Medium Term Load Forecasting Using Statistical Feature Self Organizing Maps (SOM)

N.N. Atira[1], I. Azmira[1], Z.H. Bohari[2], N.A. Zuhari[1] and N.F.M. Ghazali[1]
[1]*Fakulti Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka*
[2]*Fakulti Teknologi Kejuruteraan Elektrik dan Elektronik, Universiti Teknikal Malaysia Melaka*
intan.azmira@utem.edu.my

***Abstract*— Load forecasting is an essential tool for power system activity and planning. With the increase in development and the expansion of power system, it is important for the electrical utility to make a decision in ensuring that there would be enough supply of electricity to deal with the increasing demand. This research presents the Medium Term Load Forecasting using the artificial neural networks: Kohonen's Self-organizing Maps. The main purpose of this paper was to understand the ability of Self-Organizing Maps in forecasting the load demand and to train and test via Self-Organizing Maps method using the selected features. Using data provided by the Global Energy Forecasting Competition (GEFCom2012), this paper focused on the missing data from the year 2005 and 2006 for the load forecasting. The loaded data were trained, tested, and forecasted using SOM Toolbox in MATLAB software. The accuracy of the forecasted data was determined by calculating the error of each forecasted data by comparing them with the actual data. Then, the Mean Absolute Percentage Error was computed to determine the accuracy of the results.**

***Index Terms*— Artificial Neural Network; Load Forecasting; Medium Term; Self-Organizing Maps**

I. INTRODUCTION

Forecasting is the development of making a decision about an event, wherein the actual outcome has not yet been observed, and it is the basic facet of making a decision. Load forecasting is a critical device for power system activity and planning [1], [2]. The development of power system and its increased complexity have affected various aspects of the electric power generation and consumption, such as the load management, energy exchange, spot pricing, and etc. In this case, the forecasting procedure has turned out to be much more complex, and more precise forecasts are required [3].

Load forecasting can be classified into three categories, which are the short term, medium term, and long-term. It is essential to have forecast for various time horizons according to the different application within a utility company. Additionally, the nature of this forecast is different as well [4]. For the short-term load forecasting (STLF), the operation of a power system, such as the unit commitment, economic dispatch, security assessment and etc. is important. The long-term load forecasting (LTLF) is frequently used in power system expansion and planning such as, the construction of new power generator, while the mid-term load forecasting is normally involved in the operative planning of power systems such as, the schedule of maintenance and power generation coordination [2], [5].

For the medium term load forecasting (MTLF), a lot of variables are contributing to the load. This causes an exact prediction of load forecast becomes a complicated process since the variables are characterized to be a non-linear and non-stationary process. The process is complicated since the load can encounter rapid changes due to many factors and variables such as weather, seasonal and macroeconomic variations; thus, the load forecasting using the classical prediction models are not suitable [6], [7].

It is important for the forecasting to be emphasized at all level as the after-effect of under and over forecasting will affect all the stakeholders of electricity utilities. In this regard, detailed research on forecasting method is required to forecast the load so that the after-effect of under and over forecasting, especially the power utility could be minimized.

A. Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) methods are considered as the more advanced forecasting methods, which are useful for a multivariable model. It has been widely used in electricity load forecasting since 1990 due to its ability to forecast the non-linear and non-stationary load. ANNs are electronic clone based on the structure of neural in the brain. The artificial neuron process is motivated by neural models, which recognize and use the pattern to utilize and affect the formation of huge parallel networks, and coach those networks to solve specific problems [8], [9].

The ANNs are essentially non-linear circuits, in which their output is in linear or non-linear mathematical functions. The input of the data may be the outputs or inputs of the other network elements. Normally, the ANNs have three layers, as shown in Figure 1. The first layer is connected to the input variables known as the input layer. The third layer is connected to the output variables known as the output layer. The layer in-between the input and output layer is known as the hidden layer. The hidden layer can exist in more than one layer [10], [11].

