Pertanika J. Sci. & Technol. 27 (3): 1227 - 1245 (2019)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Application of Levenberg Marquardt Algorithm for Short Term Load Forecasting: A theoretical investigation

Manish Kumar Singla[1]*, Parag Nijhawan[1], Amandeep Singh Oberoi[2] and Parminder Singh[3]

[1] Department of Electrical and Instrumentation Engineering, Thapar Institute of Engineering and Technology Patiala, 147004 India
[2] Department of Mechanical Engineering, Thapar Institute of Engineering and Technology, Patiala, 147004 India
[3] Chemical Engineering Department, Thapar Institute of Engineering and Technology, Patiala, 147004 India

ABSTRACT

The load forecasting aims at the energy management in the field of power supply systems. It helps to diminish the production cost, spinning reserve capacity and enhance the reliability of the power system. It is tremendously essential for financial institutions, electric utilities and other participants in electric energy market, be it for transmission, generation or distribution. The economic allotment of electricity generation plays a vital role in short term load forecasting. This paper presents a solution methodology based on Levenberg Marquardt algorithm of an artificial neural network technique for short term load forecasting. The system data for forecasting the load includes the parameters like dry-bulb temperature, dew point temperature, humidity and load data. The live load data was recorded from the 66kV substation located at Bhai Roopa, Bathinda in Punjab state of India. The corresponding weather data was collected from the Indian Meteorological Department "IMD" at Pune in Maharashtra state for the years 2015 and 2016. The Levenberg Marquardt algorithm had been implemented to minimize the error function derived on the basis of computed load and actual load. This work had been carried out using the

ARTICLE INFO

Article history:
Received: 03 October 2018
Accepted: 26 March 2019
Published: 24 July 2019

E-mail addresses:
msingla0509@gmail.com (Manish Kumar Singla)
parag.nijhawan@rediffmail.com (Parag Nijhawan)
oberoi@thapar.edu (Amandeep Singh Oberoi)
parminder.singh@thapar.edu (Parminder Singh)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

MATLAB software. The obtained results would support an effective and accurate load forecasting in future.

Keywords: Electrical energy, feed forward network, Levenberg Marquardt, neural network, short term load forecasting

INTRODUCTION

Power utilities are expected to supply reliable power supply to their consumers. With the ever increasing load demand, it becomes necessity for the electric utilities to predict the future load requirements of their consumers using effective load forecasting methods and/or tools. The effective load forecasting will definitely ensure the fruitful profits for the electric utilities. This would also enhance client satisfaction level and future monetary process in their space (Singla, 2018). For efficient operation and planning of utility company, correct models of power load prediction are necessary. Load forecasting is a very essential tool for an electrical utility to form necessary choices together with choices on the purchase and for banking of power (with alternative corporations or identical state utilities or with the neighboring states) (Singla & Hans, 2018). It also helps in the adequate generation of power at each and every instant of time with the development of infrastructure and in the continuously variable load environment (Singla & Hans, 2018). It is absolutely necessary for the existing energy suppliers as well as for other alternative participants within the electrical energy transmission, generation, distribution networks, and markets.

The neural network (NN) approach was first time developed for the problem of load forecasting in the year 1990. With parallel and distributed units for processing, the NN can be defined as the set of arrays including series of the repetitive uniform processor while connected to the grid. In a neural network, the two important key terms are learning and training. The learning in NN can be done by various methods like interconnecting the various processors with each other (Ranaweera et al., 1996). Using the Neuroshell-2 in literature (Khotanzad et al., 1997) short-term load forecasting (STLF) had been carried out. Different methods like expert systems, Grey system theory and artificial neural network (ANN) to solve the short term load forecasting problem (Tayeb et al., 2013) have been reported in literature. Comparing the forecasting system in real time with the available data, it can be safely concluded that NN tool gives fairly accurate and reliable results.

ANN can only perform operations according to the trained data whereas in case of STLF, the selection of training sets is quite complicated. The selection was based on the similarity of characteristics of the training pairs present in the training set that must be same as those to the forecasted in that particular day. To get smart forecasting results, day type data should be taken under consideration. A technique is to construct the various ANNs for everyday type and feed every ANN with the corresponding day type training sets (Ho et al., 1992). The opposite is to use only one ANN, however, contain the day type data within the input variables (Dillon et al., 1991; Ranaweera et al., 1996; Chow & Leung, 1996). The previous method uses a variety of comparatively small size networks, whereas the latter has only one network of a comparatively giant size. A typical classification given in literature (Ranaweera et al., 1996) categorizes the historical loads into 5 categories. These are a Monday, Tuesday-Thursday, Friday, Saturday and Sunday/Public vacation. The traditional way of observation and comparison (Ranaweera et al., 1996; Raza et al., 2017) supports unsupervised ANN that ideates and selects the training set automatically (Yang & Huang, 1998) based on the area considered and taking into account the day type classification.

SHORT TERM LOAD FORECASTING

Background

In power system planning, generation and transmission, operation and control, the load forecasting plays a crucial part (Singla & Gupta, 2018). Forecasting signifies the estimation of active load at numerous load buses prior to actual load prevalence. Application of load forecasting in planning and operation needs an exact 'lead time' also known as 'forecasting intervals'. Categorization of load forecasting with respect to lead time is presented in Table 1.

