

SELECTION OF HIDDEN LAYER NEURONS AND BEST TRAINING METHOD FOR FFNN IN APPLICATION OF LONG TERM LOAD FORECASTING

Navneet K. Singh — Asheesh K. Singh — Manoj Tripathy *

For power industries electricity load forecast plays an important role for real-time control, security, optimal unit commitment, economic scheduling, maintenance, energy management, and plant structure planning *etc.* A new technique for long term load forecasting (LTLF) using optimized feed forward artificial neural network (FFNN) architecture is presented in this paper, which selects optimal number of neurons in the hidden layer as well as the best training method for the case study. The prediction performance of proposed technique is evaluated using mean absolute percentage error (MAPE) of Thailand private electricity consumption and forecasted data. The results obtained are compared with the results of classical auto-regressive (AR) and moving average (MA) methods. It is, in general, observed that the proposed method is prediction wise more accurate.

Key words: load forecasting, feed forward neural network (FFNN), auto-regressive (AR) method, moving average method

1 INTRODUCTION

In recent years, especially in last decade, after deregulation of electricity market, the electric load profile is led to non-linear and a deteriorating or chaotic time series is observed. As the time is passing year by year, the electrical load demand is increasing continuously in a non-uniform fashion. The conventional resources of power, on the earth, are limited and therefore, other options are being searched to fulfill the future load demand. These new resources are called non-conventional resources. Besides it the conventional resources are also under research so that they may work optimally with maximum power demand output. Load forecasting is not only used to know future load, but also for various electrical engineering applications associated with this as listed in Tab. 1. The literature survey exhibits that even 1 % forecasting error may results in hundred of thousand or even more currency units [1, 2]. Besides it, in the fast developing countries the load demand is varying drastically. No planning of power system is possible without forecasting. Therefore, it becomes necessary to forecast the load demand with least error so that power supply industries may make proper arrangements [3, 4]. Keyhani and El-Abiad were the first to present very short term load forecasting using Auto-regressive Moving Average (ARMA) model [5] in 1975. The classical approaches [6–12] used for forecasting electrical demand were regression method; statistical tools forecasting, autoregressive models, Moving Average (MA) models, ARMA, Autoregressive Integrated Moving Average (ARIMA) model, ARMA model with exogenous variables (ARMAX), *etc.* Basically the time series fore-

casting method is subdivided into two basic parts. One is conventional statistical/mathematical models and other one is modern artificial intelligence (AI) based models [13]. Recent research provides a proof that AI techniques are drawing more attention of the researchers in the area of load forecasting as they produce better results over the classical models. In general, there are five main categories [16, 17] of load forecasting. These categories along with their applications are documented in Tab.-1.

The ANN has undoubtedly emerged as promising AI tool due to its ability to extract the implicit non-linear relationship among input-output variables when learning from training data. Besides it ANN has good generalization ability. ANN is used for myriad problems in the areas of pattern recognition, curve fitting, data mining, load forecasting, system/process modeling, controls, system identification and many more problems of nonlinear fitting [14] as it does not require any complex mathematical formulations or quantitative correlation between inputs and outputs. In last twenty years a lot of research is put on the platform of electric load forecasting. Daneshi *et al* [15] introduced electric load and corresponding load factor model and stated different types of parameters influencing the load i.e. weather, time, population, economic and demographic data, energy supply and price, regional development, facilities, sales of electrical appliances and some other random disturbances.

Year by year the load pattern becomes highly mismatched due to regularly increasing annual demand as industrial load is increased more in past years. Environmental disturbances i.e. average atmospheric temperature increase creates a great nonlinearity [18–20]. Demo-

* Department of Electrical Engineering, Motilal Nehru National Institute of Technology Allahabad, 211004 India, navneet_dei@rediffmail.com, navneet.dei@gmail.com

Table 1. Main categories of load-forecasting with its applications

LF	Duration of Load Forecasting	Applications of Concerned Forecasting
VSTLF	Few minutes to an hour ahead	1. Modern protective relays and circuit breakers operation 2. Inrush current stabilizers 3. Security assessment 4. Economic dispatching 5. Real-time control and security evaluation 6. Sensitivity analysis of electrical equipments
STLF	Hourly forecasts for one day to one week ahead	1. Optimal generator unit commitment 2. Fuel allocation 3. Maintenance scheduling 4. buying and selling of power 5. Economic scheduling of generating capacity 6. Scheduling of fuel purchases 7. Security analysis 8. Short-term maintenance scheduling 9. Load flow estimations
MTLF	A few months to 1 year	1. Maintenance scheduling 2. Load dispatching coordination 3. Setting of prices, to meet the demand with fixed capacity
LTLF	1 year to 4–5 years	1. System planning 2. Purchasing of generating units 3. Staff hiring
VLTLF	More than 5 years	1. Scheduling construction of new generating capacity 2. Searching for renewable resources 3. Environmental policies planning 4. Staff Recruitment

LF = Load Forecasting
VSTLF = Very Short Term Load Forecasting
STLF = Short Term Load Forecasting
MTLF = Medium Term Load Forecasting
LTLF = Long Term Load Forecasting
VLTLF = Very Long Term Load Forecasting

graphic variables like population, living status, per person income, economic development, religious values *etc* were also found to put their effect over load forecasting [21, 22]. The demand for accurate load forecasting is increased in deregulated environment because of enhanced competition among power industries.

Selecting an optimized architecture of the ANN is a tedious task as a small structure does not exhibit a proper non-linear relationship among input and output variables. However, a large structure creates a complex structure and takes a lot of time in training and may not properly generalized for ill-posed problems. In the literature it is illustrated that there is no exact procedure to select the number of hidden layer and number of neurons in hidden layer. However, some of the researchers have proposed some thumb rules for selecting the number of neurons in the hidden layer [15]. The paper discusses a new methodology of selecting number of neurons in hidden layer of FFNN.

In Section 2, basic description of ANN architecture and detailed discussion of proposed methodology to select

optimal ANN architecture as well as suitable training method used for forecasting are presented. The case study along with the conventional AR and MA method [23] is described in Section 3. Besides it the Section 3 carries all the simulative information. Forecasting results and their comparison are presented in Figs. 10–14 and Tab. 4. The conclusion is then given in Section 4.

