SHORT TERM LOAD FORECASTING USING NEURAL NETWORKS

L. B. NIGRINI & G.D. JORDAAN

Abstract

Several forecasting models are available for research in predicting the shape of electric load curves. The development of Artificial Intelligence (AI), especially Artificial Neural Networks (ANN), can be applied to model short term load forecasting. Because of their input-output mapping ability, ANN's are well-suited for load forecasting applications.

ANN's have been used extensively as time series predictors; these can include feed-forward networks that make use of a sliding window over the input data sequence. Using a combination of a time series and a neural network prediction method, the past events of the load data can be explored and used to train a neural network to predict the next load point.

In this study, an investigation into the use of ANN's for short term load forecasting for Bloemfontein, Free State has been conducted with the MATLAB Neural Network Toolbox where ANN capabilities in load forecasting, with the use of only load history as input values, are demonstrated.

Keywords: Short Term Load Forecasting (STLF), Artificial Neural Network (ANN), time series, multilayer feed forward network, real time load data.

1. INTRODUCTION

Forecasting is usually based on identifying, modelling, and extrapolating the patterns found in historical data. Because historical data do not normally change dramatically or very quickly, statistical methods, e.g. neural networks, can be useful for short term forecasting [6, p.302] using historical occurrences as input.

Many neural network training methods and architectures exist that can be used to model a forecasting problem, but only the applications and principles used in this investigation will be discussed.

1.1 Time series prediction with a sliding widow

Most forecasting problems involve the use of time series data [15], which are amongst the oldest methods applied in load forecasting [7, p.904]. An electric load pattern is principally a time series [1, p.2]. A time series can be described as a sequential set of hourly, daily, or weekly data measured at regular intervals. A time series forecasting takes an existing series of data $x_{t-n}\ldots$, x_{t-3}, x_{t-2}, x_{t-1}, x_t and forecasts the x_{t+1} data value where:

x_{t+1} is the target value of x predicted by the neural network model;
x_t is used to indicate the value of x for the previous observation;
x indicates the observation; and
t represents the index of the time period.

Quantitative forecasting techniques involve the use of time series data (historical) and a forecasting model, i.e. an artificial neural network capable of representing complex nonlinear relationships. The model summarises patterns in the data and expresses a statistical relationship between previous and current values of the variable x, in this case. Patterns in the data can then be projected into the future, using this model [12, pp.1-5].

In neural network time series forecasting, the sliding window approach is available in updating forecasting models over time. The sliding window approach adds a new training value and disregards the oldest one from the training sample set, which is used to update the neural model. The input sample set size, presented to the neural network, stays fixed as it slides forward with time.

As an example of a sliding window technique, Figure 1 shows a standard, trained back- propagation neural network performing time series prediction using four equally time spaced input data points, sliding over the full set of trained data to predict the next output data point value [3].

