necessity of adjusting a learning constant in every application [3]. In order overcome these drawbacks PSO based design of neural networks has been proposed.

A.5 Validation of the trained neural network

Since acceptable training errors do not always guarantee similar network performance for a different set of data, for example due to the lack of representativeness of the training set or the improperly selected network size, it is necessary to validate the network performance after it is trained. This is usually done by randomly selecting 10-20% of the total training data and setting it aside for testing. Based on the above discussions, test and validation data is randomly extracted by selecting 20% and 10% from the entire training data, respectively and the rest of entire data (about 70%) is used for the networks training.

If the testing errors are unacceptable, possible causes should be identified and corrected, and the network should be retrained. The old test set should be included into the training set. If new relevant data is available, a new test set should be collected. Otherwise, a new test set is again randomly extracted from the entire training data.

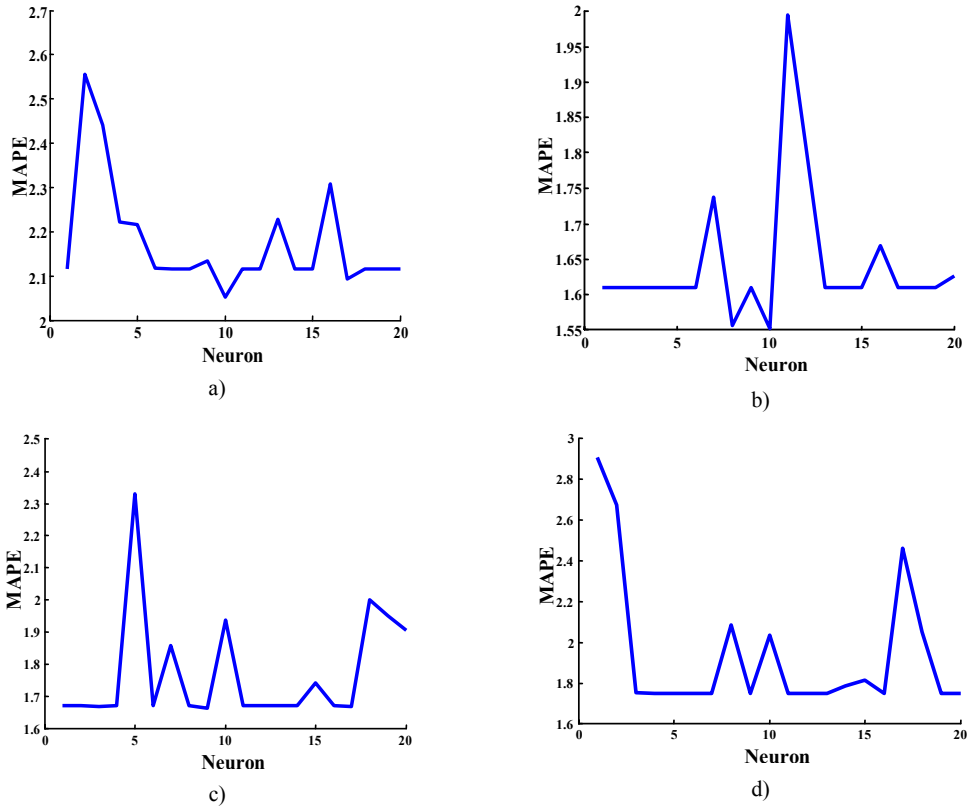


Fig. 5. The optimal number of neurons based on the MAPE performance index for each of the four-day classes
a) Saturday b) Sunday-Wednesday c) Thursday d) Friday

IV. PSO Evolve Neural networkS

In this section, the proposed neural network is employed to learn the input-output relationship of an application using the BP and PSO.

The particle swarm optimization is an evolutionary computation technique developed by Eberhart and Kennedy [21], inspired behavior of bird flocking. The PSO is a kind of algorithm to search for the best solution by simulating the movement and flocking of birds. The algorithm works by initializing a flock of birds randomly over the searching space, where every bird is called as a "particle". These "particles" fly with a certain velocity and find the global best position after some iteration. At each iteration, each particle can adjust its velocity vector, based on its momentum and the influence of

its best position as well as the best position of its neighbors, and then compute a new position that the "particle" is to fly to [15]. Some of the attractive features of the PSO include ease of implementation and the fact that no gradient information is required. It can be used to solve a wide array of different optimization problems; some example applications include neural network training and function minimization.