2 ARTIFICIAL NEURAL NETWORK

2.1 Architecture

An ANN is a dataflow-model, made up of non-linear computing units, and motivated by the complex connections of cells in a biological brain. The ANN provides an alternative computational approach inspired by the biological nervous systems process [24]. It has advantages viz. adaptive in nature, capability to handle complex systems and iterative learning process by adjustments of their weights and biases *etc*.

A specific type of ANN, *ie* fully connected FFNN [25] with a single hidden layer is exercised in the work presented here. The proposed FFNN has three layers, namely, input, hidden and output layers. Numbers of neurons in input layer (N_i) and output layer (N_o) layers are correspondingly selected on the basis of number of inputs available in one input vector and output required, whereas the nonlinearity of the system decides the number of neurons in the hidden layer (N_h). The data flow starting from input layer and traversing through a hidden layer arrives at the output layer. The adjustable parameters of ANN, which provide different mappings from input to output, are called weights. ANNs have the ability to learn by adjusting weights to be estimated by iterative training algorithm based on mathematical rules. The most used architecture is the so-called multi-layer, and the back-propagation algorithm is the most frequent algorithm for the training of the multi-layer networks [25].

A basic structure for exercising and experimentation of FFNN is depicted in Fig. 1. Here, matrix \mathbf{W} and \mathbf{V} contains weights of hidden layer and output layer, respectively; to predict the i^{th} month load $\mathbf{I}(i)$, historical load is used as input in a specific sequence as shown in Appendix 1. Inputs are indexed from $i - 1$ upto $i - 61$. If input $\mathbf{I}(i - 61)$ is unavailable then $\mathbf{I}(i - 60)$ is treated as the last (seventeenth) input. Intermediate outputs of the network used are calculated using equation

$$P_q^N = f\left(\sum_{k=1}^z P_k^{N-1} w_{qk} + b[N - 1] \times w_{qk[N-1]}\right). \quad (1)$$

Here N is the number of hidden layers in FFNN; P_q^{N-1} is the input at the q^{th} neuron of the input layer, if $N = 1$; P_q^N is the summed output of the q^{th} neuron of the N^{th} layer; y is the output of the q^{th} neuron of the output layer, if N is the last hidden layer; w_{qk} is the weight

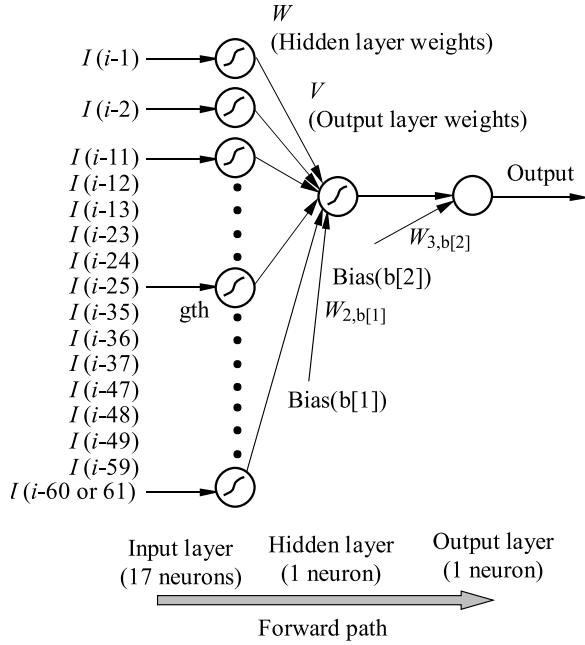


Fig. 1. Basic structure of feed forward neural network

between the q^{th} neuron of the N^{th} layer and the k^{th} neuron of the $(N-1)^{\text{th}}$ layer; z is the number of neurons in the $(N-1)^{\text{th}}$ layer; $b[N-1]$ is the bias value at the $(N-1)^{\text{th}}$ layer.

In this paper the primary attention is paid towards the selection of neural network (NN) architecture, especially the number of neurons in the hidden layer neurons and training method. The specific pattern of data assures that using FFNN, forecasting can be done with higher accuracy. In this paper the FFNN is selected in such a way that the forecasted results are as exact as real load.

2.2 Methodology proposed: FFNN architecture and number of hidden layer neuron selection

Network type, architecture, number of neurons in different layers and the training method are the key factors for a good mapping and hence results in good forecasting using FFNN. Proper input-output sequence arrangement of the training data is mandatory for accuracy in forecasting. For the case study FFNN as shown in Fig. 1 is used. It consists of three layers; the first layer contains seventeen neurons, which are fixed from the specified input sequence as listed in Appendix 1, hidden layer contains one neuron because this led to lesser prediction error. Output layer contains one neuron as single month data is predicted for one input vector. As it is obvious that the data can never contains a negative value so it will be beneficial to normalize the data in between zero and one using equation (2).

$$d = \frac{(s - s_{\min})}{(s_{\max} - s_{\min})} \quad (2)$$

where S_{\max} and S_{\min} are the maximum and minimum value from the data series used for training of NN respec-

tively, S is the value which is to be normalized and d is the normalized value between 0 and 1.

In this work, the training data are kept in a specific sequence such that these 72 months data makes a column matrix. The input selection scheme is given in Appendix 1. For forecasting data mean absolute percentage error (MAPE), given in equation (3) is used [26]. Non-absolute error value may lead to ambiguity. For better realization of forecasting results absolute error is selected, because in certain conditions, mean may be zero, for wrongly fitted data as positive errors are compensated with the negative errors. In literature MAPE is found to be the most suitable base for the same.

$$MAPE = \frac{1}{M} \sum_{k=1}^M \frac{|X_o - X_f|}{X_o} \times 100 \quad (3)$$

where M represents number of forecasted data to calculate error, K is index, X_0 shows original value and X_f expresses forecasted value corresponding to X_0 .

To reduce the MAPE, the architecture of FFNN should be optimized. For selecting the N_h of a 3 layer FFNN, researchers used a relation [27, 28] that is equation (4). It gives an approximation to select number of neurons in hidden layer (N_h).

$$N_h = \sqrt{N_i \times N_o} \quad (4)$$

In the proposed scheme a wide range of number of neurons in hidden layer *ie* 1 to 30, and training methods(TM) [29] *ie* seventeen as tabulated in Appendix 2, are picked up to incur more accurate and promising architecture of FFNN. The step-wise sequence of the proposed approach, in scientific order, is evinced in pictorial form in Fig. 2.

