



WAVELET-BASED SHORT-TERM LOAD FORECASTING USING OPTIMIZED ANFIS

M. W. Mustafa, M. Mustapha, S. N. Khalid and I. Abubakar

Faculty of Electrical Engineering, Universiti Teknologi Malaysia (UTM), Johor Bahru, Johor, Malaysia

E-Mail: wazir@fke.utm.my

ABSTRACT

This paper focuses on forecasting electric load consumption using optimized Adaptive Neuro-Fuzzy inference System (ANFIS). It employs the use of Particle Swarm Optimization (PSO) to optimize ANFIS, with aim of improving its speed and accuracy. It determines the minimum error from the ANFIS error function and thus propagates it to the premise part. Wavelet transform was used to decompose the input variables using Daubechies 2 (db2). The purpose is to reduce outliers as small as possible in the forecasting data. The data was decomposed in to one approximation coefficients and three details coefficients. The combined Wavelet-PSO-ANFIS model was tested using weather and load data of Nova Scotia province. It was found that the model can perform more than Genetic Algorithm (GA) optimized ANFIS and traditional ANFIS, which is been optimized by Gradient Decent (GD). Mean Absolute Percentage Error (MAPE) was used to measure the accuracy of the model. The model gives lower MAPE than the other two models, and is faster in terms of speed of convergence.

Keywords: shot-term load forecasting, ANFIS, PSO, wavelet transform.

1. INTRODUCTION

The need to use limited resources with maximum efficiency, together with current situation of electricity market, make it necessary to implement a speedy and accurate load forecasting. Electric load forecasting affects the general operation of electrical power system [1], [2], thus make it important in power system operations such as economic dispatch, unit commitment, load shedding etc. Also, the practice in energy consumption in which the load profile is valley in the early morning hours and grown up in the afternoon (Figure 1), make it necessary to determine when and where the generation is needed. Through this residential customers can be advice on when to use certain machines without compromising their tariff. There are two major classes of load forecasting methods; parametric method and Artificial intelligence (AI) methods [3]. Because of increase in the complexity of the power system, AI methods are now receiving more attention compared to statistical (parametric) methods [3]. But most of these AI methods are associated with computational difficulties, over fitting and non-evident selection of variables [1], [4], [5]. These subsequently result to erroneous results. They are therefore, need to optimize so as to reduce the error and increase their speed of convergence.

In this paper, db2 of wavelet family is used to remove the outliers and wide variation between the data points. This will improve the accuracy of the forecast. PSO, on the other hand is used to fasten the ANFIS training through replacing the GD algorithm in the backward path. This will not only improve the accuracy, also increase the speed of convergence of the ANFIS.

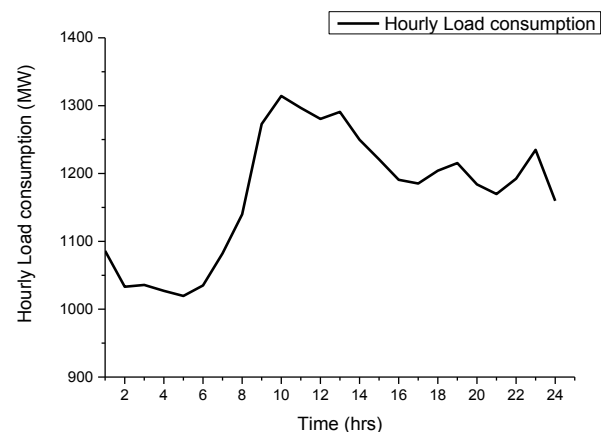


Figure-1. One day hourly load (MW).

A lot of researches were conducted based on AI methods [1], [3], [6]–[12], but there is room for improvements. As presented in [6] GA and PSO are used in training multi-layer perceptron NN (MLPNN) and compared with back propagation NN (BPNN). It was found that the GA trained NN is more accurate and slower in convergence than the PSO trained NN, but both are more accurate than the BPNN. A radial basis function (RBF) NN is proposed to forecast the load without considering the price factor [7], then the RBF-NN forecast is adjusted with real-time price using ANFIS. One-hour-ahead forecast using ANFIS is presented by Thai Nguyen and Yuan liao [8]. Next hour temperature, next hour dew point, day of the week, hour of the day and current day load are used as model inputs to the ANFIS model. M. Hanmandlu and B. K. Chauhan [9] presented a two hybrid NN models comprised of Fuzzy NN (FNN) and wavelet fuzzy NN (WFNN). Fuzzified wavelets inputs from WFNN are used in the FNN, which employed Choquet Integral through q-measure to simplify the learning process, and used reinforced learning to speed the



convergence. In an effort to minimize training errors, GA is used in selecting training variables for an ANFIS model [11]. The system is used in an automobile factory and the data used is being updated time to time for a real-time forecast. A method referred to as lower upper bound estimation was used to produce prediction interval NN based model. PSO was used to determine the optimal weights which are essential in determining the coverage width-based criterion of the prediction intervals. A hybrid of Support Vector Regression (SVR) and krill Herd (KH) is presented to forecast the load within the short-term time frame [3]. The model involved the use of KH algorithm to optimize the SVR parameters while training.