The PSO definition is presented as follow:

1) Each individual particle i has the following properties: a current position in search space w_i, a current velocity v_i, and a personal best position in search space p_i.
2) The personal best position p_i corresponds to the position in search space, where particle i presents the smallest error as determined by the objective function f, assuming a minimization task.
3) The global best position denoted by $\tilde{p}$ represents the position yielding the lowest error among all the p_i's.

(4) and (5) define how the personal and global best values are updated at time, respectively. It is assumed below that the swarm consists of particles. Thus, $i \in 1, \cdots, s$.

$$p_i(t+1) = \begin{cases} p_i(t), & if \quad f(p_i(t)) \leq f(w_i(t+1)) \\ w_i(t+1), & if \quad f(p_i(t)) > f(w_i(t+1)) \end{cases} \quad (4)$$

$$\tilde{p}(t) \in \{p_0(t), p_1(t), \cdots, p_s(t)\} \text{ and}$$
$$\tilde{p}(t) = \min\{f(p_0(t)), f(p_1(t)), \cdots, f(p_s(t))\} \quad (5)$$

During each iteration, every particle in the swarm is updated using (6) and (7). Tow pseudorandom sequences $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$ are used to affect the stochastic nature of the algorithm. For all dimensions $j \in 1 \cdots n$, let $w_{i,j}$, $p_{i,j}$ and $v_{i,j}$ be the current position, current personal best position, and velocity of the j^{th} dimension of the i^{th} particle. The velocity update step is

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + c_1 r_{1,i}(t)(p_{i,j}(t) - w_{i,j}(t)) + c_2 r_{2,i}(t)(\tilde{p}_{i,j}(t) - w_{i,j}(t)) \quad (6)$$

The new velocity is then added to the current position of the particle to obtain its next position as follows:

$$w_i(t+1) = w_i(t) + v_i(t+1) \quad (7)$$

The value of each dimension of every velocity vector v_i is clamped to the range $[-v_{max}, v_{max}]$ to reduce the likeli- hood of the particle leaving the search space. The value of v_{max} is usually chosen to be $v_{max} = k \times w_{max}$, where $0.1 \leq k \leq 1$ and w_{max} denotes the domain of the search space. Note that this does not restrict the value w_i of to the range $[-v_{max}, v_{max}]$. Rather than that, it merely limits the maximum distance that a particle will move.

The acceleration coefficients c_1 and c_2 control how far a particle will move in a single iteration. Typically, these are both set to a value of 2, although it has been shown that setting $c_1 \neq c_2$ can lead to a good performance. The inertia weight ω in (6) is used to control the convergence behavior of the PSO. Small values of ω result in more rapid convergence usually on a suboptimal position, while a too large value may prevent divergence. Typical implementations of the PSO adapt the value of during the training stage, e.g., linearly decreasing it from 1.0 to near 0 over the execution. Convergence can be obtained with fixed values. In general, the inertia weight ω is set according to the following equation:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{iter_{\max}} . iter \quad (8)$$

Where, $iter_{max}$ is the maximum number of iterations, and $iter$ is the current iteration number.The PSO system combines two models: a social-only model and the cognition-only model. These models are represented by the velocity update, shown in (6). The second term in the velocity update equation is associated with cognition since it only takes into account the particle's own experiences. The third term in the velocity update equation represents the social interaction between the particles. It suggests that individuals ignore their own experience and adjust their behavior according to the successful beliefs of individuals in the neighborhood [22]. The detailed flowchart of the implementation of PSO algorithm for training and testing the proposed ANN architectures is shown in Fig. 6.

The fitness of the i^{th} particle is expressed in term of an output mean squared error (MSE) of the neural networks as follows:

$$f(w_i) = \frac{1}{N} \sum_{k=1}^{N} [\frac{1}{O} \sum_{l=1}^{O} \{T_{kl} - P_{kl}(w_i)\}^2] \quad (9)$$

Where, f is the fitness value, T_{kl} is the target output; P_{kl} is the predicted output based on w_i N is the number of training set samples; and, O is the number of output neurons. The (9) serves as the subjective function for identifying suitable parameters for use in the PSO model.

