Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(6), 150-159
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Artificial Neural Network-Based Electric Load Forecasting

Dolores De Groff, Perambur Neelakanta

Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA
Email: degroff@fau.edu

How to cite this paper: De Groff, D. and Neelakanta, P. (2025) Artificial Neural Network-Based Electric Load Forecasting. *Journal of Computer and Communications*, **13**, 150-159.
https://doi.org/10.4236/jcc.2025.136010

Received: December 29, 2024
Accepted: June 22, 2025
Published: June 25, 2025

Abstract

This paper proposes a unique approach to load forecasting using a fast convergent artificial neural network (ANN) and is driven by the critical need for power system planning. The Mazoon Electrical Company in Oman provided the real data for the study of monthly load forecasting using ANNs, which are presented in this paper. The link between past, present, and future temperatures, loads, and humidities is learned by the artificial neural network (ANN). The test ANN predicts reasonably accurate results of predicted power loads. The underlying exercise uses a traditional multilayer ANN architecture with feed-forward and backpropagation techniques in addition to a recently proposed fast-convergence algorithm that is deduced in terms of eigenvalues of a Hessian matrix associated with the input data of temperature and humidity changing over time. The anticipated results are cross verified with actual power load data obtained.

Keywords

Load Forecasting, Artificial Neural Network, Backpropagation Algorithm, Eigenvalues, Fast Learning Rate, Power System

1. Introduction

Power load forecasting using an artificial neural network (ANN) that allows for fast convergence with precise forecasts is the focus of this study. To ascertain how much power will be required at a specific moment to supply end customers and how that demand will impact the utility grid, accurate load forecasting is crucial. By using the power load forecast, waste and inefficiency may be prevented and sufficient power can be made available to fulfill consumption demands.

As explained by Neelakanta and De Groff in [1], an artificial neural network (ANN) is a mathematical model that has been developed as a computational tool

based on the image of the biological brain complex. The Hessian matrix of the pertinent input data serves as the foundation for supervised training. Details provided by the Mazoon Electrical Company in Oman provide the pertinent data for this investigation. For the years 2007 and 2008, monthly load data is gathered for a specific Oman region known as Al Batinah. In other words, twenty-four months' worth of temperature and humidity data—two whole years—must be entered into the ANN during training. The maximum, minimum, and average temperatures, as well as the maximum, minimum, and average humidity, are the six inputs. Megawatts of load data are the outputs. In other words, this load data serves as the training's teacher. The intended output is comparable to that of a supervisory teacher.

The mathematical method used by the artificial neural network is one that converts data from an input space to an output space. In order to minimize the error—that is, the difference between the network's actual output vector and the intended output vector—supervised training aims to iteratively update a set of connectivity weights that are introduced between the neural layers in the ANN.

A multi-layered feed-forward perceptron (MLFP) consisting of an input layer, one hidden layer, and one output is the test ANN used in the simulations (Figure 1). Through linked inner and hidden layers, the input values advance. In order to squash the total output to a limited level, it is fed into a nonlinear sigmoidal function. The sigmoid-compressed output is then compared with a teacher value, which stands for the intended output aim.

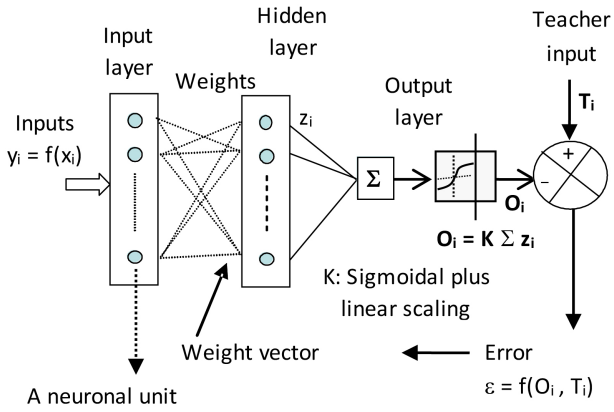


Figure 1. Test ANN architecture constructed with 6 input neuron units (NUs), 1 hidden layer with 6 NUs, 1 output unit, and hyperbolic tangent sigmoid.

