ELECTRICITY FORECASTING FOR SMALL SCALE POWER SYSTEM USING ARTIFICIAL NEURAL NETWORK

By
SALWA BINTI SOLAHUDDIN

FINAL REPORT

Submitted to the Electrical and Electronics Engineering Programme
In Partial Fulfillments of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

MAY 2011

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

ELECTRICITY FORECASTING FOR SMALL SCALE POWER SYSTEM USING ARTIFICIAL NEURAL NETWORK

by

Salwa Binti Solahuddin

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Dr. Zuhairi Hj. Baharudin
Project Supervisor

DR. ZUHAIRI BAHARUDIN
Senior Lecturer
Electrical & ... Engineering Department
Universiti ... PETRONAS
31750 Tro... ...k Darul Ridzuan
Tel: 05-368 ... 7443

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

May 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Salwa Binti Solahuddin

ABSTRACT

This project presents a practical short term load forecasting (STLF) for small scale power system using artificial neural network (ANN) method. The project applies a generic three-layered feedforward network. The network is trained in a supervised manner and used backpropagation as a learning technique. In addition, a configuration consisting of a hidden layer that uses a hyperbolic tangent sigmoid transfer function and the output layer with a pure linear transfer function is applied. Gas District Cooling (GDC) is chosen as a case study for small scale power system since this plant was designed to produce electrical power supply and chilled water for Universiti Teknologi PETRONAS (UTP) campus and in-plant use. As a sole customer of GDC power plant, the load data from 2006 till 2010 are gathered and utilized for model developments. There are two models developed based on UTP normal operating semester (Semester On) and during break (Semester Off). The developed models can forecast electricity load for the one week ahead. The computation experimental of the proposed network applies MATLAB software and its toolbox. The mean absolute percentage error (MAPE) is used as the measurement for the forecasting performance. At the end of this project, the proposed method using ANN manages to get average MAPE of 6.72 % for Model 1 (Semester Off) and 3.92 % for Model 2 (Semester On) which is considered relatively good result.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious, the Most Merciful.

I would like to express the deepest appreciation to my supervisor, Dr. Zuhairi Hj Baharudin, Lecturer of Electrical & Electronic Department, UTP. The supervision and support he gave truly help me throughout completing this project. He has guided and corrected various documents of mine with attention and care.

I am heartily thankful to my co-supervisor, Ir Mohd Fatimie Irzaq Khamis, Senior Electrical Executive of Maintenance Department, UTP, whose encouragement, guidance and support from the initial to the final level enable me to develop an understanding of this project.

Special thanks to Gas District Cooling (GDC) power plant for their generosity in providing the historical UTP daily load data. This project would not have been possible without the invaluable data provided.

I sincerely thank Maryam Jamela Ismail and Muhammad Hanif Amaran for all their valuable assistance in this project.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Electricity load forecasting is very essential in planning for electricity resource management. It can predict the future demand and manage the stored electricity for future usage. The electricity generation cost also can be minimized as well as the electricity tariff can be controlled [1].

Load forecasting can be categorized into three categories which are short term, medium term and long term. The duration for prediction of short term load forecasting (STLF) usually ranging from one hour to one week. Medium term load forecasting means the load forecasting usually from a week to a year. It deals with the scheduling of fuel supplies and maintenance operations. Long term load forecasting, it means the load forecasting is usually more than a year. This type of load forecasting is useful for planning operations [2-4].

As such, this project deliberates method to forecast STLF electricity demand for small scale power system. Many methods have been used for forecasting in the past. These include statistical methods such as fuzzy logic [5-7],[15], regression [8-13], expert systems [11],[14], support vector machines, econometric models and end-use model [19],[25].

In this project, artificial neural network (ANN) is used as a method to model complex relationship between the input and output patterns [1]. Basically, the networks learn through the examples, which consist of the input signals and the desired output. Neural networks capable to model with a greater range of relationships unlike the ordinary least squares linear regressions [16].

Gas District Cooling (GDC) is chosen as a case study for small scale power system since this plant was designed to produce electrical power supply and chilled water for Universiti Teknologi PETRONAS (UTP) campus and in-plant use. The plant is equipped with two units of 4.2 MW Solar Taurus 60S gas turbines, two heat recovery steam generators (HRSG), one package boiler (AGB), two steam absorption chillers (SAC) and four electric chillers (EC). Under normal condition, the generators are operating in island mode but during contingency period, generators are connected in parallel to Tenaga Nasional Berhad [17].

This project gives a significance contribution to the GDC in their daily operation. One of them is to study the electricity behavior of GDC UTP load demand. It is important to analyze the electricity behaviors in order to understand and be familiar with the pattern of the load. Once the pattern is identified, the predictive model can be formulated and then developed using ANN method. Therefore, this project proposes a method of a multilayer perceptron neural network and it is trained and simulated by using MATLAB.

1.2 Problem Statement

The GDC UTP faced the complexity electricity demand values due to many factors such as temperature and weather. Both factors affect during generation which made the electricity demand values to be complex and nonlinear and therefore, hard to analyze.

As of the fact that the electricity demand values make the electricity load forecasting become challenging, the project is being made using ANN models to help the GDC UTP to forecast the electricity demand accurately.

1.3 Objectives

The objectives of this project are:

i. To understand the principle of ANN method
ii. To analyze the data gathered from GDC UTP.
iii. To design the load forecasting models using ANN.
iv. To forecast the GDC UTP load demand.
v. To present the load forecasting models in a user-friendly environment.

1.4 Scope of Work

The scope of work of this project is to understand the principle of ANN by carry out the literature review as well as brief research about the topic. In addition, the data gathered from GDC UTP is analyzed and then used to design the load forecasting models. As a result, the GDC UTP load demand forecasting can be applied in the short term time scale. Also, the load forecasting models built are ensured to be presented in a user friendly environment.

1.5 Significance of Study

This project gives a significant contribution to the GDC power plant in their daily operation. One of them is to study the electricity behavior of GDC UTP load demand. It is important to analyze the electricity behaviors in order to understand and be familiar with the pattern of the load. Once the pattern is identified, the predictive model can be formulated and then developed using ANN method.

The predictive model developed is used to forecast the electricity demand in advance. This helps exact amount of electricity generated at the exact time. The earlier prediction leads to the optimization in power generation and eliminate the power wastage during the generation.

CHAPTER 2

LITERATURE REVIEW

2.1 Artificial Neural Network

Many methods have been used for load forecasting in the past. These include statistical methods such as fuzzy logic [5-7],[15], regression [8-13], expert systems [11],[14], support vector machines, econometric models and end-use model [19],[25]. The descriptions of some of the methods are tabulate below:

Table 1 Load Forecasting Methods

Method	Description
Regression Analysis	• Focus on the relationship between a dependent variable and one or more independent variables. • Most commonly, the analysis estimates the conditional expectation of the dependent variable given the independent variables [28-29].
Fuzzy Logic	• A form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than accurate [30].
Expert System	• Provide an answer to problem, or clarify uncertainties where normally one or more human experts would need to be consulted [31].
Support Vector Machines	• A set of related supervised learning

	method that analyzes data and recognize patterns. • It predicts for each given input, which of two possible classes the input is a member of [32].
Artificial Neural Network	• A mathematical model inspired by the structure and functional aspects of biological neural networks. • Used to model complex relationships between inputs and outputs to find patterns in data [23].

Table 1 shows the description of some of the methods used for load forecasting. In this project, ANN is used instead of other method since it is capable of modeling complex relationship between the input and output patterns [1]. ANN derives meaning from complicated or imprecise data and therefore used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques [35].

