

Globalized Nelder Mead Trained Artificial Neural Networks for Short Term Load Forecasting

Engr. Aamir Nawaz^{1,*}, Prof. Dr. Tahir Nadeem Malik², Engr. Nasir Saleem³,
Engr. Ehtasham Mustafa³

^{1,3,4}Institute of Engineering and Technology, Gomal University, Dera Ismail Khan, Khyber Pakhtunkhwa, Pakistan

²Department of Electrical Engineering, University of Engineering and Technology, Taxila, Punjab, Pakistan

Received: January 1, 2014
Accepted: February 12, 2015

ABSTRACT

Load forecasting has always been one of difficult tasks to achieve with minimum error due to its non-linear behavior. Artificial neural networks (ANN) are easily caught in local optima and its convergence is much slower. To get around these problems, hybrid techniques are being used effectively. That's why ANN merged with other techniques, is one of most active research topic. This paper presents a solution to train neural networks with Globalized Nelder Mead algorithm. The proposed hybrid model is employed for the prediction of hourly load demand using Australian Energy Market Operator and California data. The inputs of this model are past hourly load data, maximum and minimum temperatures of that day and day type variable to specify load demand of that day. Results indicate that proposed model provides better results than other techniques discussed in this paper.

KEYWORDS: Artificial Neural Networks, Short Term Load Forecasting, Globalized Nelder Mead

1. INTRODUCTION

Short-term load forecasting (STLF) is way of forecasting electric loads for different periods of time ranging from minutes to weeks. It is one of the most vital steps of power system operation and planning. It is approximated that 1% increase in prediction error leads to increase of a million pounds in operating costs per annum for electricity utility[1].

STLF has become a very attractive field of research for scholars. Therefore, different type of techniques and methods have been developed and proposed. They can be categorized in three major categories which are parametric, non-parametric and artificial intelligence base methods[2]. Parametric methods are those which formulate a mathematical or statistical model for load by investigating qualitative relationships between the load and load effecting factors e.g. explicit time functions, time series models[3], ARMA models, ARX models[4], polynomial functions, multiple regression and Fourier series. Non-parametric methods calculate forecast data directly from historical data presented e.g. using non-parametric regression, a load forecast can be calculated as a local average of observed past loads with the size of local neighborhood and the specific weights on the loads defined by multivariate product kernel. Artificial intelligence based methods such as artificial neural networks, fuzzy logic and expert systems [5] are used for load forecasting because of their abilities to model and solve non-linear problems.

Neural Networks have gained much more importance from some recent years for load forecasting[6]. The main advantage of neural networks is that it does not require any mathematical formulation or quantitative correlation between inputs and outputs[7]. In spite of this advantage, NN converges very slowly and it is very difficult to set the optimal weights/biases.

Yudong in [8], proposed bacterial foraging algorithm to train neural networks (BFO-NN) for short term load forecasting. In this research, Yodong has compared proposed algorithm with Genetic algorithm based neural networks (GA-NN). Results show that BFO-NN has outperformed GA-NN for short term load forecasting problem.

Shih-Hui Liao in [9], has presented a hybrid algorithm using Nelder mead and particle swarm optimization (PSO) to train neural networks. NM is used for local search while PSO is used to refine that search by exploring different regions in search space. This hybrid method outperforms other methods compared in this research article.

Williamoski in [10], has presented improved nelder mead to train neural network for controlling robotic arm kinematics. In this article, nelder mead performance is enhanced by incorporating quasi gradient in it. Further,

*Corresponding Author: Engr. Aamir Nawaz, Institute of Engineering and Technology, Gomal University, Dera Ismail Khan, Khyber Pakhtunkhwa, Pakistan. engr.aamir09@gu.edu.pk

Willaimoski has compared back-propagation algorithm, improved nelder mead algorithm and Levenberg Marquardt algorithm. Here, 2500 input patterns have been utilized to train neural networks. Back propagation depends on number of input patterns. With so many patterns, back propagation algorithm is inapplicable to train neural network. While Lavenberg Marquardt is quite fast algorithm but its training capability depends upon number of input patterns P, weights W, and outputs O. So calculating Jacobian matrix with dimensions $P \times W \times O$ where inputs patterns are 2500 and 7 weights are involved, is almost impossible, exceeding limit of computing machines. Computing capability of Improved Nelder mead depends on number of weights, not on input patterns. It's one of the advantages of improved nelder mead over back propagation and Levenberg Marquardt algorithms.

In research article [11], Ahmed has investigated degenerated simplex search method for optimizing neural network weights. Simplex methods face serious problems regarding degeneracy which is being addressed in this paper and also used that for training neural networks. Results have indicated that simplex methods outclassed back-propagation algorithm and random simplex methods for training neural networks.

Another hybrid algorithm using ant colony optimization and nelder mead is presented in [12]. This research is used for prediction of bankruptcy in banks. Nelder Mead and PSO is used as hybrid approach in [13] by Barzinpour for economic dispatch problem.

All above discussed articles show that nelder mead can be used as local search algorithm effectively with global algorithms. Luersen in [14], for the first time, has presented globalized nelder mead for exploring search space globally. Globalized Nelder Mead is modeled by restarting NM probabilistically in different search spaces using probability density function. Ghiasi improved Luersen globalized nelder mead method in [15]. He has presented results of globalized nelder mead which has outperformed GA and other methods discussed. For this reason, globalized nelder mead is considered in this research as global search method.

2. CONTRIBUTIONS

In this research, a new hybrid approach is proposed for improvement in load prediction. Our contribution in this research can be briefly described as follows:

- 1) We have proposed a new hybrid approach using Globalized Nelder Mead trained artificial neural network (NM-ANN). Globalized Nelder Mead is used for training weights of neural networks. This hybrid algorithm is not utilized in any field of power system till now.
- 2) Luersen and Ghiasi has used simple Nelder Mead algorithm with probabilistic restarts while we have used improved Nelder Mead presented by Pham [10], with probabilistic restarts which is new development in Globalized Nelder Mead for training neural networks.
- 3) We have developed four different models of artificial neural networks for case studies. These models include NM-ANN 4-3-1, NM-ANN 6-3-1, NM-ANN 4-2-1 and NM-ANN 6-2-1.
- 4) We have evaluated performance of above models using three different case studies. Case study I involves comparison of models using New South Wales (AEMO) data and California data. Case study II involves comparison of models for load prediction of randomly selected day of California data for year 2013. Case study III involves comparison of models with latest research using New South Wales (AEMO) data. Combining Pattern sequence similarity with neural networks[16] is used for comparison with proposed models. Case study IV contains performance comparison of proposed models with combined model of SARIMA (Seasonal ARIMA) and BP (Back-Propagation) algorithm [17].