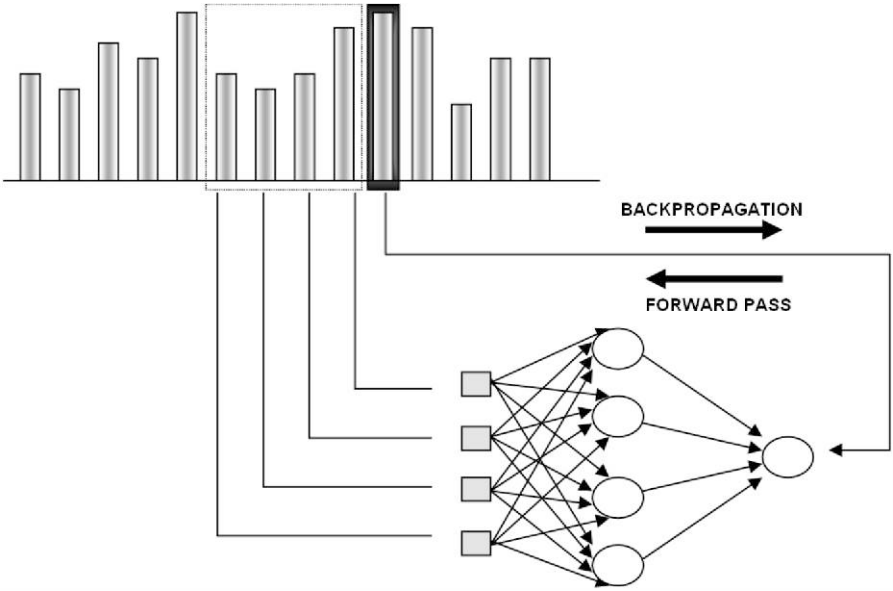

Figure 1: Time series prediction using a fully connected feed forward neural network and the "sliding window" approach.

It is not easy to determine an appropriate input sample set size that is useful for training the forecasting network. Too small a sample input set may restrict the power of the neural network to do proper forecasting by not adjusting the network parameters properly. However, the sliding window approach, with the changing starting point in the training sample set, is more suitable to contemplate changes occurring in the underlying process of the time series [13, pp.3-8].

The rate at which the samples are taken will dictate the maximum resolution of the model. Once the network is trained with this set, the same technique is used with new data points to predict the future electric load.

1.2 A typical load pattern to be used for forecasting purposes.

In Figure 2 it can be seen that the valleys in the daily pattern fluctuate little in magnitude and the base load floats at more or less 150 MW. The Direct Load Control (DLC) peaks shown in Figure 2 are not significant for forecasting, as they are not driven by consumer demand.

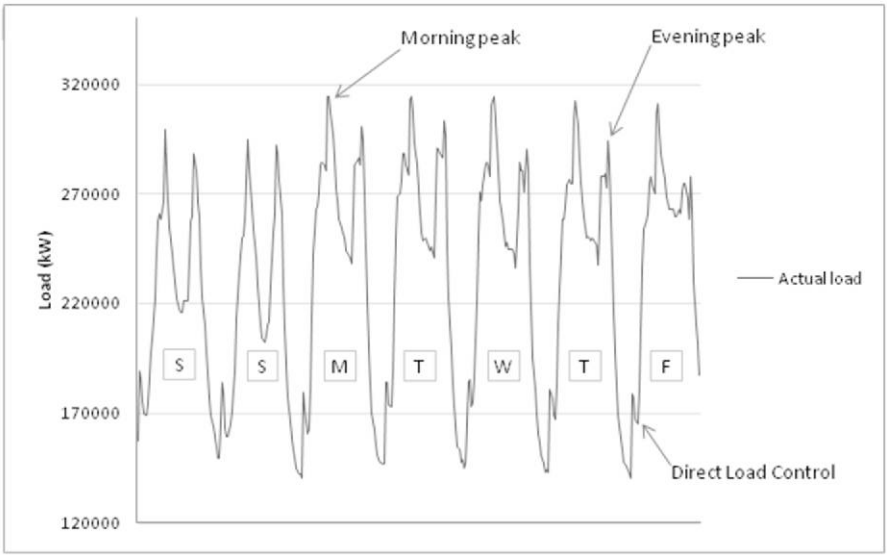

Figure 2: Load consumption of Week 4, July 2009 showing the actual load and different peaks.

Also, from Figure 2, the morning daily peak for the Bloemfontein weekly load demand normally occurs between 11:00 to 12:00 and the evening daily peak electric consumption varies between 19:00 and 21:00. These “twin peak” shaped load levels, in essence, symbolise the main daily peaks, namely: morning and evening peaks, typical also in countries like Portugal [17, p.8] and Canada [8, p.529].

These “twin peaks” confirm the non-linear features of the load and often affect the accuracy of load predictions negatively due to 'very sharp transitions in the load curve shape'.

The peak load levels for each day, Saturday to Friday, indicate the Maximum Demand used for that day. It varies between 300 MW and 320 MW depending on the day-type one is considering. An underlying repeating cyclic pattern, with random magnitudes at random periods, can be observed in the data.