Also, a neural network can be trained of as an expert in the class of information given to analyze. This expert is then used to provide projections when a new situation of interest is given [35]. Other advantages of ANN include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience
- Self-organization: ANN can create its own organization or representation of the information it receives during learning time
- Real time operation: ANN computations may be carried out in parallel, and special hardware devices are designed and manufactured for this capability
- Fault tolerance via redundant information coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage

ANN refers to a class of models inspired by the biological nervous system. The models are composed of many computing elements, usually denoted neurons; each neuron has a number of input and one output [19-26]. Each of these neurons forms a weighted sum of its input, to which a constant term called bias is added. This sum is then passed through a transfer function [20] as illustrates as follows:

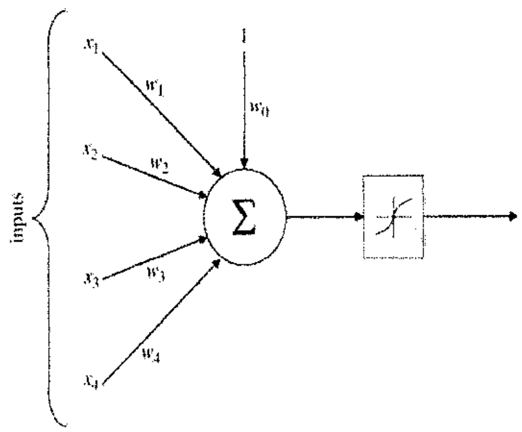


Figure 1 Internal Structure of a Neuron [20]

Figure 1 illustrates the internal structure of a neuron. Commonly, a neural network is train to perform a particular function by adjusting the values of the weights between elements [21] as shown below:

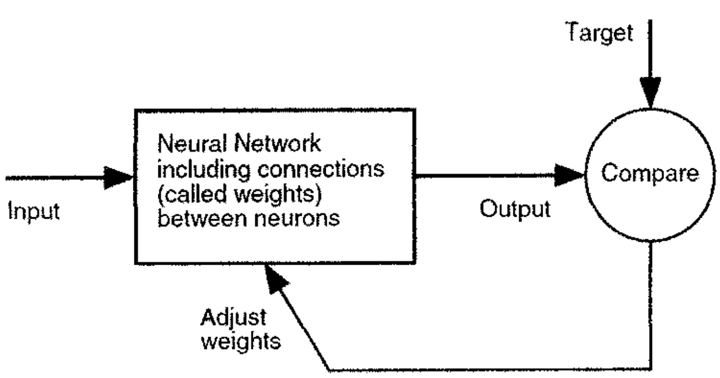


Figure 2 Neural Network Function [21]

Figure 2 shows the neural network function determined largely by the connections between the elements. Generally, neural network are adjusted, or trained, so that a particular input leads to a specific target output. The network is adjusted based on the comparison of the output and the target until the network output matches the target [21].

Batch training of a network proceeds by making weight and bias changes based on the entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. The incremental training is sometimes referred as "adaptive' training [21].

The success in the application of ANN lies in the fact that when these networks are properly trained and configured they are capable of accurately approximating any measurable function. The neurons learn the patterns hidden in data and make generalizations of these patterns even in the presence of noise or missing information. Predictions are performed by the ANN based on the observed data [22].

Among many of the forecasting approaches, ANN has the ability to experiment with theoretically poor, but data rich, models that can identify the complex non-linear relationships in the data and infer future behavior.

2.1.1 Network Architecture

The neurons are organized in a way that defines the network architecture. Networks with interconnections that do not form any loops are called feedforward, while recurrent or network with cycles in which there is one or more loops of interconnections are used [20]. The comparison between single layer feedforward, multilayer feedforward and recurrent networks are tabulate below:

Table 2 Comparison between Network Architecture

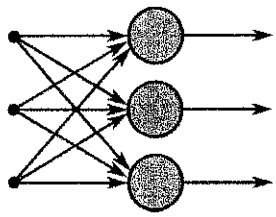
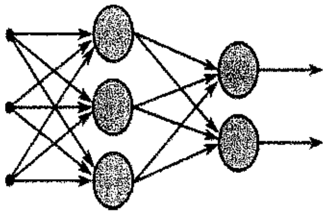
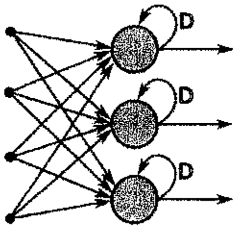
Network Architecture	Descriptions
Single Layer Feedforward	• Consist of input layer and output layer • The inputs are directly connected to the output layer 
Multilayer Feedforward	• Consist of input layer, hidden layer and output layer • Hidden layer enables greater processing power and system flexibility 
Recurrent Networks	• Networks with feedback connections, cycles are present in the network • Difficult to train than feedforward networks 

Table 2 shows the comparison between the network architecture of ANN. In this project, a generic three-layered feedforward network is used. The neurons in each layer may share the same inputs, but not connected to each other. Typically, the neurons in the input layer serve only for transferring the input pattern to the rest of the network, without any processing. The information is processed by the neurons in the hidden and output layers [20].

In order to find the optimal network architecture, several combinations which includes network with different number of hidden layers, different number of neurons in each layer and different types of transfer function are evaluated. Three of the most commonly used transfer functions are describe in table below:

Table 3 Description of Transfer Functions [21]

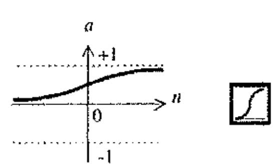
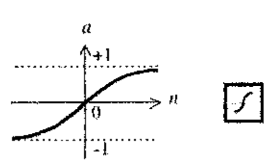
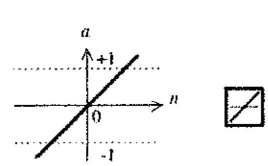
Transfer Functions	Descriptions
Logsig	• The function generates outputs between 0 and 1 as the neuron's net input goes from negative to positive infinity  $a = logsig(n)$
Tansig	• The function generates outputs between -1 and +1 as the neuron's net input goes from negative to positive infinity  $a = tansig(n)$
Purelin	• Linear transfer function  $a = purelin(n)$

Table 3 shows the description of the three most commonly used transfer functions. A configuration consisting of a one hidden layer that uses a hyperbolic tangent sigmoid transfer function and the output layer with a pure linear transfer function is applied throughout this project [20].

This configuration is proven to be universal mapper, provided that the hidden layer has enough neurons. On the other hand, if there are too few neurons, the network may overfit the data [20]. In this project, the number of neurons in the hidden layer is chosen by trial and error.

2.1.2 Network Training

There are three different paradigms used to train neural networks, each corresponding to a particular abstract learning task. There are supervised learning, unsupervised learning and reinforcement learning. However, supervised and unsupervised learning are the most common [33]. The descriptions of the learning paradigms are as follow:

Table 4 Description of Learning Paradigms [33]

Learning Paradigms	**Descriptions**
Supervised	• A technique where the input and expected output of the system are provided and ANN is used to model the relationship between the two. • The network output is compared to the desired output; the error signal is used to update the network weight vectors. • Useful for network headed for reproduce the characteristics of a certain relationship.
Unsupervised	• The data and a cost function provided is a function of the system input and output. • ANN is trained to minimize the cost function by finding the suitable input-output relationship.

	• Useful in situations where a cost function is known.

Table 4 shows the description of the most common learning paradigms used to train a neural network. A feedforward networks is normally trained in a supervised manner. In the learning process, a neural network constructs as input-output mapping, adjusting the weights and biases at each iteration based on the minimization of some error measure between the output produced and the desired output [20].

2.1.3 Network Learning

Multilayer networks used variety of learning techniques and the most common used is backpropagation algorithm. The descriptions of the techniques used are as follows:

Table 5 Description of Learning Algorithm [33]

Learning Algorithm	Descriptions
Gradient Descent	• Used to minimize the error function through the manipulation of a weight vector. • Works by taking the gradient of the weight space to find the path of the steepest descent.
Backpropagation	• Passes error signal backwards through the network during training to update the weights of the network.
Hebbian Learning	• A synapse between two neurons is strengthened when the neurons on either side of the synapse have highly correlated outputs.
Competitive Learning	• A rule based on the idea that only one neuron from a given layer will fire at a time. • Weights adjusted such that only one neuron in a layer.

Table 5 shows the descriptions of the learning algorithm of the neural network. In this project, the backpropagation algorithm is used since it is the most popular and robust tools in the training of ANN [33].

Backpropagation passes error signals backwards through the network during training to update the weights of the network. Also, the backpropagation algorithm specifies the tap weights of the network are updated iteratively during training to approach the minimum of the error function [33].

2.2 Data Collected From GDC

Upon developing the model, the data had been gathered from GDC UTP. The data of the electricity demand by UTP had been gathered starting 1st January 2006 till 31st December 2010.

The data gathered are based on daily interval data. The data have been categorized into two conditions based on UTP's Academic Calendar, as attached in Appendix A. The graphs below represent the daily data gathered from the two conditions; Semester On and Semester Off.

