

Electric Energy Forecasting by WEB-Based Method

Balanthi Beig, Majid Poshtan and Rajesh Ramanand

The Petroleum Institute, Abu Dhabi, U.A.E.
mposhtan@pi.ac.ae

Abstract

The paper presents a web-based system to forecast the demanded Electric Energy in power systems. The first method used was the linear regression method to find the slope and y-intercept, and then uses those values to predict the load for the next time interval. The second method was a back propagation neural network. This method learns patterns from the existing data set, and the trained network is then used to predict future values of the load. Overall, this program is a practical program for load forecasting for planning and operating of power systems. It has also the advantage of an easy access to the system load from the Internet for operating centers. This feature will give an open access to the future load for load marketing in deregulated systems. It may also help the interconnected utilities to predict the demanded load from each other with a reasonable error.

1. Introduction

This paper presents web-based load forecasting for power systems. Two methods, linear regression and artificial neural networks, were used to predict the load. The regression method helps to find the best-fit line for the training data, so as to calculate the slope and the y-intercept, and then use these values to forecast the load for any new input to the system. The artificial neural network provides a one-step-ahead load prediction after learning with a set of data. Load forecasting is of main concern to smooth power system operation and planning. As a result, it is required that load forecasting systems provide a better prediction value and that the non-linear relationship between input and output variables is clarified. In recent years, intelligent load forecasting is quite popular due to the good approximation of non-linear functions.

The rest of the paper is broken down as follows. First the linear regression approach to power load forecasting is presented. In this the idea behind linear regression, our implementation of this method, and then some screen shots of the working of the program are shown. The next section describes the neural network approach power load forecasting. Finally, some results from the two approaches and comparison of the output from both of them is presented.

2. Linear Regression

Linear regression is a method to best fit a linear equation (straight line) of the form $y(x) = \mathbf{a} + \mathbf{b}x$ to a collection of N points (x_i, y_i) . x is usually known as the independent variable, and y as the dependent variable. \mathbf{b} is often referred to as the slope and \mathbf{a} as the y axis intercept.

The result will be stated below without derivation that requires minimization of the sum of the squared distances from the data points and the proposed line. This function is minimized by calculating the derivatives with respect to \mathbf{a} and \mathbf{b} and setting these to zero. The solution is clearer if we define the following:

$$s_{xx} = \sum_{i=0}^{N-1} (x_i - \bar{x})^2 \quad (1)$$

$$s_{yy} = \sum_{i=0}^{N-1} (y_i - \bar{y})^2 \quad (2)$$

$$s_{xy} = \sum_{i=0}^{N-1} (x_i - \bar{x})(y_i - \bar{y}) \quad (3)$$

Then,

$$\text{slope (b)} = \frac{s_{xy}}{s_{xx}} \quad \text{and} \quad \text{y intercept (a)} = \bar{y} - b \bar{x} \quad (4)$$

And finally, the regression coefficient is:

$$r = \frac{s_{xy}}{\sqrt{s_{xx} s_{yy}}} \quad (5)$$

The regression coefficient is zero if there is no linear trend, and one for a perfect linear fit. Note that:

- This discussion assumes there is no known variance for the x and y values. There are solutions which can take this into account; this is particularly important if some values are known with less error.
- The solution above requires that the slope is not infinite, S_{xx} is not zero.

3. Artificial Neural Networks

Back propagation is the basis for training a supervised neural network. Static back propagation is used to produce an instantaneous mapping of a static (time-independent) input to a static output [1–5]. These networks are used to solve static classification problems such as load forecasting.

A) Description

To give a short explanation of back propagation, all back propagation methods use an application of the chain rule for ordered partial derivatives to calculate the sensitivity that a cost function has with respect to the internal states and weights of a network. In other words, the term back propagation is used to imply a backward pass of error to each internal node within the network, which is then used to calculate weight gradients for that node. Learning progresses by alternately propagating forward the activations and propagating backward the instantaneous errors. We will explain how back propagation is derived.