This paper is organized in such a manner as described. Section 3 introduces the basic model and detail of Nelder Mead algorithm and its improvements. Section 4 gives model and training method of NM-ANN for STLF. In section 5, models are applied on Australian Electricity Market Operator (AEMO) data of New South Wales (NSW) and California electricity data for short term load forecasting. The effectiveness and validity of the proposed approach is proved by results. In last, section 6 is dedicated for conclusion and future research areas in this field.

3. NELDER MEAD ALGORITHM AND ITS IMPROVEMENTS

2.1 Nelder Mead Model Description

Nelder Mead is one of the most fast and simple algorithms for local minimum and multidimensional optimization problems. It is different from gradient methods as it does not have to calculate derivatives[14]. This method forms a simplex which converges to local minima. A simplex is basically a geometrical figure which is formed of $N+1$ vertex where N is number of variables in a function to be optimized. In each iteration, different points are calculated from worst point by using reflection or extension and contraction or shrink in order to form new simplex. There are three parameters that can affect convergence of simplex i.e. α (reflection coefficient to define reflected point distance from centroid), β (contraction coefficient) and γ (expansion coefficient).

Nelder Mead's simplex method steps can be summarized as following[18]:

Step 1: Select α, β, γ , select an initial simplex with random vertices $x_0, x_1 \dots x_n$ and calculate their function values.

Step 2: Sort the vertices $x_0, x_1 \dots x_n$ of the current simplex so that $f_0, f_1 \dots f_n$ in the ascending order.

Step 3: Calculate the reflected point x_r, f_r

Step 4: If $f_r < f_0$:

- a. Calculate the extended point x_e, f_e
- b. If $f_e < f_0$, replace the worst point by the extended point $x_n = x_e, f_n = f_e$
- c. If $f_e > f_0$, replace the worst point by the reflected point $x_n = x_r, f_n = f_r$

Step 5: If $f_r > f_0$:

- a. If $f_r < f_i$, replace the worst point by the reflected point $x_n = x_r, f_n = f_r$
- b. If $f_r > f_i$:
 - (i) If $f_r > f_n$: calculate the contracted point x_c, f_c
 - i. If $f_c > f_n$ then shrink the simplex
 - ii. If $f_c < f_n$ then replace the worst point by the contracted point $x_n = x_c, f_n = f_c$
 - (ii) If $f_r < f_n$: replace the worst point by the reflected point $x_n = x_r, f_n = f_r$

Step 6: If the stopping conditions are not satisfied, the algorithm will continue at step 2.

2.2 Nelder Mead method with quasi gradient

As discussed earlier, Nelder Mead's simplex method is one of the best algorithms in simplicity and fastness. However, it is limited to number of problems with two or three variables because of its poor convergence. It does not define its moving directions properly to be followed to get to optimum point in high geometrical dimensions. That is why it fails to optimize multidimensional problems.

Pham and Williamowski has altered Nelder Mead method by involving quasi gradient in it[10]. This improved method still is very simple as it does not involve any derivative calculations. In this method, an extra point is calculated using previous dimensions to approximate gradients. Although, its accuracy is highly dependent on linearity of function but its computing cost does not increase considerably with size of problem to be optimized.

This method approximates gradients of $n+1$ dimensional space which is created from a geometrical simplex [10]. An extra point is created using coordinates of n vertices of a simplex and uses this point to calculate gradients. Its steps are given as follows:

Assume an optimized function $f: P_n \rightarrow P, x \in P_n$

Step 1: Initialize a simplex with n random vertices $x_1, x_2 \dots x_n$

Step 2: Select an extra point x_s with its coordinates composed from n vertices in the simplex. In other words, coordinates of the selected point are the diagonal of matrix X from n vertices in the simplex.

$$x_s = \text{diag} \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \dots & \dots & \dots & \dots \\ x_{n,1} & x_{n,2} & \dots & x_{n,n} \end{bmatrix} \quad (1)$$

Or

$$x_s = [x_{1,1} \quad x_{2,2} \quad \dots \quad x_{n,n}] \quad (2)$$

Step3: calculate quasi gradients based on the selected point x_s and other n points in the simplex.

for $i = 1: n$,
if $\text{mod}(i, 2) == 0$

$$g_i = \frac{\partial f}{\partial x_i} = \frac{f(i-1) - f(x_s)}{x_{i-1,1} - x_{s,i}} \quad (3)$$

else

$$g_i = \frac{\partial f}{\partial x_i} = \frac{f(i+1) - f(xs)}{x_{i+1,1} - xs_i} \quad (4)$$

end

end

Step 4: calculating the new reflected point R' based on the best point B and the approximate gradients. Parameter σ is the learning constant or step size.

$$R' = B - \sigma * G \quad (5)$$

Step 5: if the function value at R' is smaller than the function value at B , it means that BR' is the right direction of the gradient then R' can be expanded to E' .

$$E' = (1 - \gamma)B + \gamma * R' \quad (6)$$

The quasi gradient method using numerical methods have just been presented above is much simpler than analytical gradient methods. This method does not have to derive derivatives of a function which is usually very difficult for complicated functions. Generally, the improved simplex method with the quasi gradient method is similar to Nelder Mead's simplex method except the way it calculates the reflected point and the extended point in case the Nelder Mead's simplex method cannot define its moving directions.

2.3 Globalized Nelder Mead

A local optimizer can be restarted more than once, to perform a global optimization. Luersen in his article [14], presented a probabilistic restart procedure. It provides points far from previous local optima and previous initial points. So, points from new region of search space get more chance to be selected as initial point for the succeeding local search.

Luersen has utilized a multi-dimensional probability (MDP) density for assigning the sampling probability to a point. Here, restart strategy chooses Nr points randomly and proceeds to the point where the solution has the maximum likelihood to take place. Advantage of Luersen restart procedure is that it can give global optimum but Luersen restart is computationally expensive because of the computational time needed for MDP.

Hossein ghiasi in [15], has proposed an adaptive probability density to replace the MDP, called the variable variance probability (VVP) while following Luersen algorithm.

