# Applying Wavelets to Short-Term Load Forecasting Using PSO-Based Neural Networks

Z. A. Bashir and M. E. El-Hawary, *Fellow, IEEE*

***Abstract*—The paper addresses the problem of predicting hourly load demand using adaptive artificial neural networks (ANNs). A particle swarm optimization (PSO) algorithm is employed to adjust the network's weights in the training phase of the ANNs. The advantage of using a PSO algorithm over other conventional training algorithms such as the back-propagation (BP) is that potential solutions will be flown through the problem hyperspace with accelerated movement towards the best solution. Thus the training phase should result in obtaining the weights configuration associated with the minimum output error. Data are wavelet transformed during the preprocessing stage and then inserted into the neural network to extract redundant information from the load curve. This results in better load characterization which creates a more reliable forecasting model. The transformed data of historical load and weather information were trained and tested over various periods of time. The generalized error estimation is done by using the reverse part of the data as a "test" set. The results were compared with traditional BP algorithm and offered a high forecasting precision.**

***Index Terms*—Hourly load forecasting, neural networks, particle swarm optimization, wavelet transform, weighted multiple linear regression.**

## I. Introduction

SHORT-TERM load forecasts of power system loads with lead times ranging from one hour to several days ahead have a significant impact on the efficiency of operation of electrical utilities. Many operational decisions such as generation scheduling, load management and system security assessment are based on such forecasts. Conventional (statistical) methods have been used to predict the hourly load such as autoregressive moving average, linear regression, stochastic time series, and general exponential smoothing [1]. Such methods can not properly represent the complex nonlinear relationships between the load and a series of stochastic factors such as daily and weekly time rhythms, special events and correlation properties of weather conditions which can cause high unpredictable variations in power demand.

Since the mid 1980s, artificial neural networks (ANNs) methods have received a great deal of attention and were proposed as powerful computational tools to solve the load forecasting problem. ANNs are able to give better performance in dealing with the nonlinear relationships among their input variables [2]–[4]. ANNs could extract implicit nonlinear relationships among input variables by learning from training data. Using trained supervised networks requires a measure of the discrepancy between the network output value and desired value. The difference between these values yields an error function.

The conventional back propagation algorithm trains neural network parameters (weights and biases) using gradient descent or conjugate gradient decent methods by calculating the partial derivative of the performance with respect to the weights and biases values. However, using this technique adds some constraints to a neural network where the transfer function of each neuron must be differentiable. Moreover, 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 [5].

The advantage of using particle swarm optimization (PSO) algorithm over other techniques is that it can be computationally inexpensive, easily implemented, 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 in the training phase, to obtain a set of weights that will minimize the error function in competitive time. Weights are progressively updated until the convergence criterion is satisfied. The objective function to be minimized by the PSO algorithm is the predicted error function.

Combining the wavelet transform with other techniques such as autoregressive modeling, and neural networks for forecasting electrical load demand has been studied [6]–[9]. Wavelet and wavelet packets were applied to the power load data to decompose and reconstruct it to several scales with different levels using orthogonal and bi-orthogonal wavelets. Based on these investigations, wavelets were demonstrated to be able to extract redundant information from load data and improve forecasting accuracy.

There is a strong correlation between the behavior of electric power consumption and weather variables such as temperature, cloud cover, wind speed and humidity. Because of their uncertainty, these variables have stochastic effects on the load shape. As a result, weather data are introduced to the neural network on the input side and combined with historical load data to predict hourly loads.

To obtain better results and ensure that the adopted model results in convergence, this work investigates factors that influence the overall performance of neural networks such as the determination of the right network topology, good selection of data patterns, and the selection of appropriate and efficient training algorithm parameters.

This paper utilizes the synergies between three powerful techniques which when used as a combination result in enhancements of the forecasting capability. For example the load behavior is better characterized by wavelet transforms.

Manuscript received January 10, 2008; revised August 27, 2008. First published January 13, 2009; current version published January 21, 2009. Paper no. TPWRS-00003-2008.

The authors are with the Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS B3J 2X4, Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2008.2008606

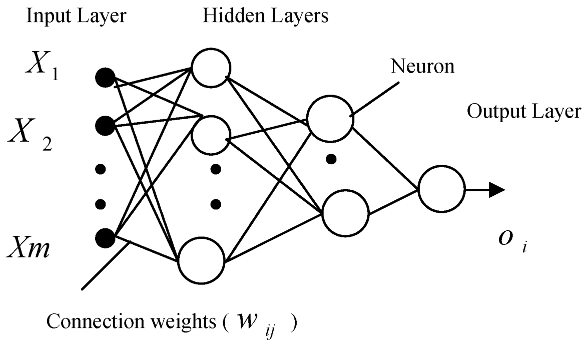


Fig. 1. Multilayer feed-forward neural network architecture.

## II. Neural Network Topology

A 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 learning process. 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. Thus, the parameters of data flow from the input neurons, forward through any hidden layer neurons, eventually reaching the output layer neurons. All of the layers are fully interconnected with each other by weights as shown in Fig. 1.

A neural network uses a learning function to modify the variable connection weights at the inputs of each processing element according to some neural based algorithm. Multiple layers of neurons with nonlinear activation functions allow the network to learn relationships between the input and output of the network. The training process requires a set of examples of proper network behavior. The network can be trained for function approximation. During the training process, the weights define a vector in a multidimensional space. The weights are updated iteratively to minimize the error function. A supervised learning rule is adopted. Thus the neural network is provided with example cases and desired responses. The weights and biases are then adjusted in order to minimize the difference between the network output and its corresponding desired output vector. The performance function of the neural network is normally chosen to be the mean squared error for each pattern on the training set

$$E_p = \frac{1}{N}\sum_{i=1}^{N}(t_{pi} - o_{pi})^2 \tag{1}$$

where $t_{pi}$ is the target at $i^{th}$ pattern, $o_{pi}$ is the predicted of the network's output at $i^{th}$ pattern and $N$ is number of training set samples.