2. FORECASTING WITH A 6:1:1 ANN FEED FORWARD MODEL

2.1 Methodology

The network was developed using three stages: the training stage, the testing stage and the evaluation stage. The workflow for this methodology had six primary steps:

1. Collect load data.
2. Create the network.
3. Configure the network.
4. Train the network.
5. Test the trained network with the training data used in step 4.
6. Validate the network with new data.

2.2 The ANN used: Type and arrangement

Numerous network topologies are available; each with its inherent advantages and disadvantages. Some networks sacrifice speed for accuracy, while others are capable of handling static (batch) or continuous input variables.

For this particular task, the data sets were not large and did not consist of multiple arrays of input variables such as temperature, active power, reactive power or day type, which means that there was no use for the design of a complex network topology. Hence, in order to arrive at an appropriate robust network topology, a multilayer feed-forward network was used. It consisted of one input layer, one hidden layer and one output layer. Only one neuron is used in the hidden layer and one in the output layer, as shown in the MATLAB representation in Figure 3.

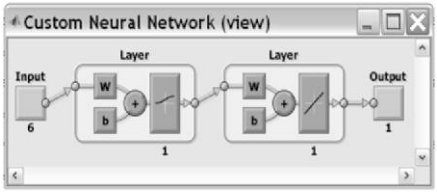

Figure 3: The 6:1:1 feed forward ANN used for the simulations

2.3 Neural network input variables

Electric load data consisting of active and reactive power were obtained from the two 132 kV feeder voltage sources at Harvard substation, west of Bloemfontein. The active power component was extracted from the load data and used to develop the artificial neural model to predict power demand with a lead time of half an hour.

There is no general rule that can be followed when selecting the number of network input variables. It depends on engineering judgement and experience and is carried out almost entirely by trial and error. As shown in Figure 3, six input variables were selected heuristically from past experience with STLF ANN models.

2.3.1 Data pre-processing

Data pre-processing was done in three phases:

1. Selected data were copied to files in the EXCEL format to make importing and exporting the data to MATLAB more efficient.
2. The data were checked for outliers and missing data. Minimal adjustments were identified and discarded or rectified.
3. Data entering the MATLAB environment were normalised to the range [-1; +1] for the ANN to facilitate training and prevent "squashing" by the sigmoid activation function [14, p.465].

2.4 Training Algorithm

Back propagation training algorithms are often too slow for practical problems, so one can use several high performance algorithms that can converge faster than back propagation algorithms. These faster algorithms fall into two main categories: heuristic techniques (variable learning rate back propagation, resilient back propagation) and numerical optimization techniques (conjugate gradient, quasi-Newton, Levenberg-Marquardt). The Levenberg-Marquardt (LM) algorithm was used in this case.

2.4.1 A brief description of the Levenberg-Marquardt modification to the back propagation training algorithm

This high-performance algorithm is one of the faster methods for training moderate-sized feed-forward neural networks up to several hundred weights. It was designed for minimizing functions that are the sums of squares of other nonlinear functions. This is beneficial to neural network training where the performance index is the mean square error (MSE). It is a variation of Newton's method.

Newton's method requires calculation of the second derivative, so it is only used when it is feasible to calculate the Hessian matrix H [18, p.15].

The Levenberg-Marquardt algorithm approaches second-order training speed without having to compute the Hessian matrix, which contains the second derivatives of the performance index along the network weights and biases axis. When the performance function has the form of a sum of squares, the Hessian matrix can be approximated as

$$\mathbf{H} = \mathbf{J}^{T}\mathbf{J}$$

and the gradient can be computed as

$$\mathbf{g} = \mathbf{J}^{t}\mathbf{e}$$

Where

J is the Jacobian matrix that contains the first derivatives of the network errors with respect to the weights and biases and,
e is a vector of network errors [9, p.25].

The Jacobian can be calculated using the standard backpropagation algorithm, which is less complicated than calculating the Hessian matrix.