A backpropagation gradient technique computes and applies the resulting error to the interconnection weights. In other words, an output, $\mathbf{O_i}$, is indicated by the sigmoid-compressed value and compared to a teacher/supervisory (reference) value, $\mathbf{T_i}$, which represents the intended output aim. After that, the error corresponding to $(\mathbf{O_i} - \mathbf{T_i})$ is backpropagated. This error is stated in terms of an error function, ε, which represents the mean-squared value of $(\mathbf{O_i} - \mathbf{T_i})$. When the error function is applied to the inter-connection weights, $\mathbf{W}_{ij}$, the backpropagation (BP) technique, usually allows a gradient based on steepest descent, which alters the weight vector

values (either increasing or decreasing).

The rth ensemble of inputs $\{y_i\}$, weighted across the input-layer (i = 1, 2, 3, ..., I = 6 units) and the hidden-layer (j = 1, 2, 3, ..., J = 6 units), is represented by the summed value ($\mathbf{\Sigma z}_i$) at the output unit in Figure 1. A hyperbolic tangent (sigmoidal) function, f(.), then squashes its linearly scaled value, $\mathbf{K \times \Sigma z_i}$ (where K is a linear-scaling constant), producing the result $\mathbf{O_i = f(K \times \Sigma z_i)}$. Additionally, the set $\{W_{ij}\}$ specifies the coefficients of weighting of the connections between the input- and hidden-layers. Additionally, as shown, the ANN's topology incorporates backpropagation and supervised learning enabled by a teacher value T_i. The gradient of an error, ε, which depends on ($\mathbf{O}_i$, $\mathbf{T}_i$), is defined by: ($\pm\Delta W_{ij}$): and the entity $\alpha \times (\pm\Delta W_{ij})$ is then applied iteratively to change the current value of W_{ij} until the error (ε) hits zero or a designated low, "stop" value. In this case, α stands for a learning coefficient that can be selected to attain a desired (quick) rate of convergence of the enforced repetition.

One effective method for training feed-forward neural networks is the backpropagation algorithm. However, it has a slow convergence rate and may produce less-than-ideal results because it updates the weights using the steepest descent approach [2]. As a result, a method that speeds up convergence is employed in this work [3] [4]. In other words, the authors have created a generalized process that, when the learning coefficient is chosen well, causes the ANN to converge to more accurate values more quickly [3] [4].

The number of iterations needed to train the net is significantly reduced when this quicker method is used. The authors have demonstrated that the greatest eigenvalue of the Hessian matrix should be inversely proportional to the learning rate α [3] [4]. There are I = 6 input units, $y = \{y_1, y_2, \ldots y_I\}$ in Figure 1. In the current example, $\mathbf{y^T y}$ is an I × I (6 × 6) square matrix, and it corresponds to the Hessian matrix. This Hessian matrix can be expressed diagonally [HD]. The Hessian matrix's symmetry results in a single, distinct eigenvalue, λ_{II}, in the diagonal form, as illustrated below (all other eigenvalues are zero):

$$[HD] = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \lambda_{II} \end{bmatrix} \quad (1)$$

To facilitate faster convergence, the learning rate (α) applied to the test ANN would be equivalent to the reciprocal of the biggest, single eigenvalue, λ_{II}, of the Hessian matrix, as mentioned before. The input data for the current study relates to each ensemble of the test power load profile under investigation. The information provided by the Mazoon Electrical Company in Oman is the basis for the power load data that is taken into consideration. The meteorological department of Oman provided the temperature and humidity data. As a result, the effectiveness of the research conducted here using the suggested ANN-based approach and the power load predictions are cross-checked against specifics of the actual power

load values.

This research predicts a monthly necessary power demand using the suggested neural network model and compares it with actual data. Using the learning rate as the reciprocal of the greatest eigenvalue of the Hessian matrix, it will be demonstrated that convergence towards accurate prediction is achieved rapidly.