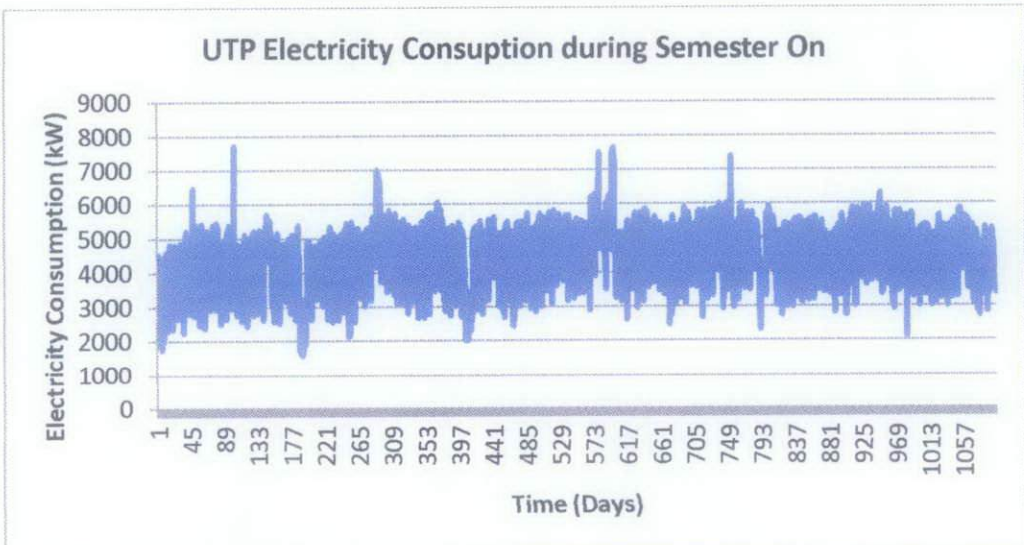


Figure 3 Graph of UTP Electricity Demand during Semester On

Figure 3 shows the UTP electricity consumption during Semester On. The average of the electricity demand is 4.6MW. Also, at certain time, the electricity demand is higher due to the special occasions organized in UTP [24].

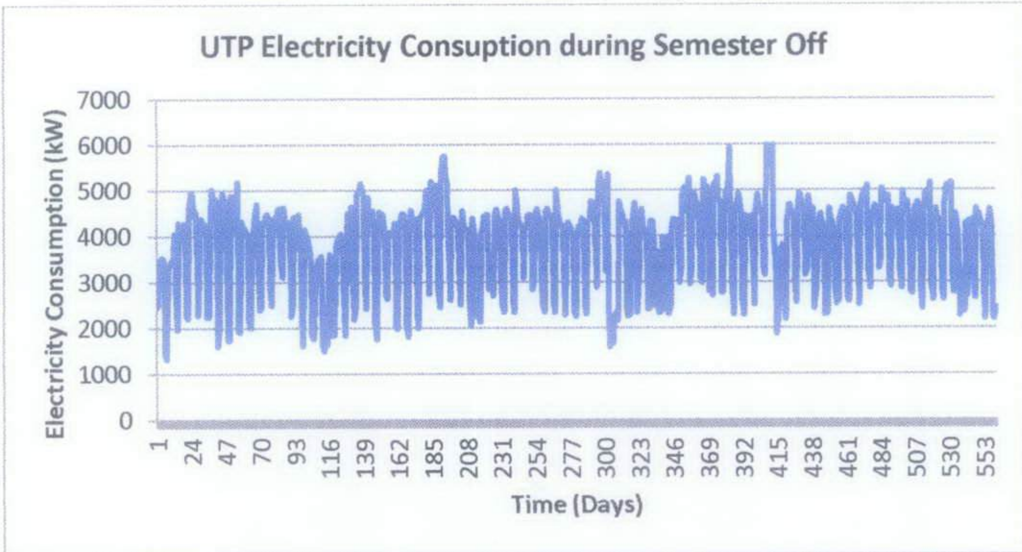


Figure 4 Graph of UTP Electricity Demand during Semester Off

Figure 4 shows the UTP electricity consumption during Semester Off. The average of the electricity demand is 3.8MW. Similarly, at certain time, the electricity demand is higher due to the special occasions organized in UTP [24].

2.3 Data Analysis

The daily data gathered from the GDC UTP are then analyzed and the results for both Semester On and Semester Off are shown below:

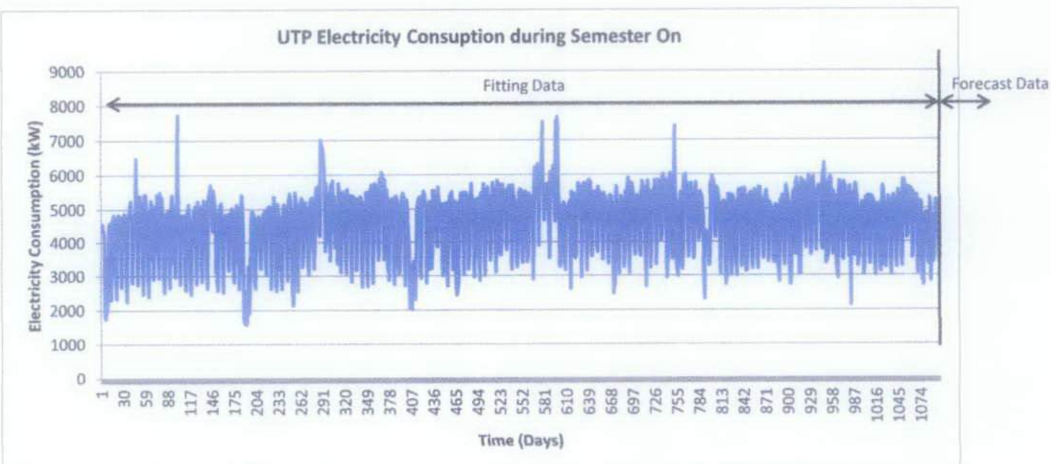


Figure 5 Graph of Fitting Data and Forecast Data during Semester On

Figure 5 shows the UTP electricity demand during Semester On. The graph is labeled with the fitting data as well as the forecast data. The fitting data is the data used for the purpose of training, validation and testing.

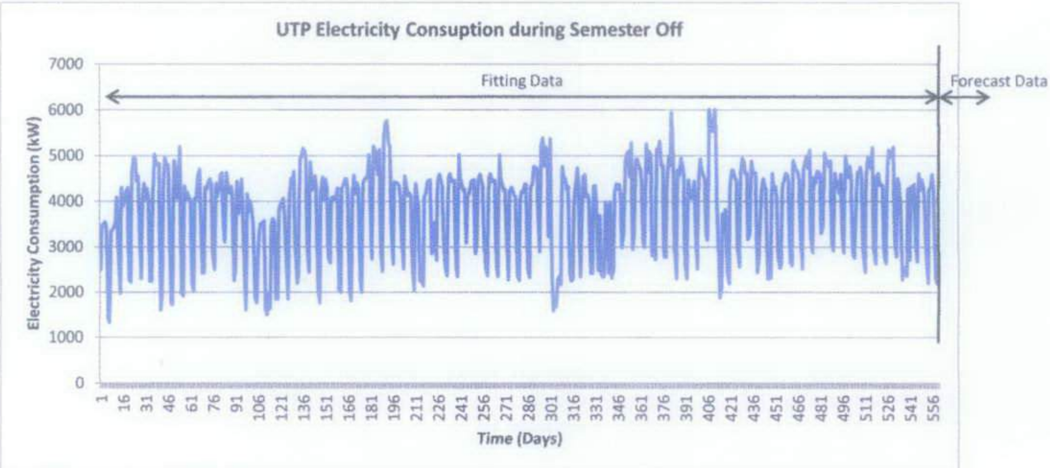


Figure 6 Graph of Fitting Data and Forecast Data during Semester Off

Figure 6 shows the UTP electricity demand during Semester Off. The graph is labeled with the fitting data as well as the forecast data. Similarly, the fitting data is the data used for the purpose of training, validation and testing.

CHAPTER 3

METHODOLOGY

3.1 Research Methodology

In order to achieve the objectives of this project, more specifically to understand the principle of ANN, literature reviews as well as brief research about the topic are carried out on several resources such as books, journal and internet.

Later, the UTP daily load data is gathered from GDC UTP and therefore analyzed for the purpose of training, validation and testing. The forecast models are then developed based on Semester On and Semester Off of UTP Academic Calendar.

There are two models developed based on the conditions in UTP. Model 1 is developed to forecast for one week ahead for Semester Off while Model 2 is developed to forecast for one week ahead for Semester On.

All the models have been simulated for twenty simulations and the average value is calculated to obtained accurate forecasting result. From the result obtained the error between the forecasted and actual load are calculated. The forecasting performance is generally evaluated using mean absolute percentage error (MAPE) and the best forecast model is chosen.

