ELECTRICITY FORECASTING FOR SMALL SCALE POWER SYSTEM USING ARTIFICIAL NEURAL NETWORK

By

MUHAMMAD IZZAT BIN ABD. MALIK

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme

in Partial Fulfilment of the Requirements

for the Degree

Bachelor of Engineering (Hons)

(Electrical & Electronics Engineering)

SEPTEMBER 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

Muhammad Izzat Bin Abd. Malik

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

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

September 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.

Muhammad Izzat Bin Abd. Malik

ABSTRACT

Short-term load forecast is an essential part of electric power system planning and operation. For this project, the main focus will be on the Gas District Cooling Plant (GDC) which acts as the primary source of energy for Universiti Teknologi PETRONAS (UTP). This project is looking into weekly forecast of the electricity production for the GDC plant using Artificial Neural Network Approach. This forecasting method will be very useful to support plant operation as the trending of load demand for an educational centre such as UTP is very much dependent on the university activities itself. The project involve MATLAB program for the STLF with Artificial Neural Network prediction model. The obtained results showed that introducing Multilayer Perceptron (MLP) Neural Network architecture improve the prediction significantly by obtaining a very small value of Mean Absolute Percent Error (MAPE). Besides that, by getting the smaller value of MAPE, it represents higher forecast accuracy of the model itself. The report consists of an introduction, problem statement, objectives, literature review and methodology used to solve the problem. It further looks into the obtained results with consistent discussion.

ACKNOWLEDGEMENTS

Firstly, my utmost gratitude to ALLAH the All-Mighty for his uncountable graces upon me and for the successful completion of this project in due course of time.

I would like to express the inmost appreciation to my supervisor, Dr. Zuhairi Hj. Baharudin, Senior Lecturer of Electrical & Electronics Department, UTP. The supervision and continuous support that he gave truly help me throughout completing this project. His guidance especially in correcting various documents of mine with attention and care.

My respectful gratitude goes to my co-supervisor, Ir. Mohd Fatime Irzaq Khamis, Senior Electrical Executive of Maintenance Department, UTP, for his full support in the completion of this project. His constant guidance, helpful comments and suggestions has helped me not only to complete but also to enhance the expected results of the project. His kindness, valuable advice, friendly approach and patience will always be appreciated.

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

I sincerely thank Nik Adli Hakimi Nik Mohamad Shukri for his valuable assistance in developing the finest forecasting model throughout this project.

Lastly, great appreciation tomy friends, who were a constant source of support during my work. To allUTP lecturers, students and staff and to all whose their names are not mentioned here but they provided help directly or indirectly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

ANN	Artificial Neural Network
GDC	Gas District Cooling
MLP	Multilayer Perceptron
UTP	Universiti Teknologi PETRONAS
MAPE	Mean Absolute Percentage Error

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Recently there are two sources that generate the electricity to Universiti Teknologi PETRONAS. The primary source is the gas district cooling (GDC) while the utility company Tenaga Nasional Berhad (TNB) provides the backup source. In addition, the capacity of the GDC plant is rated at 8.4 MW that consists of two gas turbine generators that generate 4.2 MW each. The objective of this project is to forecast the short term electricity demand for this university itself by using Artificial Neural Network (ANN). Forecast model will be designed using ANN with a few goals, which are to help GDC for pre-planned scheduling, to determine the suitable time for the plant maintenance and last but not least as the guidance throughout the whole year of plant operation. The proposed ANN model will be introduced by using previous load demand data as the set of input while the output would be the future load demand data. The accuracy of the forecast data will be determined by calculating the mean absolute percentage error (MAPE).

1.2 Problem Statement

1.2.1 *Problem Identification*

Load forecasting using conventional methods are currently being practiced among several power companies in Malaysia. However, since the relationship between load power and factors influencing load power is nonlinear, it is difficult to identify its nonlinearity by using conventional methods [1]. In this particular case, GDC UTP will be the main subject while factors that lead to the complexity electricity demand values are such as temperature and weather.

Due to the complicacy and uncertainty of load forecasting, electric power load is difficult to be forecasted precisely if no analysis model and numerical value algorithm model is applied [2].

1.2.2 *Significance of Project*

The artificial neural network used in short-time load forecasting can grasp interior rule in factors and complete complex mathematic mapping [2].

By modeling a prediction model using ANN, it will help to improve the current electricity management by GDC UTP in terms of scheduling the plant maintenance and as the part to support plant operation. Load forecasting model for this project will consider all possible inputs which are then being employed in Short Term Load Forecasting (STLF). This model will use the previous load demand data as the set of input while the output would be the future load demand data.