This paper proposed a method of optimizing ANFIS using PSO. Wavelet transform was employed to decompose the data in to details and approximate coefficients using db2. This reduce the number of outliers and give more stable variance [1]. The PSO is used to training the ANFIS by minimizing the error difference between the predicted and the actual load data. In a traditional ANFIS, GD is used in the premise part. GD involved a lot of differentiations before determining the premise parameters, as presented in section 2.2.2. It also involved passing the error through every node in layer-by-layer approach. This makes the network and the training more complex [13]. The PSO will only determines the minimum error and propagate it back to the premise part and update the membership functions.

2. ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS)

ANFIS is a Fuzzy Inference System (FIS) that combines the advantages of fuzzy systems and neural-networks. It is developed by J. S. Roger in 1993 [14]. It is a network-based structure that uses the Sugeno-type IF.....THEN rules through human reasoning to approximate non-linear systems [15]. ANFIS employs two learning algorithms-The LSE and GD. In the forward pass LSE is used to estimate the consequent parameters, and in backward pass GD is used to compute the premise parameters. Figure 2 shows a typical ANFIS structure [16] with only two inputs (x and y) and one output (z). The structure consists of five layers with several nodes (depending on the number of inputs).

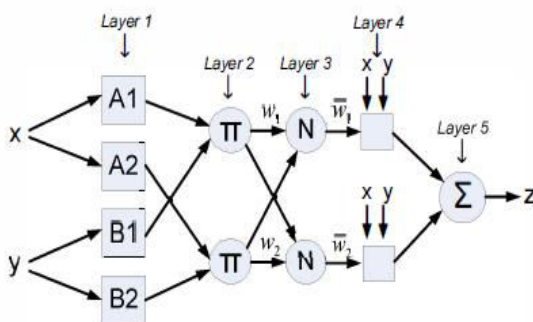


Figure-2. Typical ANFIS structure with two inputs.

For first order Sugeno-type fuzzy system with only two inputs, the following two rules hold [16];

If x is A₁ and y is B₁, then f₁ = p₁x + q₁y + r₁

If x is A₂ and y is B₂, then f₂ = p₂x + q₂y + r₂

Where p_i, q_i and r_i are the consequent parameters. In Figure 2, all square nodes are called adaptive nodes and require update of their parameters, and the circular ones are fixed nodes

If o_i^j is the output of node i in layer j [8], the function of each node is explained below;

Layer 1: Each node in this layer is an adaptive node, whose output is determined by the membership function. For node A₁ the output is given by

$$o_i^1 = \mu_{A_1}(x_i) \tag{1}$$

Where μ_{A_i} is the membership function (MF).

Depending on the complexity of the problem, many MFs are available. They include linear MF, triangular MF, Gaussian MF, trapezoidal MF pi MF and bell-shape. All these exhibits different expression and different parameters. For Gaussian MF;

$$\mu_{A_i}(x_i) = e^{-\frac{1}{2}\left(\frac{x-c_i}{\beta_i}\right)^2} \tag{2}$$

Layer 2: Output of this layer is the firing strength of all the signals entering the node from the previous layer. In other words, the output is the product of all the signals from node of the previous layer. Thus;

$$o_i^2 = w_i = \mu_{A_1} \cap \mu_{B_1} \cap \mu_{C_1} \dots \tag{3}$$

Here ∩ donates product operation

Layer 3: This is normalization layer. The output of each node here is the ratio of the node's fairing strength to the sum of all the firing strength of the nodes connected to this node, thus

$$o_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2 + \dots} \tag{4}$$

Layer 4: Output of each node in this layer is

$$o_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \tag{5}$$

Layer 5: In this layer, the only output node will sum up all the output signals of layer 4, thus

$$o_i^5 = f = \sum_i \bar{w}_i f_i \tag{6}$$

For i = 1,2, the output f is given by



$$f = \bar{w}_1 f_1 + \bar{w}_2 f_2 \tag{7}$$

$$\Rightarrow f = (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2 \tag{8}$$

Where w_i is the firing strength of the signal in node i .