Table 1

Categorization of Load Forecasting.

Nature of forecast	Lead time	Applications
Very short term	few seconds to few minutes	Scheduling of generation and distribution, power system security analysis
Short term	Half an hour to the number of hours	Unit commitment and spinning reserve allocation
Medium term	Few days to a number of weeks	Planning for seasonal peak winter, summer
Long-term	Up to one year	Planning generation growth.

There are mainly three categories for load forecasting: short-term load forecasting generally carried out for the duration ranging from few hours to one week, medium-term load forecasting generally carried out for the duration ranging from few weeks to a year, and long-term load forecasting generally carried out for the duration ranging for more than one year. In an organization, it is necessary to forecast load at various time horizons for various operations. These forecasts are distinct in nature. Most of the strategies employ statistical methods or artificial intelligence algorithms like fuzzy logic, regression, expert system and neural networks. For medium and long-term load forecasting, end-use econometric technique is widely used. For STLF, various strategies such as fuzzy logic, different regression models, statistical learning techniques, time series, expert systems and similar day methods, are employed.

Statistical approaches usually require a mathematical model that represents load as function of different factors such as time, weather, and customer class. The two important categories of such mathematical models are: additive models and multiplicative models. They differ in whether the forecast load is the sum (additive) of a number of components or the product (multiplicative) of a number of factors. For example, Chen et al. (2001) presented an additive model that took the form of predicting load as the function of four components:

$$L(t) = L_n(t) + L_w(t) + L_s(t) + L_r(t) \tag{1}$$

Where L(t) is the total load at time t; $L_n(t)$ is the normal or trend component which is set of standardized load shapes; $L_w(t)$ is the weather sensitive component; $L_s(t)$ is the special event component which create a substantial deviation from the usual load pattern and $L_r(t)$ is the completely random term or noise.

A multiplicative model may be of the form of

$$L(t) = L_n(t) \,.\, F_w(t) \,.\, F_s(t) \,.\, F_r(t) \tag{2}$$

Where $L_n(t)$ is the normal load and the correction factors $F_w(t)$, $F_s(t)$ and $F_r(t)$ are the positive numbers that can increase or decrease the overall load. The correction factor are based on current weather ($F_w(t)$), special event ($F_s(t)$) and ($F_r(t)$) is the random fluctuation.

Forecasting Strategies

Statistical approaches completely require the mathematical model that can represent a dependency of load on various factors such as time, weather and customer. The mathematical model is further sub-divided into two categories including additive model as well as a multiplicative model. These models differ in the way the load is forecasted. They consider either the multiplicative or addictive nature of various factors for the load prediction.

Medium and Long-Term Load Forecasting Strategies. The previously discussed modeling approaches such as economic modeling, end-use modeling and the combination of both, are used for the medium and long-term load forecasting.

End-Use Models. This approach utilizes the direct measurement of energy consumption on the basis of information based on several factors such as customer use, the size of the houses and the customer age (Engle et al., 1992). Statistical information concerning customers besides the dynamics of the amendment is considered as the basis of the forecast.

Econometric Models. The electricity demand is forecasted by the effective combination of two approaches including statistical approach as well as economic theory approach (Gupta & Pal, 2017). These approaches are further utilized for the representation of the relationship between the factors that affect the consumptions and the energy consumption itself. The estimation for the relationship parameters between these approaches depends upon the least square method and sometimes time's series method.

Statistical Model. Based upon the learning, the previously discussed strategies, and the end users are dependent upon the factor like economics and the customers. The active participation is also needed for the various applications related to these approaches. The statistical model basically used is multiple linear regression (Haida & Muto, 1994; Gupta & Pal, 2017).

Short-Term Load Forecasting Strategies. A large type of statistical and artificial intelligence techniques are developed for short-term load forecasting. There are a variety of techniques that can be used for the STLF such as fuzzy logic, regression model, neural network, statistical learning algorithm and time series.

Similar Day Approach. These approaches are assumed on the basis of extensive information for the days. These approaches are also considered for the forecasting of the weeks as well of the year on the basis of forecasted data which was used for the one year. The same rules are further applied for the forecasting of weekdays due to which these methodologies also consider as one of the benchmark function for the forecasted model (Mu et al., 2010).

Regression Methods. The regression approaches were used for the statistical techniques. The application of multiple regressions to find the hidden relations between dependent as well as independent parameters is also reported in literature (Mu et al., 2010). The least square method is highly considered for these approaches including the variations in the sum of the square of expected values as well as determined one.

Time Series. This approach has also been reported in literature for the measurement and the estimation of the forecasting values. They can also be used for various factors such as electrical load forecasting and also for economics (Peng et al., 1992). There are some other approaches such as Auto-regressive Integrated Moving Average (ARIMA), Auto-regressive Moving Average with Exogenous Variables (ARMAX), Auto-regressive Moving Average

(ARMA) and Auto-regressive Integrated Moving Average with Exogenous that can also be used for the electrical load forecasting.