2. ANN-Based Load Forecasting Method

2.1. Description of the Data Set and Motivation for the ANN to Perform Load Forecasting

As everyone knows, there is a relationship between temperature, humidity, and load characteristics. In warmer climates, more energy is required to cool. Air temperature is a necessary, but not sufficient, variable for adequately predicting electricity demand. Humidity plays an important role as well in electricity use. A combination of temperature and humidity affects necessary power load. Forecasting electric load is essential for power systems' operational planning and for preventing disruptions. Predictions from load forecasting might be short-term (for the next few hours or days) or long-term (for the next few months or years). The cost and dependability of the entire power system are strongly impacted by how accurate these projections are. Accurate load forecasting keeps the power system stable and balanced by ensuring that there is always an enough supply of electricity to fulfill demand. Utilities can also prevent the additional expenses that come with producing too much or too little electricity by using demand forecasts.

Therefore, a way to forecast electricity load is proposed here. ANNs can be used to do this task of estimating needed future power loads by using historical load data as well as historical data on temperature and humidity. The results of this study are confirmed by comparing them with real data, and it will be shown the ANN model used was able to predict correctly and give closely matching numbers on monthly power loads. The ANN used was successfully able to spot a pattern between inputs of temperature and humidity and output power load.

The ANN ability to learn from experience (existing data) makes this method very useful in forecasting power load. An underlying relationship between the inputs and the outputs is assumed by ANN forecasting [5]. The MLFP ANN used in this work has an inherent capability of arbitrary input-output mapping, and this makes it successful in forecasting power load.

2.2. Prescription of the Teacher Value

The data used to train the neural network is taken from the Mazoon Electrical Company, Oman [6]. Load data for twenty-four months (years 2007 and 2008) are collected for a particular region called Al Batinah in Oman. This monthly load data was then used as teacher values for the ANN (after normalization). The information that is inputted into the input layer of the neural network comprises normalized historical weather information on humidity and temperatures over the same twenty-four months. The test ANN architecture shown in **Figure 1** needs

a teacher value (T_i) in both training as well as in prediction phase. It is prescribed as follows: Month 1 data consisting of 6 normalized inputs (maximum temperature Tmax, minimum temperature Tmin, average temperature Tavg, maximum humidity Hmax, minimum humidity Hmin, average humidity Havg) is paired with a teacher value consisting of month 1 power load requirements Loadavg. Similarly, month 2 data consisting of 6 corresponding normalized weather inputs is paired with a teacher value consisting of month 2 power load requirements. Weather information for twenty-four months is thus inputted into the ANN and is paired with corresponding power load output requirements. In this method, the ANN is trained to spot the monthly pattern between weather parameters (maximum temperature, minimum temperature, average temperature, maximum humidity, minimum humidity, average humidity) and the required load output power.

Table 1 summarizes the initial data procured on weather conditions of temperature and humidity over the twenty-four-month period.

Table 1. Initial data procured on temperature and humidity and power load.

Month/year	Tmax ˚C	Tmin ˚C	Tavg ˚C	Hmax	Hmin	Havg	Loadavg MW
Jan/07	30	12	21	89	25	57	96.65
Feb/07	36	16	26	94	17	55.5	94.95
March/07	36	17	26.5	95	10	52.5	115.85
April/07	42	16	29	95	10	52.5	199.6
May/07	44	26	35	88	7	47.5	247.8
June/07	46	24	35	98	15	56.5	192.95
July/07	46	28	37	98	16	57	258.9
Aug/07	45	27	36	92	20	56	247.4
Sept/07	43	25	34	98	11	54.5	228
Oct/07	38	20	29	98	12	55	173.2
Nov/07	61	18	39.5	94	22	58	121.5
Dec/07	33	16	24.5	89	33	61	96.65
Jan/08	29	13	21	98	28	63	106.725
Feb/08	36	11	23.5	88	18	53	84.225
March/08	38	17	27.5	99	8	53.5	143.075
April/08	43	21	32	94	9	51.5	208.925
May/08	45	23	34	95	9	52	266.525
June/08	48	29	38.5	92	9	50.5	296.175
July/08	45	27	36	94	8	51	270.1
Aug/08	46	25	35.5	98	25	61.5	237.15
Sept/08	44	26	35	99	15	57	278.05
Oct/08	41	24	32.5	93	14	53.5	256.025
Nov/08	35	18	26.5	94	21	57.5	166.95
Dec/08	30	14	22	99	39	69	109.575

Weather and load-normalized parameters are shown in Table 2.

Table 2. Normalized initial data procured on temperature, humidity and power load.