1.3 Objective and Scope of the project

1.3.1 *Main Objective*

To assess the performance of the Load Forecasting Model using Artificial Neural Network (ANN) by getting the lowest value of mean absolute percentage error (MAPE) that will be used to measure the performance of a model.

The sub objectives of the project are listed as the following:

i. To outline the importance of the load forecast
ii. To outline the fundamental of ANN
iii. To construct forecast load of GDC UTP load demand using the suggested model

1.3.2 *Scope of Project*

This project will start with some literature review related to load forecasting and also the theory of ANN. Next will be the data gathering from GDC UTP which are then will be used as the input for the load forecasting model designed using MATLAB software. STLF will be used to predict one week ahead prediction. Further testing will be carried out to design the load prediction model with the highest accuracy.

1.4 Relevancy of Project

Since maximum UTP load and in-plant used was recorded at 6.2 MW and 1.2 MW respectively compared with maximum generation at 8.4 MW, STLF model will become appreciable [3]. This model will help to save the electricity generation from GDC UTP by predicting the future electricity demand so that the real maximum generation can be controlled within the prediction value. Another important aspect is that the accuracy of the prediction that will ensure the continuous and reliability supply to the consumers itself.

1.5 Feasibility of Project

The project will be done in two semesters which include three area which are research, development and also improvement of the model itself. The objective is to reduce MAPE error as compared to previous project. Input of the model will be gathered from GDC UTP. Besides that, MATLAB will be used as the tools to develop the ANN algorithm. System testing and implementation also will be using this software. Based on the description above, it is very clear that this project will be feasible to be carried out within the time frame.

CHAPTER 2
LITERATURE REVIEW

2.1 Importance of Load Forecasting

There are several techniques for load forecasting that been widely used for the last few decades. This shows the importance and the usefulness of the load forecasting. Load forecasting can be divided into three categories which are short term, medium term and long term [3]. Short term load forecasting (STLF) means the load forecasting usually from one hour to a week [4]. It represents a great saving potential for economic and secure operation of power system [5]. However, medium term load forecasting (MTLF) means the load forecasting usually from a week to a year [6]. Medium term load forecasts usually incorporate several additional influences – especially demographic and economic factors [7]. Long term load forecasting (LTLF) is usually more than a year [6] which is very useful for a longer duration of planning operations.

Up to now, the main focus in load forecasting has been on STLF since it is an important tool in the day to day operation of utility systems [8]. With a good forecast model, it will help GDC in scheduling generators operation e.g. to start, to stop and maintenance works [3]. It is evident that improvement in load forecast has a direct positive impact on system security and also cost of operation [10]. The primary application of the STLF function is to drive the scheduling functions that determine the most economic commitment of generation sources consistent with reliability requirements, operational constraints and policies, and physical, environmental and equipment limitations [4]. They are also necessary for power system security studies, including contingency analysis and load management [9]. If applied to the system security assessment problem, it can provide valuable information to detect many vulnerable situations in advance [11]. The objective of load forecasting is to get the smallest MAPE as possible although the decrement of the particular value is only around 1%. In fact, the costs of the error are so high that research that could help reducing it in a few percent points would be amply justified [12].

2.2 Artificial Neural Network Overview

Neural networks are highly interconnected simple processing units designed in a way to model how the human brain performs a particular tasks [14]. ANN traces previous load patterns and predicts (i.e. extrapolates) a load pattern using recent load data [11]. Their basic unit is the artificial neuron, schematically represented in Fig. 1. The neuron receives (numerical) information through a number of input nodes (four, in this example), processes it internally, and puts out a response [12]. Before getting the output response, the information will be passed through a transfer function which is linear, sigmoid or hyperbolic tangent. The basic building blocks of the ANN are such as Network Architecture (connection between neurons), Training or Learning (determining weights on the connections) and Activation function. More details explanation on these three points will be discussed in the next page.