1. The error for a pattern p is half of the sum of the squares of the deviations of the activation of each output unit j from the target for that unit
 2. The change in the weight on the connection from unit i to unit j for pattern p should be proportional to the negative of the slope of the error with respect to that weight. Eta is the learning rate.
 3. Using the chain rule, we can express the partial derivative as a product of partial derivatives. I_{pj} is the input to unit j for pattern p.
 4. Using the formula for input, the second derivative in (3) simplifies to the activation of unit i.
 5. We define delta for pattern p and unit j thus.
 6. We can then express the weight change in terms of the delta for the destination unit on the connection, and we only need to figure out how to calculate delta for different units.
 7. Using the chain rule again, we express delta as a product of partial derivatives.
 8. The second of the derivatives in (7) is just the derivative of f, the activation function for the unit.
 9. For the first derivative in (7), there are two cases: where j is an output unit and where it is a hidden unit. For the output unit case, the derivative is just the difference of the activation and target.
 10. This is (7) for output units.
 11. For hidden units, the chain rule was used. Using (5), the expression simplifies to the negative of the sum of the products of the deltas of the units in the layer above j and the weights connecting j to units.
 12. This is (7) for hidden units.
- Thus, like any classifier, the network learns from a given set of training data, and then uses this trained network to predict the values for any new test data.

B) The proposed Implementation

The proposed implementation of the back propagation neural network for power load forecasting is designed in C++, therefore it was not built in PHP.

4. Demonstration

A) Working System

For this method, the developed software uses MySQL and PHP. MySQL was the database used to store the data from past months and PHP was the language used to write our linear regression program. This could have also been developed in C++, but the primary objective of using PHP was to be able to put this on the web. This gave us access to our information from any computer that had access to the Internet.

B) Data Representation

The data values were stored in a database in a table called load-forecast. The table was created in MySQL and can be found on our website. PHP (using SQL commands) was then used to import the values from the database and linear regression was performed on them. A few screen shots of the working of the proposed program are shown below.

The system consisted of eight options and is discussed independently.

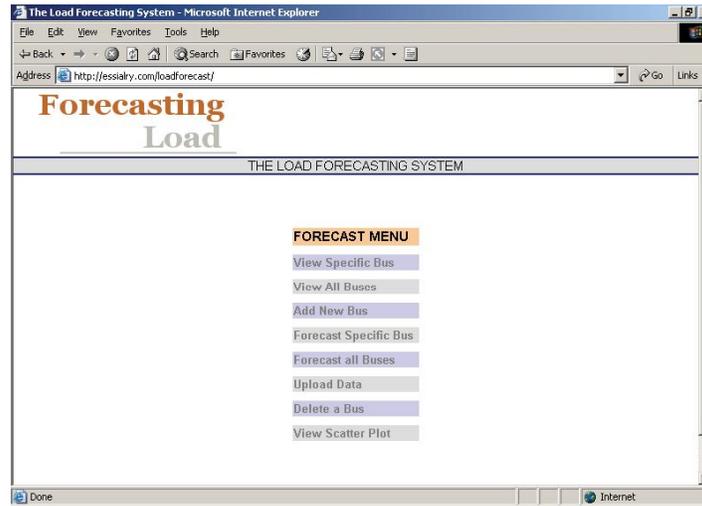


Figure 1: Load Forecasting Main Menu.

C) View Specific Bus

This was used to display the power load factor values for twelve months for any particular bus. Clicking on “View Specific Bus” gave the following menu:

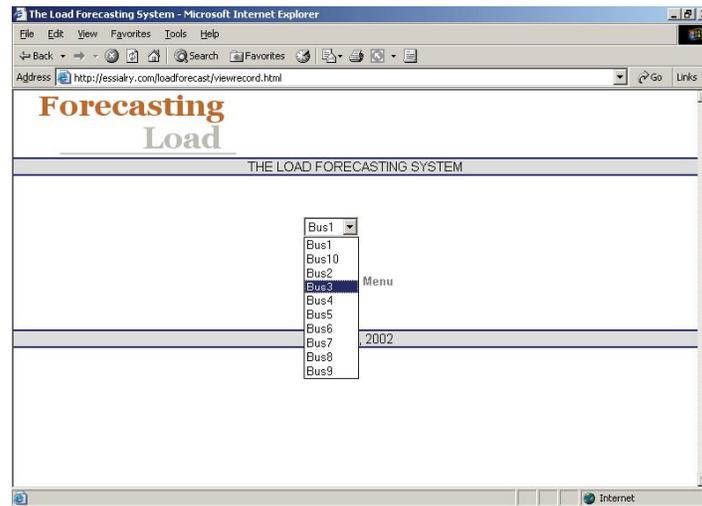


Figure 2. View Specific Bus.