## III. Wavelet Transform

Wavelet analysis provides time and frequency representation (time-scale analysis) simultaneously. Based on this feature, wavelets provide a way of analyzing a nonstationary signal in power systems such as electric load and price time-series signals [10]–[14], voltage and current waveforms [15], [16]. The

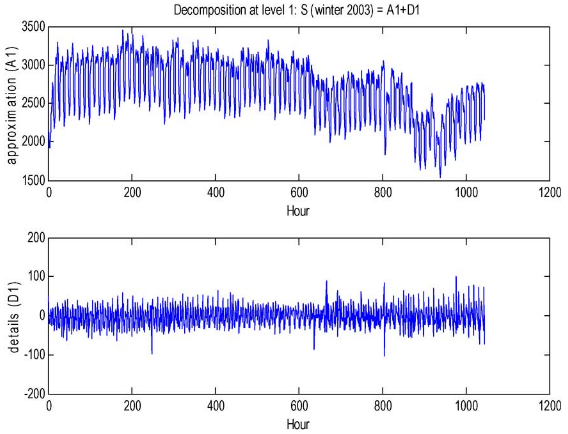


Fig. 2. One level of load signal decomposition by DB2.

wavelet transform is applied to decompose the original time domain signal into several other scales with different levels of resolution in what is called multiresolution decomposition [17], [18]. This allows extracting the irregular information from the original signal that is most likely produced by residuals and has high frequency changes. Technically, the load data are transformed into low and high coefficients. The low coefficients are an approximated version that is associated with low pass filtering while the latter is a detailed version that is associated with high pass filtering. The selection of the basic or mother wavelet has a significant effect on the results obtained. Hence, Daubechies DB2 wave is chosen over other Daubechies wavelet families to decompose data into two levels in order to reflect the uncertain factors on daily load characteristics. The orthogonality feature of DB2, allows the reconstruction of the original signal from its transform coefficients without losing any information [19].

In general, the wavelet transform of time series signal $f(t)$ and its reconstruction formulae are defined by

$$WT_{(a,b)} = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right)dt$$

where

$$\mathrm{a} = 2^{-j},\ \mathrm{b} = \mathrm{k}2^{-\mathrm{j}} \in R,\ \text{and } a \neq 0 \tag{2}$$

$$f(t) = \frac{1}{c_\psi^2}\int_a\int_b \Psi_\psi(a,b)\frac{1}{a^2}\psi\left(\frac{t-b}{a}\right)da\,db \tag{3}$$

$\psi(t)$ is the mother wavelet, and $a$ and b are the scale and time-shift parameters, respectively. The decomposition procedure has been done in this work as stated in [14].

As shown in Fig. 2 the approximation coefficients describe the repeated load behavior and carry most trendy information of the original load signal, whereas details coefficients show most of the characteristics of influential factors that affect the load demand.

## IV. Particle Swarm Optimizer

Particle swarm optimization is a heuristic approach first proposed by Kennedy and Eberhart in 1995 [20] 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 [21]. Each particle is flown through the multidimension 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 (4) and (5) 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 $(gbes,)$ trying to search for a better position. Thus, particles communicate with each other and distribute their information among each other during their search

$$\begin{aligned} v_i(t+1) =& wv_i(t) + \eta_1 * r_1 \left[pbest - x_i(t)\right] \\ &+ \eta_2 * r_2 \left[gbest - x_i(t)\right] \quad \mathrm{i} = 1, 2, \ldots\ldots \mathrm{N} \quad (4) \\ x_i(t+1) =& x_i(t) + v_i(t+1). \quad (5) \end{aligned}$$

Here $w$ is an inertia weight, which provides a balance between the local and global exploration. $v_i$ and $\mathrm{v_i}(t+1)$ are current and modified velocity for each iteration, respectively. $\eta_1$ and $\eta_2$ 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.

The connection weights between input, hidden, and output layers are represented in three matrices $w_1$, $w_2$, and $w_3$ of size $(m \times n)$. According to (4) and (5), the current position $x_i(t)$ for each particle will be represented by $w_i(t) = \{w_1, w_2, w_3\}$. Therefore, the position of each particle represents a set of weights for the current iteration. Thus, updating velocity and then position for each particle will be determined for each weight matrix so as to minimize the error function of the network. The dimension of the search space for each particle will be the number of weights associated with the network. According to (1), the fitness value for each particle will be determined based on the new position value of each weight matrix.

## V. Input Data Selection

The daily load patterns for weekdays are almost similar while those for weekends are different from weekdays due to social activities, etc. Therefore, loads and weather information data for weekdays and weekends are forecasted separately [22]. The study reported in this paper, uses hourly historical load data and weather information for an area in the State of New York collected by New York Independent Operator System and Weather Source Stations, Buffalo, NY. 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]. As shown in Table I, neural networks models are trained with 3 to 12 neurons as input data while 8 and 4 neurons for the first and second hidden layers, respectively, and 1 neuron for an output layer. Regarding the number of neurons in hidden layer, there was no such general rule that could be followed to determine the exact number of neurons in the hidden layer. Hence, the optimum number of neurons in these layers has been determined based on "trial and error" experimentation. This network structure could give the best performance in terms of accuracy. The data cover the year of 2001 and 2003. The neural network is trained for several periods of time 22, 66, 131, and 261 workdays for year 2001 whereas model testing is investigated by using year 2003, to test the generalized error estimation of neural network.