The new probability function is based on the minimum distance to the points already sampled and represented as:

$$\phi(x) = \frac{1}{2\pi} \left(1 - e^{-\left(\frac{d_{\min}^2}{2\sigma^2}\right)} \right) \quad (7)$$

$$d_{\min} = \min_{i=1,2,3\dots m} \left\{ d_i = \sqrt{\sum_{k=1}^n \left(\frac{x_{k,i} - x_k}{x_{ku} - x_{kl}} \right)^2} \right\} \quad (8)$$

where $\phi(x)$ is the sampling probability of a point x , n is the number of design variables, x_i is a point previously sampled, and m is the number of points already sampled. Length d_i is the non-dimensional distance between point x and point x_i .

The variance of the normal probability density, updated in each restart, is given by:

$$\sigma = \frac{1}{3^n \sqrt{m}} \quad (9)$$

The variance is gradually decreased when the number of sampled points is increased.

Equation (9) provides points located in the one-third of the middle of the line connecting two previously sampled points about 65% chance of selection. This property will be preserved all along the restart process. In contrast, the MDP tends to a uniform distribution when the number of sampled points is increased.

The restart procedure is not only to assign a probability density, but also includes a selection procedure to pick a new point based on assigned probability. The VVP restart uses a selection procedure different than that of Luersen restart. Nr points are randomly selected to create a selection pool, which is a set of points whereby each has a number of copies proportional to its probability value. A new point is randomly selected from this pool. In this procedure, the probability of sampling a new point is not affected by the number of selected points, Nr .

Instead of probabilistic restarts of Globalized Nelder Mead, it is same as Nelder Mead that is why NM is used

to represent Globalized Nelder Mead in this research.

3. NM-ANN MODEL

3.1 Structure

In this research, two layer neural networks have been chosen. Four models are prepared with size of 4x3x1, 6x3x1, 4x2x1 and 6x2x1 as shown in Fig.1, Fig.2, Fig.3 and Fig.4 respectively. Here, 1st digit represents input layer neurons, 2nd digit represents hidden neurons and 3rd digit represents output layer neurons. Input layer neurons correspond to following features that has been extracted from past actual load in given models [8]:

- (i) Day Type: 0 denotes Saturday, Sunday, and holidays. 1 denotes Monday and other workdays after holiday. 2 denote Tuesday, Wednesday, Thursday, and Friday.
- (ii) Min. Temp.: The minimum forecast temperature of current day.
- (iii) Max. Temp.: The maximum forecast temperature of current day.
- (iv) L(t-1): The last actual load
- (v) L(t-2): The penultimate actual load
- (vi) L(t-T): The actual load at the same time last day. If the time interval is 1h, then T=24.

For the development of the short term load forecasting models, data of following standard utility companies have been used in this research:

1. Australian Energy Market Operator
2. California Electricity Market

Figure 3 shows four ANN models that have been prepared and used in this research. Models' input, hidden and output neurons are properly shown in figure below. Here, L(t) is forecasted load of next hour. All these models inputs are normalized for fast convergence.

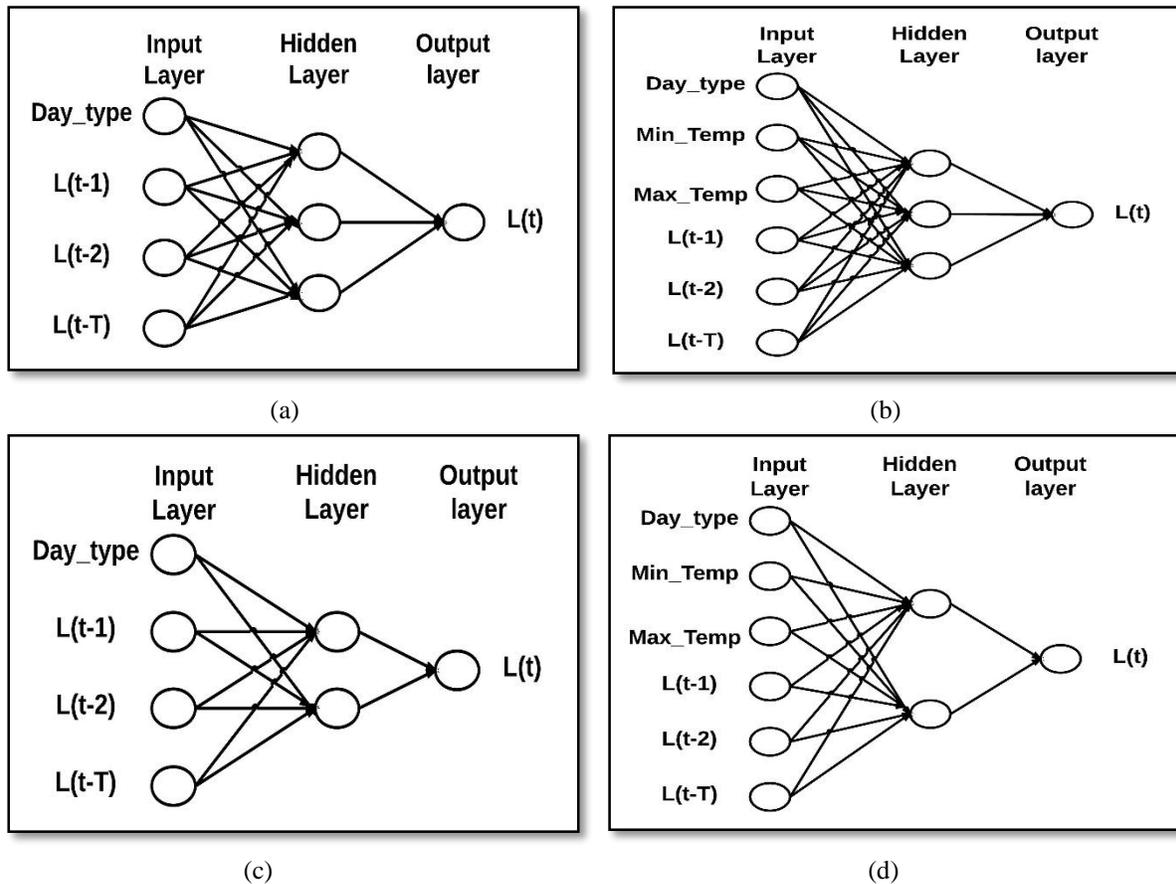


Figure 1: (a) 4x3x1 ANN Network, (b) 6x3x1 ANN Network, (c) 4x2x1 ANN Network, (d) 6x2x1 ANN Network

3.2 Transfer Function

Hidden layer outputs are calculated with following formula:

$$y_i = f_H(\text{node}_i) \quad i = 1, 2 \text{ or } i = 1, 2, 3 \quad (10)$$

Here, node_i is the activation value for the i th node in the hidden layer and f_H is the activation function of hidden neurons which is usually sigmoid function[8].

$$f_H(x) = \frac{1}{1 + \exp(-x)} \quad (11)$$

Output of neuron is calculated with the following formula:

$$L(t) = f_o(\text{node}_o) \quad (12)$$

Here, node_o is the activation function of the output neuron in the neural network models and f_o is the transfer function given in above equation.