3.2 Procedure Identification

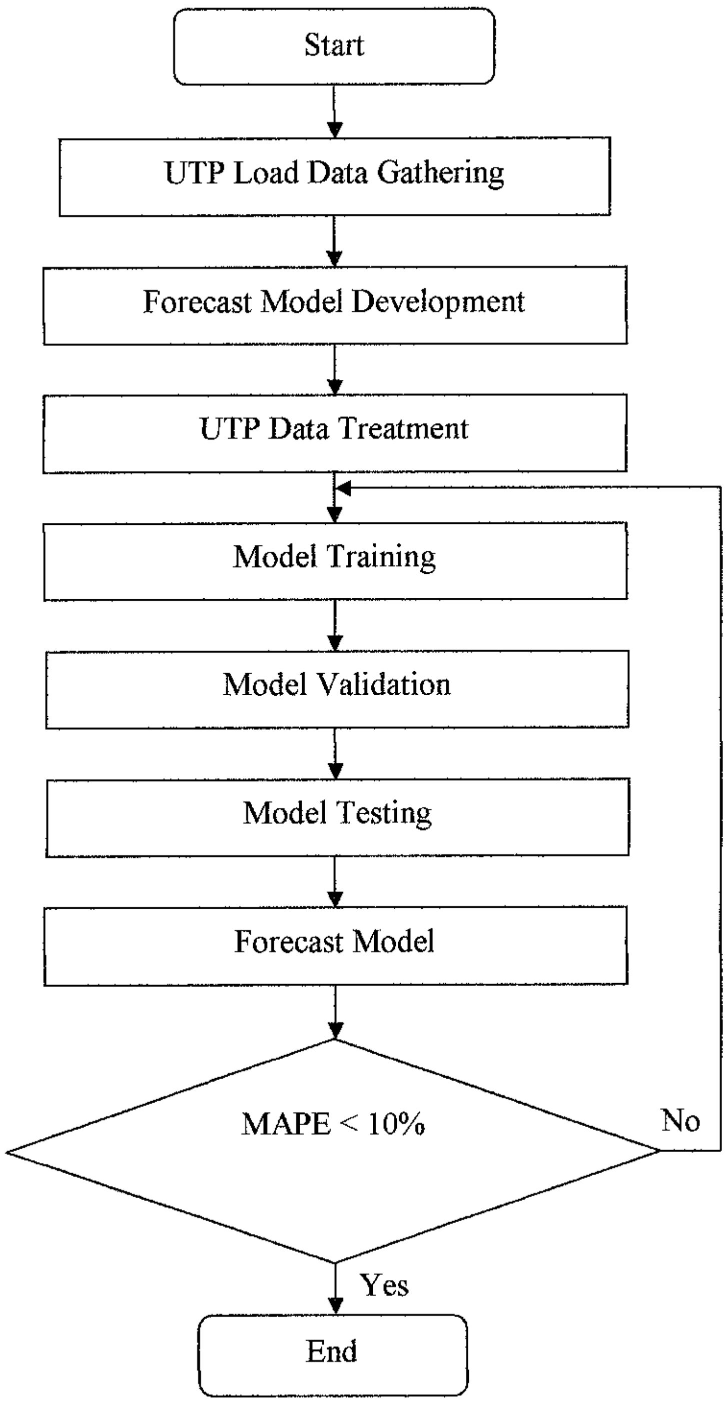


Figure 7 Project's Methodology

3.2.1 UTP load data gathering

The historical UTP daily load data have been gathered from GDC UTP ranging from 1st January 2006 until 31st December 2010. The data have been categorized into Semester On and Semester Off based on UTP Academic Calendar [24].

3.2.2 Forecast model development

The forecast models have been developed using the data gathered. There are two models that have been developed. The models are based on Semester On and Semester Off of UTP and the duration of the load forecasting. The summary of the two models are as follows:

Table 6 The Forecast Models

Model	Semester Type	Forecasting Duration
Model 1	Off	7 days
Model 2	On	7 days

Table 6 shows the two models developed. In addition, the model used a supervised technique where the input and expected output of the network are provided. The summary is shown below:

Table 7 Model's Input and Output

Model	Input (Days)	Output (Days)
Model 1	14	7
Model 2	14	7

Table 7 shows the input of the models are the load data for the last 14 days, and the output is the load data for the following 7 days. Moreover, as the data for each model is different, each of the models is built with different number of hidden layer and hidden neuron. The summary is as below:

Table 8 Model's Hidden Layer and Hidden Neuron

Model	Hidden Layer	Hidden Neuron
Model 1	1	3
Model 2	2	9, 11

Table 8 shows Model 1 has one hidden layer with three hidden neurons, while Model 2 has two hidden layer with nine and eleven hidden neurons respectively. The MATLAB coding used for the two models is included in Appendix B and C.

3.2.3 UTP data treatment

The data treatment has been done in order to create robust models. The data treatment consists of data normalization, data partitioning and data arrangement.

a) Data normalization

The input of an ANN is recommended to be normalized before training. The normalized value is defined as follows:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where X = inputs to the ANN

X_{min} = minimum values of X

X_{max} = maximum values of X

This normalization converts the original input values to a normalized value that is in the range of 0 to 1. This assists the backpropagation method better determine the weights of the ANN [34].

b) Data partitioning

The gathered data has been partitioned into three (3) partitions for the purpose of training, validation and testing as follows:

i) Training data – 40%

ii) Validation data – 30%

iii) Testing data – 30%

The partitioning has been done based on the total data of 560 days for Semester Off and 1099 days for Semester On. The partitioning of the data is based on the non-randomization data that need to done to the historical data in order to obtain accurate result [24].

c) Data arrangement

The data are also arranged in order that the first day of the week starts on Monday and end on Sunday. This provides neural network an organize information and therefore helps the network predict correctly.

3.2.4 Model training

The training of the new models involves 40% of the gathered data. The numbers of data for each of the model are shown below:

Table 9 Number of Training Data

Model	Training Data
Model 1	221
Model 2	437

3.2.5 Model validation

The purpose of validation is to guide the training of the model. The training should stop as the validation reaches the performance goal. The validation data for the models consists of 30% of the entire data. The numbers of data for validation are tabulated below:

Table 10 Number of Validation Data

Model	Validation Data
Model 1	166
Model 2	328

3.2.6 Model testing

The testing data for the new models consists of 30% of the total data. The testing is done to observe the efficiency of the developed models and the numbers of testing data are as follows:

Table 11 Number of Testing Data

Model	Testing Data
Model 1	166
Model 2	327

3.2.7 Forecast model

The forecast model is developed to forecast for seven days ahead. Table below shows the actual data for both models

Table 12 Actual Data Range

Model	**Actual Data**
Model 1	13^{th} – 19^{th} December 2010
Model 2	4^{th} – 10^{th} October 2010

3.2.8 MAPE

MAPE or mean absolute percentage error is calculated once the forecasting load obtained. The program will compare the actual load and forecast load and therefore calculate error based on the following formula:

$$\text{Relative Error} = \frac{Forecast\ Load - Actual\ Load}{Actual\ Load} \times 100\%$$

$$\text{Absolute Error} = \frac{|Forecast\ Load - Actual\ Load|}{Actual\ Load} \times 100\%$$

The expected MAPE value in this experiment is set to be less than 10% since MAPE value of 10% is considered very good while MAPE value in range from 20% to 30% or even higher is quite common [36].