If the weight and bias update from the standard backpropagation algorithm is equal to some value δ

$$\delta = \Delta\mathbf{W}^{m}(k) + \Delta\mathbf{b}^{m}(k)$$

Then the Levenberg-Marquardt algorithm approximates a function that can be solved by:

$$(\mathbf{J}^{T}\mathbf{J} + \mu\mathbf{I})\delta = \mathbf{J}^{t}e$$

Where

J is the Jacobian matrix for the network,
μ is the Levenberg –damping factor Mu,
I is the identity matrix,
δ is the weight update vector and
e is the error vector containing the output errors for each input vector used for training the network.

The weight update δ tells us by how much the network weight and bias parameters should be adjusted to achieve a near zero performance goal.

The μ is increased, e.g. by a factor of 10, it is a step closer to the gradient descent direction.

Newton's method is faster and more accurate near an error e minimum, so the intention is to change toward Newton's method as quickly as possible. As a result, μ is decreased after each successful step (closer to a zero performance goal) and is increased only when a tentative step would increase the performance function. In this way, the performance function (sum of the squares) is always reduced after each iteration cycle of the algorithm. The algorithm is assumed to have converged when some performance goal, in this case, zero, is reached.

Typical Levenberg-Marquardt algorithm training parameters, with their default values, are shown in Table 1:

Table 1: LM training parameters

Maximum number of epochs to train	100
Performance goal	0
Maximum validation failures	5
Minimum performance gradient	1e-10
Initial μ	0.001
μ decrease factor	0.1
μ increase or adjustment factor	10

2.4.2 Training sequence used

To obtain the experimental results shown in this paper, the performance of the selected neural model was measured by training and testing the network with the four week data set of July 2010. The NN was then evaluated with the four week data set of July 2011, measuring the Mean Absolute Percentage Error (MAPE), one week at a time, shown in the sequence:

1. Input July 2010 load data ► train a new network ► compare forecasted and actual July 2010 load data to validate the trained network.

2. Input July 2011 load data ► present to the taught network ► compare forecasted and actual July 2011 load data to validate the trained network.

3. MEASURING THE NEURAL NETWORK MODEL PERFORMANCE

The most important measure of a trained neural network's performance is its forecasting accuracy using data other than the training data. A frequently used "measures of accuracy" in the forecasting literature [22, p.51], [16, p.6], and [10, p.370] is the:

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|A_t - F_t|}{A_t} \times 100\ \%$$

Where

A_t = actual value and At ≠0;
F_t = forecast value;
n = number of fitted data points.

In addition, besides the MAPE criterion other statistical measures are available to further evaluate the network's performance, using linear regression or scatter plots. The success of a trained network can be considered to some degree by evaluating the MAPE on the training, validation, and test sets, but it is often useful to look into the network response in more detail. One option is to calculate the Pearson's correlation coefficient (the R value) between the outputs and targets [2, pp.298-318], [11, p.64]. It is a measure of how well the variation in the output is explained by the targets. The formula for R is:

$$R = \frac{\sum_{t=1}^{n}(A_t - \bar{A})(F_t - \bar{F})}{\sqrt{\sum_{t=1}^{n}(A_t - \bar{A})^2}\sqrt{\sum_{t=1}^{n}(F_t - \bar{F})^2}}$$

Where

At = actual value;
$\bar{A}$ = actual data average;
Ft = forecast value;
$\bar{F}$ = forecasted data average;
t = data point number;
n = number of fitted data points.

If R=1, this indicates that there is an exact linear relationship between outputs and the actual values (targets). If R is close to zero, then there is no linear relationship between outputs and targets.

4. SIMULATION RESULTS

4.1 The forecasting model performance for July 2011

The purpose of this paper was to observe the following:

- Measurement of the trained network's performance, i.e. the Forecasting Accuracy, which was to check that the prediction error is repeatable from week to week. Accuracy is limited by systematic (repeatable) errors.

- How well the network models detect the daily peak load levels [21, p.1397]. From a practical point of view, the forecasting error is usually less critical at off-peak load levels, e.g. the DLC levels indicated in Figure 2, compared to peak load levels. Therefore, only the seven daily peak load levels were calculated, using the MAPE, adapted from [4].