Neural Networks. Load forecasting can also be done in quite an effective manner by the application of artificial neural network algorithms. The output of neural network must be linear or non-linear on the basis of the input data that can also be considered as the output of previously designed neural network (Vapnik, 2013). The organization of neural network within the range can be accomplished by the effective use of input-output data. Sometimes, feedback can also be used to improve the performance of the complete network. During the implementation of a neural network for the forecasting, one should consider various parameters like the size of the neuron, relative connectivity between the layers and the elements and the utilization of uni-directional or bi-directional link within the network. Therefore, the pre-operational training needs to be considered for unsupervised learning.

Expert Systems. Various rules, as well as different procedure, are considered in this approach is completely related to the field of the system forecast. The rule-based forecasting is highly effective for its implementation in the load forecasting (Gupta & Pal, 2017). These approaches work best whenever the data is considered by the human expert for its incorporation within the software for system forecasting.

Fuzzy Logic. This technique is considered as one of the most effective methods for the mapping of the input to the output. The absence of the mathematical modeling within the system makes this technique more effective in comparison to other technique as its output is highly precise (Saxena et al., 2010). For the effective utilization of Boolean logic for the digital output, fuzzy logic is considered as one of the best technique for this particular application.

Support Vector Machines. This technique is highly effective for the minimization of issues related to the regression, and also considered one of the modern technology in the field of forecasting. It basically emerges from the statically learning theory (Keyhani, 2016).

Purpose of Load Forecasting

The load forecasting is generally carried out for the following purposes:
The utility enables the company to plan well because they understand the demand for future consumption or load.

(i) Maximum use of power generation plants. The forecasting avoids under generation or over generation.

(ii) The forecasting helps in planning the location of the site and the size of the plant. It also helps in reducing the transmission and distribution losses.

(iii) Deciding and planning for the maintenance of the power system.

(iv) The risk for the utility company is reduced.

MATERIALS AND METHODS

Overview of Levenberg Marquardt

This depicts the well-ordered strategy for training the neural network to learn from the recent one month weather and temperature data. For outlining the network architecture the MATLAB ANN toolbox was used. The MATLAB ANN tool box was used for the Training, Testing and validation of the selected data. The selection of an optimum number of hidden layers is necessary as the increase in the number of hidden layers results in increased complexity of the ANN architecture, thereby, affecting its performance. The algorithm used for training the artificial neural network was Levenberg-Marquardt. The Levenberg-Marquardt algorithm is better than Back Propagation algorithm because it has better convergence rate, the speed of iteration is more and it is more robust. The Levenberg-Marquardt algorithm is a variation of Newton's method. This algorithm is very well suited for neural network training where the performance index is Mean Square Error. If initial guess is far from the mark, the LM algorithm can find an optimal solution. The main problem of Levenberg-Marquardt is a selection of the hidden layer size, which is selected by hit and trial method. In some cases the LM algorithm is slow to converge this particularly happens when the parameter is more than ten. It trains the network quicker as compared to the back propagation (BP) algorithm. Although it is more efficient, it requires more memory. No literature reports on the usage of the considered parameters like dry-bulb and wet-bulb temperatures which is a much simplified and accurate method compared to the parameters reported in the literature viz. rain-fall prediction, humidity and wind speed.

Methodology

There were five major steps to obtain the result or to train the network. The five steps are briefly explained below one by one and shown in the form of a flowchart in Figure 1.

Data Collection and Preparation. The chronological live load data was taken from the PSPCL, 66kV substation at Bhai Roopa, Bathinda in Punjab state of India, whereas, the corresponding weather data was acquired from the IMD, Pune. The one-month load data and weather data were used for the training the network. The sample data of one day is shown in Appendix (A.1).

Data Preprocessing. Scaling of raw input data is normally important to diminish the bias caused by various measuring units of original input variables. The approach utilized for scaling the network input and target was to standardize the mean and standard deviation of the training set.

Network Structure Design. The next step behind the training and validation of data set is to outline the structure for neural networks. This has to do with choosing a network topology and resolve the input nodes, output nodes, number of hidden layers and the number of hidden nodes. The network topology is mostly determined based on the sort of task to be performed by the planned network. The multilayer feed forward neural networks were effectively applied for prediction. The number of input nodes is usually set equal to the number of input variables.

The following are the input variables for this research:

(a) Dry bulb temperature

(b) Dew point temperature

(c) Humidity

The output of the neural network represents the forecasted load data for the forecasting day. The determination of the number of hidden layers and the number of neurons within the hidden layers is an important decision within the plan of neural networks. Too many hidden neurons cause many trainable weights, which might build a neural network to become erratic and unreliable. On the other hand, too few hidden neurons limit the learning ability of a neural network and improve its approximation performance (Olagoke et al., 2016). However, there is no distinct guideline for deciding the number of neurons in the hidden layers. The usual practice is by using trial and error which cannot yield an optimum network design and therefore the method is time-consuming.

Network Training. After the network has been outlined, the following stage is to train the network. The training of an artificial neural network is an iterative method that has to do with changing the associated weight. Several techniques are utilized to enhance the execution of back propagation, one of them being the Levenberg Marquardt technique. Levenberg Marquardt was embraced for training the neural network in this work. Levenberg Marquardt is the numerical optimization based technique in which performance index is to be optimized.