2.1 Basic ANFIS training

Hybrid learning method is used to train ANFIS parameters [17]. In the forward pass LSE is used to determine the consequent parameters (p_i , q_i and r_i). In the backward pass GD is used to update the premise parameters (membership function parameters). According to [18], LSE is used in hybrid with GD because GD is generally slow, and may be trapped in local minima. Below is the explanation of the two algorithms.

2.1.1 Least square estimation

In the forward pass, the premise parameters are fixed and consequent parameters are computed using LSE. LSE is a process of estimating parameters from minimization of discrepancies between the expected value of a data and its actual value [19]. Now equation (8) can be written as

$$f = AX \tag{9}$$

Where

$$X = [p_1 \quad q_1 \quad r_1 \quad p_2 \quad q_2 \quad r_2]^T$$

and

$$A = \begin{bmatrix} \bar{w}_1 x & \bar{w}_1 y & \bar{w}_1 & \bar{w}_2 x & \bar{w}_2 y & \bar{w}_2 \end{bmatrix}$$

Here, least square estimate of x , denoted by x^* , can be used to minimize the square errors

$$\|AX - f\|^2$$

If A is invertible

$$X = A^{-1} f \tag{10}$$

Otherwise pseudo inverse of A is computed using the relation

$$X = (A^T A)^{-1} A^T f \tag{11}$$

If and only if $A^T A$ is non-singular.

This gives the consequent parameters at each cycle.

2.1.2 Gradient decent algorithm

In backward pass, consequent parameters are fixed and premise parameters are computed using GD. As presented in [20] Gradient decent is a directional optimization method that tends to minimize a given cost function. From equation (6), the estimated output is given by

$$f = \sum_i \bar{w}_i f_i$$

If for one iteration, the estimated output for k^{th} row (data point) is f_k and the actual output is b_k , the error margin can be define as

$$E_k = (b_k - f_k)^2 \tag{12}$$

Now the objective is to minimize the overall error measure define by

$$\sum_k E_k$$

K is the total number data points.

For an error signal $e_{i,l}$ corresponds to node ‘ i ’ and layer ‘ l ’

$$e_{i,l} = \frac{\partial E_k}{\partial o_{i,l}} \tag{13}$$

$o_{i,l}$ is the output of node ‘ i ’ and layer ‘ l ’. From equation (12) the error derivative of node 1, between layer 5 and layer 4 is;

$$e_4 = \frac{\partial E_k}{\partial o_{4,1}} = -2(b_k - f_k) \tag{14}$$

Between respective nodes and respective layers, the error signal between layer ‘ l ’ and ‘ m ’ can be computed using chain rule as follows:

$$\frac{\partial E_k}{\partial o_{i,l}} = \sum_{m=1}^N \frac{\partial E_k}{\partial o_{m,l}} \frac{\partial o_{m,l}}{\partial o_{i,l}} \tag{15}$$

Thus, for premise parameter α

$$\frac{\partial E}{\partial \alpha} = \sum \frac{\partial E_k}{\partial o} \frac{\partial o}{\partial \alpha}$$

From the ANFIS structure of Figure-2

$$\frac{\partial E}{\partial \alpha} = \frac{\partial E}{\partial o_5} \frac{\partial o_5}{\partial o_4} \frac{\partial o_4}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial o_1} \frac{\partial o_1}{\partial \alpha} \tag{16}$$

Where

$$\frac{\partial o_5}{\partial o_4} = \frac{\partial (\sum f_i \bar{w}_i)}{\partial f_i \bar{w}_i} = 1$$

$$\frac{\partial o_4}{\partial o_3} = \frac{\partial (f_i \bar{w}_i)}{\partial w_i} = f_i$$



$$\frac{\partial o_3}{\partial o_2} = \frac{\partial \left(\frac{w_i}{\sum_k w_k} \right)}{\partial w_i} = \frac{\sum_k w_k - w_i}{\left(\sum_k w_k \right)^2}$$

n is the total number of the fuzzy rules

$$\frac{\partial o_2}{\partial o_1} = \frac{\partial \left(\prod A_k \right)}{\partial A_m}, \text{ for } A_k \in R(A_m), A_k \neq A_m$$

A_k is the membership function grade which makes the premise part of the rule constraining the fuzzy rule A_m .