- A calculation of the correlation coefficients (R values) for week 1 to week 4, obtained from the scatter plots, using July 2011's actual and forecasted results.

Utilising the training data (July 2010) and forecasting data (July 2011), the results are represented in two sets as follows:

1. The monthly data from July 2010 (used for training) were used to check the performance of the trained network, shown in Figure 4;

2. The data in July 2011 were used as a completely independent set to validate the trained network's performance, shown in Figure 5.

Day type and hour of the day details are irrelevant and therefore absent on the x-axis of each weekly plot in Figures 4 and 5.

The y-axis in Figures 4 and 5 represents the kW Load used in each week.

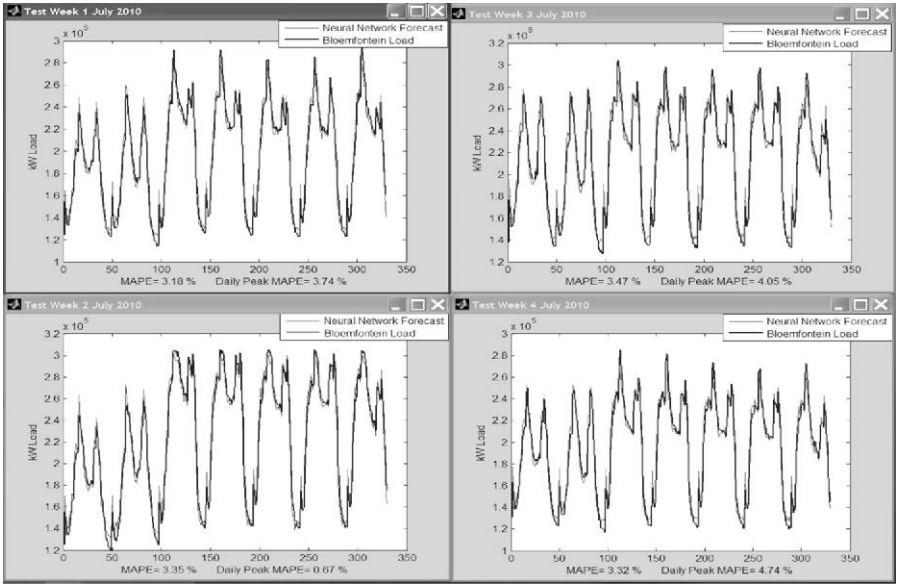

Figure 4: Plots of the superimposed actual and forecasted results of the neural network for the training and test set (four weeks from the 3rd July 2010 to 30th July 2010)

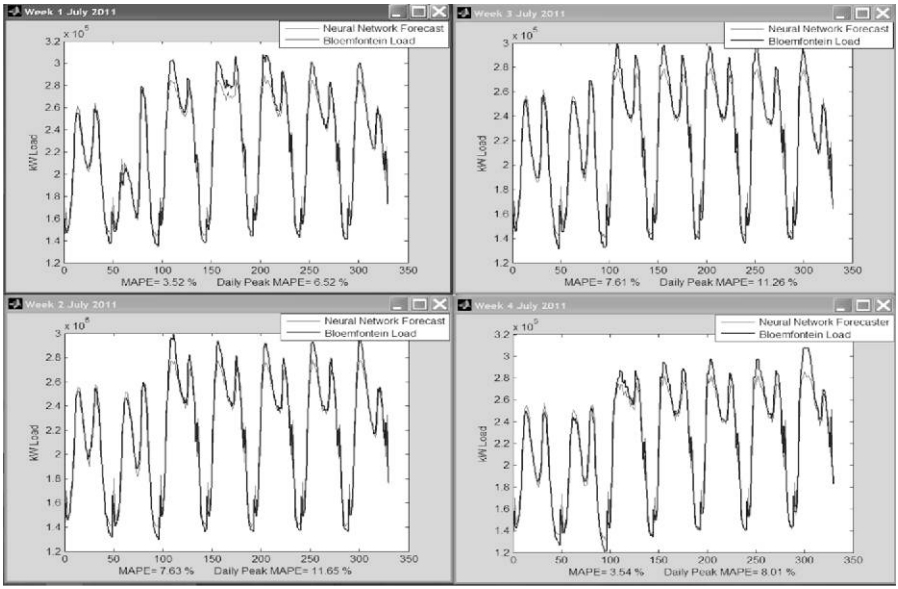

Figure 5: Plots of the superimposed actual and forecasted results of the neural network for the vailidation data set (four weeks from the 2nd July 2011 to 29th July 2011)