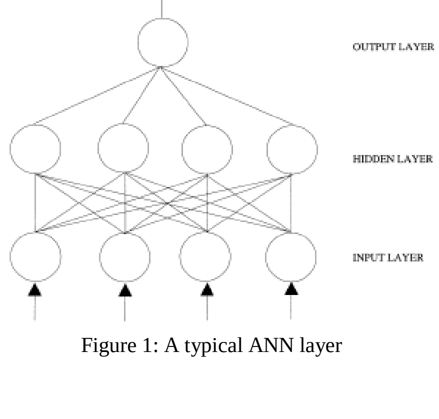

Figure 1: A typical ANN layer

The neural networks are currently the most popular methods to develop load forecasting tools. The multilayer perceptron (MLP) is the main structure used in these models although other techniques such as self-organizing maps or recurrent networks are also good candidates for promising results [12].

In a broad sense, there are two types of learning; the supervised learning and the unsupervised learning. Supervised learning means that the exact answer is well-known and the data is used to train the network for a given problem. This learning utilized both input and output variables. The input variables are used to accommodate initial data while the output variables can be used to differentiate with input data to determine the fault. The unsupervised learning means the exact answer is not known. The network needs to discover its own pattern based on input data, which purely depends on the input variables. The SOM obtains a statistical feature of the input data and is applied to a wide field of data classification. The generated output will not be used as the data from where it is learned. Further, the learning also does not need human interaction as it can be handled with a broad and/or complicated dataset [13], [14].

B. *Organizing Maps*

The Self-Organizing Maps (SOM) have been used since other architecture required supervised training, hence they do not have a favorable ability to disclose data outside of the domain of the trained data. Thus, the SOMs have been designed to overcome the following shortcomings.

The SOMs is a 'self-organizing' method since due no supervision is required. It learned based on its own through unsupervised ambitious learning. The 'Maps' means that it endeavors to map its weight to comply with a given input data [15]. The SOM made up of neurons grouped on a low-dimensional grid. Every neuron is a d-dimensional weight vector, where the d is the dimension of input vectors. The neurons are allied to the near neurons by a neighboring other's relationship, which indicates the topology and the structure of the map [16].

Before any training process happens, the initial values are given to the prototype vectors. The SOM is very strong with respect to the initialization, but when it is properly attained it allows the algorithm to converge actively to the excellent solution. There are three initializations of SOM, which are the random, linear and sample initialization.

The random initialization means that the SOM algorithm can be initialized using approximate values. It has been demonstrated that initially, the un-ordered vectors will be in order in the long run, with usual applications in a few hundred initial steps. It is selected randomly and independently from the data points [17].

The linear initialization is where the weight vectors are initialized in orderly configuration along the linear subspace by two dominant eigenvectors of the input data set. The sample initialization is where the weight vectors are initialized with random samples drawn from the input dataset [13], [18].

C. *U-Matrix*

The U-Matrix is a representation of the SOM that visualizes the distances between the neurons. The distance between the adjacent neurons is calculated and presented with different coloring between the adjacent nodes. This matrix represents the distance between each neuron and all its neighboring ones and is able to reveal the local cluster structure of the map. The farther the distance is, the higher the difference between them will be, resulting in a higher similarity [19], [20].

II. METHODOLOGY

Figure 2 shows the flowchart of the methodology for the MTLF using SOM. The data used were from Global Energy Forecasting Competition (GEFCom2012), provided by the US utility [21]. The data were organized according to temperature history, holiday list, seasons, month, day and date. The total load and average temperature were calculated for each month in the year 2004, 2005 and 2006.

The data from the year 2004 were trained to forecast the data for the year 2005 while the data from 2005 were used to train for load forecasting for the year 2006.

Two sets of data from the year 2004 and 2005 were used to train the data using the SOM Toolbox in MATLAB software. The data from the year 2004 were trained to forecast the data for the year 2005, while the data from 2005 were used to train for load forecasting year 2006.