In this work Gaussian membership function is used. To update the premise parameter α (c and ∂ in this case);

$$\frac{\partial E_k}{\partial c_i} = \frac{\partial E_k}{\partial o_{1,i}} \frac{\partial \mu_{1,i}}{\partial c_i} = e_{1,i} \frac{\partial \mu_{1,i}}{\partial c_i} \quad (17)$$

And

$$\frac{\partial E_k}{\partial \sigma_i} = \frac{\partial E_k}{\partial o_{1,i}} \frac{\partial \mu_{1,i}}{\partial \sigma_i} = e_{1,i} \frac{\partial \mu_{1,i}}{\partial \sigma_i} \quad (18)$$

Therefore, for the parameter c_i , the update formula is given by;

$$\Delta c_i = -\eta \frac{\partial E_k}{\partial c_i} \quad (19)$$

The learning rate, η is given by;

$$\eta = \frac{K}{\sqrt{\sum_{i=1}^N \left(\frac{\partial E}{\partial c} \right)^2}} \quad (20)$$

N is the number of nodes in layer 1 and K is the step size. Equations (19) and (20) applied to σ_i in the same way.

2.2 Optimized ANFIS training

In this paper, PSO is proposed to train the ANFIS. To make comparison, GA is also used to train the ANFIS differently. This involve determining the premise parameters of the membership function as presented in equation (19). In both methods, these parameters are determined without passing through the differentiation processes of section 2.1.2. It is through estimation of minimum value of the error function in equation (12), and automatically update the parameters in every cycle. Such training process was discussed in sections 2.2.1 and 2.2.2.

2.2.1 ANFIS training with GA

GA is a search stochastic algorithm based on evolutionary theory, often applied to optimization problems [21]. In GA, each candidate solution (chromosome or string) has the capacity of determining future solution if it has a good fitness (optimal solution). Such solutions generate other similar and good solutions (offspring). This continue until stopping criteria is reached. The aim here is to minimize the error difference between the actual (b_k) and the predicted output (f_k) as presented in equation (12). The objective is to replace the traditional GD algorithm with GA so as to minimize equation (12) in every iteration, and propagate the error directly to layer one. This will reduce the computational difficulties associated with ANFIS and speed up the convergence of the system [13], [14].

GA produce next generation through mutation and cross-over. Parents (current solution) produce offspring (next solution) through toggle switch of certain bits.

2.2.2 ANFIS training with PSO

In PSO a particle (problem) within a solution space is moving in search of optimal solution with reference to its position (local solution) and the space position (global solution). It is presented by Kennedy and Elbert [22]. Train ANFIS with PSO was by initializing population, N , equals to the number of membership functions associated with the input vectors. For every epoch, the consequent parameters were determined using LSE. And the premise parameters (c and ∂ in this case) are calculated using PSO. The process was through the following steps:

Step 1: Initialize particles' population, $P(t)$, with the individual particle's position within the hyperspace, such that $t = 0$

Step 2: Evaluating the performance $F(.)$ (position and velocity) of each particle, through self-experience (i.e individual position $\vec{x}_i(t)$).

Step 3: Compare the current individual position with the previous positions. If $F(\vec{x}_i(t)) < pbest_i$, then

$$\left. \begin{aligned} pbest_i &= F(\vec{x}_i(t)) \\ \vec{x}_{pbest_i} &= \vec{x}(x) \end{aligned} \right\} \quad (21)$$

Step 4: Compare the current global position with the previous positions. If $F(\vec{x}_i(t)) < gbest_i$, then

$$\left. \begin{aligned} gbest_i &= F(\vec{x}_i(t)) \\ \vec{x}_{gbest_i} &= \vec{x}(x) \end{aligned} \right\} \quad (22)$$

Step 5: Then compute the velocity vector for each particle using the relation



$$\vec{v}(t) = \vec{v}_i(t-1) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + \rho_2(\vec{x}_{gbest_i} - \vec{x}_i(t)) \quad (23)$$

Where $\rho_1 = r_1 C_1$ and $\rho_2 = r_2 C_2$ are random variables. $r_1, r_2 \sim U(0,1)$ are positive acceleration constants, such that $C_1 + C_2 \leq 4$

Step 6: Move each particle to new position $\vec{x}_i(t)$, using the equation

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t), \text{ for } t = t + 1 \quad (24)$$

Considering equation (12), equation (24) can be written as

$$e_i(t) = e_i(t-1) + v_i(t) \quad (25)$$

Step 7: go to step 2 and repeat the steps until the system converges.

When the stopping criteria is reached, the final estimated error ($e_i(t)$) is used directly to update the premise parameters of the ANFIS. The update is repeated in every epoch of the ANFIS.