Month/year	Tmax ˚C	Tmin ˚C	Tavg ˚C	Hmax	Hmin	Havg	Load MW
Jan/07	0.492	0.414	0.532	0.89	0.641	0.826	0.270
Feb/07	0.590	0.552	0.658	0.94	0.436	0.804	0.266
March/07	0.590	0.586	0.671	0.95	0.256	0.761	0.324
April/07	0.689	0.552	0.734	0.95	0.256	0.761	0.558
May/07	0.721	0.897	0.886	0.88	0.179	0.688	0.693
June/07	0.754	0.828	0.886	0.98	0.385	0.819	0.540
July/07	0.754	0.966	0.937	0.98	0.410	0.826	0.724
Aug/07	0.738	0.931	0.911	0.92	0.513	0.812	0.692
Sept/07	0.705	0.862	0.861	0.98	0.282	0.790	0.638
Oct/07	0.623	0.690	0.734	0.98	0.308	0.797	0.485
Nov/07	1	0.621	1	0.94	0.564	0.841	0.340
Dec/07	0.541	0.552	0.620	0.89	0.846	0.8841	0.270
Jan/08	0.475	0.448	0.532	0.98	0.718	0.913	0.299
Feb/08	0.590	0.379	0.595	0.88	0.462	0.768	0.236
March/08	0.623	0.586	0.696	0.99	0.205	0.775	0.400
April/08	0.705	0.724	0.810	0.94	0.231	0.746	0.584
May/08	0.738	0.793	0.861	0.95	0.231	0.754	0.746
June/08	0.787	1	0.975	0.92	0.231	0.732	0.829
July/08	0.738	0.931	0.911	0.94	0.205	0.739	0.756
Aug/08	0.754	0.862	0.899	0.98	0.641	0.891	0.663
Sept/08	0.721	0.897	0.886	0.99	0.385	0.826	0.778
Oct/08	0.672	0.828	0.823	0.93	0.359	0.775	0.716
Nov/08	0.574	0.621	0.671	0.94	0.538	0.833	0.467
Dec/08	0.492	0.483	0.557	0.99	1	1	0.307

2.3. Prescription of the Learning Coefficient and Adjustment of the Weights

To update the interconnection weights, the backpropagated error is applied using the formula w (new) = w (old) ± μ × (dε/dw). The weight coefficients $\{w_{ij}\}$ should be able to adopt values that allow the output error, ε to rapidly converge to zero (or a predetermined extremely small, "stop" value) at the required learning rate. To achieve a rapid convergence of the net, the relevant learning rate is inferred [3] [4].

In the present study, each input data set consists of six (normalized) weather parameters (as listed in Table 2), and constructing a corresponding transpose $[\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_6}]^T$, a [6 × 6] Hessian matrix [**H**] is specified as follows: [**HD**] = [$\mathbf{y_1}, \mathbf{y_2}, \ldots,$

$\mathbf{y_6}]^T$ $[\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_6}]$, A symmetric square [(I = 6) × (I = 6)] matrix with one non-vanishing eigenvalue as a diagonal element is referred to by the equivalent [**HD**]. In other words, there is only one non-vanishing eigenvalue left for every ensemble data set.

Table 3. Computed Hessian eigenvalues, λ_{66} for the ensemble data sets {r = 1, 2, …24}.

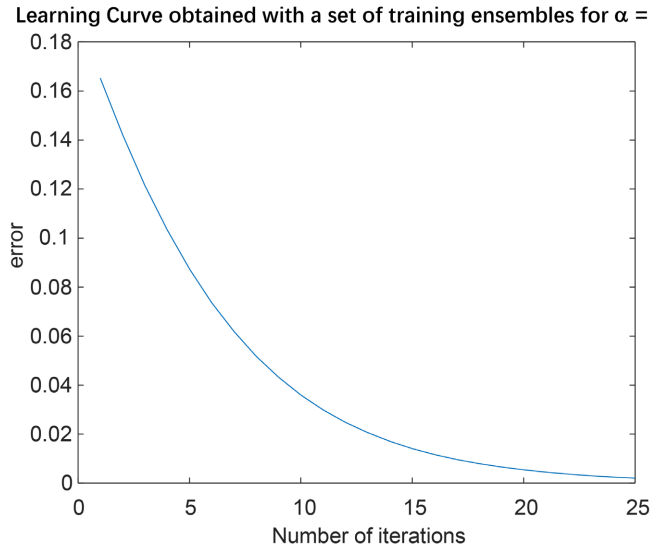
r	1	2	3	4	5	6	7	8	9	10
λ_{66}	3.77	3.03	3.41	3.92	4.28	3.71	4.00	3.44	3.17	2.84
r	11	12	13	14	15	16	17	18	19	20
λ_{66}	2.42	3.02	3.27	4.29	3.09	3.64	4.01	4.19	3.82	3.39
r	21	22	23	24						
λ_{66}	2.86	2.69	2.81	2.58						