Normalization is needed to train the data. Data normalization formed the maps to dominate map topology. There are 4 types of normalizations, which are the var, range, log, and logistic. In this case, 'var' data input normalized the variance to unity zero and the means to zero, 'range' input data scaled the variable values between zero and one, 'log' is a logarithmic transformation and 'logistic' scaled all possible values between zero and one.

The load was grouped before forecasting. Then, it was compared to the actual load. The accuracy of the results were calculated using MAPE equation:

$$Error\ (\%) = \frac{|Actual - Forecasted|}{Actual} \times 100 \qquad (1)$$

$$MAPE\ (\%) = \frac{1}{M} \times \sum_{i=1}^{M} Error_{[i]}\ (\%) \qquad (2)$$

where *Actual* is the real value of load demand, *Forecasted* is the forecasted value in the same year and M is the total forecasted data.

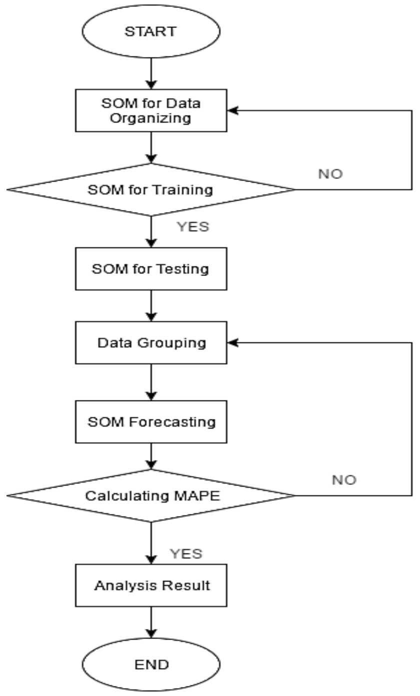

Figure 2: The methodology of the MTLF using SOM.

III. Result and Discussion

A. SOM training year 2004

Table 1 shows the comparison of the best number of neuron from every four methods of normalization. By comparing the topographic error, quantization error, and training time, the best and most suitable method of normalization is the *range*. The *range* normalization has the smallest topographic error with 0.000 while the quantization error is 0.052. The training time for the model to mapping the U-matrix is 29 seconds.

Table 1
The comparison between 4 types of normalization for the year 2004

Method	N.N	Classification Result			T.T
		M.S	Q.E	T.E	
range	1780	[56, 32]	0.052	0.000	29
var	1760	[52, 34]	0.171	0.000	26
log	1800	[47, 38]	0.103	0.005	27
logistic	1760	[53, 33]	0.035	0.005	26

Legend
N.N – Number of neurons M.S – Map size
Q.E – Quantization error T.E – Topographic Error
T.T – Training Time (s)

B. SOM Testing and Forecast year 2005

From Table 1, it can be observed that the best and the most suitable normalization method is using the range with the neuron number 1780. Using the range with the neuron 1780, U-matrix was mapped for testing and forecasting.

Figure 3 shows the U-matrix that had been mapped for testing and forecasting. It can be observed that the testing data are closed to the training data. Thus, the training and testing data can be classified into a similar group since it can be considered that the nearest data will have less MAPE value. Based on the SOM testing for the year 2005, the U-matrix visualized the distances between neighboring map units, which were between the testing cell and the nearest winning cells.

C. SOM training year 2005

Table 2 shows the comparison of the best number of the neuron from every four methods of normalization. By comparing the topographic error, quantization error, and training time, the best and the most suitable method of normalization is logistic. The logistic normalization has the smallest topographic error with 0.000, while the quantization error is 0.032. The training time for the model to mapping the U-matrix is 28 seconds.