V. Handling the Special Days

In Iran, there are tow kinds of holidays, national and religious. National holidays are fixed in time, but religious holidays are moving each year. Load curve of holidays differ from a typical weekday, also number of these days in historical information in comparison with typical weekdays is less. It is important to forecast the loads of such days as well, in order to have a complete model. It is known that electric consumption decreases on holidays, as shown in Fig. 7. If the neural networks, designed for regular load forecasting, are directly used for special day load forecasting, large errors are observed. Thus, they should be analyzed separately.

One exception can be done to single day holidays that coincide to Fridays. They are not so much different than the regular Friday data, therefore, there is no need to form a cluster for this kind of data; instead, they can be put into the Friday training set.

In this study, the entire load patterns (from 1998 to 2003 years) of special days are classified into number of holidays. Then, a separate ANN model is used for each holiday. A three layer feed-forward neural network is used for each of holidays and by using particle swarm optimization algorithm, the number of hidden layers is found. ANN architectures are as follow:

- *Input variables:* As seen from Table 2 and Fig. 8 special days have highly correlations and highly similarity daily load curves with last Friday and the same special day in the previous year. But, correlation and similarity daily load curves special day relative the previous day depends to the

special day-type. By the above analysis, the input variables are as follow:

- 24 hourly scaled load values and their temperatures data of the previous day (48 units).
- 24 hourly scaled load values of the previous Friday (24 units).
- 24 hourly scaled load values of the same special day in the previous year (24 units).
- 24 hourly scaled temperatures data of the special day in the current year (forecast day) (24 units).

- *Output variables:* 24 hourly scaled load values and maximum scaled load values of the special day in the current year (25 units).

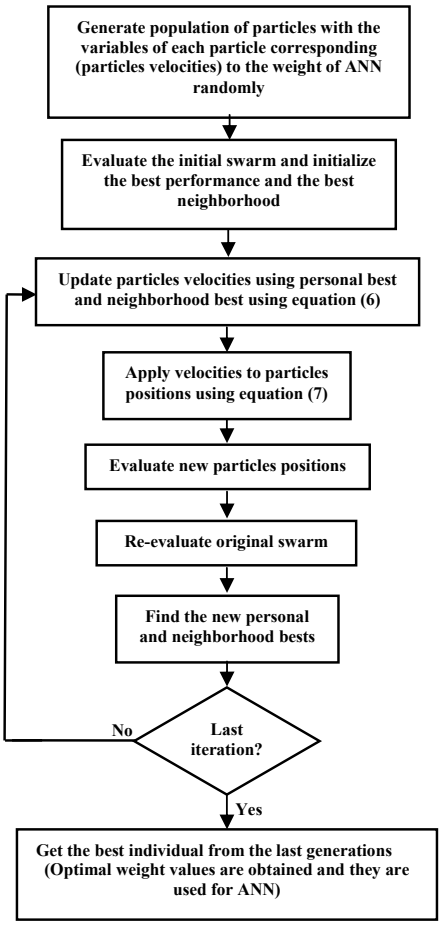


Fig. 6. Flowchart for training and testing the ANN using PSO algorithm.

TABLE II
SPECIAL DAILY LOAD CORRELATIONS IN YEAR 2003

Special days (Day of forecast)	Previous day	Previous Friday	The same special day in the previous year
Arbein (Religious holiday)	0.9420	0.9926	0.9789
14 Khordad (National holiday)	0.9727	0.9902	0.9916

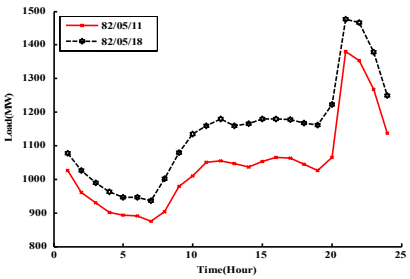


Fig. 7. Comparison of daily load curves for a special day (religious holiday-Saturday 2003.08.02) and the same day in the next week (regular day-Saturday 2003.08.09).