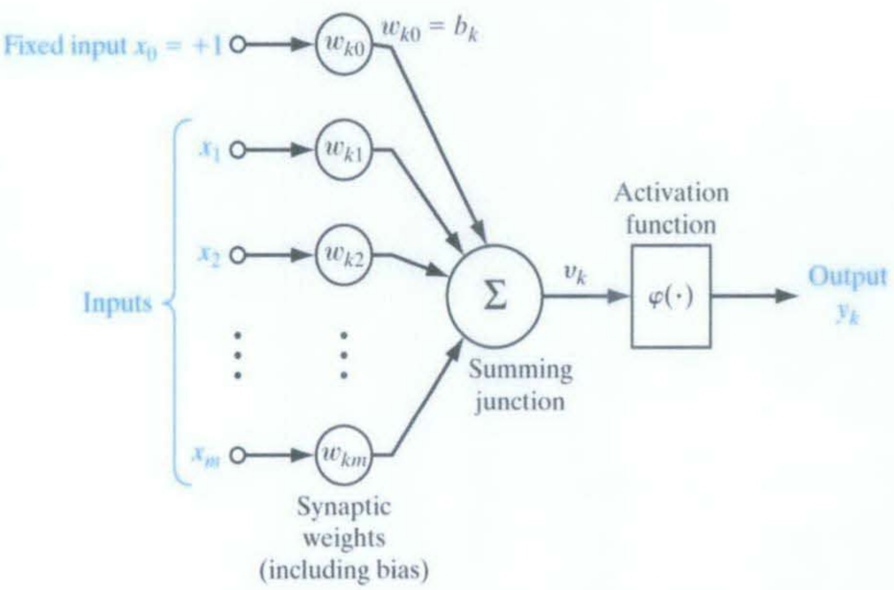

Figure 1: Nonlinear model of a neuron [12]

2.3 Artificial Neural Network Architecture

The word Architecture is used to describe the Neural Network configuration, together with other attributes of the Neural Network such as the learning rule, activation function, learning and momentum factors [15]. Networks with interconnections that do not form any loops are called Feed-forward [16]. Recurrent or non feed-forward networks in which there are one or more loops of interconnections are used for some kinds of applications [17]. The best way to describe the definition of network architecture is the manner in which the neurons of neural network are structured is intimately linked with the learning algorithm used to train the network [14]. The most popular ANN architecture is a three layered, feed-forward system with back-propagation [22]. Throughout project, Multilayer perceptrons which utilizes a supervised learning technique called back-propagation for training the network will be used. Generally, we may define the network architecture in three main classes:

2.3.1 *Single Layer Feed-forward Network*

It consists of a single layer of weighted interconnections. The inputs may be connected fully to the output units [18]. In other words, this network is strictly a feed-forward or acyclic type [14].

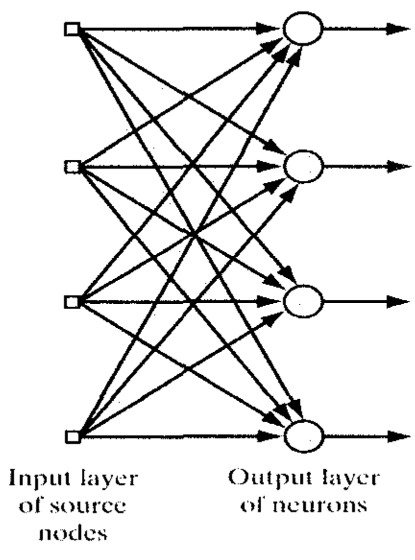

Figure 2 : Feed-forward or acyclic network with a single layer of neurons [14]

2.3.2 *Multilayer Feed-forward Networks*

It consists of single input layer, one or more hidden layers and one output layer. Each layer employs several neurons and each neuron in a layer is connected to the neurons in the adjacent layer with different weights [11]. The function of hidden neurons is to intervene between the external input and the network output in some useful manner [14]. By increasing the hidden layers, it can be used to solve more complicated problems.

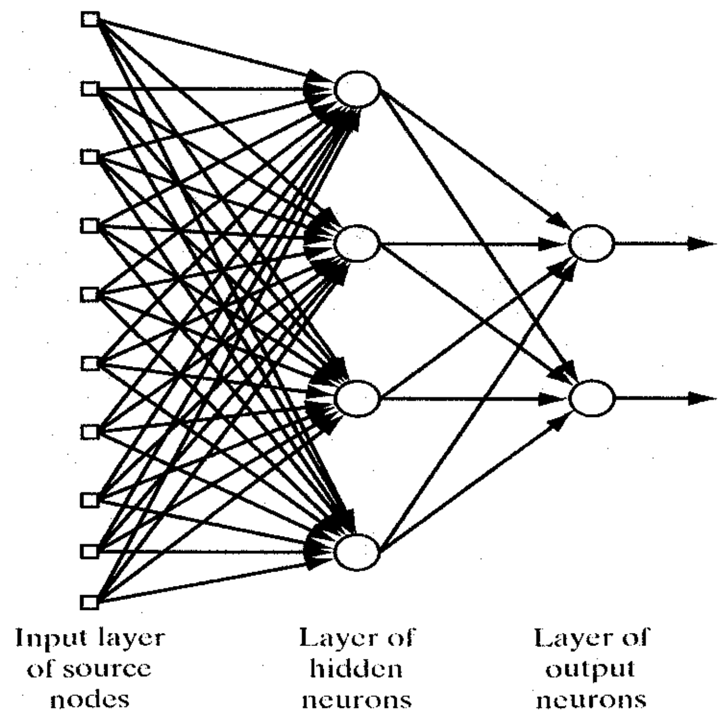

Figure 3 : Multilayer Feed-forward Networks with one hidden layer and one output layer