The performance index to be optimized for the Levenberg Marquardt (Singla, 2018) in equation 3.

$$F(w) = \sum_{p}^{p=1} \left[\sum_{k=1}^{k} \left(d_{kp} - o_{kp} \right)^2 \right] \tag{3}$$

Levenberg Marquardt algorithm consolidates the speed of Gauss-Newton's method and the stability of error in back propagation algorithm during training. When μ is large, the learning process follows the error in back propagation algorithm, and when μ is small, it follows the Gauss-Newton's algorithm. Here μ is a damping term.

$$\Delta W = \left(J^T J + \mu I\right)^{-1} J^T e \tag{4}$$

The Jacobian matrix is the matrix of all first order partial derivatives of a vectored valued function. When the matrix is a square matrix, both the matrix and its determinant are referred to as a Jacobian.

$$J = \begin{pmatrix} \frac{\partial F(x_1, W)}{\partial w_1} & \frac{\partial F(x_1, W)}{\partial w_{w1}} \\ \frac{\partial F(x_N, W)}{\partial w_1} & \frac{\partial F(x_N, W)}{\partial w_W} \end{pmatrix} \tag{5}$$

The Jacobian matrix equation in a neural network is N $\times$ W matrix.

Network Validation. After the network has been properly trained, it must be validated for its performance of generalization.

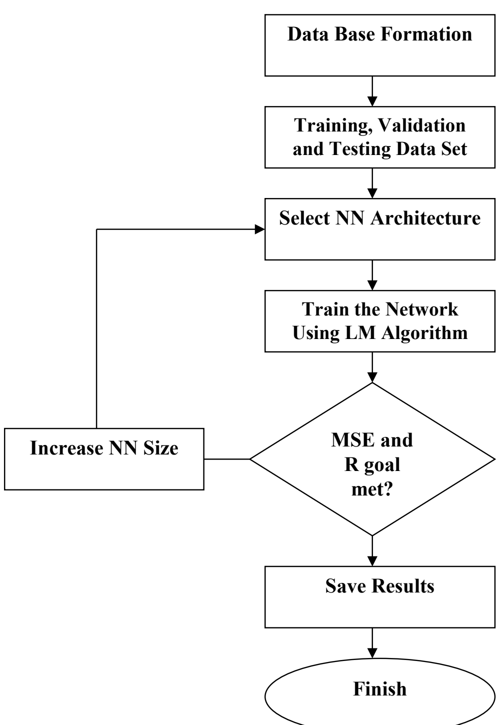

Figure 1. Testing and training flow chart

RESULTS AND DISCUSSIONS

The results of single layer feed-forward network and multilayer feed-forward network using the Levenberg-Marquardt technique are presented and discussed. The data set is divided into three parts, i.e. validation, training, and testing. The output considered is on hourly basis for the month of January. A feed-forward network consists of several successive layers of neurons with one input layer, several hidden layers, and an output layer. The neurons are connected using weight vectors and neither feedback nor intralayer connections exist. A neuron i thus takes the output of its k input neurons, computes the weighted sum, subtracts a so-called bias θ_i and applies the activation function a, the constraint functions for training the neural network is:

$$y_i = a(\sum_{k-1}^{n} w_{ik} x_k - \theta_i) \tag{6}$$

Single Layer Feed-Forward Network

In case of single layer network, there was single input layer as well as single output layer. Further, the neurons present in the input layer received the signals at the input terminal whereas the neurons present in the output layer received the output signal in a similar way. The input cells were connected to the similar output cell by the utilization of synaptic link carrying weight with it. Due to which this was considered as the feed forward neural network as the inverse operation could not be possible in this network. Despite of the fact that the network had two layers still it was considered as a single layer due to single output layer receiving signal from input layer (Rajashekaran & Vijayalksmi, 2004). The data was forecasted with different sizes of the hidden layer and the best results were observed when hidden neuron size was 8 as shown in Table 2. The Figure 2 represents the actual load and forecasted load in the hidden neuron size of eight.

Table 2

Represents the different sizes of the hidden neuron and error

Hidden Neuron size	2	4	6	8	10
MSE	0.0248	0.0130	0.0132	0.0090	0.0189
RMSE	0.1574	0.1142	0.1150	0.0950	0.1377
MAE	0.1431	0.9	0.08721	0.05711	0.1193
SSE	4.1660	2.1924	2.224	1.5190	3.1885
MAPE	17.6876	11.3278	11.0675	7.3897	13.6234

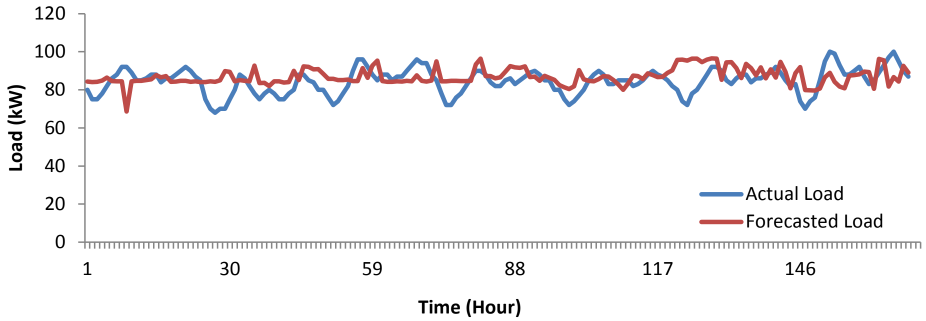

Figure 2. represents the actual load and forecasted a load of hidden neuron size eight.