3. WAVELET TRANSFORM

Wavelet is used to decompose time series signal in to approximate and details components. The load series is decomposed in to low and high coefficients. This is to extract high frequencies from the load series and reduce the variation between the load data [1], [4]. The data is decomposed in to three levels, using db2 as presented in equation (25)

$$l(t) = A_3(t) + D_3(t) + D_2(t) + D_1(t) \quad (26)$$

Where $l(t)$ is the load series, $A_2(t)$ is the approximate component and $D_2(t)$ and $D_1(t)$ are detail components. Generally, a wavelet transform of time series signal is given by equation (27)

$$WT_{(a,b)} = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt \quad (27)$$

Where $\psi(t)$ is the mother wave let, a is scale factor and b is the time-shift parameter. Following the decomposition of the load series as presented in [4], it can be observed that the approximate part describes the load pattern and the details part presents the most important components of the load series. To reconstruct the load data back after the forecasting an inverse of the same wavelet was used. The expression of the inverse wavelet is given in equation (28), with the all parameters maintaining their original definition.

$$f(t) = \frac{1}{C_{\psi}^2} \int_a^{\infty} \int_b^{\infty} \psi_{\psi}(a,b) \frac{1}{a^2} \psi\left(\frac{t-b}{a}\right) da db \quad (28)$$

4. LOAD FORECASTING IMPLEMENTATION

For accurate forecast, there is need to determine the actual variables that influence the load consumption. In this work data sets from Nova Scotia region are used in the forecasting. This data is available online for public use. Being the smallest province in Canada, and not more than 67km from the ocean, the weather is being controlled by the ocean. It is therefore difficult to determine exact variables that will affect the load consumption. Even though there is little variation between the four seasons, data of the spring season was considered. Because the weather in this season is uniform compared to other seasons. Temperature, relative Humidity, Wind speed and forecasting day load, collected in the spring season are used in this experiment. Spring starts from middle of March to the Middle of June. First eight weeks (middle of March and complete April) for training and last week for testing. Meaning that two months data for training and the subsequent for testing. Data of three days (Tuesday, Wednesday and Thursday) in each week was used for both training and testing. Tuesday and Wednesday data were used to forecast Thursday. This is because the load pattern is similar over these days.

Before the forecasting exercise, a db2 of the wavelet family is used to decompose the input variables in to approximate and detail coefficients. This will reduce the volatility of the variance and reduce the effect of outliers in the forecasting data [1].

Following the processing of the forecasting data, we start by forecasting the load using classical ANFIS. The error measure and the time of the forecast are recorded. Then, the forecasting is conducted using PSO optimized ANFIS and GA optimized ANFIS. Also, the results were recorded. The results obtained are shown in section 5.

5. RESULTS DISCUSSIONS

In this experiment we consider training ANFIS with both GA and PSO differently, and compare with the basic ANFIS. This give three different models - M1, M2 and M3. All the three models are trained and tested based on the data explained in section 3.

M-1: Here we used ANFIS with its default training algorithm. That is ANFIS trained with Gradient Decent (GD) algorithm [18]. Figure-3 shows the plot of actual vs the forecasted load. An MSE of 1509, RMSE 38.84 and MAPE of 3.0% are obtained. Also, this model converged within 1541.56 seconds.

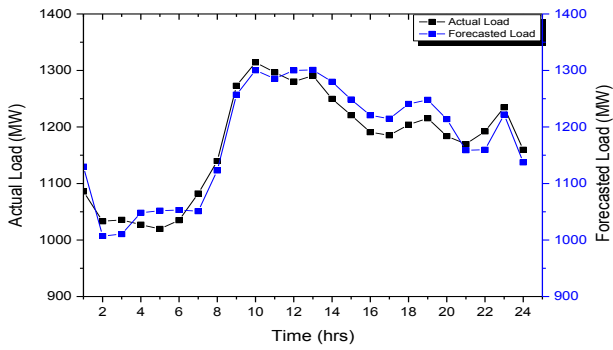


Figure-3. Forecasted and actual load for ANFIS model (M1).