3.3 User-friendly Features

A user-friendly feature is added to allow users to run MATLAB program of both Model 1 and Model 2 as one. This help users save their time from running the MATLAB program separately. The features processes are as illustrate as follows:

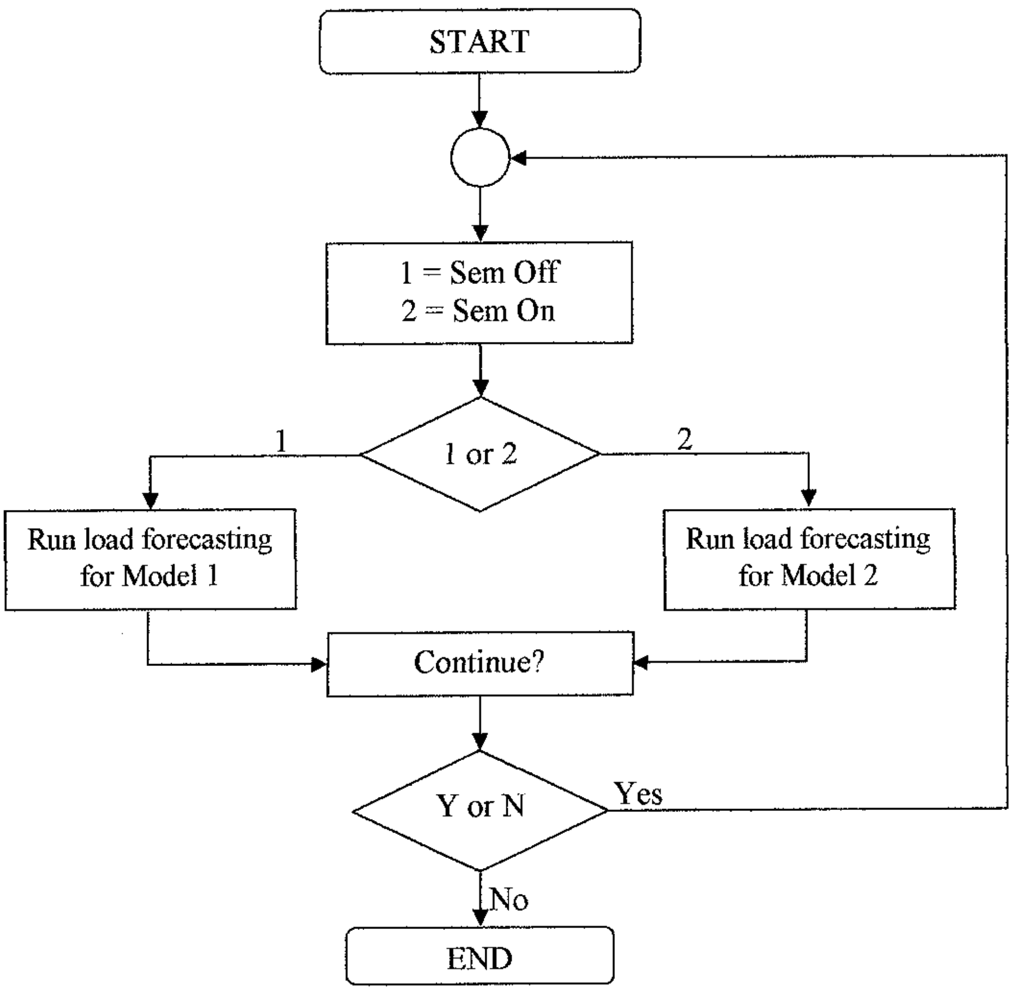


Figure 8 User-friendly Features Flow Chart

The flow chart in Figure 8 shows the user-friendly features processes. The build in instruction will easily let users to select which model to run. Once the model is selected, the program will run the forecasting model and the load demand for the next seven days is then predicted. The MATLAB coding of these user-friendly features is attached in Appendix D.

3.4 Project Duration

In order to effectively monitor the progress of this project, a Gantt chart consist of one year duration planning had been construct. See Appendix E.

3.5 Tool Required

The *MATLAB R2007b* software is used as the main tool for the new load forecasting model developments. MATLAB is an ideal tool for working with ANN since it is highly efficient in performing vector and matrix calculations. Also, MATLAB comes with a specialized Neural Network Toolbox® which contains a number of useful tools for working with ANN [33].

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results

The forecast models are simulated to determine the result of one week forecast load. The comparison of the forecast load and the actual load demand are shown in Figure 9 and Table 13:

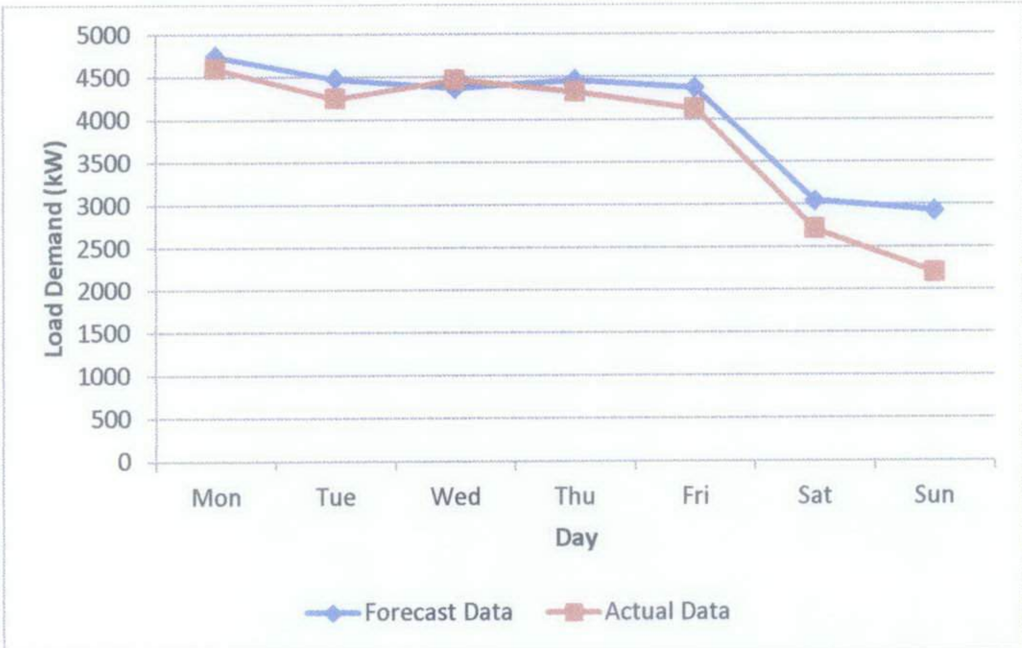


Figure 9 Forecast and Actual Load Demand for Model 1 (13th – 19th December 2010)

Table 13 Comparison between Forecast and Actual Load Demand for Model 1 (13^{th} – 19^{th} December 2010)

Day	Load Demand (kW)		MAPE (%)
	Forecast	Actual	
Monday	4751.1	4604	7.1546
Tuesday	4481.7	4252	
Wednesday	4369.9	4472	
Thursday	4466.7	4328	
Friday	4372.8	4128	
Saturday	3041.5	2716	
Sunday	2926.8	2204	

For Model 1, the MAPE value calculated for this simulation is 7.1546, while for Model 2, the MAPE value is 3.7042. The comparison for Model 2 is as Figure 10 and Table 14 below:

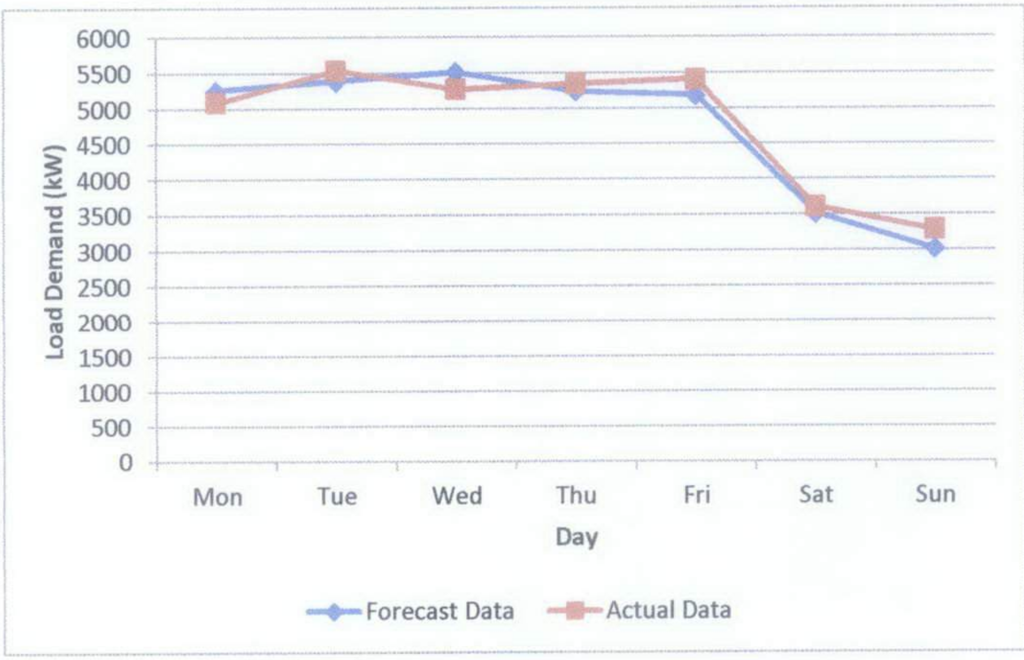


Figure 10 Forecast and Actual Load Demand for Model 2 (4^{th} – 10^{th} October 2010)