In Fig. 2, various parameter used are number of layers in FFNN (N_L), index (i and j) for the loop which are set to be at 0 initially. N_i and N_o are number of neurons in input and output layer respectively. MAPEmin is the value of minimum MAPE, N_{hmin} is the value of N_h when MAPE is minimum. N_{hold} and TM_{old} show the value of N_h and TM in previous loop respectively. TMmin is the value of TM when minimum MAPE is achieved.

The neural network is trained and tested by normalized data which are spitted into three parts that are training data, validation data and testing data. For this intent, data from January 2000 to December 2006 is used. Primarily selected TM is Levenberg-Marquardt (LM) due to its efficient and fast convergence [30] towards the goal. In this deal, keeping the LM training method fixed, value of N_h is varied and hence, value of N_{hmin} is obtained. In succession, keeping value of N_h constant value of TM is varied from 1 to 17 as demonstrated in Appendix 2. This practice will lead to find new value of N_h . If the current value of the current variable is found to be the same as previous value, next to it, recent value of TM as well as N_h are selected to be the final value and the stopping criteria of either value of TM or the N_h to be same is met, the optimized architecture of the FFNN is selected and trained. Further the forecasting is performed for months January 2007 to December 2009.

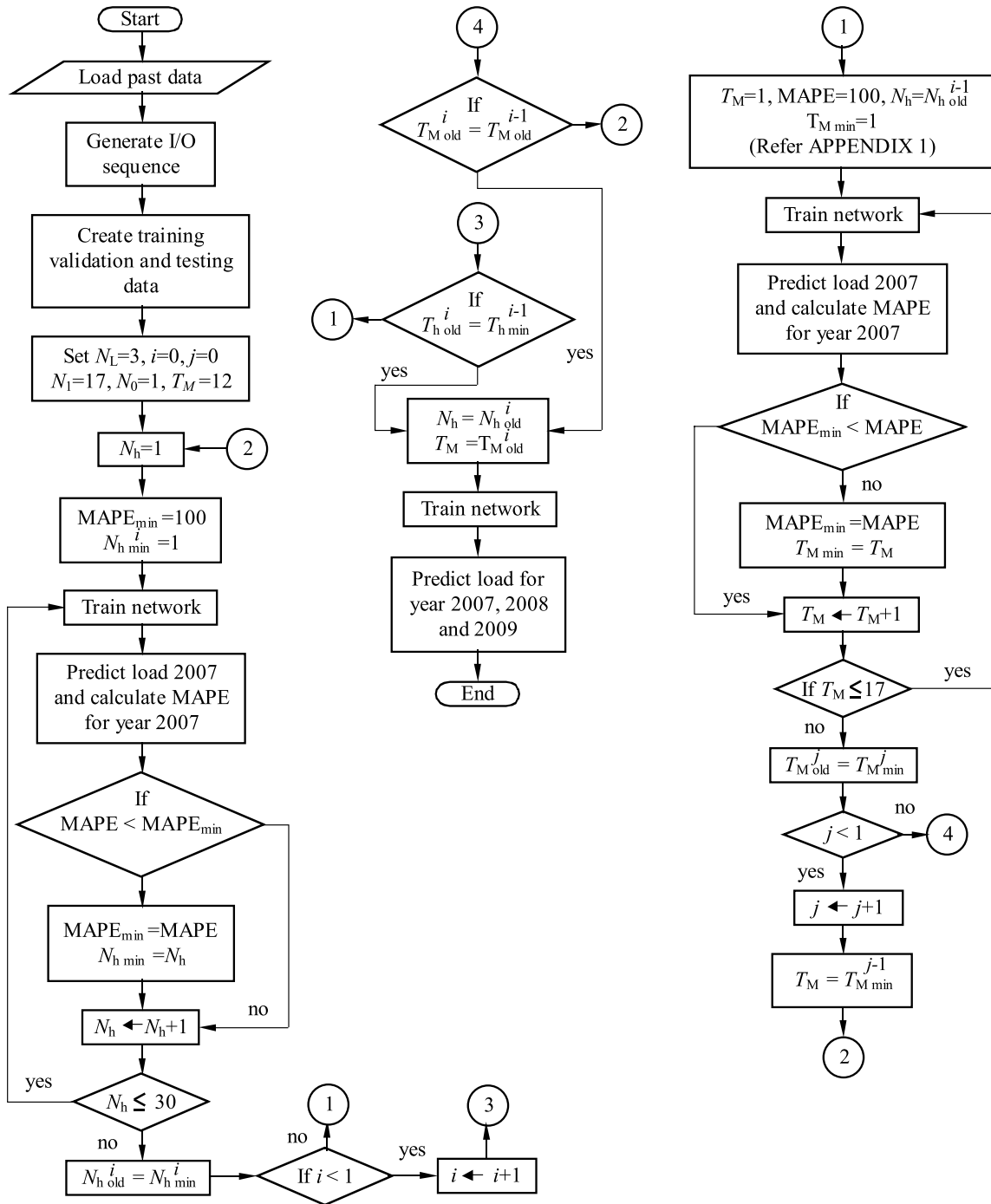


Fig. 2. Flow-Chart of the proposed methodology

3 CASE STUDY, CLASSICAL MODELS AND SIMULATION

3.1 Case study

To investigate the effectiveness of the proposed method a study is performed on 120 months data representing private consumption (in Million KWHrs) of duration January 2000 to December 2009 at Thailand [31]. Data of interval January 2000 to December 2006 is utilized for training, validation, testing and remaining data is used

for comparison with the simulated data. The trained network is tested for forecasting the data from January 2007 to December 2009, assuming this data as unknown. Finally, this forecasted data is compared with the original data.