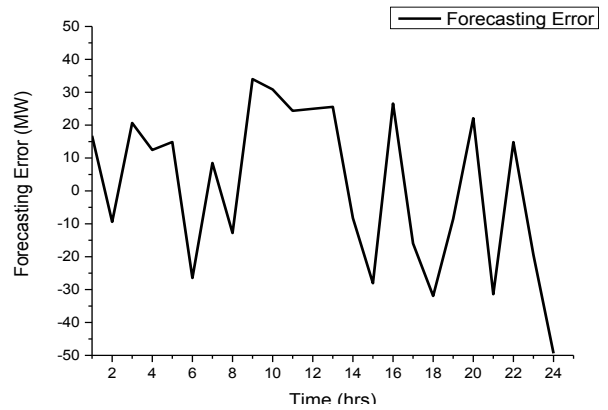


Figure-6. Plot of Forecasted errors for GA-ANFIS Model (M2).

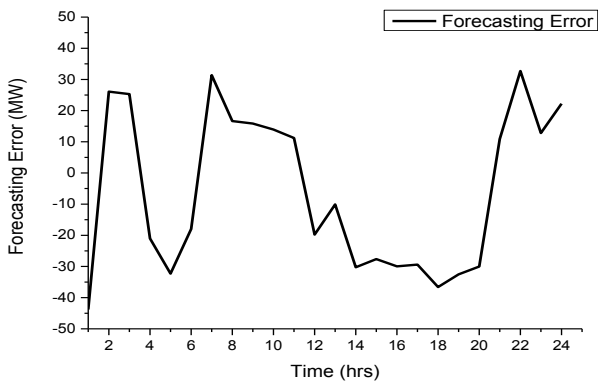


Figure-4. Plot of forecasted errors for ANFIS model (M1).

M-2: Here we used GA to train ANFIS. The GA was used in finding the variables (Δc_i in equation 19) associated with the premise part of the ANFIS. This is through replacing the GD with GA. Figures-4 is the graph of the forecasted and actual load, and Figure-5 is the forecasting error obtained. An MSE of 992.62, RMSE of 31.51 and MAPE of 2.6% are obtained. The maximum time for this model to converge was 250.47 seconds.

M 3: Here we used PSO to train the ANFIS. The PSO was used in finding the variables (Δc_i in equation 19) associated with the premise part of the ANFIS. This is also, through replacing the GD with PSO. Figures 6 is the graph of the forecasted and actual load, and Figure-7 is the forecasting error obtained. An MSE of 62.63, RMSE of 25.74 and MAPE of 2.1% are obtained. This model is faster than the other two. It converged within 223.35 seconds.

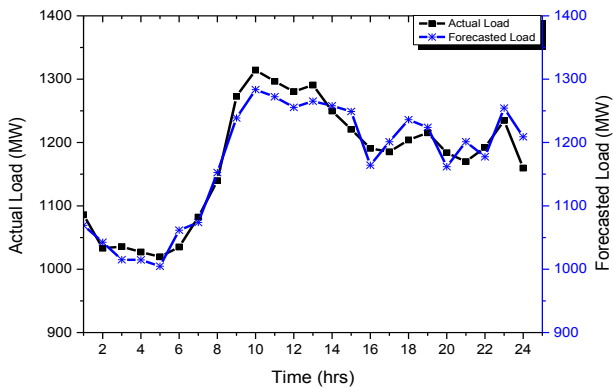


Figure-5. Plot of actual vs forecasted load for GA-ANFIS model (M2).

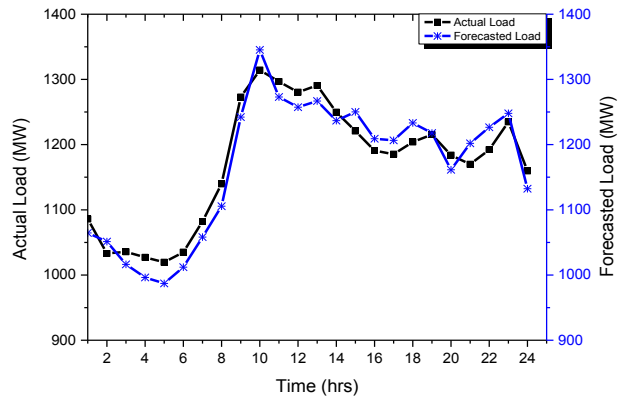


Figure-7. Plot of actual vs forecasted load for PSO-ANFIS model (M3).