4.2 Comparing the weekly performance of the test- and the validation sets using the MAPE results in Figures 4 and 5

From the daily peak MAPE values, which vary between 0.67 % and 4.74 %, shown in Figure 4, the ANN under predicts the actual daily peak load levels. The goodness of fit of the rest of the forecasted and actual load graphs, superimposed on each other, is visually clear.

In Figure 5, the actual daily peak load levels are significantly underpredicted. This is indicated in the daily peak load MAPE values, which varies between 6.52 % and 11.65 %, shown in each subgraph in Figure 5.

Visually comparing the peak values of the daily loads in the validation set data shown in Figure 5 with the peak loads in the weeks in the training data shown in Figure 4, it can be seen that the training data's daily load peaks were, on average, slightly lower than the daily load peaks shown in Figure 5. So, although the neural network shows acceptable MAPE values during training, it did not predict the new data well (see Table 2). Considering the fact that neural networks can only “model” data given to it within its training limits, these large peak load level errors are to be expected during performance measurements [20, p.109], [19, p.249].

Table 2: The weekly MAPE values obtained from Figure 4 and Figure 5, showing the trained and forecasted MAPE results for the months in July 2010 and July 2011

	July	Week 1	Week 2	Week 3	Week 4
Test set	**2010**	**3.18%**	**3.35%**	**3.47%**	**3.32%**
Validation set	**2011**	**3.52%**	**7.63%**	**7.61%**	**3.54%**

Examining the overall test set's weekly MAPE performance results for the network in Table 2, the difference in the MAPE performance of the test set is less than 1%. The small discrepancies between the forecasted results indicate that overfitting is a minimum [5, p.731] for the test set.

Comparing the overall validation set's weekly MAPE performance results for the same network in Table 2; the difference in the MAPE performance can vary up to more than 4%, which indicates that the ANN is not generalising well.

4.3 Comparing the weekly performance of the validation set using correlation coefficient results from the scatter plots in Figure 6

The scatter plots of the network outputs versus the targets (plotted as open circles) are superimposed on the best linear fit (solid line) and the perfect fit (dashed line).

The Target (x-axis) and Output (y-axis) represent the actual and predicted kW Load respectively. The R values (correlation coefficients) taken from Figure 6 are shown grouped in Table 3. The R value for the total response is better than 0.98

TABLE 3: The weekly correlation coefficients taken from Figure 6, rounded to three decimal places.

	July	Week 1	Week 2	Week 3	Week 4
Corr. coefficient (R)	**2011**	**0.987**	**0.986**	**0.986**	**0.986**

For a perfect fit, all the data should fall along a 45° line, where the network outputs are equal to the targets. In Figure 6, week 1 to week 4, it is easy to distinguish the best linear fit line from the perfect fit line, which indicates that the trained network could perform better.