3.2 Auto-regressive(AR) model [32]

Compact Mathematic form of AR model is

$$\phi(B)Z_t = a_t. \quad (5)$$

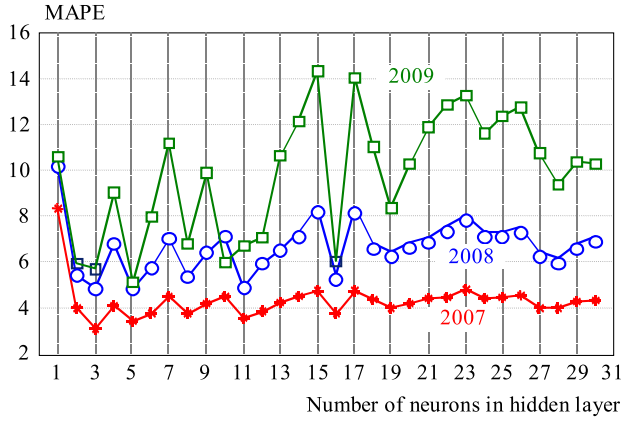


Fig. 3. MAPEs plot with respect to hidden layer neuron keeping $TM = LM$

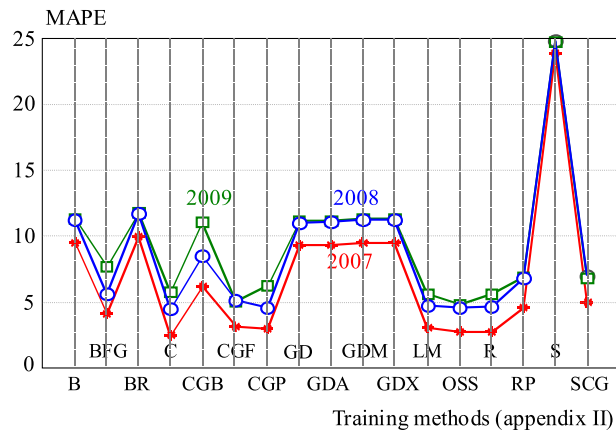


Fig. 4. MAPEs plot with respect to training methods keeping $N_h = 3$

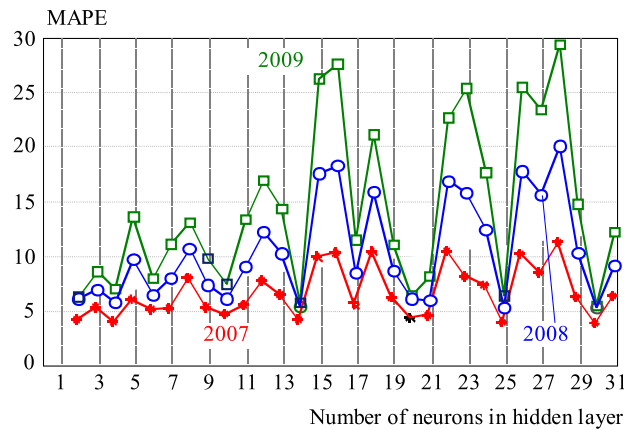


Fig. 5. MAPEs plot with respect to hidden layer neuron keeping $TM = C$

Here,

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p, \quad (5a)$$

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t, \quad (5b)$$

$$B^m Z_t = Z_{t-m}, \quad (5c)$$

where Z_t is the parameter value at instant t , B is lag operator, a_t is random shock for p previous values from stochastic time series are selected for forecasting.

3.3 Moving Average Method [23]

In MA method it is assumed that the next predicted value will be the average of some past values. In case of 6 months MA prediction six recent values of the past data is used and the new value is forecasted using equation (3). Similarly 10 and 12 months MA prediction can be done using the same equation. With the help of equation (3) load is predicted and shown in Fig. 13.

$$L_i^C = \frac{L_{(i-C)}^C + L_{(i-C+1)}^C + \dots + L_{(i-2)}^C + L_{(i-1)}^C}{C}. \quad (6)$$

Here, L_i^C is the MA (in this case energy consumption in Million KWh) of past C months data for the i^{th} month.

3.4 Observations During Simulation

According to the proposed methodology, specified in Section 2.2, the graphical results after every round are exhibited in Tab. 2 and the graphical diagrams are depicted in Figs. 3 to 8.

Figure 3 shows that minimum MAPEs reached in the diagram are 3.02 %, 4.76 % and 5.04 % for N_h equals to 3, 3 and 5 neurons for year 2007, 2008 and 2009, respectively. This exhibits that combination with $N_h = 3$ is optimum.

In Fig. 4 minimum MAPEs are shown as 2.46 %, 4.48 % and 4.83 % for training methods C, C and OSS as detailed in Appendix 2, for year 2007, 2008 and 2009 respectively. This exhibits that with $N_h = 3$, optimum results can be achieved through C method.

Figure 5 explains that minimum MAPEs are obtained in the diagram are 2.40 %, 4.06 % and 4.29 % for N_h equals to 29, 24 and 29 neurons for year 2007, 2008 and 2009, respectively. This shows, $N_h = 29$ is working to be optimum for the case study.

With minimum MAPEs 2.26 %, 3.85 % and 4.29 % keeping N_h to be 29 corresponding to SCG, SCG and C for year 2007, 2008 and 2009, respectively. Figure 6 displays optimum results with SCG method.

Observing Fig. 7, a conclusion is obtained that $N_h = 1$ is the best architecture when, training method is SCG and minimum MAPEs corresponds to year 2007, 2008 and 2009 are 2.05 %, 3.44 % and 4.63 % for hidden neurons to be 1, 25 and 1, respectively.

For the last round when, stopping criteria is met, is depicted through Fig. 8. Here keeping N_h to be 1 various training methods are examined as listed in Appendix 2 and it is found that SCG with minimum MAPEs as 2.05 %, 3.61 % and 4.63 % for year 2007, 2008 and 2009, respectively are acquired.