3.3 Training Method for ANN with NM

The block diagram of training method used for these networks is shown in Fig. 5. In this figure, error $e(k)$ is fed to NM which further search for optimized weights/ biases to reduce this error $e(k)$. Here, $d(k)$ is actual load and $o(k)$ is forecasted load.

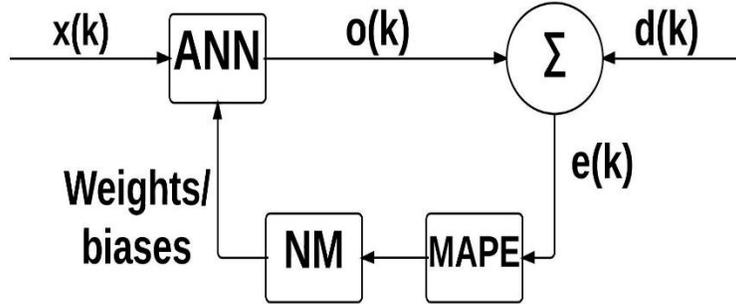


Figure 2: NM Training Method for optimization of ANN weights

MAPE (Mean Absolute Percentage Error) is selected for error calculation [19]. It is used as function for error search space where Globalized Nelder Mead finds optimized solution. It is calculated from all actual and forecasted load value using formula:

$$MAPE = \frac{1}{n} \sum_{L=1}^n \left| \frac{A_L - F_L}{A_L} \right| \times 100 \quad (13)$$

Where A_L is actual load and F_L is forecasted load and n is total number of hours of prediction.

Figure 3 shows flowchart of NM trained ANN. Weights of ANN are generated by NM simplex and are optimized using probabilistic restarts, large test and small test. Many local points are analyzed to get optimized weights for ANN. Analyzing large number of local points in search of global optima, gives good convergence of hybrid algorithm.

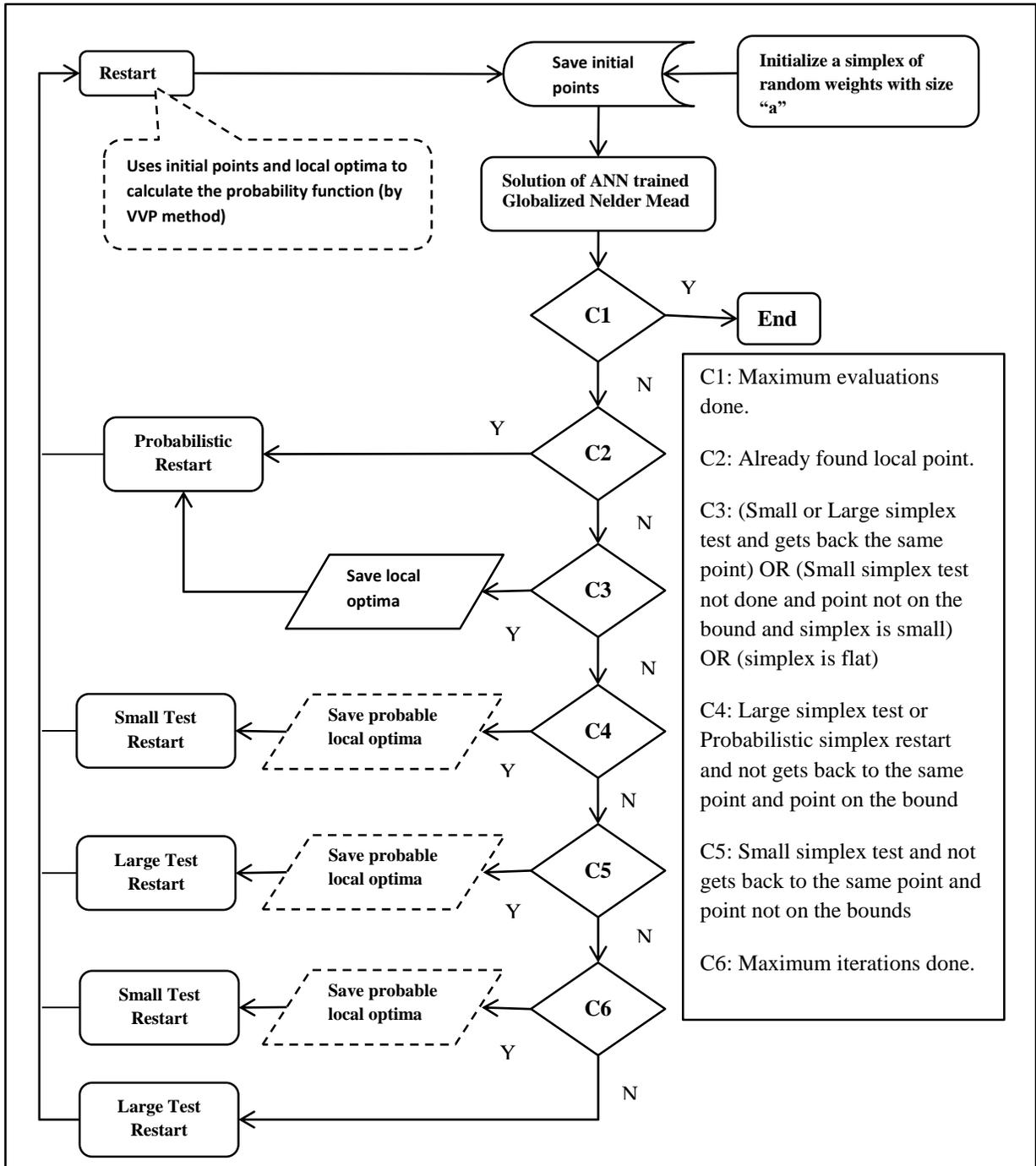


Figure 3: Flowchart of Globalized NM trained ANN

4. EXPERIMENTS AND DISCUSSIONS

These experiments have been performed on platform of intel core i5 2.5GHz and 4GB memory, running windows 8 OS. The algorithm of neural networks are totally self-developed and not used from Matlab built-in toolbox. These programs can be run on any platform, running Matlab in it.