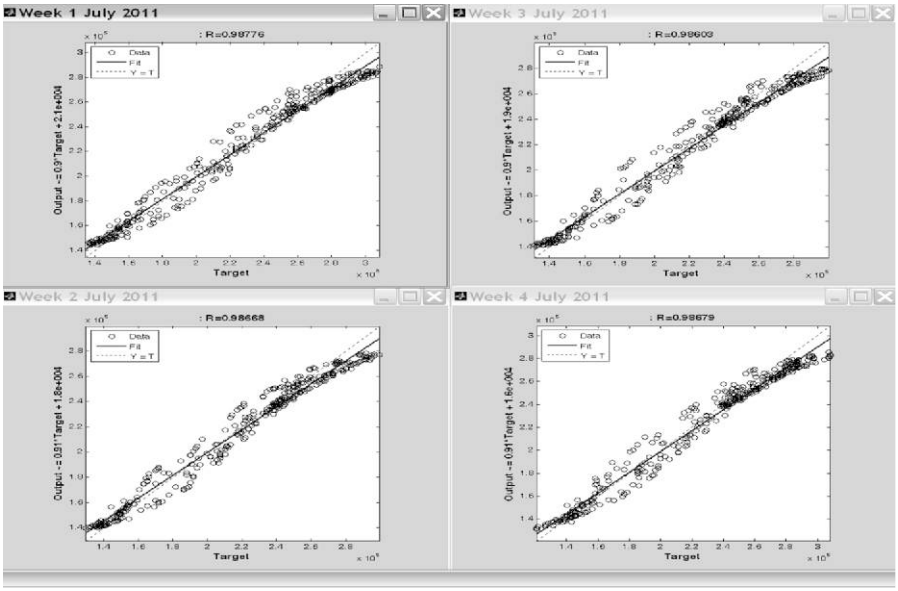

Figure 6: Plots of the Target values (actual kW load used) versus the Output values (neural network forecast) of the neural network for the four weeks from the 2nd July 2011 to 29th July 2011

5. CONCLUSION

A compact ANN model was trained to predict the load for July 2011 using the historical data of July 2010 as a training set.

The input vector and number of neurons in the hidden layer were kept to a minimum to avoid model over-parameterization.

The MAPE results obtained from Table 2 show that the forecasting error is not repeatable for each week of July 2011. The trial-to-trial validation accuracy varies between 3.52% and 7.63%.

Although the R values in Table 3 look consistent, generalisation was not complete as there is some dispersion around the 45° line as seen in the scatter plots in Figure 6.

In this case, the network response is not entirely satisfactory. Further investigation into the network architecture and the size of the input vector needs to be done.

6. ACKNOWLEDGEMENT

The authors wish to thank Mr. WR Kleyn of Eskom Bloemfontein - Electricity Delivery Systems Support for supplying the load data.

7. REFERENCES

Almeshaiei, E. and Soltan, H. A methodology for Electric Power Load Forecasting, Alexandria Engineering Journal, In Press, Corrected Proof, Available online 27 July 2011, ISSN 1110-0168, DOI: 10.1016/j.aej.2011.01.015. http://www.sciencedirect.com/science/article-/pii/S1110016811000330 (accessed August 2011)

Berk, K.N. and Carey, P. Data Analysis with Microsoft Excel. Toronto: Brooks/Cole, 2004.

Crone, S.F. EVIC'05 Slides - Forecasting with Neural Networks Tutorial. 15 December 2005. (accessed July 2008).

Du Plessis, L. System Optimisation and the Impact of the Short Term Load Forecast. http://www.eepublishers.co.za/article/system-optimisation-and-impact-of-short-term-load-forecast.html. (accessed 14/6/ 2011).

Fidalgo, J.N. and Matos, M.A. Forecasting Portugal Global load with Artificial Neural Networks. Vol. 2, in Artificial neural networks - ICANN 2007: 17th International Conference, edited by J. Marques de Sá, 731. Berlin: Springer-Verlag, 2007.

Ghiassi,M., Zimbra, D.K. and Saidane, H. Medium term system load forecasting with a dynamic artificial neural network model. Electric Power Systems Research, Volume 76, Issue 5, March 2006, Pages 302-316. a. ISSN 0378-7796, DOI: 10.1016/j.epsr.2005.06.010.
http://www.sciencedirect.com/science/article/pii/S0378779605001951
(accessed July 2011).

Hahn, H., Meyer-Nieberg, S., and Pickl, S. Electric load forecasting methods: Tools for decision making, European Journal of Operational Research, Volume 199, Issue 3, 16 December 2009, Pages 902-907, ISSN 0377-2217, DOI: 10.1016/j.ejor.2009.01.062. http://www.sciencedirect.com/science/-article/pii/S0377221709002094.
(accessed May 2010).