2.3.3 *Recurrent Network*

All units are connected to all other units and every unit is both input and an output [18]. It has self-feedback loops which ensure the output of neuron is fed back into its own input. The presence of feedback loops has a profound impact on the learning capability of the network and on its performance [14].

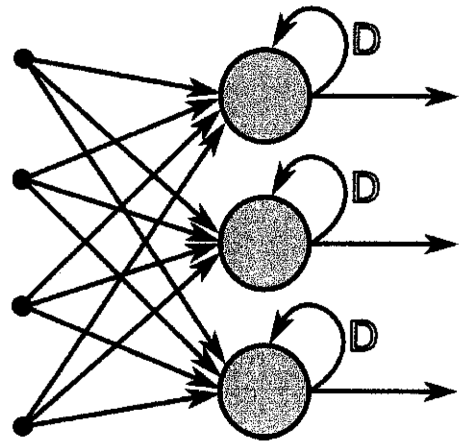

Figure 4 : Recurrent Networks

2.4 Learning Paradigms

Learning Paradigm refers to a model of the environment in which the neutral network operates [14]. Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place [19]. The success of this project configuration dwells in the fact that it can learn from retrospective information in a supervised learning [22]. In other hand, supervised learning will be preferred for this project's learning paradigm. In general, there are two types of learning paradigms as follows:

2.4.1 *Supervised Learning*

Is a process of providing the network with a series of sample inputs and comparing the output with the expected responses and the training continues until the network is able to provide the expected response [18]. Besides that, in supervised learning the network is trained using historical data derived from the system, the relationship between input and output is to be determined [22]. By considering learning with a teacher is also referred to as supervised learning, teacher will have the knowledge of the environment that is represented by a set of input-output examples. Suppose now that teacher and the neural network are now both exposed to a training vector drawn from the environment, by virtue of built-in knowledge, the teacher is able to provide the neural network with a desired response for that training vector [14]. Based on figure 5, learning system will deliver its actual response with the objective to get the lowest error signal based on the difference with the desired response. This feedback loop will continue until the neural network can emulate the teacher which in this case the value of the error signal will be zero.

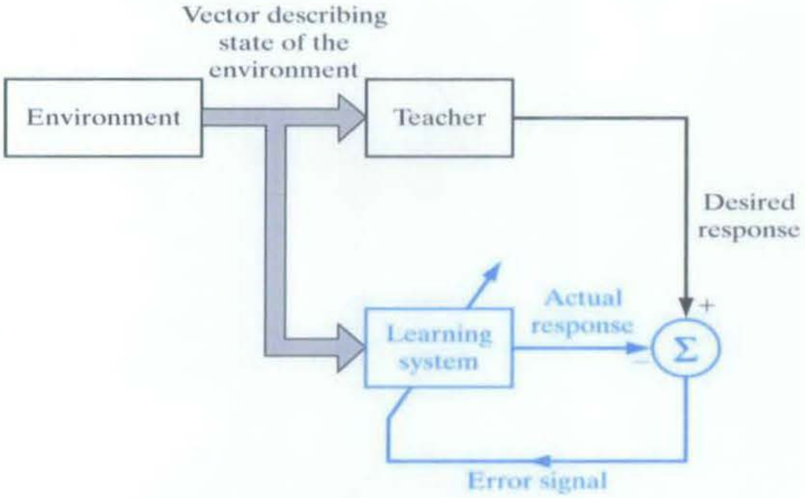

Figure 5 : Block Diagram of learning with a teacher [14]

2.4.2 *Unsupervised Learning*

In a neural network, if for the training input vectors, the target output is not known, the training method adopted is called as unsupervised training [18]. In this type of learning, there will be no teacher to supervise the learning process as illustrated in Figure 6. Modification of the weight by network will be done to ensure the most similar input vector is assigned to the same output unit. Because of the teacher's unavailability in this type of learning, provision is made for a task independent measure of the quality of representation that the network is required to learn, and the free parameters of the network are optimized with respect to that measure [14]. Once the network has become tuned to the statistical regularities of the input data, it develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically [20].