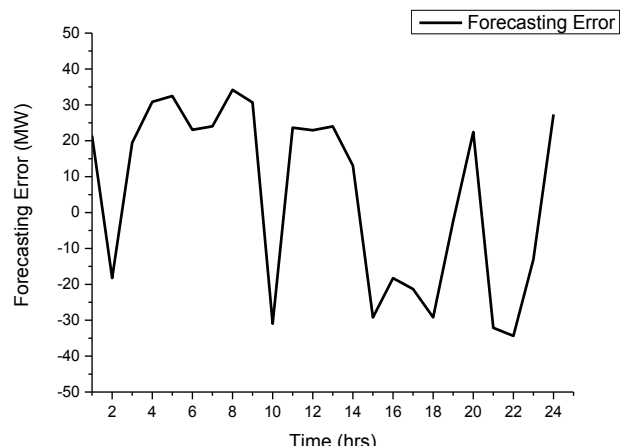


Figure-8. Plot of forecasted errors for PSO-ANFIS Model (M3).



The Absolute Percentage Error (APE) of the three models (M1, M2 and M3) are compared in the bar-graphed of Figure-9. It can be observed that M1 gave APE of about 4.0% at first hour, M2 gave APE of 4.25% at 24th hour, and M3 gave an APE of 3.25%.

The results of all the three Models is presented in Table-1. It can be observed that training ANFIS with PSO give best results and is faster in terms of convergence. This was followed by GA optimized ANFIS model and then finally the traditional ANFIS model. It is therefore important to decompose the data, or reduce the variance of the data and remove outliers before the forecasting. This will improve the accuracy of the forecasting. Also optimizing ANFIS with PSO will reduce the converging time, because all the mathematical complexity of GD are removed from the ANFIS. This improves the accuracy and speeds up the forecasting.

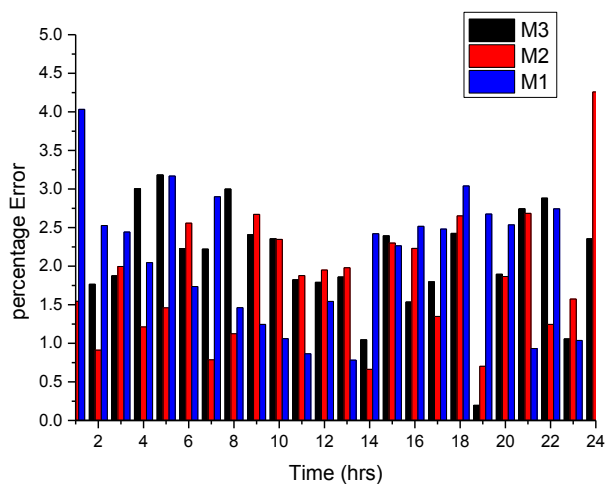


Figure-9. Comparing the absolute percentage error of the three models.

Table-1. Performance evaluation and accuracy of the three models.

Model	Error measurement			Time of convergence (sec)
	MSE	RMSE	MAPE (%)	
M1	1509.07	38.84	3.0	1541.56
M2	992.62	31.51	2.6	250.47
M3	662.63	25.74	2.1	223.35

6. CONCLUSIONS

This work focused on load forecasting using optimized ANFIS. Three different models are produced and tested using historical load and weather data. Data of Nova Scotia province during spring season was considered for both training and testing the models. Db2 wavelet was used to decompose the data in to one approximate and three details coefficients, which are used in the forecasting exercise. First model (M1) involved the use of ANFIS train with the traditional GD algorithm, second model (M2) is an ANFIS train with GA (GA optimized ANFIS)

and the last model (M3) is an ANFIS train with PSO (PSO optimized ANFIS).

Applying Wavelet transform to refine the data helped in reducing the volatility of the data variance, and reduces the outliers from the data. Among the three models, PSO optimized ANFIS model found to be more accurate and converges faster than the other three. This helps in reducing the computational complexity, which is prone to error in the traditional ANFIS, and speeds up the forecasting exercise. It is therefore necessary to refine forecasting data prior to the forecasting exercise. This will reduce the number of outliers in the data. Also, applying optimization methods in AI models is very essential in obtaining good and accurate load forecasting results.

ACKNOWLEDGEMENTS

The authors acknowledge Malaysian Ministry of Higher Education and Universiti Teknologi Malaysia for supporting this work.