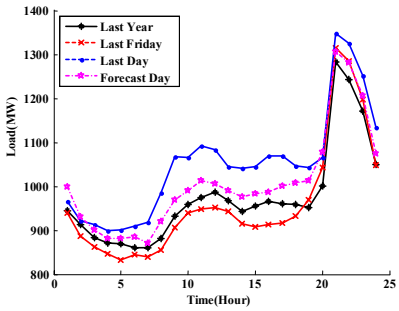


Fig. 8. Daily load curves for a special day, previous day before Friday and the same special day in the previous year

V. SIMULATION RESULT AND EVALUATION

To evaluate, the performance of the proposed neural networks is tested on real data of a local utility in Iran power system. Four and six years of historical data is used to train the regular days and special day neural network based STLF model, respectively. Actual weather data is used. The evaluation of the forecasting accuracy is accomplished by a MAPE and Absolute Percentage Error (APE), which are given by:

$$MAPE = \frac{1}{N}\sum_{1}^{N}\left|\frac{Actual_i - Forecast_i}{Actual_i}\right| \times 100 \quad (10)$$

$$APE = \left|\frac{Actual_i - Forecast_i}{Actual_i}\right| \times 100 \quad (11)$$

Where, N is the total number of hours, $Actual_i$ and $forecast_i$ are the actual load and forecasting load at hour I, respectively. The mean squared error is used as the fitness function that is optimized by the PSO and BP algorithms.

To demonstrate the effectiveness of the proposed ANN based model for solution of the STLF problem, some simulations are carried out. ANN model is designed for each of four-day classes and special days, we ran the two training algorithms (PSO and BP), respectively. Samples from the evaluation results are shown in Figs. 9 and 10 for four-day classes. Fig. 9 shows the actual and forecasted daily load. Fig. 10 depicts the APE. Also, Table 3 summaries the MAPE for different day classes and Table 4 summaries the APE of maximum daily load for different day classes. Fig. 11 shows the actual and forecasted daily load of two typical of special day (Arbein and 14 Khordad). Fig. 12 depicts the absolute percentage error of two typical of special day. Also Table 5

summaries the mean absolute percentage errors of two aforesaid special days with PSO and BP training methods. From the figures, it can be seen that the PSO-based STLF is very close to the actual load compared with the BP method and achieves higher accuracy than the BP algorithm. Thus, the PSO approach is more effective and economical, and can effectively improve the forecasting precision, and has very less the forecasted load errors.

TABLE V
MAPEs OF TWO SPECIAL DAYS WITH PSO AND BP ALGORITHMS

Special days (Day of forecast)	MAPE	
	BP	PSO
Arbein (Religious holiday)	2.039	1.76
14 Khordad (National holiday)	2.033	2.013

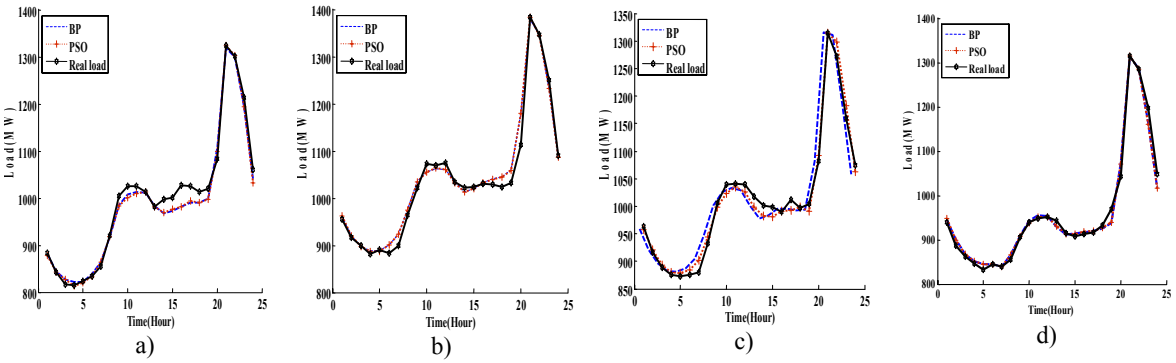


Fig. 9. ANN forecasted loads for sample four-day classes: (a) Saturday (b) Remaining working days (c) Thursday (d) Friday