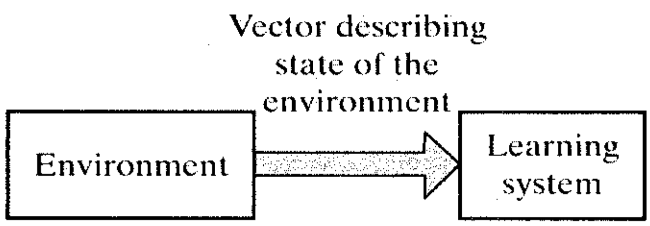

Figure 6 : Block Diagram of unsupervised learning

2.5 Learning Algorithms

The process of modifying the weights in the connections between networks layers with the objective of achieving the expected output is called training a network [18].The adequate selection of inputs for neural network training is highly influential to success the training [16]. In the learning process a neural network constructs an 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 [16]. Each learning algorithm differs from the other in the way in which the adjustment to a synaptic weight of a neuron is formulated. There are various learning rules that available such as:

2.5.1 *Hebbian Learning Algorithm*

This learning can also be called as correlational learning. The rules can be expand into two parts based on the original meaning:

i. If two neurons on either side of a synapse (connection) are activated simultaneously, then the strength of that synapse is selectively increased.
ii. If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated [14].

This Hebbian learning rule represents a purely feed forward, unsupervised learning [18].

2.5.2 *Competitive Learning Algorithm*

In this learning, the output neurons of a neural network compete among themselves to become active. It is this feature that makes competitive learning highly suited to discover statistically salient features that may be used to clarify a set of input patterns [14]. Besides, this rule has a mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron, or only one neuron per group, is active at a time [18]. Winner-takes-all-neuron title will be given to the winning neuron.

2.5.3 *Back-propagation learning Algorithm*

In this algorithm, the input is passed layer through layer until the final output is calculated, and it is compared to the real output to find the error [16]. According to the difference between the produced and target outputs, the network's weights are adjusted to reduce the output errors [11]. However, the standard back propagation learning algorithm is not efficient numerically and tends to converge slowly [16]. In order to accelerate the learning process, two parameters of the back-propagation, the learning rate and another parameter, momentum can be adjusted [21]. The learning rate is proportion of error gradient by which the weights should be adjusted. Larger values can give a faster convergence to the minimum but also may produce oscillation around the minimum [16]. The momentum determines the proportion of the change of past weights that should be used in the calculation of the new weights [22].

2.6 Activation Function

The activation function is used to calculate the output response of a neuron [18]. The behavior of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (Activation Function) that is specified for the units [22]. In choosing the activation function, this function must be differentiable and non decreasing; most papers used either logistic (sigmoid) or the hyperbolic tangent functions, and it is not clear whether the choice has any effect on the forecasting accuracy [12, 24].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 [25, 16]. Two of most commonly used functions are shown below:

2.6.1 *Sigmoid Functions*

The sigmoid function which has s-shaped graph is by far the most common form of activation function used in the construction of artificial neural network. It is defined as strictly increasing function that exhibits a grateful balance between linear and nonlinear behavior [14]. Two types of sigmoid functions will be used throughout this project:

2.6.1.1 Logistic Function (Log-Sigmoid Transfer Function)

An example of the sigmoid function is the logistic function which is defined as:

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

Where a is the slope parameter of the sigmoid function. The sigmoid transfer function shown below takes the input, that have value between plus and minus infinity, and squashes the output into the range 0 to 1.

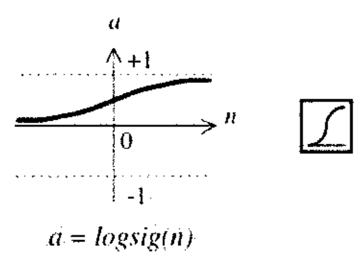

Figure 7 : Log-Sigmoid Transfer Function

2.6.1.2 Hyperbolic Tangent Function

For the corresponding form of a sigmoid function we may use the hyperbolic tangent function, defined by:

$$\varphi(v) = \tanh(v)$$

The function generates outputs between -1 and +1 as the neuron's net input goes from negative to positive infinity.