TABLE I
Structure of ANNs Adopted Models

| layer | Data Type | No. of neurons & type of T.F |
|---|---|---|
| **Input** | Load for current hour and previous two hours | 3 |
| **Input** | Temp. for current hour and previous two hours | 3 |
| **Input** | Wind speed for current hour and previous two hours | 3 |
| **Input** | Humidity for current hour and previous two hours | 3 |
| **Hidden** | N/A | 8,4 / Tansig |
| **Output** | Normalized Load | 1 / Purelin |

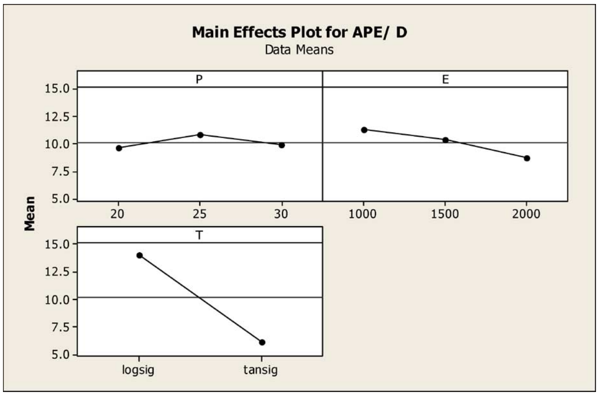


Fig. 3. Experiment design results for ANNs and PSO.

## VI. Training Algorithm

The training algorithm is a form of supervised learning where the weights are updated in order to reduce the error function whenever the network output does not match the desired values. The training function "*TRAINPSO*" for neural networks *WNNs-PSO* [23], [24] is used. As shown in Fig. 3, an experimental design "*Full Factorial Design*" [25] was built to investigate the main modified parameters of both ANNs and PSO that effectively influence their performance. The nonlinear hyperbolic tangent function (tan-sigmoid) has been chosen over the log-sigmoid function for hidden layers. Regarding the

PSO parameters, the swarm size was set equal to 20, 25, and 30 particles where the optimal size is found to be 20. The number of iterations is run for 1000, 1500, and 2000 epochs. The search space range is taken between [−25 25], the acceleration values ($\eta_1 = \eta_2 = 2$), and the inertia weight is gradually decreased from 0.9 to 0.4 in order to reduce the influence of past velocity.

The training function "*TRAINGDM*" is used to train the back propagation neural network *WNNs_PB* model. "*TRAINGDM*" adjusts the weight and bias values according to gradient descent with momentum constant [26]. Initialization weights and biases for all models were done by using a layer initialization function according to the ideas of "*Nguyen-Widrow*" [27]. The training functions repeatedly apply a set of input vectors to a network, updating the network each time until certain stopping criteria are met. In order to meet the performance goal of *WNNs_BP* model when the maximum error goal of 1e-05, the number of iterations for its training was set for 5000 epochs and the optimal learning rate value was 0.16 and momentum constant was 0.9. For both models, the tan-sigmoid function as shown in (6) is used as transfer function for hidden layers and linear function "*purelin*" is used for output layer. This type of structure has proven successful for function approximation studies [26]

$$\tan sig(k) = \frac{2}{(1+\exp(-2*k))-1} \tag{6}$$

where $k$ is the neuron's activation function $\sum_{i=1}^{N} w_{ij}x_i$.

## VII. Simulation Results

The numerical simulations are run on a laptop *PC* (*Centrino Duo*, 1.8 GHz, 1 GB RAM). The simulation models and design of experiment are implemented in "*MATLAB 7*" and "*MANITAB 15*", respectively. The measure of error between the actual and predicted load is obtained using mean absolute percentage error $MAPE$ (7) as the prediction error as follows:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{A_i - P_i}{A_i}\right] \times 100\% \quad \text{i} = 1, 2, \ldots \text{N} \tag{7}$$

where $A_i$ and $P_i$ are the actual price and predicted price, respectively. $N$ is the number of hours.

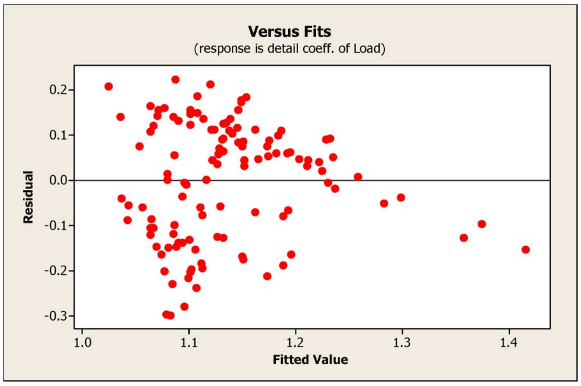


Fig. 4. Nonconstant variance of prediction errors by MLR.

TABLE II
APE (%) and MAPE (%) for Different Weather Variables Using **WNN-PSO** Model

| Hour | L | L & T | L,T,& W | L,T,,H |
|---|---|---|---|---|
| 0 | 3.2844 | 3.9149 | 4.0422 | 0.0057 |
| 1 | 3.8998 | 4.8252 | 5.3577 | 1.8512 |
| 2 | 4.1436 | 5.0744 | 5.9406 | 2.6807 |
| 3 | 2.2109 | 3.2826 | 3.9438 | 0.7912 |
| 4 | 4.6829 | 5.8515 | 6.4457 | 3.4735 |
| 5 | 5.0366 | 5.7555 | 6.1565 | 2.9008 |
| 6 | 2.6634 | 3.0602 | 3.2907 | 0.0336 |
| 7 | 2.4032 | 3.8654 | 3.7656 | 1.2217 |
| 8 | 2.7932 | 0.7759 | 1.1106 | 3.1897 |
| 9 | 2.4594 | 0.6122 | 1.6211 | 2.6395 |
| 10 | 1.3226 | 0.5995 | 0.9839 | 1.1577 |
| 11 | 3.1108 | 1.0408 | 1.5009 | 2.6376 |
| 12 | 2.4542 | 0.2389 | 0.0455 | 1.6627 |
| 13 | 1.5784 | 0.4292 | 0.2491 | 1.0350 |
| 14 | 2.1794 | 0.3010 | 0.5787 | 1.7370 |
| 15 | 3.8859 | 2.2166 | 2.9567 | 4.0333 |
| 16 | 0.8009 | 0.6875 | 0.4309 | 1.4190 |
| 17 | 1.3080 | 3.0458 | 0.7333 | 1.0228 |
| 18 | 3.5867 | 1.7844 | 4.9564 | 3.7485 |
| 19 | 3.7965 | 1.8471 | 4.2924 | 3.5907 |
| 20 | 2.1310 | 0.0087 | 2.2162 | 1.6192 |
| 21 | 0.7923 | 0.8607 | 1.4560 | 1.4098 |
| 22 | 0.5240 | 0.8133 | 1.5162 | 1.9220 |
| 23 | 0.1179 | 0.1786 | 0.4427 | 2.2867 |
| MAPE | 2.5486 | 2.1279 | 2.6681 | **2.0029** |