Multi Layer Feed-Forward Network

As its name indicates, it is formed from multilayers. So, architecture of a multilayer feed-forward network possessing an auxiliary layer is considered between the input layer and the output layer. The hidden neurons present within the middle layer were considered for the computational purpose only. The major importance of hidden layer is that the computational work is performed by this layer before the input signal is received by the output terminal (Rajashekaran & Vijayalksmi, 2004). The input hidden layer weight is basically, a synaptic weight links formed by the combination of input neurons and the hidden neurons. In a similar way, whenever the output neurons formed a combination with the hidden layer neurons, it is considered as the hidden output layer weight. In the multi layer feed-forward network some cases are discussed below.

Case 1. In this case of multi layer feed-forward network, the hidden layer 1 is variable and hidden layer 2 is constant and this was used for error calculation. In the hidden layer 2, the constant value is 2. In this case, the lowest value is kept constant. In this case, the lowest error is observed, when hidden layer 1 size is 4 and hidden layer 2 size is 2. Table 3 represents the different size of neuron and error and Figure 3 shows the actual and forecasted load for the hidden layer having minimum error.

Table 3

Different sizes of Hidden neuron and error

Hidden layer 1	2	4	6	8	10
Hidden layer 2	2	2	2	2	2
MSE	0.0262	0.0149	0.0209	0.0169	0.0285
RMSE	0.1621	0.1223	0.1447	0.1303	0.1688
MAE	0.1484	0.09957	0.1267	0.1110	0.15552
SSE	4.4173	2.5147	3.5203	2.8548	4.7907
MAPE	18.3185	12.5050	15.7495	13.8465	19.1667

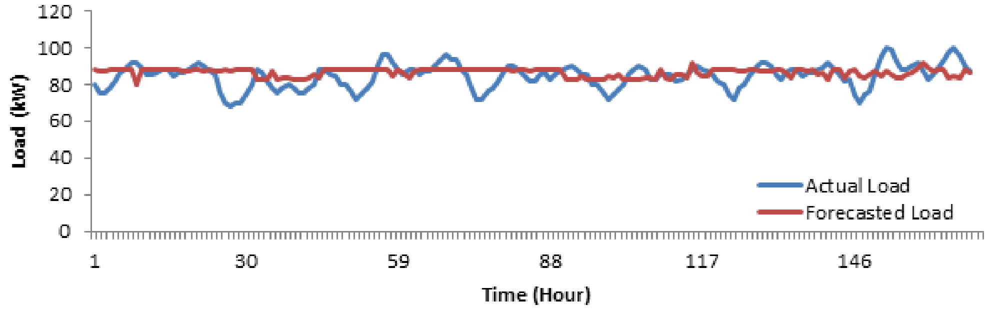

Figure 3. Actual load and Forecasted load of the hidden layer having less error.

Case 2. In this case also, hidden layer 1 is variable and hidden layer 2 is fixed. The hidden layer 2 size is fixed at 6. In this case, the middle value is kept constant. In this case, the error is minimum, when the hidden layer 1 size is 8 and hidden layer 2 size is 6. Table 4 shows the different sizes of hidden neuron and error. Figure 4 represents the actual and forecasted load of the hidden layer having minimum error.

Table 4

Different sizes of hidden neuron and error

Hidden layer 1	2	4	6	8	10
Hidden layer 2	6	6	6	6	6
MSE	0.0122	0.0135	0.0170	0.0066	0.0276
RMSE	0.1106	0.1164	0.1304	0.0814	0.1662
MAE	0.08923	0.0947	0.1091	0.0304	0.1514
SSE	2.0573	2.2767	2.8591	1.1133	4.6429
MAPE	11.2599	11.8770	12.4325	4.2341	18.7025

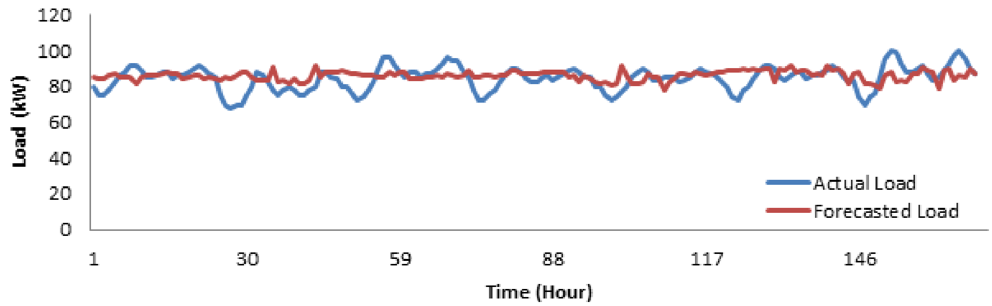

Figure 4. Actual load and Forecasted load of the hidden layer having less error

Case 3. In this case also, the hidden layer 1 size is variable and hidden layer 2 size is fixed. Now, the fixed value of hidden layer 2 is 10. In this case, the highest value is kept constant. In this case, the error is minimum, when the hidden layer 1 size is 4 and hidden layer 2 size is 10. Table 5 shows the different sizes of hidden neuron and error and Figure 5 represents the actual and forecasted load of the hidden layer having minimum error.