As indicated earlier, the desired learning rate can be set inversely proportional to the maximum eigenvalue of the data set. From data in Table 3, this maximum eigenvalue, λ_{max}, occurs for the ensemble, r = 14 and has a value, λ_{max} = 4.2936. Hence, corresponding learning rate α is: $(1/\lambda_{max}) = 0.2329$.

3. Results and Discussion

3.1. Learning Curves

The learning rate used in the training plans mentioned above is $\alpha = 1/\lambda_{max}$. By determining pertinent learning curves, it is shown how effective it is to use $(\alpha = 1/\lambda_{max})$ for the learning rate (instead of the conventional, arbitrarily defined number, say, $\alpha = 0.001$) to achieve a fast convergence. Figure 2 shows examples of these comparisons using learning curves derived for an exemplary training ensemble with r = 1. By first determining an appropriate value for the learning rate based on $\alpha = 1/\lambda_{max.}$, you can see that the convergence rate is significantly faster.



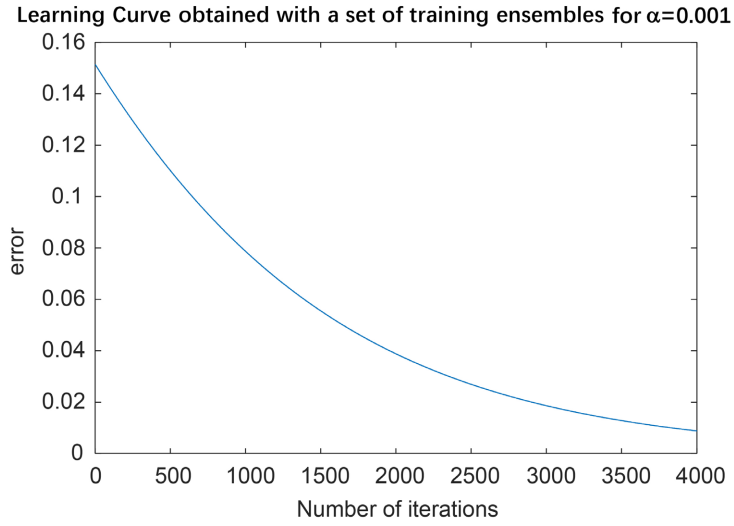


Figure 2. Learning curves obtained with the set of training ensemble r = 1 using an arbitrary learning coefficient, $\alpha = 0.001$ and adopting an optimum value of $\alpha = 0.2329$.

3.2. A Note on Learning Coefficient

Selecting a learning coefficient (also known as learning rate) in an ANN involves finding a value that allows the network to learn effectively by updating its weights without getting trapped in local minima or diverging due to excessively large, iterative steps of updates. Mostly, the choice of learning coefficient is done by trial and error with a small value in the beginning and gradually increasing it to an optimal value by monitoring the training loss on a validation set.

In order to facilitate a rapid convergence towards a desirable level of output prediction, it is suggested that the learning rate (α) applied to the test ANN, should correlate with the reciprocal of the biggest, single eigenvalue, λ_{II} of the aforementioned Hessian matrix. Choosing the largest, single eigenvalue, λ_{II} is specific to enforcing the derivative of e to zero as required at the global minimum. The Hessian representation specifies the average over all inputs of $\mathbf{y}^T\mathbf{y}$, with $\mathbf{y}^T$ being the transpose of $\mathbf{y}$ in multidimensional cases. Additionally, the cost surface's form is represented by the Hessian, whose eigenvalues indicate how steep the surface is along the curvature directions; a steep curvature is indicated by a big eigenvalue. As such, the learning rate being inversely proportional to this large eigenvalue implies an optimally small value towards attaining the global minimum needed. Relevant single learning rate chosen thereof will not cause any divergence along the steep directions (specifically pertinent to the large eigenvalue direction). In essence, denoting λ_{max} as the largest eigenvalue of the Hessian matrix, the learning rate of the order of $(1/\lambda_{max})$, would lead to optimal convergence towards the global minimum.