The tabulated form of various observations during different experiments, are listed in Tab. 2. In this table minimum MAPE obtained and the MAPE obtained concerned

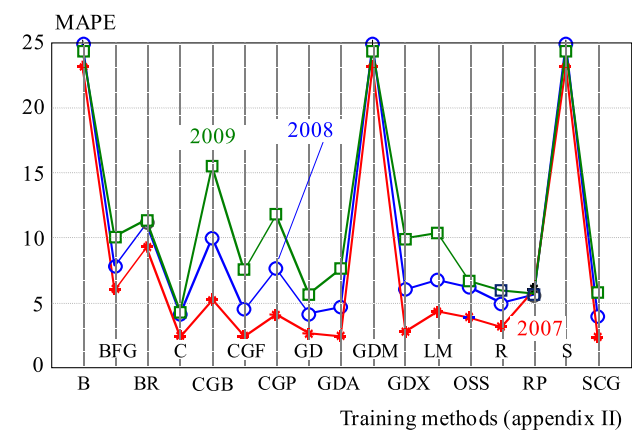


Fig. 6. MAPEs plot with respect to training methods keeping $N_h = 29$

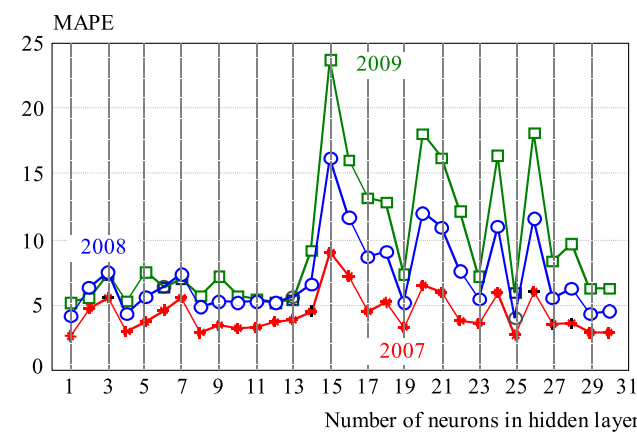


Fig. 7. MAPEs plot with respect to hidden layer neuron keeping $TM = SCG$

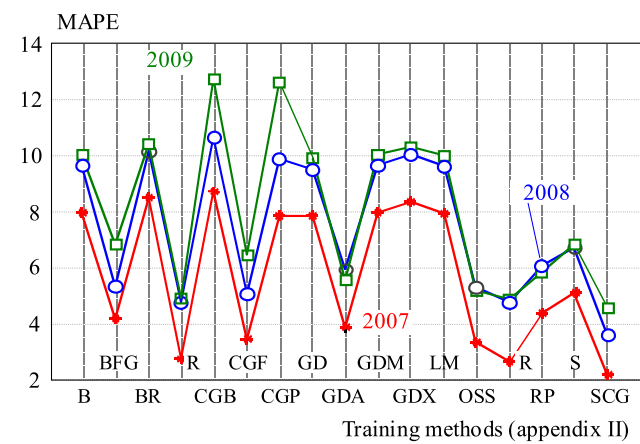


Fig. 8. MAPEs plot with respect to training methods keeping $N_h = 1$

with next round constant parameter *ie* training method or the N_h .

From the observations as detailed in Tab. 2, there were total 6 rounds to reach to stopping criteria. In the sixth round the TM is found to be same as the previous, i.e. SCG. At this stage of experiment, it is obvious that optimized structure of FFNN is 17-01-01 and should be trained with training method SCG. The upper row of a round displays the results for a specific TM or N_h for year 2007, 2008 and 2009 respectively. However, lower row presents least values of the MAPE throughout a round for year 2007, 2008 and 2009 explicitly for every year along with the corresponding N_h or TM respectively. During the experimentation of the case study, various rounds are as follows:

1. Keeping $TM = LM$, $N_h = 3$ is found to be the most efficient with minimum $MAPE = 3.02\%$ for year 2007.
2. Keeping $N_h = 3$, $TM = C$ is found to be the most efficient with minimum $MAPE = 2.46\%$ for year 2007.
3. Keeping $TM = C$, $N_h = 29$ is found to be the most efficient with minimum $MAPE = 2.40\%$ for year 2007.

4. Keeping $N_h = 29$, $TM = SCG$ is found to be the most efficient with minimum $MAPE = 2.26\%$ for year 2007.
5. Keeping $TM = SCG$, $N_h = 1$ is found to be the most efficient with minimum $MAPE = 2.05\%$ for year 2007.

Table 2. Some important observations during the process of selecting optimized FFNN architecture

		Year		
TM/ N_h *	Error Type	2007	2008	2009
$TM = LM$	$MAPE(N_h)$	3.02(3)	4.76(3)	5.60(3)
Round1	$MAPE_{min}(N_h)$	3.02(3)	4.76(3)	5.04(5)
$N_h = 3$	$MAPE(TM)$	2.46(C)	4.48(C)	5.81(C)
Round2	$MAPE_{min}(TM)$	2.46(C)	4.48(C)	4.83(OSS)
$TM = C$	$MAPE(N_h)$	2.40(29)	4.07(29)	4.29(29)
Round3	$MAPE_{min}(N_h)$	2.40(29)	4.06(24)	4.29(29)
$N_h = 29$	$MAPE(TM)$	2.26(SCG)	3.85(SCG)	5.75(SCG)
Round4	$MAPE_{min}(TM)$	2.26(SCG)	3.85(SCG)	4.29(C)
$TM = SCG$	$MAPE(N_h)$	2.05(1)	3.61(1)	4.63(1)
Round5	$MAPE_{min}(N_h)$	2.05(1)	3.44(25)	4.63(1)
$N_h = 1$	$MAPE(TM)$	2.05(SCG)	3.61(SCG)	4.63(SCG)
Round6	$MAPE_{min}(TM)$	2.05(SCG)	3.61(SCG)	4.63(SCG)

*TM=Training Method (Appendix 2);
* N_h =Number of neurons in hidden layer

Table 3. Various training parameters used to train optimized FFNN

Parameter	Value
FFNN Structure	17-01-01
Training Method*	SCG'
Epochs reached/Set	40/100
Gradient reached/Set	0.0343426/ 1×10^{-6}
MSE performance reached/Set	0.00469397/0
Stopping Mode while training	Validation Stop
Tr., Val. And Testing Data %	70%-15%-15%

*Appendix 2

6. Keeping $N_h = 1$, $TM = SCG$ is found to be the most efficient with minimum $MAPE = 2.05\%$ for year 2007.

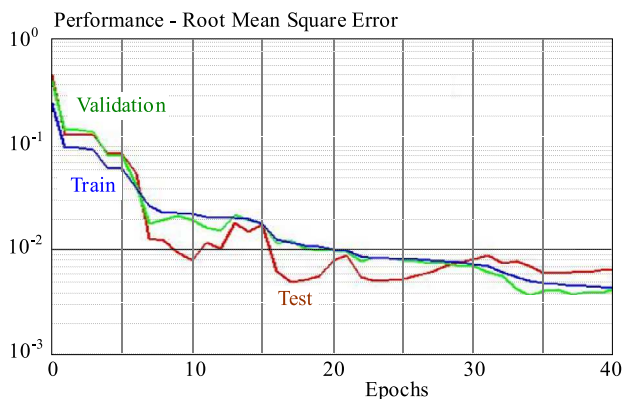


Fig. 9. Training performance curve while training the optimized FFNN with training method SCG (Appendix 2)