4.1 Database of research

The load and weather data is divided into three case studies to study the performance of each model. The interval of load data is 1hour and for weather data, minimum and maximum temperature values of data are chosen

per day. Three case studies are as follows:

Case Study I: The load and weather data of AEMO was taken from 01/01/2009 to 01/31/2009 and California was taken from 01/01/2013 to 01/31/2013. There are total of 20 days and 24 samples per day are taken for training the models. Thus, total samples are $20 \times 24 = 480$.

Case Study II: We used 365 days of California data for year 2013. Thus, total samples are $365 \times 24 = 8760$. Out of these, 80% are chosen for training, 10% for validation and 10% for testing. For next day 24 hour load prediction, previous 20 days are used for training.

Case Study III: For comparison with other research, we used New South Wales data taken from AEMO (Australian Energy Market Operator) for year of 2009, 2010 and 2011. Data of respective month of 2009 is used for training the models, respective month data of 2010 is used for validation and respective month data of 2011 is used for testing. January, February and December are difficult to forecast according to [16], so we have used these months for comparing performance of proposed models with Pattern Sequence forecasting (PSF) models [16].

Case Study IV: We have utilized South Australia (SA) data taken from AEMO for year of 2005, 2006 and 2007. Data of June and July months are taken for comparison as they are difficult to predict. Comparison of proposed models is done with SARIMA (Seasonal ARIMA) and BP (Back-propagation) combined model [17].

4.2 Case Study I: Proposed NM-ANN Models Comparison

First four models were developed by using the California Electricity Market data and rest of the four models were developed using Australian Electricity Market Operator data (New South Wales). All models have been implemented in environment of Matlab programming and different results are obtained.

In the table given below, models were compared on the basis of MAPE (%). Among these, 4x3x1 model was found to give best results for California data and NSW AEMO data as shown in table 1. In this research, these models were further used for comparison with different techniques discussed later.

Table 1: Comparison of different models of this research for different load company's data

S No	Proposed Model	MAPE (%)	Load Data Company
1	Model-1: NM-ANN 4-3-1	3.776	California Data
2	Model-2: NM-ANN 6-3-1	5.032	
3	Model-3: NM-ANN 4-2-1	4.833	
4	Model-4: NM-ANN 6-2-1	4.658	
5	Model-5: NM-ANN 4-3-1	5.418	AEMO Data
6	Model-6: NM-ANN 6-3-1	8.854	
7	Model-7: NM-ANN 4-2-1	9.037	
8	Model-8: NM-ANN 6-2-1	8.070	

Figure given below shows the results comparison more elaborately of above table.

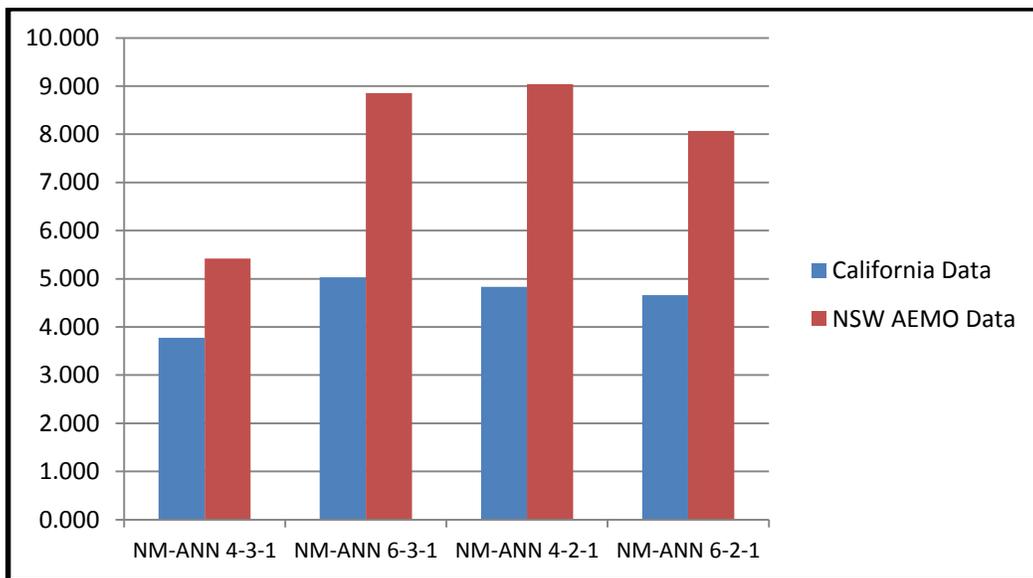


Figure 4: Bar chart for comparison of MAPE of different developed models for California and AEMO data

4.3 Case Study II: Proposed Hybrid Models Comparison for Random Selected Day

In table 2, each of the four models is applied on California data for year of 2013 and random days are selected for comparison. Models are trained on previous 20 days and tested on next day. For January, 1st 20 days are being used while for all other months, 15th of every month is predicted while trained on previous 20 days. Summer and winter months are difficult to forecast as compared to other months. As shown in table, MAPE of January, February, and December in winter and June, July and August in summer is greater compared to other months.

Table 2: Comparison of models for California data, 2013

Day	NM ANN 4-3-1	NM ANN 6-3-1	NM ANN 4-2-1	NM ANN 6-2-1
21-Jan	4.918	5.866	6.772	6.663
15-Feb	2.991	6.039	1.832	5.774
15-Mar	3.143	6.234	2.114	8.878
15-Apr	4.866	8.745	8.606	5.040
15-May	1.549	2.668	1.657	6.420
15-Jun	4.312	4.373	4.826	5.066
15-Jul	5.589	7.801	5.442	4.819
15-Aug	6.408	9.642	7.034	8.935
15-Sep	2.989	11.715	7.928	5.728
15-Oct	1.764	8.439	2.815	2.801
15-Nov	3.789	6.309	2.071	4.845
15-Dec	4.124	7.130	2.944	5.510

Below figure shows the bar graph for graphical representation of table 2 results.