Now if we chose Bus 3, the load values for this bus were returned (Figure 3).

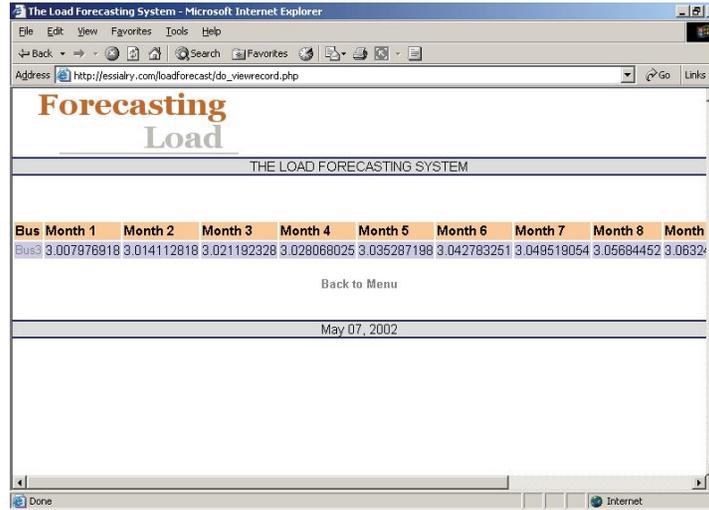


Figure 3. View Load for Specific Bus.

D) View All Buses

In this option, the same task was performed as Option 1, except that it returned the load values for all buses instead of a specific bus (Figure 4).

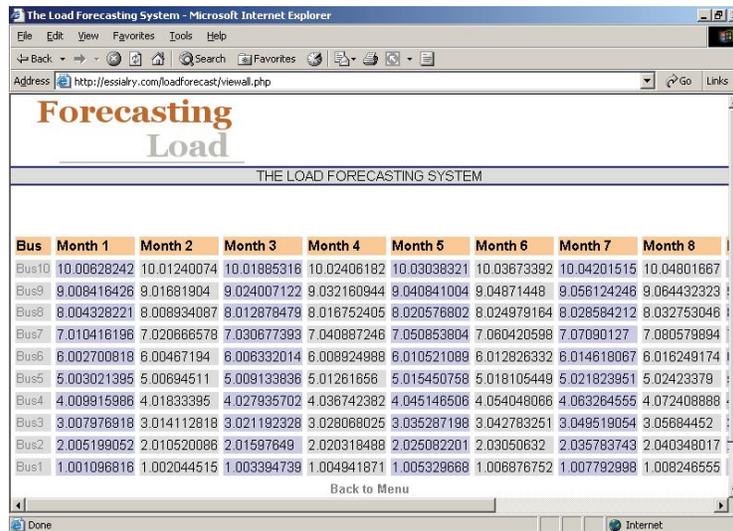


Figure 4. View All Bus.

E) Add A New Bus

This menu option allowed the user to add a new bus or node to the database. This option is required to grow the system. Over time, as the number of buses increases, it is needed to constantly update our database. This permits such an update to take place.

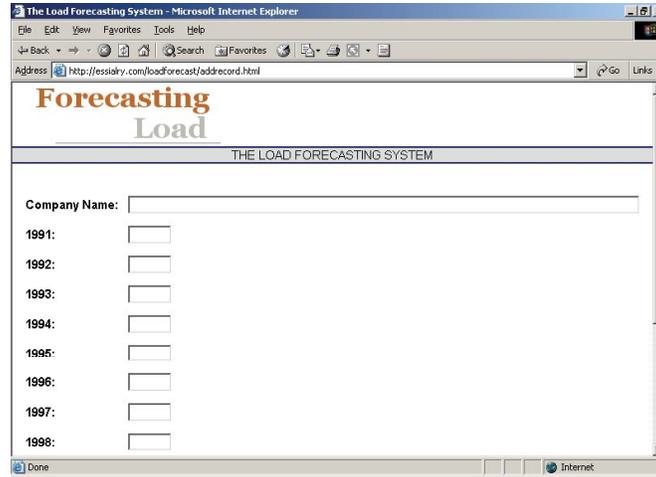


Figure 5. Add New Bus.