Table 14 Comparison between Forecast and Actual Load Demand for Model 2 (4^{th} – 10^{th} October 2010)

Day	Load Demand (kW)		MAPE (%)
	Forecast	Actual	
Monday	5263.8	5092	3.7042
Tuesday	5393.1	5540	
Wednesday	5514.6	5276	
Thursday	5250.6	5352	
Friday	5190.5	5412	
Saturday	3522.1	3616	
Sunday	3018.4	3288	

In order to obtain the best result, the models are run for 20 simulations. Figure 11 and Figure 12 depict the MAPE values of each simulation for both models:

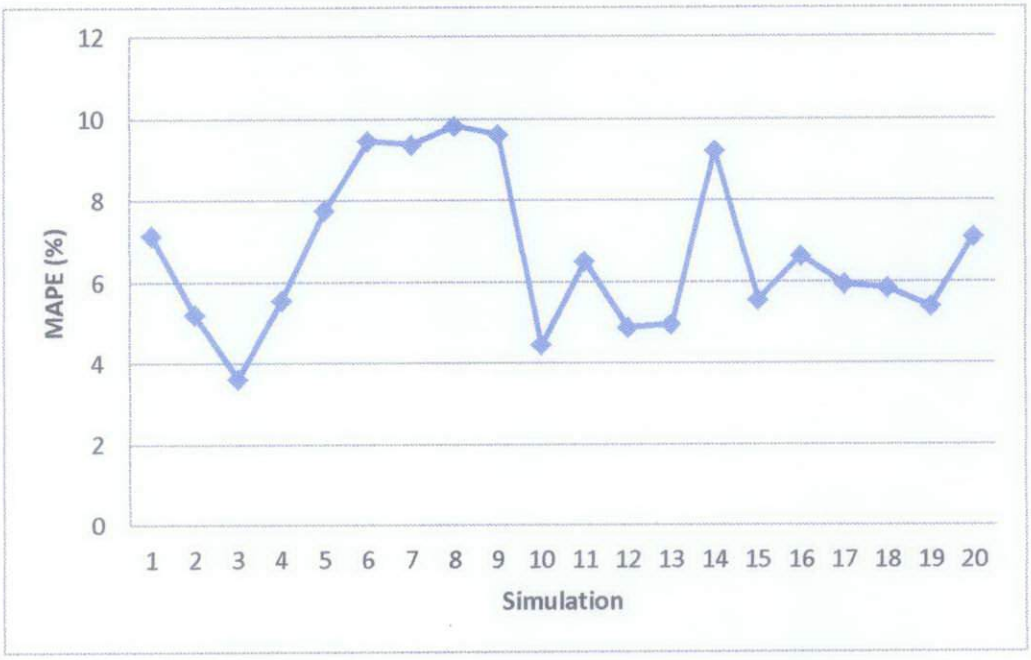


Figure 11 MAPE Values for Model 1

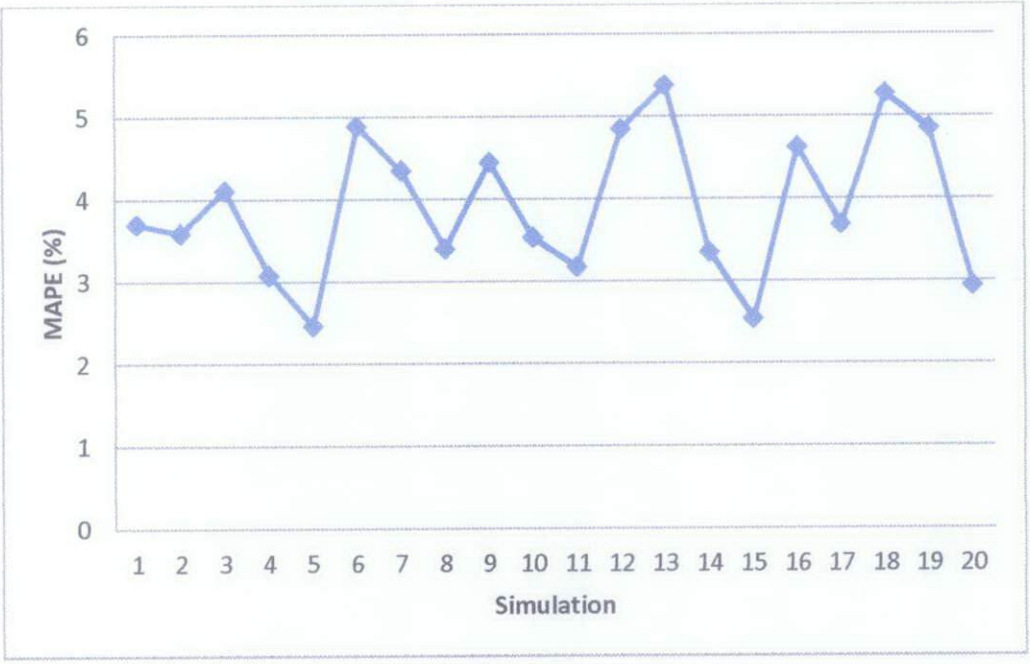


Figure 12 MAPE Values for Model 2

Table 15 MAPE Value of Forecast Models

Simulation	MAPE Value (%)	
	Model 1	Model 2
1	7.1546	3.7042
2	5.2386	3.5971
3	3.6287	4.1158
4	5.5743	3.0855
5	7.7631	2.4734
6	9.4669	4.889
7	9.3826	4.3525
8	9.8419	3.4071
9	9.6389	4.4519
10	4.4725	3.5399
11	6.5133	3.1771
12	4.9058	4.8466
13	4.9813	5.3818

14	9.2332	3.3560
15	5.5859	2.5488
16	6.6534	4.6263
17	5.9580	3.6896
18	5.8613	5.2790
19	5.4148	4.8582
20	7.1163	2.9513
MAPE Average Value (%)	**6.72**	**3.92**

4.2 Discussion

The configuration of Model 1 consist of one hidden layer with three hidden neurons, and Model 2 consisting of two hidden layer with nine and eleven hidden neurons respectively. Both of these models give relatively good result and prove that the configurations are applicable for short term load forecasting. Also, the result shows that both models are efficient and represents a high degree of accuracy in the load forecasting.

However, the result summarized in Table 15 shows Model 2 has better result than Model 1. This is due to the fact that there are many university events conducted during the Semester Off which lead to non steady state load during this period compared to Semester On period. Moreover, a configuration consist of two hidden layer gives the network enough neurons and avoid the data to be over fitted.

CHAPTER 5

CONCLUSIONS & RECOMMENDATIONS

5.1 Conclusions

Accurate electricity forecasting is essential for GDC operation since its can help in pre-planned scheduling and maintenance of the power plant. Once the future demand is known, the exact amount of electricity can be generated at exact time. Therefore, when the demand is less than 4.2MW, one generator can be shut down for maintenance works. This helps exact amount of electricity generated at the exact time.

Also, when only one generator is operated, means the energy is used effectively. Consequently, it leads to the optimization in power generation and eliminate the power wastage during the generation. An efficient energy management will reduced the operating cost as well as the negative impact to the environment. This helps GDC UTP management team to develop a great image in line with Malaysia's strategy to moderate trends in increasing energy intensity and avoid wasteful energy usage.

Besides, electricity forecasting allows earlier notification to the power generation system in order to avoid an unbalanced system. Unbalanced system happens when the electricity demand is more compared to the electricity generated. This situation can leads to fault or malfunction of the system.

In this project, the proposed method using ANN manages to get average MAPE of 6.72 % for Model 1 and 3.92 % for Model 2 which is considered relatively good result. From the experiments, it is found that the forecasting method for the GDC UTP is well-suited with multilayer perceptron feedforward neural network.

The models developed have different configuration which Model 1 with one hidden layer and Model 2 with two hidden layer. During the experiment, both models are configured with different number of hidden layer in order to determine the model's best architecture.

The experiment of Model 1 with two hidden layer gives higher MAPE value. This is because a configuration of two hidden layer make the structure of the model become complex and result in the fitting of the signal in the data as well as the noise, or known as over-training of the ANN model [34].

On the other hand, Model 2 with two hidden layer gives lower MAPE values compared to a configuration consist of only one hidden layer. This shows that Model 2 is well-trained and the architecture provides high degree of accuracy.

5.2 Recommendations

From the result obtained, it is believe there are still rooms for improvement. Future studies on these forecasting models can incorporate information about the effects of the public holiday as well as weather into the neural network as to obtain more representative forecast of the future demand.