Kandil, N., Wamkeue, R., Saad, M. and Georges, S. An efficient approach for short term load forecasting using artificial neural networks, International Journal of Electrical Power & Energy Systems, Volume 28, Issue 8, October 2006, Pages 525-530, ISSN 0142-0615, DOI: 10.1016/j.ijepes.2006.02.014.
(http://www.sciencedirect.com/science/article/pii/S0142061506000676)
(accessed May 2011).

Madsen, K., Nielsen, H.B. and Tingleff, O. Methods for Non-linear Least Squares Problems. 2nd Edition. Informatics and Mathematical Modelling, Technical University of Denmark, DTU. April 2004.
www2.imm.dtu.dk/pubdb/views/edoc_download.../imm3215.pdf.
(accessed December 2010).

Mandal, P., Senjyu, T., and Urasaki, N. Toshihisa Funabashi, A neural network based several-hour-ahead electric load forecasting using similar days approach, International Journal of Electrical Power & Energy Systems, Volume 28, Issue 6, July 2006, Pages 367-373, ISSN 0142-0615, DOI: 10.1016/j.ijepes.2005.12.007.
http://www.sciencedirect.com/science/article/pii/S0142061506000275
(accessed March 2011).

Middleton, M.R. Data Analysis Using Microsoft Excel. 3. Toronto: Brooks/Cole, 2004.

Montgomery, D.C., Jennings, C.L. and Kulahci, M. Introduction to Time Series Analysis and Forecasting. Hoboken: John Wiley & Sons, Inc, 2008.

Morantz, B.H., Whalen, T.G., Zhang, P. A Weighted Window Approach to Neural Network Time Series Forecasting. Georgia State University, 2003. 3-8.

Palmer-Brown, D., Draganova, C., Pimenidis, E. and Mouratidis, H. Engineering Applications of Neural Networks. 11th International Conference, EANN. London: Springer, 2009. 465.

Plummer, E.A. Time Series Forecasting with Feed-Forward Neural Networks: Guidelines and Limitations. University of Wyoming, Laramie, Wyoming, USA. 2000.

Santos, P, J, Martins, A. G. and Pires., A.J. Designing the input vector to ANN-based models for short term load forecast in electrical distribution sytems. 2004.http://www.inescc.pt/documentos/14_2004.pdf (accessed June 2011).

Santos, P.J., Martins, A.G., Pires, A.J., Martins, J. F. and Mendes, R. V. Short-Term Load Forecast Using Trend Information and Process Reconstruction. www.inescc.pt/documentos/14_2004.pdf (accessed May 2009).

Shan, S. A Levenberg-Marquardt Method For Large-Scale Bound-Constrained Nonlinear Least-Squares. The University of British Columbia, Computer Science. Vancouver. Canada (2008): 1-8.

Slade, P. and Gedeon, T.D. Bimodal Distribution Removal. In New Trends in Neural Computation: International Workshop on Artificial Neural Networks, IWANN ,93 Sitges Spain June 1993, edited by Joan Cabestany, Alberto Prieto José Mira, 249. Berlin: Springer-Verlag, 1993.

Swingler, K. Applying Neural Networks: A Practical Guide. San Francisco: ACADEMIC PRESS, 1996.

Yalcinoz, T. and Eminoglu, U. Short term and medium term power distribution load forecasting by neural networks, Energy Conversion and Management, Volume 46, Issues 9-10, June 2005, Pages 1393-1405, ISSN 0196-8904, DOI: 10.1016/j.enconman.2004.07.005.
http://www.sciencedirect.com/science/article/pii/S019689040400192X (accessed January 2011).

Zhang, G., Patuwo, B.E. and Hu, M.Y. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting Volume 14, Issue 1, 1 March 1998, Pages 35-62, ISSN 0169-2070, DOI: 10.1016/S0169-2070(97)00044-7. http://www.sciencedirect.com/science/article/pii-/S0169207097000447 (accessed August 2008).