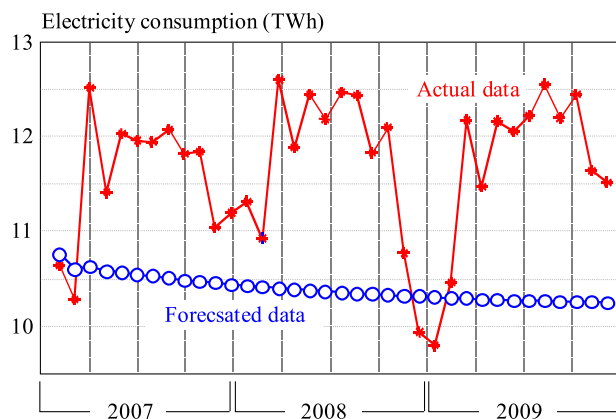


Fig. 10. Comparison between original and forecasted monthly loads for year 2007 using proposed methodology

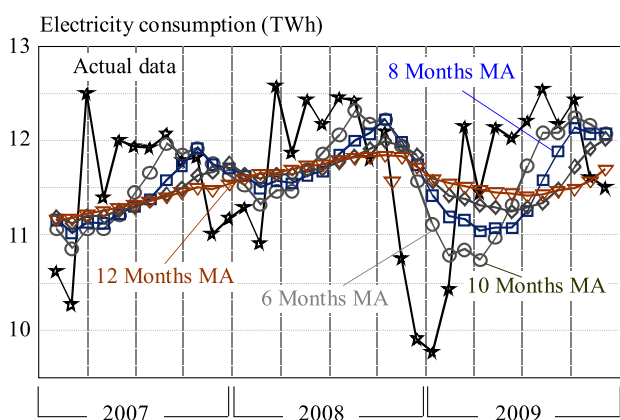


Fig. 11. Comparison between original and forecasted monthly loads for year 2007, 2008 and 2009 using MA methodology

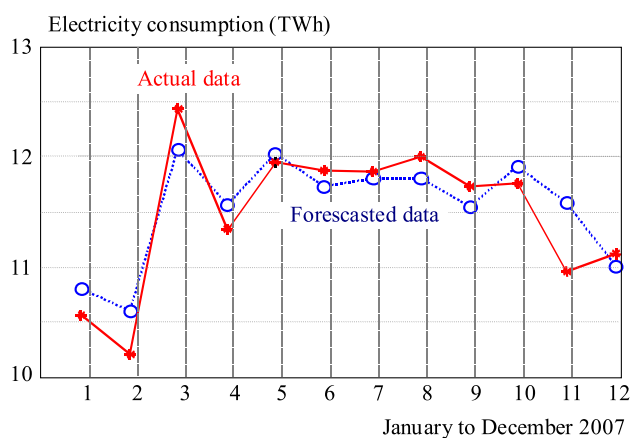


Fig. 12. Comparison between original and forecasted monthly loads for year 2007 using proposed methodology

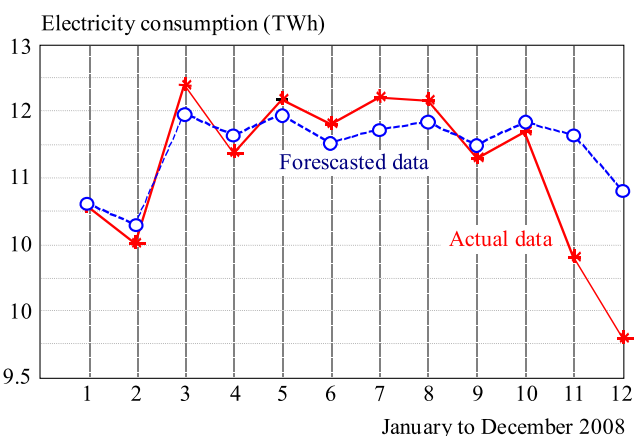


Fig. 13. Comparison between original and forecasted monthly loads for year 2008 using proposed methodology

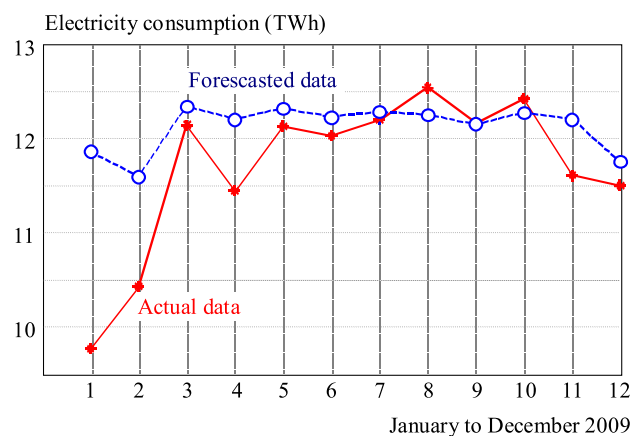


Fig. 14. Comparison between original and forecasted monthly loads for year 2009 proposed methodology

The proposed FFNN architecture is trained by SCG training algorithm as stated in Appendix 2, keeping N_h as 1 and the forecasted data for year 2006, 2007 and 2008 are demonstrated in Figs. 10, 11 and 12 in Section 4. Parameters of training, during the training of optimized FFNN architecture are mentioned in Tab. 3.

Training performance (root mean square error) curve using the training parameters, obtained is appearing in Fig. 9. It shows the performance of the training after 40 epochs to be 0.0343426.