Also, the central idea of neural networks is that such parameters can be adjusted so that the network exhibits some desired or interesting behavior. Thus, a way to train the network to do a particular job by adjusting the weight or bias parameters should be considered, instead of the network itself will adjust these parameters to achieve the desired end.

REFERENCES

[1] Lai Mei Yan and Ong Hong Choon, "Neural networks forecasting on electricity consumption in Malaysia", Universiti Sains Malaysia

[2] G. Gross and F. D. Galiana, "Short-term Load Forecasting," Proceedings of the IEEE, vol.75, pp. 1558-1573, 1987.

[3] S. Chentur Pandian, K. Duraisamy, C. Christober Asir Rajan and N. Kanagaraj. "Fuzzy approach for short term load forecasting," Electric Power System Research, vol. 76, issue 6-7, pp 541-548, 2006.

[4] E. A. Feinberg and D. Genethliou, "Load Forecasting," State University of New York,Stony Brook , 2004.

[5] A. M. Al-Kandari, S. A. Soliman and M. E. El-Hawary, "Fuzzy short-term electric load forecasting," International Journal of Electrical Power & Energy Systems, vol. 26, pp. 111-122, 2, 2004

[6] T. Senjyu, S. Higa and K. Uezato, "Future load curve shaping based on similarity using fuzzy logic approach," Generation, Transmission and Distribution,IEE Proceedings-, vol. 145, pp. 375-380, 1998.

[7] H.-Wu and C. Lu, "Automatic fuzzy model identification for short-term load forecast," Generation, Transmission and Distribution, IEE Proceedings-, vol. 146, pp. 477-482, 1999.

[8] H. S. Hippert and C. E. Pedreira, "Estimating temperature profiles for short-term load forecasting: neural networks compared to linear models," Generation, Transmission and Distribution, IEE Proceedings-, vol. 151, pp. 543-547, 2004.

[9] P. J. Brockwell and R. A. Davis, Introduction to Time Series and Forecasting. Springer-Verlag New York, Inc., 2002.

[10] J. Contreras, R. Espinola, F. J. Nogales and A. J. Conejo, "ARIMA models to predict next-day electricity prices," Power Systems, IEEE Transactions on, vol. 18, pp. 1014-1020, 2003.

[11] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," Power Systems, IEEE Transactions on, vol. 16, pp. 498-505, 2001.

[12] A. D. Papalexopoulos and T. C. Hesterberg, "A regression-based approach to short-term system load forecasting," Power Systems, IEEE Transactions on, vol. 5, pp. 1535-1547, 1990.

[13] T. Haida and S. Muto, "Regression based peak load forecasting using a transformation technique," Power Systems, IEEE Transactions on, vol. 9, pp. 1788-1794, 1994

[14] I. Moghram and S. Rahman, "Analysis and evaluation of five short-term load forecasting techniques," Power Systems, IEEE Transactions on, vol. 4, pp. 1484-1491, 1989.

[15] H. C. Wu and C. N. Lu, "A Data Mining Approach for Spatial Modeling in Small Area Load Forecast," Power Engineering Review, IEEE, vol. 22, pp. 60-60, 2002.

[16] James W. Taylor, Lilian M. de Menezes, Patrick E. McSharry, "A Comparison of Univariate Methods for Forecasting Electricity Demand Up to a Day Ahead" , University of Oxford, May 2005.

[17] Mohd Fatimie Irzaq Khamis, Zuhairi Baharudin, Nor Hisham Hamid, Mohd Faris Abdullah and Siti Sarah Md Yunus, "Electricity Forecasting for Small Scale Power System using Fuzzy Logic," The 9th International Power and Energy Conference (IPEC), 2010.

[18] Guoqiang Zhang, B. Eddy Patuwo, Micheal Y. Hu, "Forecasting with artificial neural networks: The state of the art," International Journal of Forecasting 14, vol 14 ,pp 35-62,1998.

[19] RF Engle, C. Mustafa, and Rice, "Modelling Peak Electricity Demand", Journal of Forecasting, 1992

[20] J. P. S. Catalao, S. J. P. S Mariano, V. M. F. Mendes, and L. A. F. M. Ferreira, "Short-term electricity prices forecasting in a competitive market: A neural network approach", Elsevier B.V, 2006

[21] Howard Demuth, Mark Beale, "Neural Network Toolbox", The Math Work, 2004.

[22] D.Ortiz-Arroyo, M.K. Skov, and Q. Huynh, "Accurate Electricity Load Forecasting with Artificial Neural Networks", University Esbjerg Denmark, 2005

[23] G. A. Adepoju, S. O. A. Ogunjuyigbe, and K.O. Alawode, "Application of Neural Network to Load Forecasting in Nigerian Electrical Power System", The Pacific Journal of Science and Technology, May 2007

[24] Noorazliza Binti Sulaiman, "Electricity Load Forecasting For the Small Scale Power System Using Artificial Neural Network", Universiti Teknologi PETRONAS, June 2010

[25] Eugene, A. F. and Dora, "Load forecasting", State University of New York, 2004

[26] Bakirtzis, A. G., Petridis, V., Kiartzis, S. J., Alexiadis, M. C., and Maissis, "A Neural Network Short Term Load Forecasting Model for Greek Power Systems", IEEE Transactions on Power Systems, 1996

[27] Senjyu, T., Sakihara, H., Tamaki, Y., and Uezato, "Next Day Peak Load Forecasting using Neural Network with Adaptive Learning Algorithm Based on Similarity", Electric Machines and Power Systems, 2000

[28] David A. Freedman, "Statistical Models: Theory and Practice", Cambridge University Press, 2005

[29] R. Dennis Cook, Sanford Weisberg, "Criticism and Influence Analysis in Regression", Sociological Methodology, Vol 13. Pp 313-361, 1982

[30] Novak, V., Perfilieva, I and Močkoř, J., "Mathematical principles of fuzzy logic", Dodrecht: Kluwer Academic, 1999

[31] RF Expert. "RF Expert systems" Retrieved April 27, 2011 from http://www.rfexpert.org/index.php?id=37&L=1

[32] Corinna Cortes and V. Vapnik, "Support-Vector Networks", Machine Learning, 1995 Retrieved April 27, 2011 from http://www.springerlink.com/content/k238jx04hm87j80g/

[33] Wikimedia. "Artificial Neural Network" Retrieved October 17, 2010 from http://en.labs.wikimedia.org/wiki/Artificial_Neural_Networks/Print_Version

[34] T. Fomby, "Artificial Neural Networks (ANNs)", Southern Methodist University, 2008

[35] C. Stergiou and D. Siganos, "Neural Networks", Retrieved May 9, 2011 from http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

[36] iPredict, "Time-series Forecasting Error Statistics", Retrieved May 9, 2011 from http://www.ipredict.it/ErrorStatistics.aspx

APPENDIX A

UTP ACADEMIC CALENDAR

JANUARY 2006 SEMESTER (UNDERGRADUATE)

PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	14 Jan 2006	22 Jan 2006
Registration OF Existing Students	1 day	22 Jan 2006	
Lecture	7	23 Jan 2006	10 March 2006
Mid Semester Break	1	11 March 2006	19 March 2006
Lecture	7	20 March 2006	5 May 2006
Study Week	1	6 May 2006	14 May 2006
Examination Week	3	15 May 2006	2 June 2006
End of Semester Break	7	3 June 2006	23 July 2006

JULY 2006 SEMESTER (UNDERGRADUATE)

PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	15 July 2006	23 July 2006
Registration OF Existing Students	1 day	23 July 2006	
Lecture	7	24 July 2006	8 Sept 2006
Mid Semester Break	1	9 Sept 2006	17 Sept 2006
Lecture	7	18 Sept 2006	3 Nov 2006
Study Week	1	4 Nov 2006	12 Nov 2006
Examination Week	3	13 Nov 2006	1 Dec 2006
End of Semester Break	7	2 Dec 2006	21 Jan 2007

JANUARY 2007 SEMESTER (UNDERGRADUATE)

PARTICULARS	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	13 Jan 2007	21 Jan 2007
Registration of Existing Students	1 day	21 Jan 2007	
Lecture	7	22 Jan 2007	9 March 2007
Mid-Semester Break	1	10 March 2007	18 March 2007
Lecture	7	19 March 2007	4 May 2007
Study Week	1	5 May 2007	13 May 2007
Examination Week	3	14 May 2007	1 June2007
End of Semester Break	7	2 June 2007	22 July 2007

JULY 2007 SEMESTER (UNDERGRADUATE)