REFERENCES

- [1] S. Li, P. Wang, S. Member, and L. Goel. 2015. A Novel Wavelet-Based Ensemble Method for Short-Term Load Forecasting with Hybrid Neural Networks and Feature Selection. *IEEE Trans. POWER Syst.* PP(99): 1-11.
- [2] T. G. Manohar and V. C. V. Reddy. 2008. Load Forecasting by a Novel Technique using ANN. *ARPN J. Eng. Appl. Sci.* 3(2): 19-25.
- [3] A. Baziar and A. Kavousi-fard. 2015. Short Term Load Forecasting Using A Hybrid Model Based On Support Vector Regression. *Int. J. Sci. Technol. Res.* 4(05): 189-195.
- [4] Z. A. Bashir and M. E. El-Hawary. 2009. Applying Wavelets to Short-Term Load Forecasting using PSO-Based Neural Networks. *IEE Trans. POWER Syst.* 24(1): 20-27.
- [5] A. S. Khwaja, M. Naeem, A. Anpalagan, A. Venetsanopoulos and B. Venkatesh. 2015. Improved Short-Term Load Forecasting using Bagged Neural Networks. *Electr. Power Syst. Res.* 125: 109-115.
- [6] S. Mishra and S. K. Patra. 2008. Short Term Load Forecasting using Neural Network trained with Genetic Algorithm and Particle Swarm Optimization Sanjib Mishra. In: *First International Conference on Emerging Trends in Engineering and Technology.* pp. 606-611.
- [7] Z. Yun, Z. Quan, S. Caixin, L. Shaolan, L. Yuming and S. Yang. 2008. RBF Neural Network and ANFIS-



- Based Short-Term Load Forecasting Approach in Real-Time. *IEEE Trans. Power Syst.* 23(3): 853-858.
- [8] T. Nguyen and Y. Liao. 2011. Short-Term Load Forecasting Based on Adaptive Neuro-Fuzzy Inference System. *J. Comput.* 6(11): 2267-2271.
- [9] M. Hanmandlu and B. K. Chauhan. 2011. Load Forecasting Using Hybrid Models. *IEEE Trans. Power Syst.* 26(1): 20-29.
- [10] H. Quan, D. Srinivasan, and A. Khosravi. 2014. Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals. *IEEE Trans. Neural Networks Learn. Syst.* 25(2): 303-315.
- [11] K. Kampouropoulos, F. Andrade, A. Garcia and L. Romeral. 2014. A Combined Methodology of Adaptive Neuro- Fuzzy Inference System and Genetic Algorithm for Short-term Energy Forecasting. *Adv. Electr. Comput. Eng.* 14(1): 9-14.
- [12] N. Ding, C. Benoit, G. Foggia, Y. Bésanger, S. Member, and F. Wurtz. 2015. Neural Network-Based Model Design for Short-Term Load Forecast in Distribution Systems. *IEEE Trans. Power Syst.*, vol. (in press). pp. 1-10.
- [13] M. Teshnehlab, M. A. Shoorehdeli, and A. K. Sedigh. 2008. Novel Hybrid Learning Algorithms for Tuning ANFIS Parameters as an Identifier Using Fuzzy PSO. In: *IEEE International Conference on Networking, Sensing and Control.* pp. 111-116.
- [14] V. S. Ghomsheh, M. A. Shoorehdeli, and M. Teshnehlab. 2007. Training ANFIS Structure with Modified PSO Algorithm. In: *Mediterranean Conference on Control and Automation.* pp. 1-6.
- [15] A. T. Azar. 2010. Adaptive Neuro-Fuzzy Systems. In *Fuzzy Systems*, no. February, A. T. Azar, Ed. Rijeka: In Tech. pp. 85-110.
- [16] M. S. Gaya, N. Abdul Wahab, Y. M. Sam and S. I. Samsudin. 2014. ANFIS Modelling of Carbon and Nitrogen Removal in Domestic. *J. Teknol.* 67(5): 29-34.
- [17] J. R. Jang, C.-T. Sun and E. Mizutani. 1997. *Neuro-Fuzzy and Soft Computing.* NJ: Prentice Hall.
- [18] J. R. Jang. 1993. ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Trans. Syst. Man Cybern.* 23(3): 665-685.
- [19] B. S. Everitt and D. C. Howell. 2005. Least Squares Estimation. in *Encyclopedia of Statistics in Behavioral Science*, Volume 2., B. S. Everitt and D. C. Howell (Eds.). Chichester: John Wiley. pp. 1041-1045.
- [20] P. Siarry, E. C. D. P. E. C. P, and V. Vignes. 1993. Gradient Descent Method for Optimizing Various. In: *Second IEEE International Conference on Fuzzy Systems.* (1): 1241-1246.
- [21] A. Gupta and P. K. Sarangi. 2012. Electrical Load Forecasting using Genetic Algorithm Based Back-Propagation Method. *ARPJ J. Eng. Appl. Sci.* 7(8): 1017-1020.
- [22] R. Eberhart and J. Kennedy. 1995. A New Optimizer Using Particle Swarm Theory. In: *Six International Symposium on Micro Machine and Human Science.* pp. 39-43.