To compare the forecasting accuracy of our models based on the training algorithm, the historical and transformed historical load data with their influential variables which has one level of approximation coefficients were forecasted by the neural network using back-propagation and PSO algorithms. It is noted that using more than one level of approximation results in a degraded ANN network performance due to over-fitting and memorization. In addition, applying ANNs to the second part which has high frequency changes and does not follow a specific pattern is not recommended. Thus, this part has been forecasted using weighted multiple-linear regression (*WMLR*) method, based on the relationship between the independent predictor variable (weather information) and dependent load

$$L_D = a_0 + b_1x_1 + b_2x_2 + \ldots .b_nx_n \tag{8}$$

where $L_D$ is the forecasted load of the detail coefficients, $a_0$, $b_n$ are regression coefficients to be estimated using weighted least square regression, and $x_n$ is the weather data such as temperature and humidity. It should be noted that applying *MLR* to the detail coefficients yielded a nonconstant variance estimations of prediction errors as can be illustrated in Fig. 4. Therefore, *WMLR* technique is used to modify the influence of large errors on the estimation of regression coefficients [25]. Finally, the forecasted approximation and details coefficients are reconstructed to form the overall predicted signal load.

The forecasting algorithm is based on load and weather variables. Theses variables are actually used to build the model, and to show how weather variables could effect load prediction. Our adopted model is trained for all seasons of the year and to observe also the significant effect of each weather variable on forecasting results. When we include wind speed variable, the

TABLE III
REGRESSION ANALYSIS: **L** VS, **T**, **W**, AND **H**

| Predictor | Coeff. | SSE | T-ratio | P-value |
|---|---|---|---|---|
| **Interception** | 1798.5 | 435.4 | 4.13 | 0.000 |
| **Temp.** | 10.037 | 5.524 | 1.82 | **0.078** |
| **W. speed** | 5.19 | 11.95 | 0.43 | 0.667 |
| **Humidity** | -6.611 | 3.124 | -2.12 | **0.041** |

TABLE IV
APE (%) AND MAPE (%) FOR DIFFERENT PERIODS OF TIME *(January 13, 2003)*

| Hour | 22 Days | 66 Days | 131 Days | 261 Days |
|---|---|---|---|---|
| | Month | Season | L. season | Year |
| 0 | 3.5821 | 2.3995 | 3.5911 | 0.1567 |
| 1 | 4.7955 | 3.2698 | 5.3002 | 1.1428 |
| 2 | 5.4431 | 3.6228 | 6.0798 | 1.5562 |
| 3 | 3.7047 | 1.7905 | 6.5538 | 1.5664 |
| 4 | 6.4211 | 4.3726 | 7.9980 | 2.6059 |
| 5 | 6.1470 | 4.2838 | 5.2308 | 1.6467 |
| 6 | 3.3471 | 1.6648 | 6.1145 | 1.8055 |
| 7 | 3.0134 | 1.9464 | 8.3171 | 2.4959 |
| 8 | 2.3273 | 2.9693 | 0.3522 | 4.8655 |
| 9 | 1.8504 | 2.6834 | 1.7585 | 5.0289 |
| 10 | 0.6273 | 1.5183 | 1.0558 | 3.4241 |
| 11 | 2.3876 | 3.1880 | 1.6385 | 4.2181 |
| 12 | 1.6815 | 2.4391 | 1.6023 | 4.1399 |
| 13 | 0.8470 | 1.6257 | 1.1333 | 3.7548 |
| 14 | 1.4707 | 2.2474 | 0.9006 | 3.6055 |
| 15 | 3.2901 | 4.1395 | 3.9014 | 6.5807 |
| 16 | 0.2919 | 1.2009 | 1.2044 | 3.9241 |
| 17 | 2.1018 | 1.0458 | 2.0033 | 0.3411 |
| 18 | 2.6433 | 3.7798 | 4.1800 | 6.1872 |
| 19 | 2.8011 | 3.9149 | 5.1808 | 7.1614 |
| 20 | 1.0249 | 2.1813 | 2.8295 | 4.8159 |
| 21 | 0.0729 | 1.1339 | 1.9724 | 4.3035 |
| 22 | 0.0681 | 1.0705 | 1.8814 | 4.4931 |
| 23 | 0.0817 | 1.3156 | 1.7251 | 4.9529 |
| MAPE/D | 2.5009 | **2.4918** | 3.1904 | 3.5322 |
| MAPE/P | 3.4768 | **2.8226** | 3.1894 | 3.7416 |
| MSE | 0.0067 | **0.0082** | 0.0187 | 0.0137 |

APE is noticeably increased in and therefore it has been dropped from the model. These are shown in Tables II and VI. As depicted in Table III a regression analysis also has been applied to test the load sample with its influential factors data (weather variables), to assess the significance of variable effects on estimated load. According to null hypothesis test, wind speed's p-value ($p > 0.1$) means the higher probability that rejecting it is a mistake. In other word, it is not significant.

Based upon the size of input data, the selected model trained for different periods of time as shown in Table IV. The predicted error is calculated on an hourly basis for four different models for October 13, 2003 and is shown in Table V. Fig. 5 shows 14-day forecasted load in the winter season, and the corresponding prediction errors plotted in Fig. 6. The maximum is well below 5%. The plots of actual and forecasted daily load curves for different season weekdays are represented in Fig. 7(a)–(d).