Table 5
Different sizes of hidden neuron and error

Hidden layer 1	2	4	6	8	10
Hidden layer 2	10	10	10	10	10
MSE	0.0232	0.0055	0.0094	0.0114	0.0133
RMSE	0.1523	0.0742	0.0973	0.1070	0.1154
MAE	0.1375	0.0093	0.0620	0.0775	0.0859
SSE	3.8986	0.9268	1.5907	1.9247	2.2388
MAPE	17.0182	0.529	7.9813	9.8251	10.8694

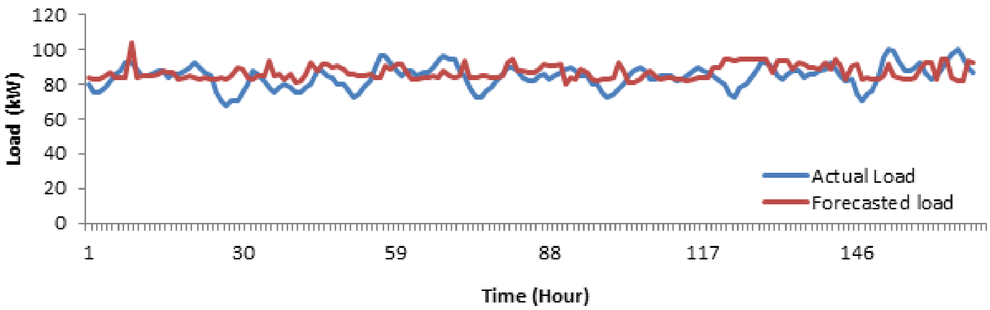

Figure 5. Actual load and Forecasted load of the hidden layer having less error

Case 4. In this case, the hidden layer 1 is fixed and the hidden layer 2 is variable. The size of hidden layer 1 is 2. In this case, the lowest value is kept constant. In this case, the error is minimum, when the hidden layer 1 size is 2 and hidden layer 2 size is 8. Table 6 shows the different sizes of hidden neuron and error. Figure 6 represents the actual and forecasted load of the hidden layer having minimum error.

Table 6

Different sizes of hidden neuron and error

Hidden layer 1	2	2	2	2	2
Hidden layer 2	2	4	6	8	10
MSE	0.0185	0.0258	0.0107	0.0086	0.0191
RMSE	0.1363	0.1606	0.1035	0.0932	0.1382
MAE	0.1177	0.1458	0.0795	0.0516	0.1209
SSE	3.1245	4.3353	1.7997	1.4596	3.2094
MAPE	14.6645	18.0127	10.1118	6.8059	15.0267

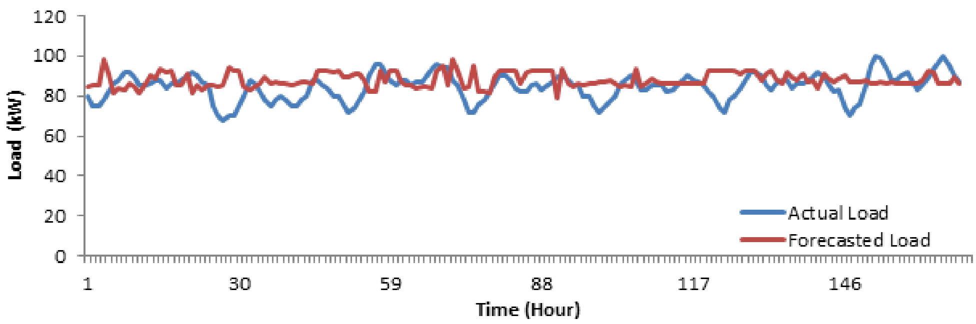

Figure 6. Actual load and Forecasted load of the hidden layer having less error

Case 5. In this case, the hidden layer 1 is fixed and the hidden layer 2 is variable. The hidden layer 1 size is 6. In this case, the middle value is kept constant. In this case the error is minimum, when the hidden layer 1 size is 6 and hidden layer 2 size is 6. Table 7 shows the different sizes of hidden neuron and error. Figure 7 represents the actual and forecasted load of the hidden layer having minimum error.