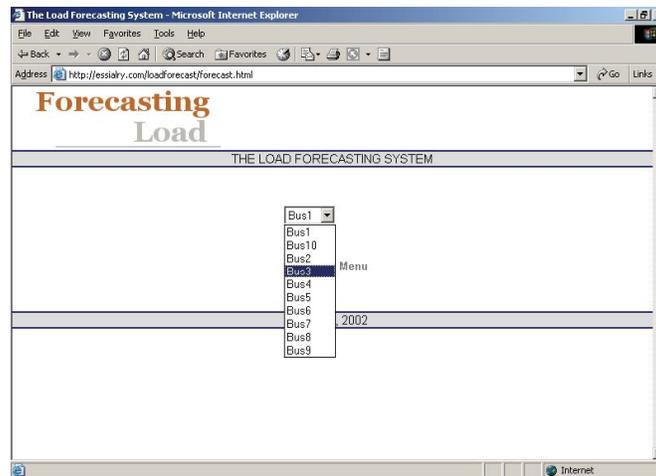


Figure 6. Forecast Specific Bus.

F) Forecast Specific Bus

This option allows the user to forecast load for the next interval, say 13th month for any specific bus (Figure 7).

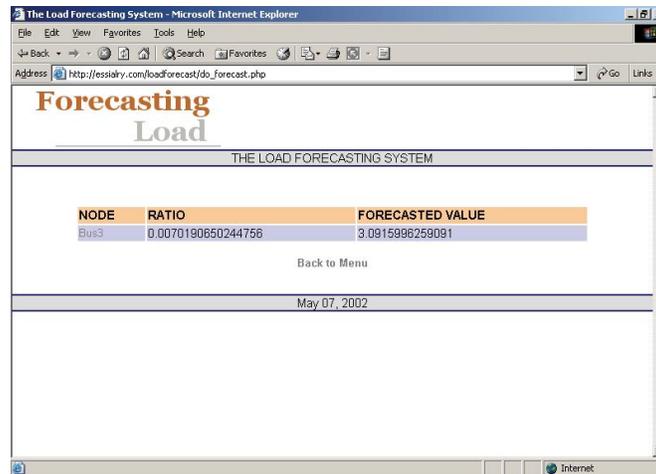


Figure 7. Load Forecast for Bus 3.

G) Forecast a Set of Buses

This option does the same thing as Option 4 except that it forecasts load values for a set of buses in the database. It sorts the output in the order of fastest growing bus (Figure 8). It also gives the user an option to graph the values to see it in a pictorial format. The graph (Figure 9) shows both the growth ration of a node and the forecasted value.

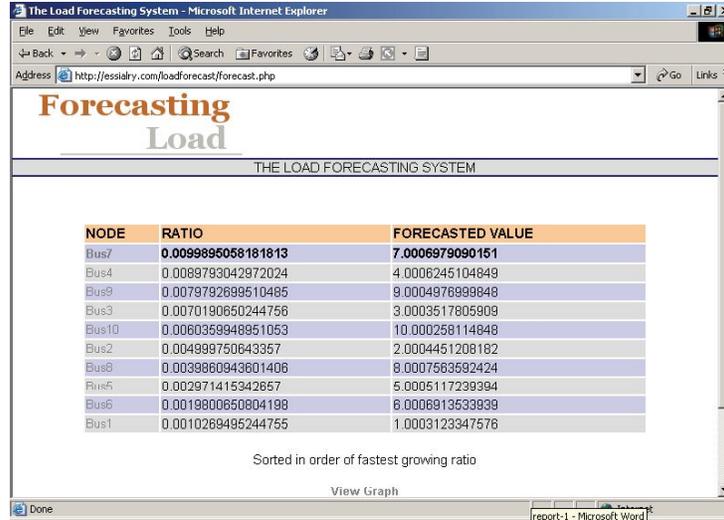


Figure 8. Forecasting a Set of Buses.