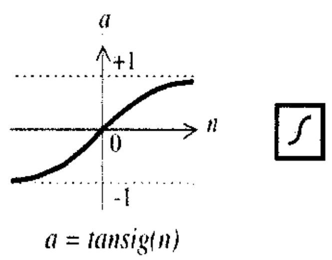

Figure 8 : Tan-Sigmoid Transfer Function

2.6.2 *Identity Function*

This is a linear transfer function: $f(x) = x$

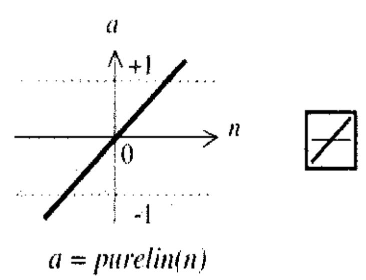

Figure 9 : Linear Transfer Function

2.6.3 *Levenberg-Marquardt back-propagation algorithm*

The standard back-propagation algorithm has been widely applied in neural network learning. However, due to the low speed of convergence, considerable research works have been done to improve it. A lately developed algorithm, the Lavenberg-Marquardt back-propagation, has been used to train feed-forward neural networks since it can yield a speed-up of large factors via limited modifications of the standard back-propagation algorithm [26].

2.7 Improving Generalization

One of the problems that occur during neural network training is called overfitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations. One method for improving network generalization is to use a network that is just large enough to provide an adequate fit. The larger network you use, the more complex the functions the network can create. If you use a small enough network, it will not have enough power to overfit the data. Unfortunately, it is difficult to know beforehand how large a network should be for a specific application. There are two other methods for improving generalization that are implemented in Neural Network Toolbox software: regularization and early stopping. The next sections describe these two techniques and the routines to implement them. A good choice for the generalization parameter is the subject matter of this section [14].

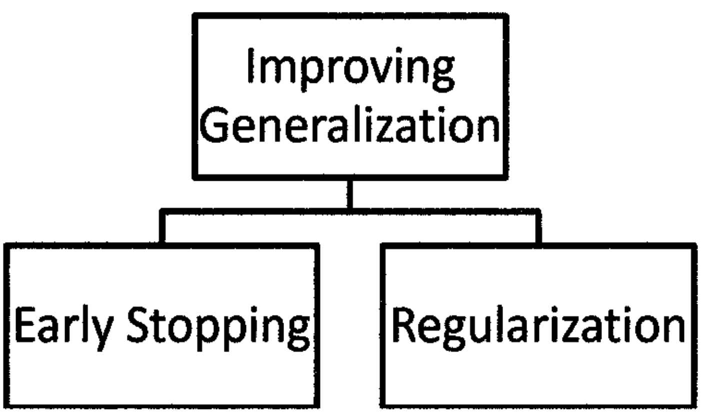

Figure 10 : Techniques on Improving Generalization

2.7.1 *Early Stopping*

In this technique the available data is divided into three subsets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set typically begins to rise. When the validation error increases for a specified number of iterations (net.trainParam.max_fail), the training is stopped, and the weights and biases at the minimum of the validation error are returned.

2.7.2 *Regularization*

Another method for improving generalization is called regularization. This involves modifying the performance function, which is normally chosen to be the sum of squares of the network errors on the training set. The problem with regularization is that it is difficult to determine the optimum value for the performance ratio parameter. If you make this parameter too large, you might get overfitting. If the ratio is too small, the network does not adequately fit the training data.

2.8 Application Example

In this section, there will be some review on the selected papers which have highest citations. The selection of these papers will be based on the objective itself which is the implementation of Artificial Neural Network in load forecasting. Those papers are listed as the following:

2.8.1 *Neural Networks for Short-Term Load Forecasting [12]*

The aim of this paper is to investigate the reasons for such skepticism which has been highlighted by some authors that believe the advantages of using ANN in forecasting have not been systematically proved yet. In order to overcome this problem, this paper examines a collection of papers (published between 1991 and 1999) that report the application of ANN to short term load forecasting. ANN network architecture and also the configuration proposed in this paper will be presented in a table as below:

Table 1 : ANN Design 1 Implementation

Architecture		*Learning Paradigm*		*Learning Algorithm*		*Activation Function*	
Single Layer FFN		Supervised	√	Hebbian		Logistic	√
Multilayer FFN	√	Unsupervised		Comparative		Hyperbolic	√
Recurrent Network				Back-propagation	√	Identity	
**FFN : Feed-Forward Network *Most of papers use Logistic or Hyerbolic as the activation function*						Lavenberg-Marquardt	

2.8.2 *Electric Load Forecasting Using Artificial Neural Network [11]*

This paper evaluates ANN approach in load forecasting compared to the previous method used such as time series and regression method. Inaccuracy of prediction and numerical instability for time series and regression method allow ANN to prove its capability in handling such load forecasting problem. The ANN is able to perform non-linear modeling and adaptation. A widely used model called the multi-layered perceptron (MLP) will be utilized in this paper. Network architecture and configuration of ANN are presented in a table as below:

Table 2 : ANN Design 2 Implementation

Architecture		Learning Paradigm		Learning Algorithm		Activation Function	
Single Layer FFN		Supervised	√	Hebbian		Logistic	
Multilayer FFN	√	Unsupervised		Comparative		Hyperbolic	√
Recurrent Network				Back-propagation	√	Identity	
*FFN : Feed-Forward Network						Lavenberg-Marquardt	

2.8.3 *Short-term electricity prices forecasting in a competitive market: A neural network approach [16]*

This paper highlighted a neural network approach for forecasting short-term electricity prices. The network architecture used in this paper will be the usual the architecture that been used for ANN except some modification for the activation function. This paper utilizes Levenberg-Marquardt algorithm as it trains a neural network 10-100 times faster than the usual back-propagation algorithm. In addition, the Levenberg-Marquardt provides a nice compromise between the speed of Gauss-Newton and the guaranteed convergence of steepest descent [16, 27]. The design of ANN for this paper is illustrated in table as below:

Table 3 : ANN Design 3 Implementation

Architecture		Learning Paradigm		Learning Algorithm		Activation Function	
Single Layer FFN		Supervised	√	Hebbian		Logistic	
Multilayer FFN	√	Unsupervised		Comparative		Hyperbolic	
Recurrent Network				Back-propagation	√	Identity	
*FFN : Feed-Forward Network						Lavenberg-Marquardt	√

2.9 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 types based on UTP's academic calendar, as attached in Appendix A. The graphs below represent the daily data gathered from two conditions; Semester Off and Semester On.

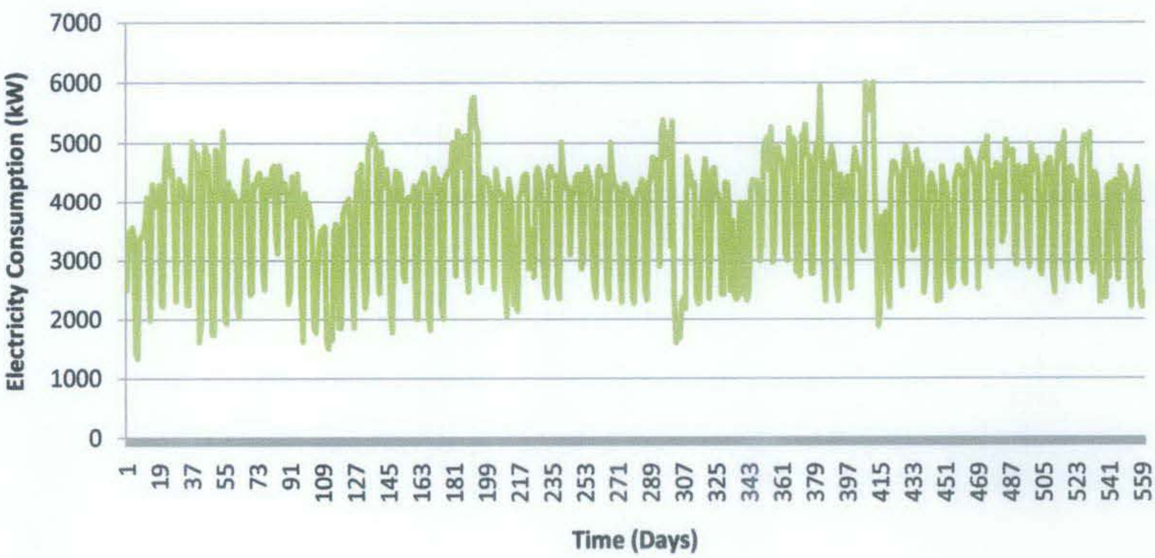

Figure 11: Graph of UTP Electricity Demand during Semester OFF

Figure 11 shows the UTP electricity consumption during Semester OFF. The average of the electricity demand is 3.8MW. At certain particular time, the electricity demand is higher due to the special occasions organized in UTP.