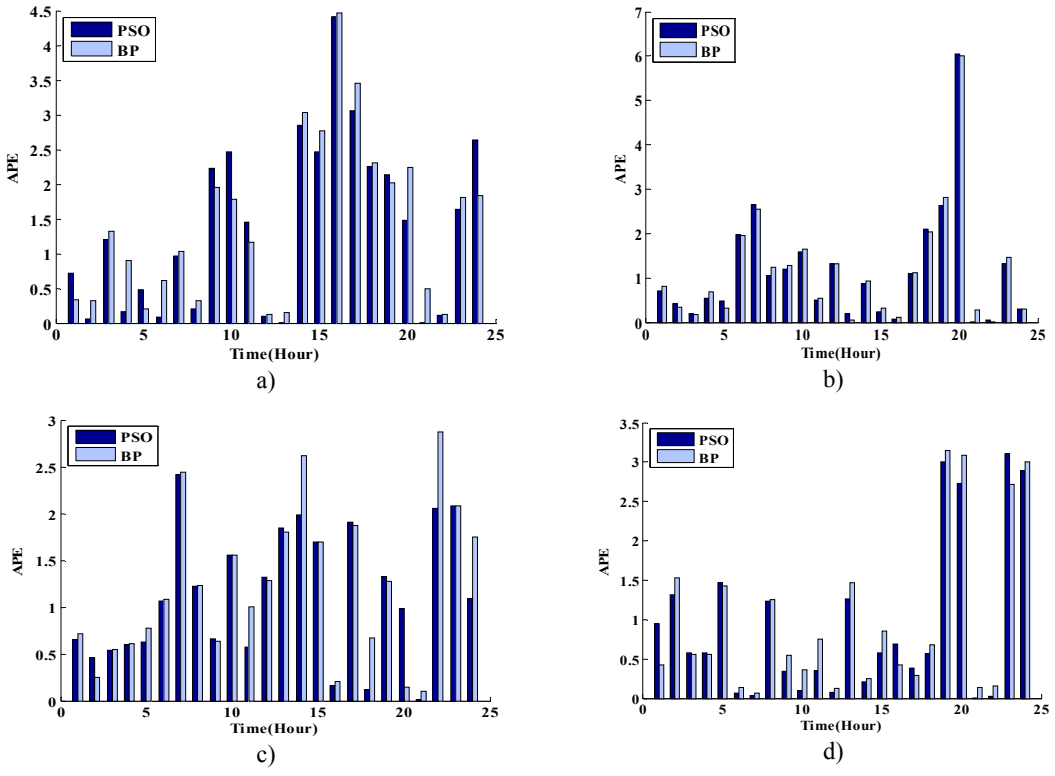


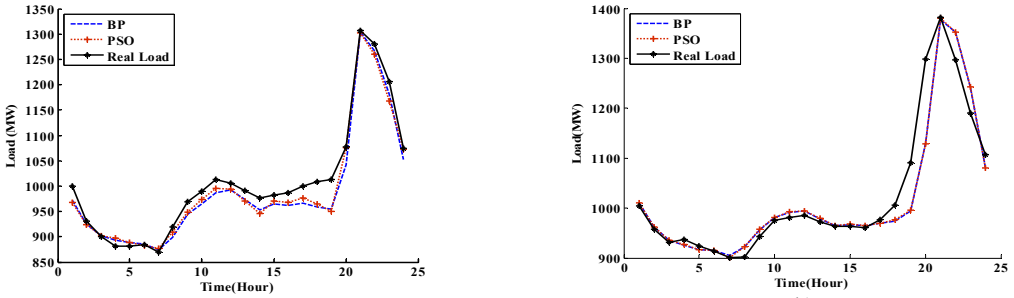
Fig. 10. Forecasted APE for sample four-day classes: (a) Saturday (b) Remaining working days (c) Thursday (d) Friday