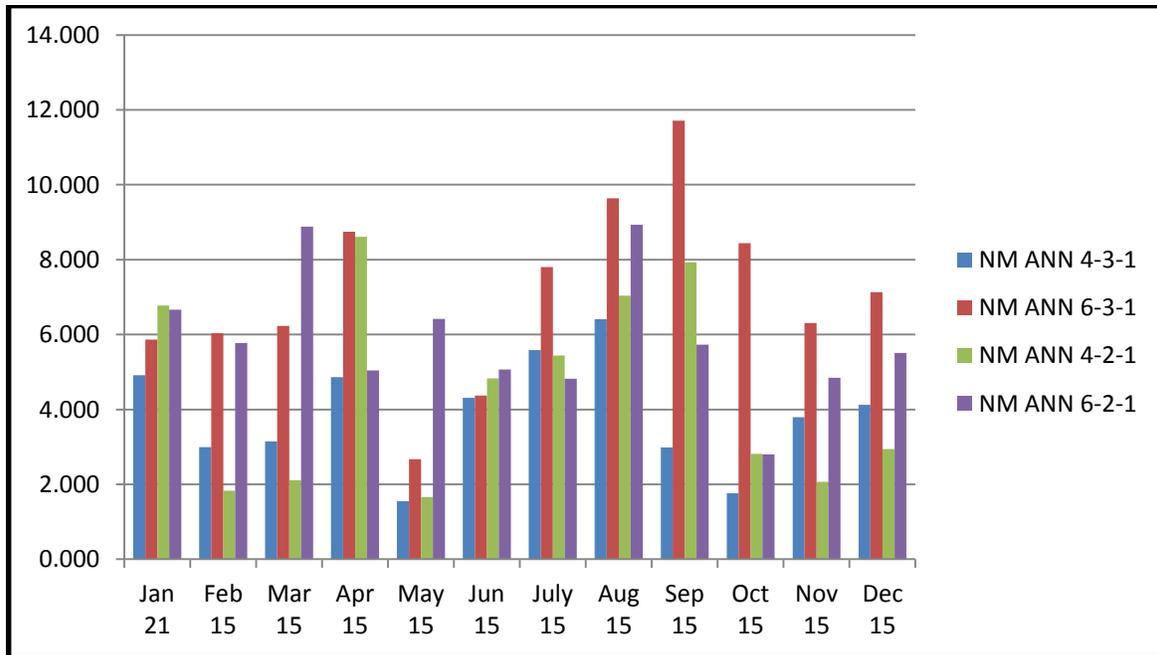


Figure 5: Comparison of models for California data, 2013

From above results, NM-ANN 4-3-1 is the best model among tested models while 4-2-1 is the second best. This proves that adding temperature as input in load forecasting, does not improve the model performance. Therefore, models without temperature input are behaving quite better than other with temperature inputs.

4.4 Case Study III: Proposed Hybrid Models Comparison with PSF

All models are applied on AEMO data of NSW and compared with Pattern Sequence-Based Forecasting (PSF) models of Irena[16]. Below figures show actual and forecasted load of NSW for February, 2011 while figures for January and December, 2011 are not shown.

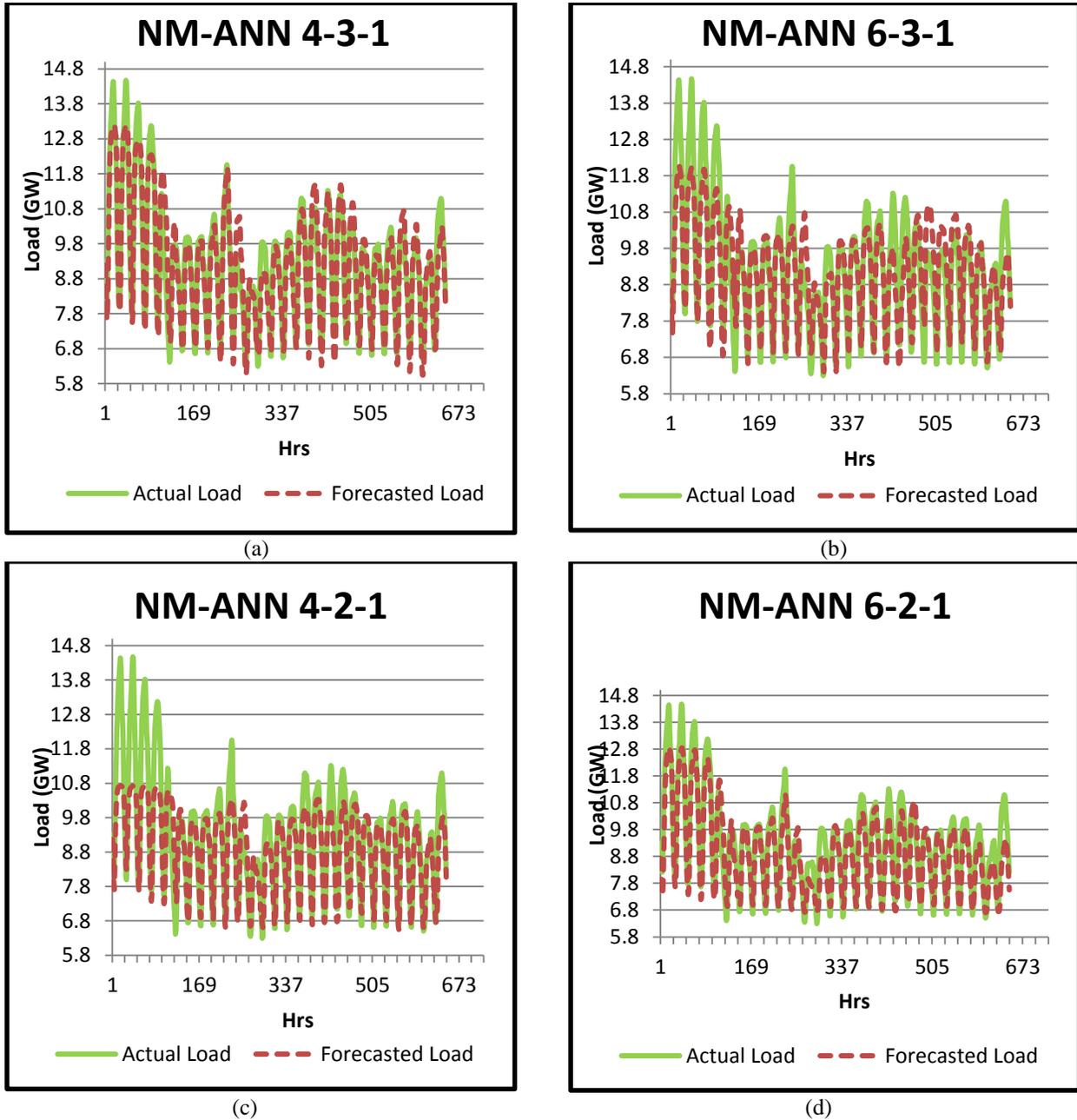


Figure 6: Actual and Forecasted Load for NSW data for Feb, 2011 (a) NM-ANN 4-3-1 model (b) NM-ANN 6-3-1 model (c) NM-ANN 4-2-1 model (d) NM-ANN 6-2-1 model

Table 3 shows comparison of PSF models with proposed models where NM-ANN 4-3-1 model shows quite promising result for February and December.