Table 2
The comparison between 4 types of normalization for the year 2005

Method	N.N	Classification Result			T.T
		M.S	Q.E	T.E	
range	1800	[55, 33]	0.049	0.000	29
var	1720	[52, 33]	0.168	0.003	32
log	1720	[46, 37]	0.107	0.005	24
logistic	1800	[55, 33]	0.032	0.000	28

Legend
N.N – Number of neurons M.S – Map size
Q.E – Quantization error T.E – Topographic Error
T.T – Training Time (s)

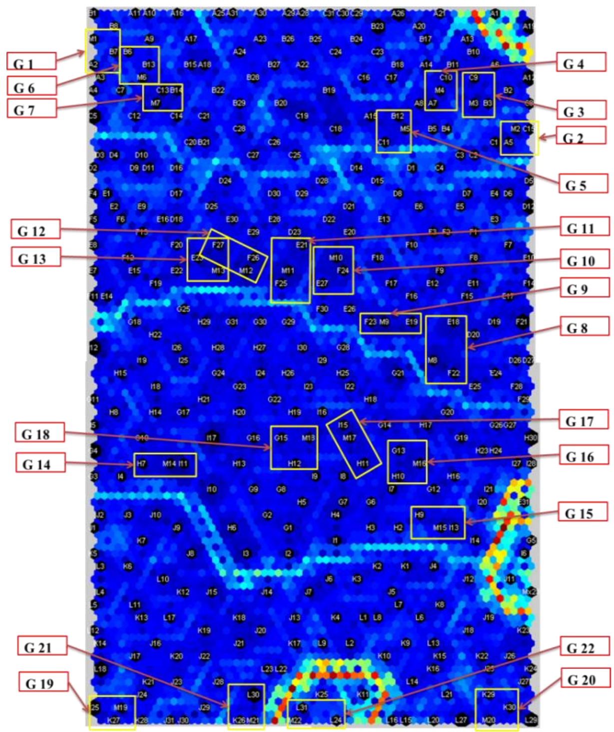
Figure 3: SOM testing for the year 2005.

D. SOM Testing and Forecast year 2006

From Table 2, it can be observed that the best and the most suitable normalization method is using *logistic* with the neuron number 1800. Using the *logistic* with the neuron number 1800, U-matrix was mapped for testing and forecasting.

Figure 4 shows the U-matrix that had been mapped for testing and forecasting. It can be observed that the testing data are closed to the training data. Thus, the training and testing data can be classified to a similar group since it can be considered that the nearest data will have less MAPE value. Based on the SOM testing for the year 2006, the U-matrix visualized the distances between neighboring map units, which were between the testing cell and the nearest winning cells.

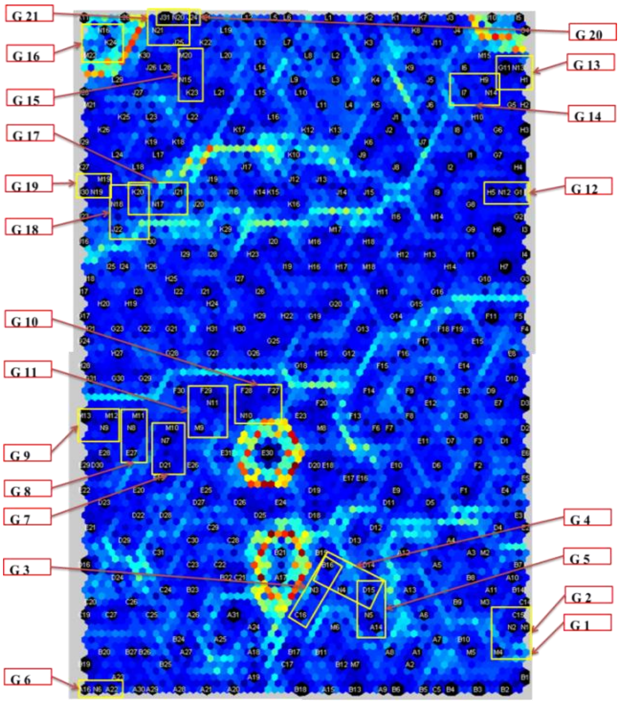

Figure 4: SOM testing for the year 2006.