TABLE V
APE (%) AND MAPE (%) FOR ALL SEASONS OF YEAR 2003 USING **WNN-PSO** MODEL

| Hour | WMLR | NN_BP | WNN_BP | WNN_PSO |
|---|---|---|---|---|
| 0 | 8.2863 | 5.8673 | 5.4627 | 3.7314 |
| 1 | 14.3170 | 6.8759 | 10.5931 | 4.0212 |
| 2 | 17.6570 | 7.1447 | 5.9282 | 4.2860 |
| 3 | 19.4527 | 5.6111 | 6.0769 | 3.7785 |
| 4 | 21.8083 | 5.7601 | 8.6054 | 4.7347 |
| 5 | 20.0846 | 4.7772 | 8.5633 | 4.3810 |
| 6 | 14.6190 | 2.7367 | 5.4564 | 3.0259 |
| 7 | 2.3178 | 0.3398 | 6.0188 | 4.3719 |
| 8 | 6.0763 | 1.8887 | 3.3052 | 0.2838 |
| 9 | 8.6702 | 2.1608 | 3.7368 | 0.7840 |
| 10 | 10.2829 | 3.3382 | 1.3852 | 0.0791 |
| 11 | 9.4484 | 4.1162 | 0.7496 | 1.7048 |
| 12 | 9.4533 | 4.0839 | 1.0207 | 0.7458 |
| 13 | 9.6096 | 3.9896 | 0.6633 | 1.2245 |
| 14 | 9.1419 | 4.1537 | 0.1667 | 0.8096 |
| 15 | 8.3402 | 3.2867 | 1.0465 | 2.3958 |
| 16 | 8.8050 | 2.9130 | 1.9488 | 1.1066 |
| 17 | 9.9588 | 2.0440 | 1.6967 | 2.2607 |
| 18 | 9.3633 | 1.0756 | 1.9147 | 0.6679 |
| 19 | 11.5329 | 1.8875 | 5.7774 | 0.2366 |
| 20 | 11.9377 | 3.5060 | 2.0680 | 1.2388 |
| 21 | 11.9840 | 3.5396 | 0.9235 | 0.3993 |
| 22 | 7.5490 | 2.5072 | 2.3820 | 1.2221 |
| 23 | 0.3378 | 0.7906 | 0.8399 | 0.2261 |
| MAPE/D | 10.8764 | 3.5164 | 3.5971 | **1.9882** |
| MAPE/S | 10.2003 | 4.4706 | 2.9093 | **2.3834** |
| MSE | ________ | 0.0241 | 0.0423 | **0.0098** |

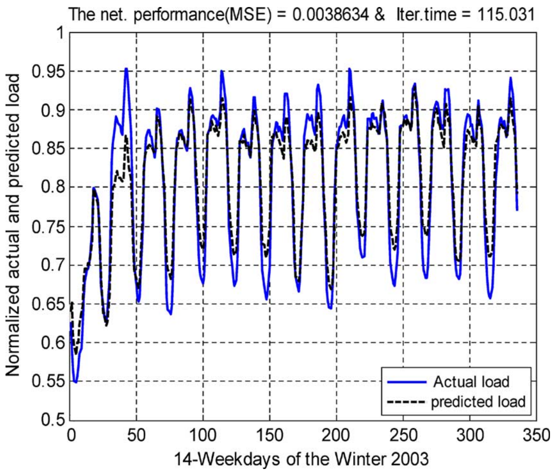


Fig. 5. Forecasted load for 14-day winter season using **WNNs_PSO** mode.

We also tested the adopted model by applying variable weather patterns (unstable weather system) to check the capability of the model to forecast load with unexpected weather changing. Table VII shows that load during fall season is predicted using winter, spring and summer weather information and yields an encouraging result. To check the quality of the neural networks training, a linear regression has been performed between neural network output and the corresponding target vector. Hence, the Pearson correlation coefficient (R-value) is determined to measure how well the variation in neural network

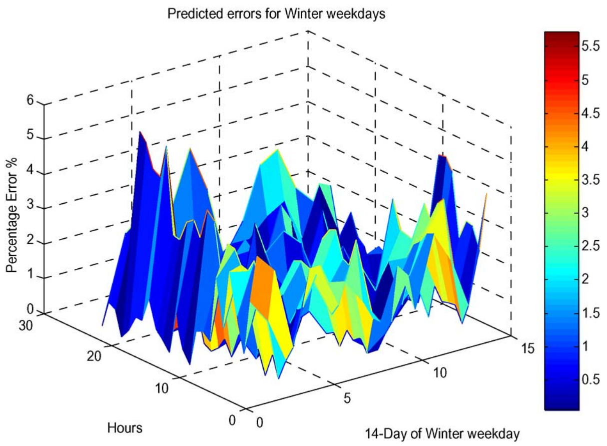
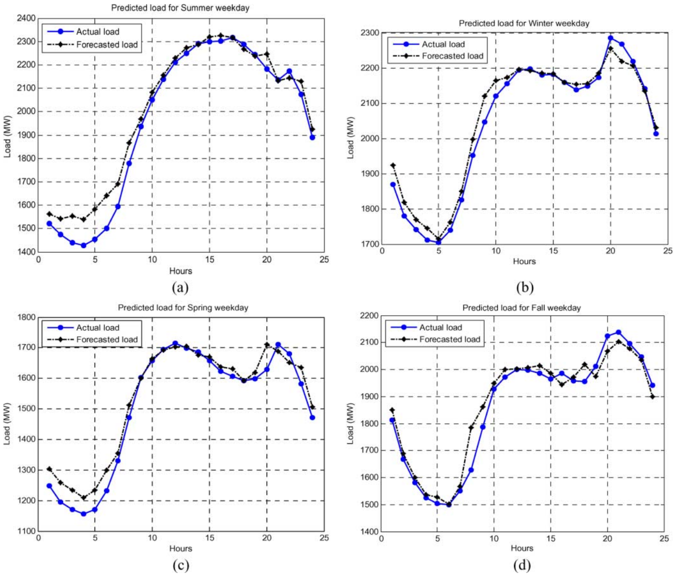


Fig. 6. Predicted errors for 14-day of winter weekdays using **WNNs_PSO** model.