TABLE III
MAPE BY PSO AND BP ALGORITHMS FOR DIFFERENT DAY CLASSES

Day-classes		MAPE	
		BP	PSO
Class 1	Saturdays	1.45	1.383
Class 2 (Remaining working days)	Sundays	1.647	1.636
	Mondays	1.8017	1.76
	Tuesdays	1.595	1.556
	Wednesdays	1.1803	1.148
Class 3	Thursdays	1.218	1.124
Class 4	Fridays	1.0	0.938

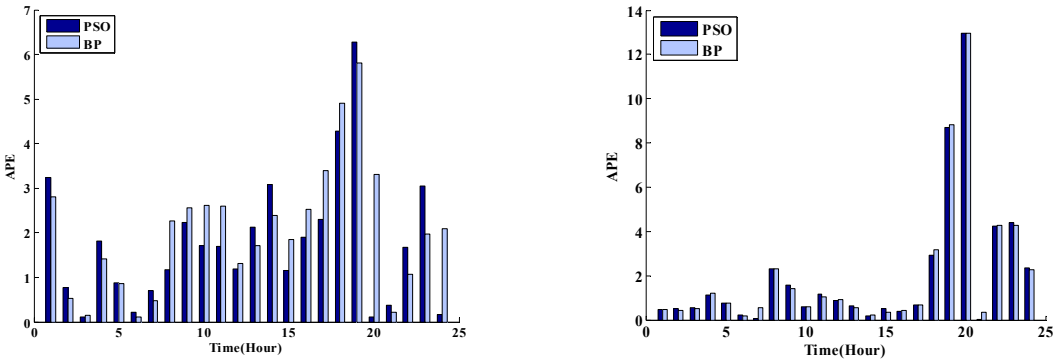
TABLE IV
APE OF MAXIMUM DAILY LOAD FOR DIFFERENT DAY CLASSES

Day-classes		APE	
		BP	PSO
Class 1	Saturdays	0.497	0.00964
Class 2 (Remaining Working days)	Sundays	0.275	0.00039
	Mondays	0.3139	0.00041
	Tuesdays	0.2764	0.00038
	Wednesdays	0.284	0.0004
Class 3	Thursdays	0.10	0.0114
Class 4	Fridays	0.1416	0.0



a) b)

Fig. 11. ANN forecasted loads for sample of two special days: (a) 14 Khordad (b) Arbein.



a) b)

Fig. 12. APE of two typical of special days: (a) 14 Khordad (b) Arbein.

X. CONCLUSIONS

This paper describes the process of developing three layer feed-forward large neural networks for short-term load forecasting based on PSO technique. It is based on clustering data analysis and correlation measures. Clustering is performed after a detailed data analysis, based on correlation measures, daily and seasonal variations, holiday behaviors, etc. Then separate large neural network models are constructed for each cluster. Currently, there is no systematic methodology for optimal design and training of an artificial neural network for short-term load forecasting. For this reason, PSO algorithm is employed to find the optimum large neural networks structure and connecting weights for one-day ahead electric load forecasting problem in this paper. The PSO technique can generate high-quality solutions within shorter calculation time and stable convergence characteristic than other stochastic methods. It exhibits significant improvement in computational efficiency.

This study also presented the research work conducted to improve the short-term load forecasting for special days in anomalous load conditions, which was a difficult task using conventional methods.

The results show that the proposed ANN-based model not only is effective in reaching proper load forecast but also it can be applied to the automatic design of an optimal forecaster

based on the available historical data. Also, it has easy implementation and good performance. The model performance evaluation in terms of 'MAPE' and APE indices reveals that the proposed PSO based ANN model produces lower prediction error and is superior to the BP based ANN method. Thus, the proposed forecasting methods could be provided a considerable improvement of the forecasting accuracy for the regular and the special days and it is recommended as a promising approach for the solution of the STLF problem.

REFERENCES

[1] A.-U. Asar, S.R.-U. Hassnain, A. Khan, "Short-term load forecasting using particle swarm optimization based ANN approach", *Proc. of IEEE Int. Joint Conf. on Neural Networks,* Orlando, Florida, USA, 2007, pp. 6.

[2] K.-H. Kim, H.-S. Youn, Y.-C. Kang, "Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference method", *IEEE Trans. on Power Systems*, vol. 15, No. 2, 2000, pp. 559-565.