E. The Error and MAPE

To evaluate forecasting accuracy of the whole procedure, the error was calculated for each cell by comparing the actual data with the forecasted data. The calculated MAPE values were tabulated, as shown in Table 3.

Table 3
The error and MAPE value

Cell	Date	Error (%)
M1	6/3/2005	3.84
M2	7/3/2005	1.10
M3	8/3/2005	8.85
M4	9/3/2005	3.35
M5	10/3/2005	3.11
M6	11/3/2005	0.58
M7	12/3/2005	3.68
M8	21/6/2005	4.35
M9	22/6/2005	2.79
M10	23/6/2005	2.55
M11	24/6/2005	7.40
M12	25/6/2005	4.76
M13	26/6/2005	8.17
M14	10/9/2005	2.43
M15	12/9/2005	0.48
M16	13/9/2005	2.10
M17	14/9/2005	11.86
M18	15/9/2005	5.41
M19	25/12/2005	2.78
M20	27/12/2005	3.02
M21	30/12/2005	3.11
M22	31/12/2005	2.62
N1	13/2/2006	14.42
N2	14/2/2006	7.82
N3	15/2/2006	1.91
N4	16/2/2006	2.43
N5	17/2/2006	2.89
N6	19/2/2006	11.31
N7	25/5/2006	13.98
N8	26/5/2006	7.06
N9	27/5/2006	1.89
N10	30/5/2006	0.93
N11	31/5/2006	0.62
N12	4/8/2006	0.36
N13	7/8/2006	1.72
N14	8/8/2006	6.57
N15	22/11/2006	6.35
N16	23/11/2006	0.05
N17	24/11/2006	0.19
N18	25/11/2006	3.01
N19	26/11/2006	2.68
N20	27/11/2006	0.87
N21	28/11/2006	7.09
MAPE (%)	**4.24**	

From Table 3, it can be observed that certain data have larger errors, which are M3, M11, M13, M17, M18, N1, N2, N6, N7, N8, N15, N15, and N21. While the M6, M15, N10, N11, N12, N16 and N17 have errors that are less than one percent. The MAPE value that was calculated based on the table above is 4.24%.

IV. CONCLUSION

As a conclusion, the Self-Organizing Maps came out as a stimulating method for medium-term load forecasting. The experimental results from this research showed that the SOM is capable to forecast the medium-term load data since the calculated MAPE is 4.24%. The load data have been trained, tested and forecasted by running the simulation using the selected features. The missing data is able to predict and analyze based on the U-matrix that had been mapped after SOM testing.

This method is considered as a good alternative method as the data is forecasted using the unsupervised learning SOM. By using the SOM, the data that had been mapped was clearly interpreted and understood as it can be visualized in the U-matrix form. The SOM converts the high dimension input to the low-dimension and discrete map, which makes it easy to observe the similarities between the input data.

For recommendation, besides the holiday influence and average temperature; the maximum and minimum temperature, rainfall, humidity and wind speed should be considered in the forecasting to achieve more accurate results. The forecasting methods also can be improved by using hybrid methods such as the combination of the SOM with SVR or the SOM with Fuzzy Algorithm.

REFERENCES

[1] I. A. Samuel, F. C. F, A. A. A, and A. A. Awelewa, “Medium-Term Load Forecasting Of Covenant University Using The Regression Analysis Methods,” vol. 4, no. 4, pp. 10–17, 2014.

[2] N. Amjady and F. Keynia, “Mid-term load forecasting of power systems by a new prediction method,” vol. 49, pp. 2678–2687, 2008.