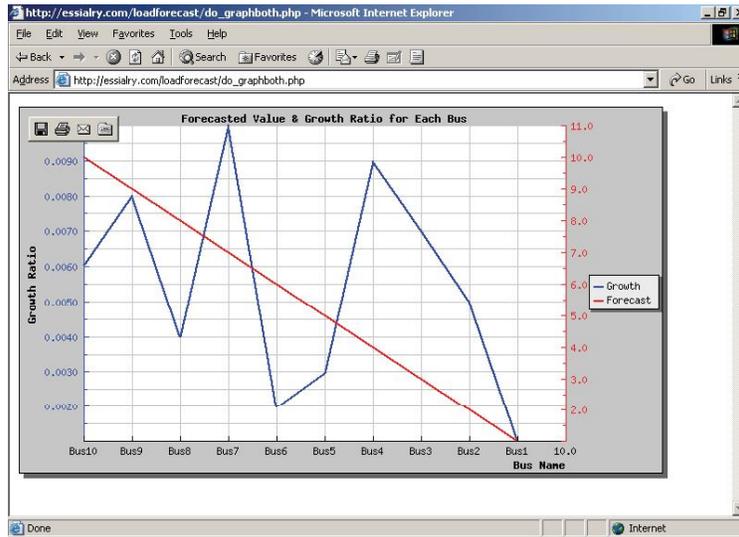


Figure 9. Graphical load growth.

This developed graphing scheme lets the graph be plotted dynamically each time a new value is entered or a modification is made to the existing value. It means if a new bus is added, there is no need to manually go into the program and add values to our graph. It will automatically pull the values from the database and graph it to give the correct graph.

H) Upload Data

This is also another useful feature of the proposed program that not only the user already have a menu option to add data in manually, by typing the values for each bus and then adding it to the database, but also upload the data from a file. The program is capable of reading the data from a text file in ASCII format and upload it to the table in the database (Figure 10).

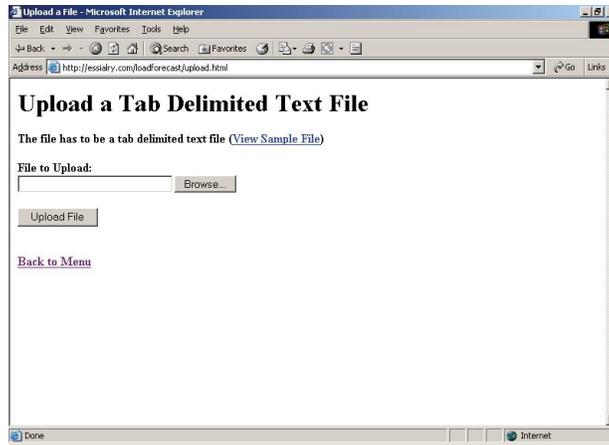


Figure 10. Uploading a data file.

I) Deleting a Bus

This is an essential option of any database to allow the user to delete a particular bus from the database. Figure 11 shows a selection for deleting a bus.

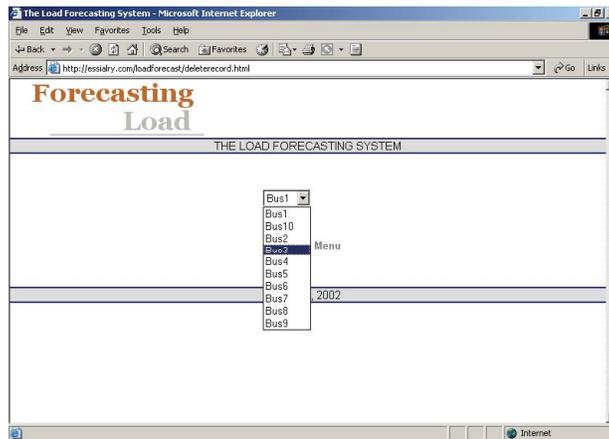


Figure 11. Deleting a Bus.

J) View Scatter Plot

This option will allow the user to view the scatter plot of any particular bus. This was particularly useful if the user wanted to see the pattern of the load factor values for a period of time for a specific bus (Figure 12).

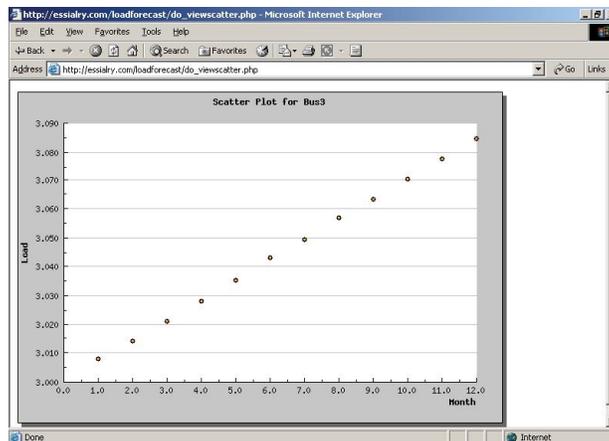


Figure 12. Scatter plot.