Month-wise predicted as well as actual loads for year 2007, 2008 and 2009 are shown in Figs. 10–12. The fore-

casting results using MA method for various period averages are pointed out in Fig. 13. Similarly Fig. 14 is drawn using the actual and forecasted data obtained from AR (2) method. AR method does not trace the actual data of the case study and hence called not suitable for this case study and produce a large MAPE during the experimentation. Table 4 gives various results obtained from classical AR method, MA method and ANN. Referring Tab. 4, MAPE with proposed technique is less than or equal to 4.63 % for three consecutive years, which is below the normal accepted forecasting MAPE [15, 26, 33] in case of medium and long term load forecasting. Table 4 shows that MAPE in the predicted data is least in case of proposed ANN architecture, and in MA method on considering more terms of the historical data the prediction goes smoother, which does not provide good forecasting.

Table 4. Comparison of mean absolute percentage error with ann and classical methods

Year	ANN	AR	MA			
			6 months	8 months	10 months	12 months
2007	2.05	9.20	4.17	4.89	5.20	4.9
2008	3.61	11.96	5.19	5.89	5.88	5.62
2009	4.63	12.80	5.74	6.78	7.00	6.32

4 CONCLUSION

An easy and cyclic methodology to explore number of neurons in the hidden layer as well as the best training method is developed, for the case study. Monthly loads from January to December is forecasted for three consecutive years 2007, 2008 and 2009 using the proposed ANN architecture with considerable and satisfactory limit of accuracy.

It is observed that there is an unusual dip in electricity consumption in Dec. 2008 and Jan. 2009 which is rare phenomenon, and normally not used while training the proposed ANN, therefore large absolute errors are observed at these points. Overall finding, from Tab.4 shows that ANN method is better than the conventional AR and MA method of forecasting.

Appendix 1

Sequence for input selection for feed forward neural network

No. of neuron in the first layer	Proposed selection scheme	No. of neuron in the first layer	Proposed selection scheme
1	(i – 1) th month	11	(i – 37) th month
2	(i – 2) th month	12	(i – 47) th month
3	(i – 11) th month	13	(i – 48) th month
4	(i – 12) th month	14	(i – 49) th month
5	(i – 13) th month	15	(i – 59) th month
6	(i – 23) th month	16	(i – 60) th month

7	(i – 24) th month	17	(i – 61) th month
8	(i – 25) th month		if available
9	(i – 35) th month		(i – 60) th month
10	(i – 36) th month		otherwise

Appendix 2

Different training methods used in proposed work [29]

Sequence No.(TM)	Training Method Name of Method	Representation
1	Batch training with weight and bias learning rules	B
2	BFGS quasi-Newton back-propagation	BFG
3	Bayesian regularization	BR
4	Cyclical order incremental update	C
5	Powell-Beale conjugate gradient back-propagati	CGB
6	Fletcher-Powell conjugate gradient back-propagation	CGF
7	Polak-Ribiere conjugate gradient back-propagation	CGP
8	Gradient descent back-propagation	GD
9	Gradient descent with adaptive learning rule back-propagation	GDA
10	Gradient descent with momentum back-propagation	GDM
11	Gradient descent with momentum and adaptive learning rule back-propagation	GDX
12	Levenberg-Marquardt back-propagation	LM
13	One step secant back-propagation	OSS
14	Random order incremental training with learning functions	R
15	Resilient back-propagation (Rprop)	RP
16	Sequential order incremental training with learning functions	S
17	Scaled conjugate gradient back-propagation	SCG

REFERENCES

[1] ALFARESM, H. K.—NAZEERUDDIN, M.: Electric Load Forecasting: Literature Survey and Classification of Methods, Int. Jour. of Systems Sc. **33** No. 1 (2002), 23–34.

[2] HINOJOSA, V. H.—HOESE, A.: Short-Term Load Forecasting Using Fuzzy Inductive Reasoning and Evolutionary Algorithms, IEEE Trans. on Power Systems **25** No. 1, 565-574, Feb. 2010..

[3] ALFARES, H. K.—NAZEERUDDIN, M.: Electric Load Forecasting: Literature Survey and Classification Of Methods, Int. Jour. of Systems Science **33** No. 1 (Jan. 2002), 23–34.

[4] HINOJOSA, V. H.—HOESE, A.: Short-Term Load Forecasting Using Fuzzy Inductive Reasoning and Evolutionary Algorithms, IEEE Trans. on Power Systems **25** No. 1 (Feb 2010), 565–574.

[5] HAGAN, M. T.—BEHR, S. M.: The Time Series Approach to Short Term Load Forecasting, IEEE Trans. on Power Systems **PWRS-2** No. 3 (Aug 1987), 785–791.