Table 3: MAPE comparison of NM-ANN with PSF-NN [16]

Models	PSF	PSF-NN1	PSF-NN2	PSF-NN3	NM-ANN 4x3x1	NM-ANN 6x3x1	NM-ANN 4x2x1	NM-ANN 6x2x1
January	4.85	5.16	5.28	3.92	5.62	6.90	4.59	8.84
February	5.81	5.68	7.35	5.05	4.71	6.66	5.47	5.71
December	5.61	4.73	9.93	7.07	4.63	4.88	8.76	6.53

Below is bar chart showing comparison of MAPE (%) of PSF [16] and NM-ANN proposed models.

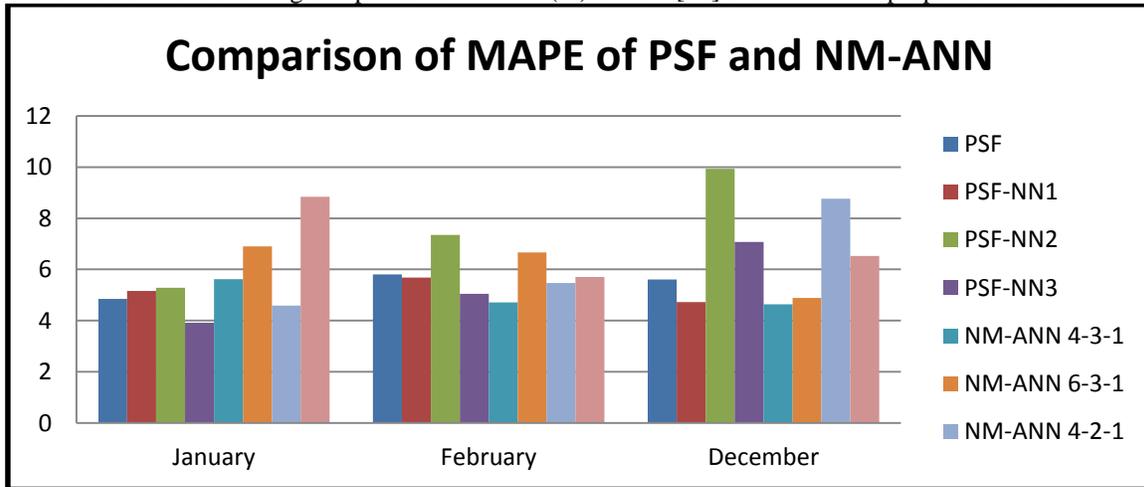


Figure 7: Bar chart for comparison of MAPE of PSF [17] and NM-ANN models

As above table and bar chart shows that NM-ANN 4-3-1 gives better MAPE than PSF models and other NM-ANN models for month of February and December while NM-ANN 4-2-1 gives second best MAPE after PSF-NN3 for month of January.

4.5 Case Study IV: Proposed Hybrid Models Comparison with SARIMA and BP Combined Model

South Australia (SA) data has been applied on all models and compared with combined model (SARIMA + BP) [17] for year 2005, 2006 and 2007. June and July month data is used for forecasting. Data of 2005 is used for training, data of 2006 is used for validation and data of 2007 is used for testing.

Table 4: MAPE comparison of proposed models with SARIMA + BP model

Models	SARIMA + BP [17]	NM-ANN 4x3x1	NM-ANN 6x3x1	NM-ANN 4x2x1	NM-ANN 6x2x1
June-July	5.13	5.05	10.95	6.07	7.90

Table 4 shows that NM-ANN 4-3-1 models gives better MAPE than SARIMA + BP model [17].

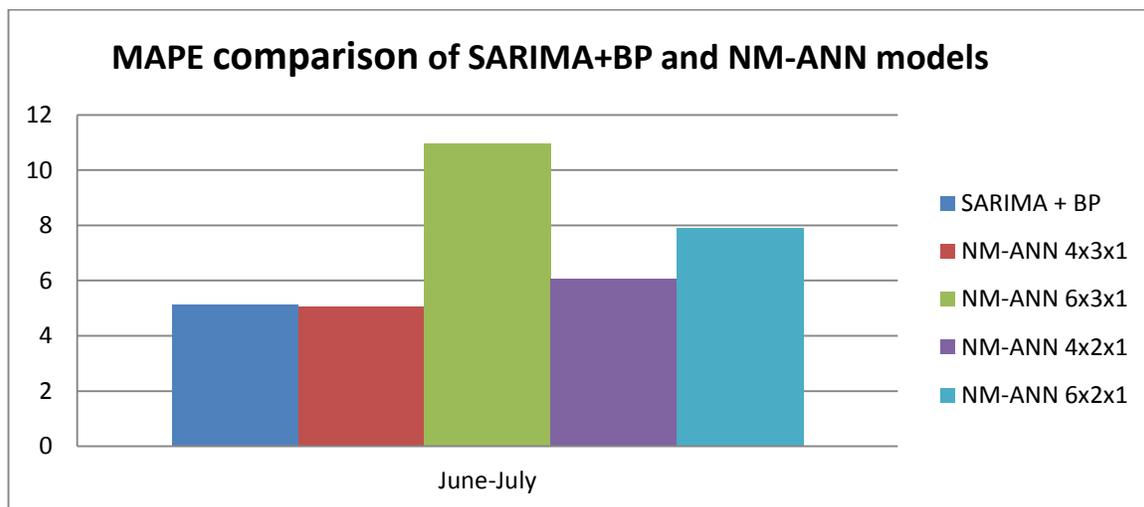


Figure 8: Bar chart for MAPE comparison of NM-ANN models with SARIMA + BP model

Bar Chart shows graphically the difference between MAPE of combined model (SARIMA + BP) and NM-ANN models. NM-ANN 4-3-1 outperforms combined model of and all other NM-ANN models.

By analyzing all above results of different case studies, NM-ANN 4-3-1 is found to be best forecasting model for different data companies discussed above.

5. CONCLUSIONS

A neural network based on Globalized Nelder Mead learning rule is proposed for short term load forecasting in this paper. Globalized Nelder Mead performance depends upon number of weights and is independent of number of samples. This property of NM outruns Back-Propagation (BP) and Lavenberg Marquardt (LM) training algorithms because it requires less training time for larger data sets.