TABLE VI
APE (%) AND MAPE (%) FOR FALL WEEKDAYS USING LOAD, TEMP., AND HUMIDITY (L, T, AND H)

| Hour | Fall | Winter | Spring | Summer |
|---|---|---|---|---|
| 0 | 3.7314 | 1.3271 | 3.1111 | 3.3560 |
| 1 | 4.0212 | 2.6436 | 4.4047 | 4.7898 |
| 2 | 4.2860 | 2.2800 | 4.6897 | 4.1412 |
| 3 | 3.7785 | 0.9326 | 3.8541 | 3.1569 |
| 4 | 4.7347 | 3.3893 | 4.3771 | 3.7632 |
| 5 | 4.3810 | 4.1877 | 3.7532 | 5.4048 |
| 6 | 3.0259 | 0.3225 | 1.8507 | 2.7795 |
| 7 | 4.3719 | 0.2754 | 3.0995 | 1.0252 |
| 8 | 0.2838 | 2.4534 | 0.8365 | 3.8817 |
| 9 | 0.7840 | 2.5722 | 2.0088 | 5.1270 |
| 10 | 0.0791 | 2.4505 | 0.2304 | 0.2757 |
| 11 | 1.7048 | 2.7313 | 0.3382 | 1.1752 |
| 12 | 0.7458 | 1.9684 | 0.3544 | 0.7308 |
| 13 | 1.2245 | 1.8531 | 0.4937 | 1.0036 |
| 14 | 0.8096 | 1.7907 | 0.5896 | 1.6464 |
| 15 | 2.3958 | 4.7122 | 0.9634 | 2.5157 |
| 16 | 1.1066 | 1.8349 | 1.8516 | 1.4937 |
| 17 | 2.2607 | 1.3960 | 0.7790 | 2.5675 |
| 18 | 0.6679 | 4.9515 | 2.4464 | 0.6284 |
| 19 | 0.2366 | 3.6595 | 7.6208 | 0.3914 |
| 20 | 1.2388 | 2.4636 | 0.0837 | 1.1842 |
| 21 | 0.3993 | 2.7331 | 1.1775 | 2.6808 |
| 22 | 1.2221 | 0.7393 | 4.2468 | 2.0949 |
| 23 | 0.2261 | 1.6242 | 2.7956 | 2.1837 |
| MAPE/D | **1.9882** | 2.3038 | 2.3315 | 2.5162 |
| MAPE/S | **2.3834** | 2.5048 | 3.1512 | 2.9830 |
| MSE | **0.0098** | 0.0085 | 0.0123 | 0.0152 |



(a) (b) (c) (d)

Fig. 7. (a) Actual and forecasted summer daily load using **WNNs_PSO** model. (b) Actual and forecasted winter daily load using **WNNs_PSO** model. (c) Actual and forecasted daily load using **WNNs_PSO** model. (d) Actual and forecasted fall daily load using **WNNs_PSO** model.

TABLE VII
APE (%) AND MAPE (%) FOR VARIABLE WEATHER PATTERNS USING **WNN-PSO** MODE

| Hour | WNN_PSO Fall/Winter | WNN_PSO Fall/Spring | WNN_PSO Fall/Summer | WNN_PSO Fall/Fall |
|---|---|---|---|---|
| 0 | 5.5594 | 6.2789 | 8.2331 | 3.7314 |
| 1 | 3.9224 | 4.2668 | 6.5350 | 4.0212 |
| 2 | 4.6747 | 5.1537 | 7.0389 | 4.2860 |
| 3 | 5.9428 | 6.4142 | 8.0825 | 3.7785 |
| 4 | 5.9832 | 6.5267 | 7.8832 | 4.7347 |
| 5 | 6.1908 | 6.7584 | 7.7886 | 4.3810 |
| 6 | 2.1512 | 2.7665 | 3.9544 | 3.0259 |
| 7 | 2.6871 | 3.3322 | 4.5365 | 4.3719 |
| 8 | 2.2918 | 3.0519 | 4.4742 | 0.2838 |
| 9 | 0.0176 | 0.8701 | 2.4580 | 0.7840 |
| 10 | 1.0341 | 0.7016 | 1.5568 | 0.0791 |
| 11 | 0.4895 | 1.9442 | 2.3574 | 1.7048 |
| 12 | 1.0623 | 1.1442 | 1.8361 | 0.7458 |
| 13 | 0.8526 | 1.3627 | 2.1360 | 1.2245 |
| 14 | 0.4860 | 2.6155 | 3.4716 | 0.8096 |
| 15 | 1.3599 | 2.4206 | 3.3634 | 2.3958 |
| 16 | 0.4922 | 2.7217 | 4.0202 | 1.1066 |
| 17 | 0.1832 | 2.7889 | 4.3743 | 2.2607 |
| 18 | 0.5843 | 3.2597 | 4.7927 | 0.6679 |
| 19 | 2.2562 | 0.0016 | 1.4815 | 0.2366 |
| 20 | 2.5083 | 0.3784 | 0.7517 | 1.2388 |
| 21 | 2.4106 | 0.3707 | 0.5423 | 0.3993 |
| 22 | 0.2155 | 2.1607 | 3.0027 | 1.2221 |
| 23 | 1.2155 | 3.8831 | 4.6012 | 0.2261 |
| MAPE./D | 2.2738 | 2.9655 | 4.1363 | 1.9882 |
| MAPE./S | 2.5375 | 2.5640 | 2.6575 | 2.3834 |

output is explained by the target vector as shown in Fig. 8. The R-value close to 1 means a good fit of data to the regression line.