- [6] WANG, B.—TAI, N. L.—ZHAI, H. Q.—YE, J.—ZHU, J. D.—QI, L. B.: A New ARMAX Model based on Evolutionary Algorithm and Particle Swarm Optimization for Short-Term Load Forecasting, *Electric Power Systems Research* **78** No. 10 (Oct 2008), 1679–1685.
- [7] AGUIRRE, L. A.—RODRIGUES, D. D.—LIMA, S. T.—MARTINEZ, C. B.: Dynamical Prediction and Pattern Mapping in Short-Term Load Forecasting, *Int. Jour. of Electrical Power & Energy Systems*, **30** No. 1 (Jan 2008), 73–82.
- [8] SRINIVASAN, D.: Energy Demand Prediction using GMDH Networks, *Neurocomputing* **72** No. 1-3 (Dec 2008), 625–629.
- [9] AZADEH, A.—SABERI, M.—GHADERI, S. F.—GITIFOROUZ, A.—EBRAHIMPOUR, V.: Improved Estimation of Electricity Demand Function by Integration of Fuzzy System and Data Mining Approach, *Energy Conversion and Management* **49** No. 8 (Aug 2008.), 2165–2177.
- [10] HAHN, H.—MEYER-NIEBERG, S.—PICKL, S.: Electric Load Forecasting Methods: Tools for Decision making, *European Journal of Operational Research* **199** No. 3 (Dec 2009), 902–907.
- [11] PAPPAS, S. S.—EKONOMOU, L.—KARAMPELAS, P.—KARAMOUSANTAS, D. C.—KATSIKAS, S. K.—CHATZARAKIS, G. E.—SKAFIDAS, P. D.: Electricity Demand Load Forecasting of the Hellenic Power System using an ARMA Model, *Electric Power Systems Research* **80** No. 3 (Mar 2010), 256–264.
- [12] QI—WU: A Hybrid-Forecasting Model based on Gaussian Support Vector Machine and Chaotic Particle Swarm Optimization, *Expert Systems with Applications: An Int. Jour.* **37** No. 3 (Mar 2010), 2388–2394.
- [13] DANESHI, H.—DANESHI, A.: Price Forecasting in Deregulated Electricity Markets – A Bibliographical Survey, Third Int. Conf. on Electric Utility Deregulation and Restructuring and Power Technologies, DRPT 2008, 6–9 Apr 2008, pp. 657–661.
- [14] AZADEH, A.—GHADERI, S. F.—SOHRABKHANI, S.: Annual Electricity Consumption Forecasting by Neural Network in High Energy Consuming Industrial Sectors, *Energy Conversion and Management* **49** No. 8 (Aug 2008), 2272–2278.
- [15] DANESHI, H.—SHAHIDEHPOUR, M.—CHOOBBARI, A. L.: Long-Term Load Forecasting in Electricity Market, *IEEE Int. Conf. on Electro/Inf. Tech.* 2008, 18–20 May 2008, pp. 395–400.
- [16] DANESHI, H.—DANESHI, A.: Real Time Load Forecast in Power System, Third Int. Conf. on Electric Utility Deregulation and Restructuring and Power Technologies, DRPT 2008, 6–9 Apr 2008, pp. 689–695.
- [17] PINDORIYA, N. M.—SINGH, S. N.—SINGH, S. K.: Forecasting the Day-Ahead Spinning Reserve Requirement in Competitive Electricity Market, *Power and Energy Society General Meeting – Conversion and Delivery of Electrical Energy in the 21st Century*, IEEE, USA, 20–24 July 2008, pp. 1–8.
- [18] HINOJOSA, V. H.—HOESE, A.: Short-Term Load Forecasting using Fuzzy Inductive Reasoning and Evolutionary Algorithms, *IEEE Trans. Power Syst.* **25** No. 1 (Feb 2010), 565–574.
- [19] PARKPOOM, S. J.—HARRISON, G. P.: Analyzing the Impact of Climate Change on Future Electricity Demand in Thailand, *IEEE Trans. Power Syst.* **23** No. 3 (Aug 2008).
- [20] DREZGA, I.—RAHMAN, S.: Input Variable Selection for ANN-Based Short-Term Load Forecasting, *IEEE Tran. on Power Sys.* **13** No. 4 (Nov 1998).
- [21] AZADEH, A.—FAIZ, Z. S.: A Meta-Heuristic Framework for Forecasting Household Electricity Consumption, *Applied Soft Computing* **11** No. 1 (Jan 2011), 614–620.
- [22] TSEKOURAS, G. J.—DIALYNAS, E. N.—HATZIARGYRIOU, N. D.—KAVATZA, S.: A Non-Linear Multivariable Regression Model for Midterm Energy Forecasting of Power Systems, *Electric Power Systems Research* **77** No. 12 (Oct 2007), 1560–1568.
- [23] SULLIVAN, W. G.—CLAYCOMBE, W. W.: *Fundamentals of Forecasting*, Reston Publishing Company, Inc., Reston Virginia, 1977.
- [24] McCULLOCH, W. S.—PITTS, W.: *Bulletin of Mathematical Biophysics* **5** (1943), 115–133.
- [25] HAYKIN, S.: *Neural Network: A comprehensive foundation*, Prentice Hall of India, New Delhi, 1999.
- [26] SHRIVASTAVA, V.—MISRA, R. B.: A Novel Approach of Input Variable Selection for ANN Based Load Forecasting, *Joint Int. Conf. on Power System Technology and IEEE Power India Conf., POWERCON 2008*, 12–15 Oct 2008, pp. 1–5.
- [27] METHAPRAYOON, K.—LEE, W. J.—RASMIDDATTA, S.—LIAO, J. R.—ROSS, R. J.: Multistage Artificial Neural Network Short Term Load Forecasting Engine with Front End Weather Forecasting, *IEEE Trans. on Industry Applications* **43** No. 6 (Nov/Dec 2007), 1410–1416.
- [28] CHARYTONIUK, W.—CHEN, M. S.: Neural Network Design for Short Term Load Forecasting, *Int. Conf. on Electric Utility Deregulation and Restructuring and Power Technologies*, Proceedings DRPT 2000, 2000, pp. 554–561.
- [29] BEALE, M. H.—HAGAN, M. T.—DEMUTH, H. B.: *Neural Network Toolbox™ 7 Users Guide*, The Math Works, Inc., Natick, MA, Sep 2010.
- [30] AMJADY, N.—DARAEPOUR, A.: Mixed Price and Load Forecasting of Electricity Markets by a New Iterative Prediction Method, *Electric Power Systems Research* **79** No. 9 (Sep 2009), 1329–1336.
- [31] <http://www2.bot.or.th/statistics/ReportPage.aspx?reportID=108&language=eng>.
- [32] BOX, G. E. P.—JENKINS, G. M.: *Time Series Analysis: Forecasting and Control*, Holden-Day Inc., 500 Sansome street, San Francisco, California, 1976.
- [33] KUNCORO, A. H.—ZUHAL, R. D.: Long-Term Load Forecasting on the Java-Madura-Bali Electricity System using Artificial Neural Network Method, *Int. Conf. on Advances in Nuclear Science and Engineering in Conjunction with LKSTN*, 2007, pp. 177–181.

Received 27 January 2011

Navneet K. Singh is research scholar at the Department of Electrical Engineering, Motilal Nehru National Institute of Technology, Allahabad, India. His research area includes load forecasting, power system planning and application of artificial intelligence (AI) techniques in power system.

Asheesh K. Singh received the PhD degree from Indian Institute of Technology (IIT), Roorkee, India. Currently he is with the Department of Electrical Engineering, Motilal Nehru National Institute of Technology, Allahabad, India. His research area includes power quality, power system harmonics, voltage unbalance, power system, energy efficiency, energy audit.

Manoj Tripathy received the PhD degree from Indian Institute of Technology (IIT), Roorkee, India. Currently he is with the Department of Electrical Engineering, Motilal Nehru National Institute of Technology, Allahabad, India. His research area includes power system protection, digital/numerical relays; artificial intelligence (AI) based relays, application of artificial neural network, fuzzy logic, neuro-fuzzy technique.