It had been demonstrated that NM-ANN 4-3-1 gives more accuracy than Pattern Sequence-Based Forecasting (PSF) models and combined model (SARIMA +BP) discussed previously. This research can be enhanced by applying on industrial data. Further exploration can be done by utilizing proposed algorithm for weekly, monthly and yearly forecasting models. Future research can be done on deeper NM tuning of parameters and appropriate Neural Network selection for particular problems to achieve better results. Also, new efficient techniques can be used to aid NM for deeper and faster search to get further improved results.

REFERENCES

1. H. S. Hippert and J. W. Taylor, *An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting*. Neural Netw., 2010. **23**(3): p. 386-395.
2. H. K. Alfares and M. Nazeeruddin, *Electric load forecasting: Literature survey and classification of methods*. International Journal of Systems Science, 2002. **33**(1): p. 23-34.
3. P. P. Balestrassi, et al., *Design of experiments on neural network's training for nonlinear time series forecasting*. Neurocomput., 2009. **72**(4-6): p. 1160-1178.
4. D. Niu, et al., *Middle-long power load forecasting based on particle swarm optimization*. Comput. Math. Appl., 2009. **57**(11-12): p. 1883-1889.
5. Q. Wu, *Power load forecasts based on hybrid PSO with Gaussian and adaptive mutation and Wv-SVM*. Expert Syst. Appl., 2010. **37**(1): p. 194-201.
6. P. Balasubramaniam and M. S. Ali, *Robust exponential stability of uncertain fuzzy Cohen--Grossberg neural networks with time-varying delays*. Fuzzy Sets Syst., 2010. **161**(4): p. 608-618.
7. R. Rakkiyappan, P. Balasubramaniam, and J. Cao, *Global exponential stability results for neutral-type impulsive neural networks*. Nonlinear Analysis: Real World Applications, 2010. **11**(1): p. 122-130.
8. Y. Zhang, L. Wu, and S. Wang, *Bacterial foraging optimization based neural network for short-term load forecasting*. Journal of Computational Information Systems, 2010. **6**(7): p. 2099-2105.
9. S.-H. Liao, et al., *Training neural networks via simplified hybrid algorithm mixing Nelder--Mead and particle swarm optimization methods*. Soft Computing, 2014: p. 1-11.
10. N. D. Pham, *Improved nelder mead's simplex method and applications*. 2012, Auburn University. p. 109.
11. S. Ahmed, *Degenerated simplex search method to optimize neural network error function*. Kybernetes, 2013. **42**(1): p. 106-124.
12. N. Sharma, N. Arun, and V. Ravi, *An ant colony optimisation and Nelder-Mead simplex hybrid algorithm for training neural networks: an application to bankruptcy prediction in banks*. International Journal of Information and Decision Sciences (IJIDS), 2013. **5**(2).
13. F. Barzinpour, et al., *A hybrid Nelder--Mead simplex and PSO approach on economic and economic-statistical designs of MEWMA control charts*. The International Journal of Advanced Manufacturing Technology, 2013. **65**(9-12): p. 1339-1348.
14. M. A. Luersen and R. L. Riche, *Globalized Nelder-Mead method for engineering optimization*, in *Proceedings of the third international conference on Engineering computational technology*. 2002, Civil-Comp press: Stirling, Scotland. p. 165-166.
15. H. Ghiasi, D. Pasini, and L. Lessard, *Constrained Globalized Nelder--Mead Method for Simultaneous Structural and Manufacturing Optimization of a Composite Bracket*. Journal of COMPOSITE MATERIALS, 2008. **42**(7): p. 717-736.
16. I. Koprinska, et al. *Combining pattern sequence similarity with neural networks for forecasting electricity demand time series*. in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. 2013.
17. Y. Yang, et al., *A New Strategy for Short-Term Load Forecasting*. Abstract and Applied Analysis, 2013. **2013**: p. 9.

18. F. Gao and L. Han, *Implementing the Nelder-Mead simplex algorithm with adaptive parameters*. Computational Optimization and Applications, 2012. **51**(1): p. 259-277.
19. R. J. Hyndman and A. B. Koehler, *Another look at measures of forecast accuracy*. International Journal of Forecasting, 2006. **22**(4): p. 679–688.



Engr. Aamir Nawaz received B.Sc Eng. and M.Sc Eng. Degrees in electrical engineering from University of Engineering and Technology, Pesawar, and University of Engineering and Technology, Taxila, Pakistan in 2009 and 2014 respectively. He has almost three year experience of working in industrial sector. He is currently working as Lecturer in Institute of Engineering and Technology, Gomal University, Dera Ismail Khan, Pakistan. His main interests include AI tools applications in power system, power electronics, Smart Grids and HVDC.



Tahir Nadeem Malik received the B.Sc Eng. and M.Sc Eng. degrees from University of Engineering and Technology, Lahore (Pakistan) in 1984 and 1993 respectively, and Ph.D from University of Engineering and Technology, Taxila (Pakistan) in 2009 all in Electrical Engineering. Since 1987, he has been a Faculty Member in the Department of Electrical Engineering and Technology, Taxila, where he is currently serving as Professor and head Electrical Power System Group. His research interests are in power system operational planning, AI application in Power System, and smart grid.

List of Figures

Figure 1: (a) 4x3x1 ANN Network, (b) 6x3x1 ANN Network, (c) 4x2x1 ANN Network, (d) 6x2x1 ANN Network ..5	
Figure 2: NM Training Method for optimization of ANN weights	6
Figure 3: Flowchart of Globalized NM trained ANN	7
Figure 4: Bar chart for comparison of MAPE of different developed models for California and AEMO data	8
Figure 5: Comparison of models for California data, 2013	9
Figure 6: Actual and Forecasted Load for NSW data for Feb, 2011 (a) NM-ANN 4-3-1 model (b) NM-ANN 6-3-1 model (c) NM-ANN 4-2-1 model (d) NM-ANN 6-2-1 model	10
Figure 7: Bar chart for comparison of MAPE of PSF [17] and NM-ANN models	11
Figure 8: Bar chart for MAPE comparison of NM-ANN models with SARIMA + BP model	11

List of Tables

Table 1: Comparison of different models of this research for different load company’s data	8
Table 2: Comparison of models for California data, 2013	9
Table 3: MAPE comparison of NM-ANN with PSF-NN [16]	11
Table 4: MAPE comparison of proposed models with SARIMA + BP model	11