## VIII. EVALUATING ADOPTED MODEL

The over-fitting phenomenon often occurs during training ANNs. Therefore, a technique called "*early stopping*" is applied for validating and testing sets to the proposed forecasting model. The data sequence set for load and influential variables during fall season 2001 weekdays is divided into three subsets, e.g., training subset period start from October 1 to November 30, December 3–14 for validation subset and December 17–31 for test subset. This technique is implemented by passing the validation data vector to the training phase vector. Results are

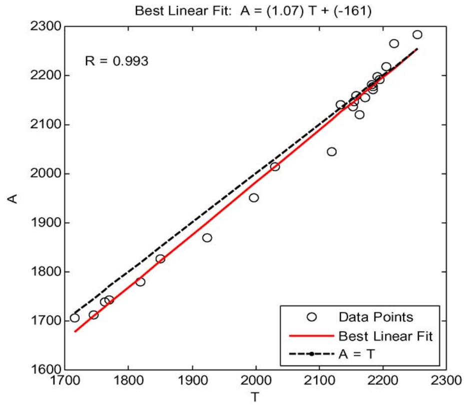


Fig. 8. Correlation coefficient between output and target value.

TABLE VIII
APE (%) AND MAPE (%) OF TRAINING, VALIDATING, AND TESTING SETS FOR FALL WEEKDAYS

| Hours | Training (43-day) | Validation (16-day) | Testing ( 16-day) |
|---|---|---|---|
| 0 | 4.6115 | 7.9865 | 10.7400 |
| 1 | 4.8712 | 7.3745 | 7.3630 |
| 2 | 4.9695 | 5.6882 | 4.3195 |
| 3 | 5.1105 | 5.7655 | 4.6197 |
| 4 | 4.2264 | 3.9210 | 4.0604 |
| 5 | 3.4573 | 0.8172 | 0.0955 |
| 6 | 1.1014 | 6.1517 | 3.5921 |
| 7 | 0.4578 | 5.6010 | 2.3203 |
| 8 | 1.1647 | 1.1372 | 10.2201 |
| 9 | 2.0365 | 4.5298 | 5.9502 |
| 10 | 2.2857 | 3.7928 | 2.0810 |
| 11 | 2.7072 | 0.5525 | 1.2211 |
| 12 | 2.4698 | 0.3832 | 2.7507 |
| 13 | 2.4099 | 1.4958 | 0.1990 |
| 14 | 2.1876 | 0.6686 | 1.1172 |
| 15 | 2.0215 | 2.5748 | 3.8870 |
| 16 | 1.7493 | 1.8517 | 1.0982 |
| 17 | 1.5193 | 6.7226 | 2.1747 |
| 18 | 2.4355 | 1.1790 | 10.9797 |
| 19 | 2.8867 | 0.1192 | 6.8433 |
| 20 | 2.3657 | 1.2659 | 4.2004 |
| 21 | 2.2427 | 2.5279 | 1.2956 |
| 22 | 0.0775 | 5.0491 | 1.7072 |
| 23 | 2.1356 | 9.2916 | 2.4321 |
| MAPE/D | 2.5625 | 3.6020 | 3.9695 |
| MAPE/ P | 2.3555 | 4.3709 | 4.1228 |

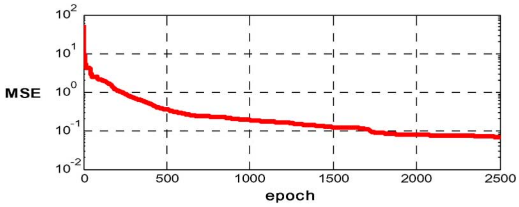


Fig. 9. Convergence of PSO algorithm.

shown in Table VIII. However, if further training on the primary set will degrade the network generalization to validation set, the validation set will stop the training early at a specified number of iterations [26]. The test vector performance can be used to measure how well the network generalizes beyond primary and validation vectors. Fig. 10 shows reasonable results where both the validation set error and test set error patterns have similar characteristic, and does not appear that any significant over-fitting has occurred.

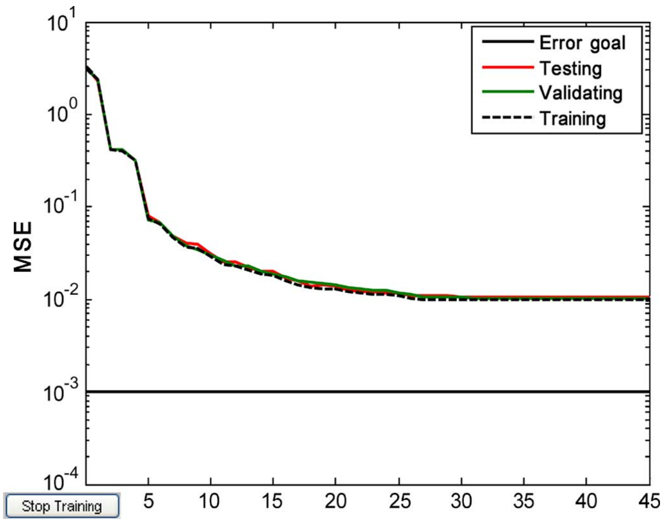


Fig. 10. Performance of training, validating, and testing sets for fall weekdays.

## IX. CONCLUSIONS

This paper demonstrated the ability of an artificial intelligence method to predict hourly load using a particle swarm optimization methodology as a training phase of the neural network, to adjust the weights and biases of the model. The learning process involves minimizing the error objective function which represents the difference between the network output and the corresponding target. According to the simulation results, it was found that selecting a proper network structure, effective input data, and suitable PSO parameters are considered and gets better performance of the designed model will be obtained. The statistical method used to obtain significant PSO parameters contributed to obtaining good convergence properties as shown in Fig. 9. The forecasted models have enhanced accuracy due to applying wavelet analysis to the network input data. Our results also show that using some weather information with historical load data positively influences the training and testing results. This paper introduces for the first time an application that utilizes the synergies between three powerful techniques which when used as a combination have resulted in enhancements of the forecasting capability. For example the load behavior is better characterized by wavelet transforms. An additional specific contribution is in ANN training, in which standard back-propagation has been replaced by PSO. As a nonconvex optimization tool, PSO is useful for training an ANN, as confirmed by results of the paper.

## REFERENCES

[1] I. Moghram and S. Rahman, “Analysis and evaluation of five short-term load forecasting techniques,” *IEEE Trans. Power Syst.*, vol. 4, no. 4, pp. 1484–1491, Nov. 1989.