Table 7

Different sizes of hidden neuron and error

Hidden layer 1	6	6	6	6	6
Hidden layer 2	2	4	6	8	10
MSE	0.0283	0.0205	0.0125	0.0142	0.0126
RMSE	0.1684	0.1433	0.1118	0.1191	0.1125
MAE	0.1551	0.1249	0.0823	0.0957	0.0841
SSE	4.7655	3.4509	2.1029	2.3858	2.1284
MAPE	19.1214	15.5359	10.4296	12.0142	10.6470

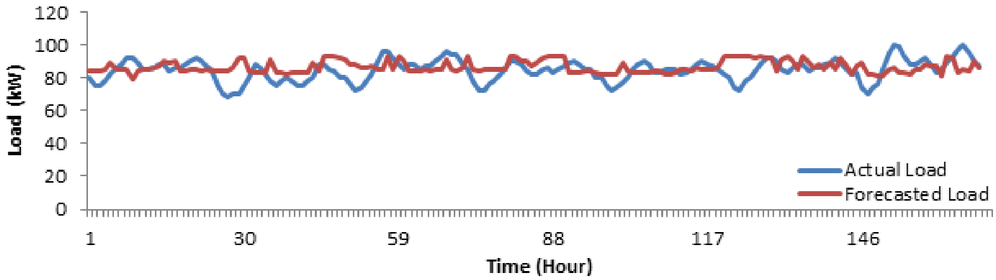

Figure 7. Actual load and Forecasted load of the hidden layer having less error.

Case 6. In this case, the hidden layer 1 is fixed and the hidden layer 2 is variable. The size of hidden layer 1 is 10. In this case, the highest value is kept constant. In this case the error is minimum, when the hidden layer 1 size is 10 and hidden layer 2 size is 6. Table 8 shows the different sizes of hidden neuron and error. Figure 8 represents the actual and forecasted load of the hidden layer having minimum error.

Table 8
Different sizes of hidden neuron and error

Hidden layer 1	10	10	10	10	10
Hidden layer 2	2	4	6	8	10
MSE	0.0115	0.0104	0.0062	0.0074	0.0071
RMSE	0.1075	0.1020	0.0790	0.0862	0.0847
MAE	0.08552	0.06928	0.0228	0.0262	0.0332
SSE	1.9443	1.7494	1.0510	1.2484	1.2061
MAPE	10.8171	8.8370	3.3250	3.7294	4.5570

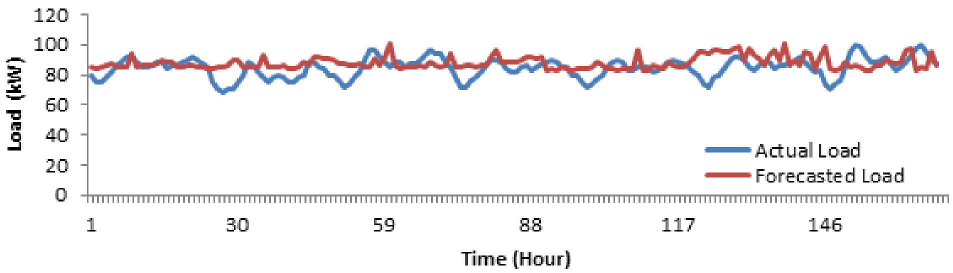

Figure 8. Actual load and Forecasted load of the hidden layer having less error

CONCLUSION

In this paper, the authors discuss about the short term load forecasting using the Levenberg Marquardt algorithm. The input data was collected from the weather station and load data was collected from the load station. The parameters used for comparing the performance of different models were MSE (Mean Square Error), RMSE (Root Mean Square Error), MAE (Mean Percentage Error), SSE (Sum of Square of Error) and MAPE (Mean Percentage Absolute Error). It is concluded that the MSE error, RMSE error, MAE error, SSE error, and MAPE error in the single layer network and multi layer network are different. No literature reports on calculation of more than two errors in load forecasting whereas, the obtained results in the presented works consists of five numbers of distinct errors and hence, provides more accurate load forecasting results compared to the literature. In the multi layer NN model, some cases have been considered taking the hidden layers constant or variable. Accurate load forecasting ensures the minimization of the production cost, spinning reserve capacity, and enhances the reliability of the power system. The Levenberg Marquardt algorithm is implemented to minimize the error function derived on the basis of computed load and actual load. The network is trained with1000 iterations and the training function used is "trainlm". The effectiveness of the applied algorithm for load forecasting is quite obvious from the results presented that would support a cost effective load forecasting in future. Hence, the presented work is surely a step forward toward estimating an error less load demand that could bring down the excess production losses to great deal.

ACKNOWLEDGEMENTS

Thapar Institute of Engineering and Technology Patiala, India has well equipped laboratories for effective academic and research work. The authors are really grateful to the Thapar Institute of Engineering and Technology for providing the necessary support and facilities to carry out the work presented in this manuscript.

REFERENCES

Chen, H., Canizares, C. A., & Singh, A. (2001, January 28 – February 1). ANN-based short-term load forecasting in electricity markets. In *2001 IEEE Conference Proceedings on Power Engineering Society Winter Meeting* (Vol. 2, pp. 411-415). Columbus, OH, USA.

Chow, T. W. S., & Leung, C. T. (1996). Nonlinear autoregressive integrated neural network model for short-term load forecasting. *IEE Proceedings-Generation, Transmission and Distribution, 143*(5), 500-506.