[3] H. Shayeghi, H. A. Shayanfar, G. Azimi, Intelligent neural network based STLF", *Int. J. of Intelligent Systems and Technologies*, vol. 4., No. 1, 2009, pp. 17-27.

[4] G. Box, G.M. Jenkins, "*Time series analysis, forecasting and control*", San Francisco: Holden-Day; 1970.

[5] J.H. Park, Y.M. Park, K.Y. Lee, "Composite modeling for adaptive short-term load forecasting", *IEEE Trans. on Power Systems*, vol. 6, 1991, pp. 450-457.

[6] A. D. Papalexopoulos, T. C. Hesterberg, "A regression-based approach to short-term system load forecasting", *IEEE Trans on Power Systems*, vol. 4, No. 4, 1990, pp. 1535-1547.

[7] Z. Baharudin and N. Kamel, "Autoregressive method in short term load forecast", *in Proc. of 2nd IEEE Int. Conf. on Power and Energy (PECON 08),* Johor Baharu, Malaysia, 2008, pp. 1603-1608.

[8] Z. A. Bashir and M. E. El-Hawary, "Short-term load forecasting using artificial neural network based on particle swarm optimization algorithm", *in Proc. of IEEE Electrical and Computer Engineering Canadian Conf.,* 2007, pp. 272-275.

[9] T. W. S. Chow, C. T. Leung, "Neural networks based short term load forecasting using weather compensation", *in Proc. of IEEE Conf.*, 1996, pp. 1736-1742.

[10] R. Afkhami, F. Mosalman Yazdi, "Application of neural networks for short-term load forecasting", *in Proc. of IEEE Power India Conf.,* 2006, pp. 5.

[11] W. Charytoniuk, M.S. Chen, "Neural network design short-term load forecasting", *in Proc. of IEEE Int. Conf. on Electric Utility Deregulation and Restructuring and Power Technologies*, City University, London, 2000, pp. 554-561.

[12] E. Banda and K. A. Folly, "Short-term load forecasting using artificial neural network", *in Proc. of IEEE Power Technology Conf.,* Lausanne, 2007, pp. 108-112.

[13] G.-C. Liao, T.-P. Tsao, "Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting", *IEEE Trans. on Evolutionary. Computation,* vol. 10, No. 3, 2006, pp. 330-340.

[14] S. H. Ling, F. H. F. Leung, H. K. Lam, Y.-S. Lee, P. K. S. Tam, "A novel genetic-algorithm-based neural network for short-term load forecasting", *IEEE Trans. on Industrial Electronics*, vol. 50, No. 4, 2003, pp. 793-799.

[15] J.-R. Zhang, J. Zhang, T.-M. Lok, M. R. Lyu, "A hybrid particle swarm optimization-back propagation algorithm for feed-forward neural network training", *Applied Mathematics and Computation*, vol. 185, 2007, pp. 1026-1037.

[16] C.-F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design", *IEEE Trans. on systems, Man and Cybernetics-Part B: Cybernetics,* vol. 34, No.2, 2004, pp. 997-1006.

[17] H. Shayeghi, A. Jalili and H. A. Shayanfar, "Multi-stage fuzzy load frequency control using PSO", *Energy Conversion and Management,* vol. 49, 2008, pp. 2570–2580.

[18] M. Clerc and J. Kennedy, "The particle swarm-Explosion, stability, and convergence in a multidimensional complex space", *IEEE Trans. on Evolutionary. Computation,* vol. 6, No. 1, 2002, pp. 58–73.

[19] C. Zhang, H. Shao, Y. Li, "Particle swarm optimization for evolving artifical network", in *Proc. of IEEE Int. Conf. on Systems., Man and Cybernetics*, vol. 4, 2000, pp. 2487-2490.

[20] R. Mendes, P. Cortez, M. Rocha, and J. Neves, "Particle swarms for feedforward neural network training", in *Proc. Int. Joint Conf. on Neural Networks*, vol. 2, 2002, pp. 1895-1899.

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

[22] C. Sun, D. Gong, "Support vector machines with PSO algorithm for short-term load forecasting", *in Proc. of IEEE Int. Conf. on Networking, Sensing and Control*, 2006, pp. 676-680