K) Neural Network Verses Linear Regression

The result of the forecasted values using linear regression (LR) and a back propagation neural network (NN) are compared in Figure 12.

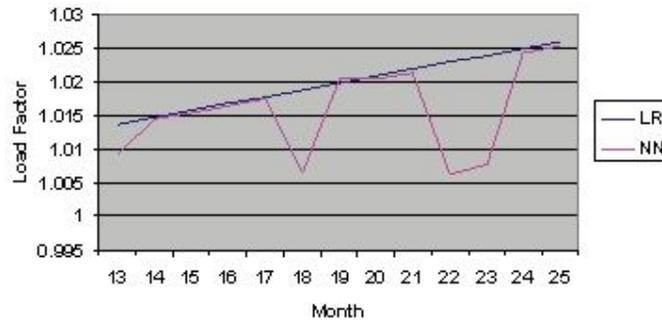


Figure 13. Load forecasted by two methods, LR and NN.

5. Conclusions

The proposed Web-based load forecasting can be used by utilities to have an easy access to the load data in future to manage the generation in interconnected systems. The easy and fast access to the Internet will allow the utilities to predict the amount of power needed to be sold/bought from the grid. There are many methods to predict the load including linear regression and artificial neural network, however the linear regression method is easier to be implemented in the Internet.

6. References

- [1] K.S. Swarup and B. Satish, "Integrated ANN approach to forecast load," IEEE Computer Applications in Power, Vol. 15, No. 2, April 2002, pp 46 -51.
- [2] T. Senjyu, H. Takara, K. Uezato, T. Funabashi, "One-hour-ahead load forecasting using neural network", Power Systems, IEEE Transactions, Vol. 17, No. 1, Feb. 2002 , pp. 113 –118.
- [3] H.S. Hippert, C.E. Pedreira, R.C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," Power Systems, IEEE Transactions, Vol. 16, Feb 2001, pp. 44 –55.
- [4] A.K. Sinha, "Short term load forecasting using artificial neural networks," Industrial Technology 2000. Proceedings of IEEE International Conference, Vol. 1, 2000, pp. 548 –553.
- [5] L. Xiaoping and Z. Hongxing , "Short-term load forecasting using neural networks combined with linear models", Intelligent Control and Automation, 2000, Vol. 3 ,2000, pp. 1633 -1637.

Author Biographies

Dr. Majid Poshtan was born in Tehran, Iran and received his B.S.in Electrical Engineering from Tehran University, Tehran, Iran, in 1988, the M.S. degree in Electrical Engineering from the University of New Brunswick, Fredericton, NB, Canada, in 1992 and the Ph.D.in Electrical Engineering from Tulane University, New Orleans, LA, U.S.A., in 2000. He is currently an Assistant Professor in the Department of Electrical Engineering, The Petroleum Institute, Abu Dhabi, U.A.E. Before joining the Institute, Dr. Poshtan has worked in different electric power projects in Entergy Corp., U.S.A.

Dr. Abdul Rahiman Beig (SM'06, M'92) was born in Mangalore, India in 1968. Dr. Beig received the B.E. degree in electrical and electronics engineering from the National Institute of Technology Karnataka, Suratkal, India, in 1989, the M.Tech, and Ph.D. degrees in electrical engineering from the Indian Institute of Science, Bangalore, in 1998 and 2004, respectively. He is currently an Assistant Professor in the Department of Electrical Engineering, The Petroleum Institute, Abu Dhabi, U.A.E. Before joining The Petroleum Institute, Dr. Beig was the member of the faculty of Electrical Engineering at the National Institute of Technology Karnataka Suratkal, India. From 1989 to 1992, he was with M/S Kirloskar Electric Company, Ltd., Mysore, India, as a Research and Development Engineer in the drives group. His research interests include ac drives and multilevel inverters. Dr. Beig received the Innovative Student Project Award for his Ph.D. work from the Indian National Academy of Engineering in 2005, and L&T-ISTE National Award for the his M.E. thesis from the Indian Society for Technical Education in 1998.