[3] O. A. S. Carpinteiro and A. P. Alves da Silva, “A hierarchical \mboxself-organizing map model in short-term load forecasting,” *Eng. Appl. Neural Networks. Proc. 5th Int. Conf. Eng. Appl. Neural Networks*, pp. 75–80, 1999.

[4] E. A. Feinberg and D. Genethliou, “Chapter 12 LOAD FORECASTING,” in *Applied Mathematics for Restructured Electric Power Systems: Optimization, Control, and Computational Intelligence*, Springer, Boston, MA, 2005, pp. 269–285.

[5] E. Gonzalez-Romera, M. A. Jaramillo-Moran, and D. Carmona, *Monthly Electric Energy Demand Forecasting Based on Trend Extraction*, vol. 21. 2006.

[6] N. Abu-shikhah, F. Elkarmi, and O. M. Aloquili, “Medium-Term Electric Load Forecasting Using Multivariable Linear and Non-Linear Regression,” *Smart Grid Renew. Energy*, vol. 2, no. May, pp. 126–135, 2011.

[7] M. Martín-Merino and J. Román, “Electricity Load Forecasting Using Self Organizing Maps BT - Artificial Neural Networks – ICANN 2006,” 2006, pp. 709–716.

[8] G. P. Papaioannou, C. Dikaiakos, A. Dramountanis, and P. G. Papaioannou, “Analysis and Modeling for Short- to Medium-Term Load Forecasting Using a Hybrid Manifold Learning Classical Statistical Models (SARIMAX , Exponential,” 2016.

[9] A. D. Papalexopoulos, S. Hao, and T.-M. Peng, “An implementation

of a neural network based load forecasting model for the EMS," *IEEE Trans. Power Syst.*, vol. 9, no. 4, pp. 1956–1962, 1994.
[10] A. Mohan, "Mid Term Electrical Load Forecasting For State of Himachal Pradesh Using Different Weather Conditions via ANN Model," vol. 1, no. 2, pp. 60–63, 2013.
[11] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks : The state of the art," vol. 14, pp. 35–62, 1998.
[12] M. López, S. Valero, C. Senabre, J. Aparicio, and A. Gabaldon, "Application of SOM neural networks to short-term load forecasting: The Spanish electricity market case study," *Electr. Power Syst. Res.*, vol. 91, pp. 18–27, 2012.
[13] T. Kohonen, *Self-Organizing Maps*, 2nd ed. Springer-Verlag Berlin Heidelberg, 2001.
[14] V. Chaudhary, R. S. Bhatia, and A. K. Ahlawat, "A novel Self-Organizing Map (SOM) learning algorithm with nearest and farthest neurons," *Alexandria Eng. J.*, vol. 53, no. 4, pp. 827–831, 2014.
[15] S. M. Guthikonda, "Kohonen Self-Organizing Maps," no. December, 2005.
[16] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "SOM Toolbox for Matlab 5," *Tech. Rep. A57*, vol. 2, no. 0, p. 59, 2000.
[17] A. A. Akinduko, E. M. Mirkes, and A. N. Gorban, "SOM: Stochastic initialization versus principal components," *Inf. Sci. (Ny).*, vol. 364–365, pp. 213–221, 2016.
[18] Z. H. Bohari, H. S. Azemy, M. N. Mohd Nasir, F. Baharom, M. F. Sulaima, and M. Jali, *Reliable short term load forecasting using self organizing map (SOM) in deregulated electricity market*, vol. 79. 2015.
[19] A.Ultsch and H. P. Siemon, "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis," in *INNC'90*, 1990, pp. 305–308.
[20] S.-L. Shieh and I.-E. Liao, "A new approach for data clustering and visualization using self-organizing maps," *Expert Syst. Appl.*, vol. 39, no. 15, pp. 11924–11933, 2012.
[21] T. Hong, P. Pinson, and S. Fan, "Global energy forecasting competition 2012," *Int. J. Forecast.*, vol. 30, no. 2, pp. 357–363, 2014.