PARTICULARS	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	14 July 2007	22 July 2007
Registration of Existing Students	1 day	22 July 2007	
Lecture	7	23 July 2007	7 Sept 2007
Mid-Semester Break	1	8 Sept 2007	16 Sept 2007
Lecture	7	17 Sept 2007	2 Nov 2007
Study Week	1	3 Nov 2007	11 Nov 2007
Examination Week	3	12 Nov 2007	30 Nov 2007
End of Semester Break	7	1 Dec 2007	20 Jan 2008

JANUARY 2008 SEMESTER (UNDERGRADUATE)

PARTICULARS	NO. OF WEEKS	DATE START	DATE ENDS
Registration and Orientation of New Students	1	12 Jan 2008	20 Jan 2008
Registration of Existing Students	1 day	20 Jan 2008	
Lecture	7	21 Jan 2008	7 March 2008
Mid-Semester Break	1	8 March 2008	16 March 2008
Lecture	7	17 March 2008	2 May 2008
Study Week	1	3 May 2008	11 May 2008
Examination Week	3	12 May 2008	30 May 2008
End of Semester Break	7	31 May 2008	20 July 2008

JULY 2008 SEMESTER (UNDERGRADUATE)

PARTICULARS	NO. OF WEEKS	DATE START	DATE ENDS
Registration and Orientation of New Students	1	12 July 2008	20 July 2008
Registration of Existing Students	1 day	20 July 2008	
Lecture	10	21 July 2008	26 Sept 2008
Mid-Semester Break	1	27 Sept 2008	7 Oct 2008
Lecture	4	8 Oct 2008	31 Oct 2008
Study Week	1	1 Nov 2008	9 Nov 2008
Examination Week	3	10 Nov 2008	28 Nov 2008
End of Semester Break	7	29 Nov 2008	18 Jan 2009

JANUARY 2009 SEMESTER (UNDERGRADUATE)

PARTICULARS	NO. OF WEEKS	DATE	
		START	END
Registration and Orientation of New Students	1	10 Jan 2009	18 Jan 2009
Registration of Existing Students	1 day	18 Jan 2009	
Lecture	9	19 Jan 2009	20 March 2009
[illegible]	[illegible]	[illegible]	[illegible]
Lecture	5	30 March 2009	1 May 2009
Study Week	1	2 May 2009	10 May 2009
Examination Week	3	11 May 2009	29 May 2009
[illegible]	[illegible]	[illegible]	[illegible]

JULY 2009 SEMESTER (UNDERGRADUATE)

PARTICULARS	NO. OF WEEKS	DATE	
		START	END
Registration and Orientation of New Students	1	11 July 2009	19 July 2009
Registration of Existing Students	1 day	19 July 2009	
Lecture	9	20 July 2009	18 Sept 2009
[illegible]	[illegible]	[illegible]	[illegible]
Lecture	5	30 Sept 2009	30 Oct 2009
Study Week	1	31 Oct 2009	8 Nov 2009
Examination Week	3	9 Nov 2009	27 Nov 2009
[illegible]	[illegible]	[illegible]	[illegible]

ACADEMIC CALENDER UNDERGRADUATE PROGRAMS
YEAR 2010

January 2010 Semester

PARTICULARS	NO. OF WEEKS	DATE	
		START	ENDS
Registration and Orientation of New Students	1	16 Jan 2010	24 Jan 2010
Registration of Existing Students	1 day	24 Jan 2010	
Lecture	7	25 Jan 2010	12 Mar 2010
Mid-Semester Break	1	13 Mar2010	21 Mar 2010
Lecture	7	22 Mar 2010	7 May 2010
Study Week	1	8 May 2010	16 May 2010
Examination Week	3	17 May 2010	4 Jun 2010
End of Semester Break	7	5 Jun 2010	25 July 2010

July 2010 Semester

PARTICULARS	NO. OF WEEKS	DATE	
		START	ENDS
Registration and Orientation of New Students	1	17 July 2010	25 July 2010
Registration of Existing Students	1 day	25 July 2010	
Lecture	6	26 July 2010	3 Sept 2010
Mid-Semester Break	1	4 Sept 2010	14 Sept 2010
Lecture	8	15 Sept 2010	5 Nov 2010
Study Week	1	6 Nov 2010	14 Nov 2010
Examination Week	3	15 Nov 2010	3 Dec 2010
End of Semester Break	7	4 Dec 2010	23 Jan 2011

APPENDIX B

MATLAB CODING FOR MODEL 1

```
clear;
clc;
echo on;
pause
load utploaddatasemoff;
p=trdat';
pt=trtgdat';
VV.P=val';
VV.T=valtg';
ts=tsdat';
tst=tstgdat';
pause
net=newff((minmax(p)),[3 1],{'tansig' 'purelin'},'trainlm');
pause
net.trainParam.epochs = 100;
net.trainParam.goal = 0.001;
net.trainparam.show=1;
%============Start training the model.%Please%wait.=============
pause
net=train(net,p,pt,[],[],VV);
pause
test1=sim(net,p);
pause
day=[1:1:221];
plot(day,test1,day,pt)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title('COMPARISON')
legend('simulation','actual',1)
grid on
error_test1=(sum(abs(test1-pt))/size(pt,2))*100;
pause
test2=sim(net,ts);
pause
error=(sum(abs(test2-tst))/size(tst,2))*100;
day=[1:1:166];
plot(day,test2,day,tst)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title('COMPARISON')
legend('testing','actual',2)
grid on
pause
ain=ain';
aout=aout';
pload=sim(net,ain);
```

```
pload=(10000*(pload));
pload
aload=(10000*(aout));
aload
pause
MAPE=(sum(abs(pload-aload))/sum(abs(aload)))*100;
day=[1:1:7];
plot(day,pload,day,aload)
xlabel('time(in days)')
ylabel('kW load')
title('COMPARISON')
legend('predicted','actual',2)
grid on
pause
```

APPENDIX C

MATLAB CODING FOR MODEL 2

```
clear;
clc;
echo on;
pause
load utploaddatasemon;
p=trdat';
pt=trtgdat';
VV.P=val';
VV.T=valtg';
ts=tsdat';
tst=tstgdat';
pause
net=newff((minmax(p)),[9 11 1],{'tansig' 'tansig' 'purelin'
},'trainlm'); %asal 9
pause
net.trainParam.epochs = 100;
net.trainParam.goal = 0.001;
net.trainparam.show=1;
%============Start training the model.%Please%wait.===============
pause
net=train(net,p,pt,[],[],VV);
pause
test1=sim(net,p);
pause
day=[1:1:437];
plot(day,test1,day,pt)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title('COMPARISON')
legend('simulation','actual',1)
grid on
error_test1=(sum(abs(test1-pt))/size(pt,2))*100;
pause
test2=sim(net,ts);
pause
error=(sum(abs(test2-tst))/size(tst,2))*100;
day=[1:1:327];
plot(day,test2,day,tst)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title('COMPARISON')
legend('testing','actual',2)
grid on
pause
ain=ain';
aout=aout';
```

```
pload=sim(net,ain);
pload=(10000*(pload));
pload
aload=(10000*(aout));
aload
pause
MAPE=(sum(abs(pload-aload))/sum(abs(aload)))*100
day=[1:1:7];
plot(day,pload,day,aload)
xlabel('time(in days)')
ylabel('kW load')
title('COMPARISON')
legend('predicted','actual',2)
grid on
pause
```

APPENDIX D

MATLAB CODING FOR USER-FRIENDLY FEATURES

```
clear;
clc;
fprintf('\nElectricity Forecasting for Small Scale Power System
using ANN \n\n')

ff = 999;

while ff > 0
    get = input ('\nEnter 1 for Semester Off or 2 for Semester On
-> ' );
    if get == 1
        run utplf1;
    elseif get == 2
        run utplf2;
    else
    fprintf('\n Incorrect reply, try again later \n\n');
    end



    repeat = 0;
    while strcmp(repeat, 'n')~=1 && strcmp(repeat, 'N')~=1 &&
strcmp(repeat, 'y')~=1 && strcmp(repeat, 'Y')~=1
    repeat = input('\n\n Do you wish to continue? Enter Y or N ->
','s');
    if strcmp(repeat, 'n')~=1 && strcmp(repeat, 'N')~=1 &&
strcmp(repeat, 'y')~=1 && strcmp(repeat, 'Y')~=1
    fprintf('\n Incorrect reply, try again \n\n'), end
    end

    if repeat == 'y' || repeat == 'Y'
    ff = 999;
    else ff = 0; end

end
```

APPENDIX E

GANTT CHART

GANTT CHART