In all, the choice of learning rate as above is based on the gradient descent optimization with an enforced architecture of the test ANN to have identical number of neurons in the input and first hidden layer so that the resulting interconnection weight coefficients are represented by a symmetric square matrix; and, the corre-

sponding Hessian format is diagonalized to obtain the highest eigenvalue as above.

The learning rate is a hyperparameter that controls the extent of changing the model, in response to the estimated error each time the model weights are updated in the iteration implied. If the learning rate is too small, it may result in a long training process and a too large a value may result in learning a sub-optimal set of weights too fast, that is, causing an unstable training process. Thus, the learning rate is the most important hyperparameter when configuring the neural network. Therefore, the proposed method is devised to offset the traditional intuitive guess or trial-and-error approach in prescribing learning rate, with a simple architectural symmetric connectivity between equal number of neurons taken at the input and the first hidden layer. The efficacy of relevant convergence performance is verified with data pertinent to predicting power load details in an actual electric power system.

3.3. Predicted ANN Output Values

The following Table 4 shows the predicted load for January 2009. The input data was for the previous 24 months (prior to January 2009).

Table 4. Predicted load compared to actual load power.

Month	Actual load	Predicted load	Percentage error
Jan 2009	103.7 MW	101.3 MW	2.3%

As can be seen in Table 4, the predicted results are very close to actual historical data.

4. Conclusions

An encouraging use of ANN for load forecasting has been shown in the current study. The outcomes, which are displayed in Table 4, are extremely encouraging. The data confirms that the ANN predicts future power loads with a high degree of accuracy. Table 4 shows that the ANN prediction of power load closely matches the historical measured values.

A list of the study's main findings is as follows:

- Forecasting electric load is an application for ANN. The findings demonstrate that the ANN can effectively interpolate between training sets' load and weather pattern data to produce future load patterns.
- It was discovered that the convergence rate can be accelerated by first determining an appropriate value for the learning rate based on ($\alpha = 1/\lambda_{max}$), where λ_{max} is the maximum eigenvalue of the corresponding Hessian matrices of the input data.
- The ANN also converges with slightly more accurate values when λ_{max} is used.
- The enhancement offered here can be regarded as a worthwhile and practical substitute for employing an arbitrary learning rate.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Neelakanta, P.S. and De Groff, D. (1994) Neural Network Modeling: Statistical Mechanics and Cybernetic Perspectives. CRC Press.

[2] Magoulas, G.D., Vrahatis, M.N. and Androulakis, G.S. (1999) Improving the Convergence of the Backpropagation Algorithm Using Learning Rate Adaptation Methods. *Neural Computation*, **11**, 1769-1796. https://doi.org/10.1162/089976699300016223

[3] De Groff, D. and Neelakanta, P. (2018) Faster Convergent Artificial Neural Networks. *International Journal of Computers & Technology*, **17**, 7126-7132.

[4] De Groff, D. and Neelakanta, P. (2018) Predicting Hurricane Direction and Intensity *via* a Fast Convergent Artificial Neural Network. *American Journal of Engineering Research*, **9**, 219-227. http://www.ajer.org/papers/Vol-7-issue-7/X0707219227.pdf

[5] Zhang, G., Eddy Patuwo, B. and Y. Hu, M. (1998) Forecasting with Artificial Neural Networks: The State of the Art. *International Journal of Forecasting*, **14**, 35-62. https://doi.org/10.1016/s0169-2070(97)00044-7

[6] Swaroop, R. and Abdulqader, H. (2012) Load Forecasting for Power System Planning and Operation Using Artificial Neural Network at Al Batinah Region Oman. *Journal of Engineering Science and Technology*, **7**, 498-504.