[2] E. Sanchez-Sinencio and C. Lau, *Artificial Neural Networks, Paradigms, Applications, and Hardware Implementations*. Piscataway, NJ: IEEE Press, 1992.

[3] B. S. Kermanshahi, C. H. Poskar, G. Swift, P. McLaren, W. Pedrycz, W. Buhr, and A. Silik, "Artificial neural network for forecasting daily loads of a Canadian electric utility," in *Proc. IEEE 2nd Int. Forum Applications of Neural Networks to Power Systems*, 1993, pp. 302–307.

[4] T. W. S. Chow and C. T. Leung, "Neural networks based short-term load forecasting using weather compensation," *IEEE Trans. Power Syst.*, vol. 11, no. 4, pp. 1736–1742, Nov. 1996.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representation by back propagating errors," *Nature*, pp. 533–536, 1986.

[6] A. K. Saha, S. Chowdhury, S. P. Chowdhury, Y. H. Song, and G. A. Taylor, "Application of wavelets in power system load forecasting," in *Proc. IEEE Power Eng. Soc. General Meeting*, 2006.

[7] Y. Bi, J. Zhao, and D. Zhang, "Power load forecasting algorithm based on wavelet packet analysis," in *Proc. IEEE Int. Conf. Power System Technology*, 2004, pp. 987–990.

[8] D. Tao, W. Xiuli, and W. Xifan, "A combined model of wavelet and neural network for short term load forecasting," in *Proc. IEEE Int. Conf. Power System Technology*, 2002, pp. 2331–2335.

[9] A. J. R. Reis and A. P. A. da Silva, "Feature extraction via multiresolution analysis for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 189–198, Feb. 2005.

[10] Z.-Y. Dong, B.-L. Zhang, and Q. Huang, "Adaptive neural network short term load forecasting with wavelet decomposition," in *Proc. IEEE Power Tech.*, 2001.

[11] C. M. Huang and H. T. Yang, "Evolving wavelet-based networks for short-term load forecasting," *Proc. Inst. Elect. Eng., Gen., Transm., Distrib.*, vol. 148, no. 3, pp. 222–228, May 2001.

[12] I. K. Yu and Y. H. Song, "Load and price forecasting via wavelet transform and neural networks," in *Operation of Market-Oriented Power Systems*, Y.-H. Song and X.-F. Wang, Eds. New York: Springer, 2003, pp. 281–313.

[13] A. J. Coneio, J. Contreras, R. Espinola, and M. A. Plazas, "Forecasting electricity prices for a day-ahead electric energy market," *Int. J. Forecast.*, vol. 21, no. 3, pp. 435–462, Jul.–Sep. 2005.

[14] A. J. Conejo, M. A. Plaza, and A. B. Molina, "Day-ahead electricity price forecasting using the wavelet transform and ARIMA models," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1035–1042, May 2005.

[15] Z. Gang, Y. Xiao-Mei, L. Hao-Wen, and L. Ding, "Classification of voltage sag based on wavelet transform and wavelet network," in *Proc. 2004 Int. Conf. Machine Learning and Cybernetics*, 2004, vol. 1, pp. 466–470.

[16] G. Takata, J. Tahara, M. Michihira, A. Tsuyoshi, K. Amako, H. Omori, and K. Yasui, "The time-frequency analysis of harmonics with wavelet transform for the power electronic systems," in *Proc. Power Conversion Conf. PCC Osaka 2002*, 2002, vol. 2, pp. 733–737.

[17] C. K. Chui, *An Introduction to Wavelets*. New York: Academic, 1992, pp. 6–18.

[18] S. Mallat, "A theory for multiresolution signal decomposition—The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[19] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 41, pp. 906–909, 1988.

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

[21] R. C. Ebehart, P. K. Simpson, and R. W. Dobbins, "Compuational intelligence," in *PC Tools*. New York: Academic Press Professional, 1996.

[22] Z. Bashir and M. E. El-Hawary, "Short term load forecasting by using wavelet neural networks," in *Proc. IEEE Can. Conf. Electrical and Computer Engineering*, 2000, pp. 163–166.

[23] B. Birage, "PSOt—A particle swarm optimization toolbox for use with Matlab," in *Proc. 2003 IEEE Swarm Intelligence Symp.*, Apr. 24–26, 2003, pp. 182–186.

[24] B. Birage, Particle Swarm Optimization ToolBox, 2005.

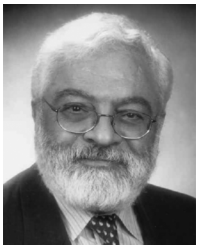
[25] D. C. Montgomery, G. C. Runger, and N. F. Hubele, *Engineering statistics*. New York: Wiley, 2007.

[26] H. Demuth and M. Beal, *Neural Network ToolBox User's Guide*. Natick, MA: The Math Works, 1998.

[27] Matlab on Help. [Online]. Available: http://www.MathWorks.com.

**Z. A. Bashir** received the M.A.Sc. degree from Dalhousie University, Halifax, NS, Canada, in 2000. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at Dalhousie University.

His research interests are in applications of artificial intelligence techniques in power systems engineering.



**M. E. El-Hawary** (S'68–M'72–F'90) received the B.Eng. degree in electrical engineering with distinction from the University of Alexandria, Alexandria, Egypt, in 1965 and the Ph.D. degree from the University of Alberta, Edmonton, AB, Canada, in 1972.

He was a Killam Memorial Fellow at the University of Alberta. He is now a Professor of electrical and computer engineering at Dalhousie University, Halifax, NS, Canada. He served on the faculty and was a Chair of the Electrical Engineering Department at Memorial University of Newfoundland for eight years. He was an Associate Professor of electrical engineering at the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, for two years and was Instructor at the University of Alexandria. He pioneered many computational and artificial intelligence solutions to problems in economic/environmental operation of power systems. He has written ten textbooks and monographs and more than 120 refereed journal articles. He has consulted and taught for more than 30 years.

Dr. El-Hawary is a Fellow of the Engineering Institute of Canada (EIC) and the Canadian Academy of Engineering (CAE).