Dillon, T. S., Sestito, S., & Leung, S. (1991). Short term load forecasting using an adaptive neural network. *International Journal of Electrical Power and Energy Systems, 13*(4), 186-192.

Engle, R. F., Mustafa, C., & Rice, J. (1992). Modelling peak electricity demand. *Journal of Forecasting, 11*(3), 241-251.

Gupta, V., & Pal, S. (2017). An overview of different types of load forecasting methods and the factors affecting the load forecasting. *International Journal for Research in Applied Science and Engineering Technology, 5*(4), 729-733.

Haida, T., & Muto, S. (1994). Regression based peak load forecasting using a transformation technique. *IEEE Transactions on Power Systems*, *9*(4), 1788-1794.

Ho, K. L., Hsu, Y. Y., & Yang, C. C. (1992). Short term load forecasting using a multilayer neural network with an adaptive learning algorithm. *IEEE Transactions on Power Systems, 7*(1), 141-149.

Keyhani, A. (2016). *Design of smart power grid renewable energy systems*. Hoboken: John Wiley & Sons.

Khotanzad, A., Afkhami-Rohani, R., Lu, T. L., Abaye, A., Davis, M., & Maratukulam, D. J. (1997). ANNSTLF-a neural-network-based electric load forecasting system. *IEEE Transactions on Neural networks, 8*(4), 835-846.

Mu, Q., Wu, Y., Pan, X., Huang, L., & Li, X. (2010, March 28-31). Short-term load forecasting using improved similar days method. In *2010 Asia-Pacific Power and Energy Engineering Conference (APPEEC)* (pp. 1-4). Chengdu, China.

Olagoke, M. D., Ayeni, A. A., & Hambali, M. A. (2016). Short Term Electric Load Forecasting using Neural Network and Genetic Algorithm. *International Journal of Applied Information Systems (IJAIS), 10*(4), 22-28.

Peng, T. M., Hubele, N. F., & Karady, G. G. (1992). Advancement in the application of neural networks for short-term load forecasting. *IEEE Transactions on Power Systems, 7*(1), 250-257.

Rajashekaran, S., & Vijayalksmi, G. A. (2004). *Neural Networks, Fuzzy Logic and Genetic Algorithms*. New Delhi: PHI Learning.

Ranaweera, D. K., Hubele, N. F., & Karady, G. G. (1996). Fuzzy logic for short term load forecasting. *International Journal of Electrical Power and Energy Systems, 18*(4), 215-222.

Raza, M. Q., Nadarajah, M., Hung, D. Q., & Baharudin, Z. (2017). An intelligent hybrid short-term load forecasting model for smart power grids. *Sustainable Cities and Society, 31*, 264-275.

Saxena, D., Singh, S. N., & Verma, K. S (2010). Application of computational intelligence in emerging power systems. *International Journal of Engineering, Science and Technology, 2*(3), 1-7.

Singla, M., & Gupta, J. (2018). Load forecasting using back propagation algorithm. *International Journal of Engineering and Techniques, 4*, 169-175.

Singla, M., & Hans, S. (2018). Load forecasting using Fuzzy Logic Tool Box. *Global Research and Development Journal for Engineering, 38*, 12-19.

Singla, M. (2018). *Load Forecasting Using Artificial Neural Network* (Doctoral dissertation). Thapar Institute of Engineering and Technology, India.

Tayeb, E. B. M., Ali, A. T., & Emam, A. A. (2013). Electrical Energy Management and Load Forecasting in a Smart Grid. *International Journal of Engineering Inventions, 2*(6), 98-101.

Vapnik, V. N. (2013). *The nature of statistical learning theory*. New York: Springer science & business media.

Yang, H. T., & Huang, C. M. (1998). A new short-term load forecasting approach using self-organizing fuzzy ARMAX models. *IEEE Transactions on Power Systems, 13*(1), 217-225.

APPENDIX

A.1 Sample of data for one day

Date	Hours	Dew Point	Dry Bulb	Humidity	Load
1/1/2015	1	6.5	7.4	94	80
1/1/2015	2	7.7	9.4	89	75
1/1/2015	3	11.9	12.6	95	75
1/1/2015	4	10.2	10.2	100	78
1/1/2015	5	8.6	9	97	82
1/1/2015	6	8.2	9	95	86
1/1/2015	7	4.9	5.4	97	88
1/1/2015	8	4	4	100	92
1/1/2015	9	8.4	8.4	100	92
1/1/2015	10	7.2	7.2	100	89
1/1/2015	11	7.1	8	94	85
1/1/2015	12	6.5	8.2	89	85
1/1/2015	13	7.8	7.8	100	86
1/1/2015	14	9.2	10.4	92	88
1/1/2015	15	6.5	7.8	91	88
1/1/2015	16	7.8	9.8	87	84
1/1/2015	17	8.2	8.2	100	86
1/1/2015	18	9.6	10	97	86
1/1/2015	19	5.2	5.2	100	88
1/1/2015	20	10	10	100	90
1/1/2015	21	7.1	8.4	91	92
1/1/2015	22	8.6	9.8	92	90
1/1/2015	23	10.6	11	97	87
1/1/2015	24	9.6	10.8	92	85