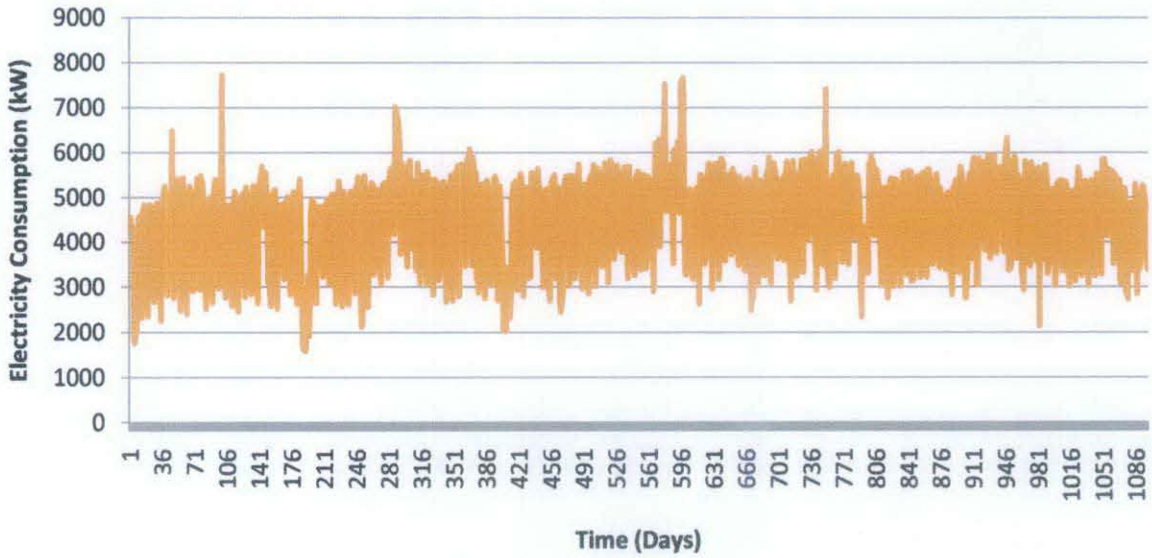

Figure 12 : Graph of UTP Electricity Demand during Semester ON

Figure 12 shows the UTP electricity consumption during Semester On. The average of the electricity demand is 4.6MW. At certain time, the electricity demand is higher due to a number of special occasions organized in UTP [23].

2.9.1 *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:

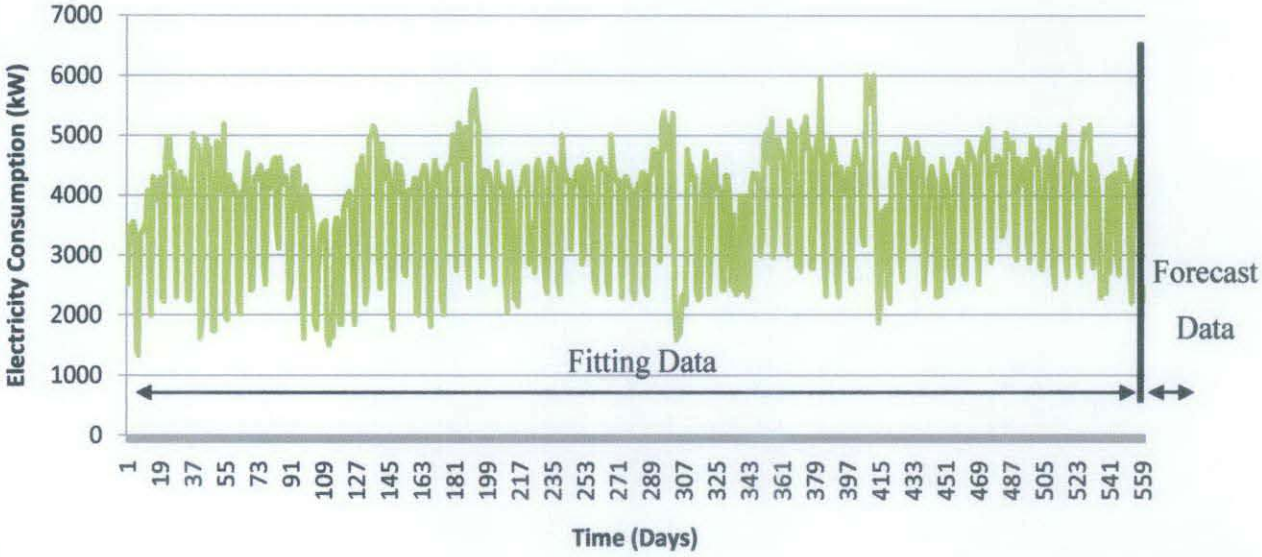

Figure 13 : Graph of Fitting Data and Forecast Data during Semester OFF

Figure 13 shows the UTP electricity demand during Semester Off. The graph is labeled with the fitting data as well as the forecast data. The fitting data will be used for the purpose of training and testing in Artificial Neural Network.

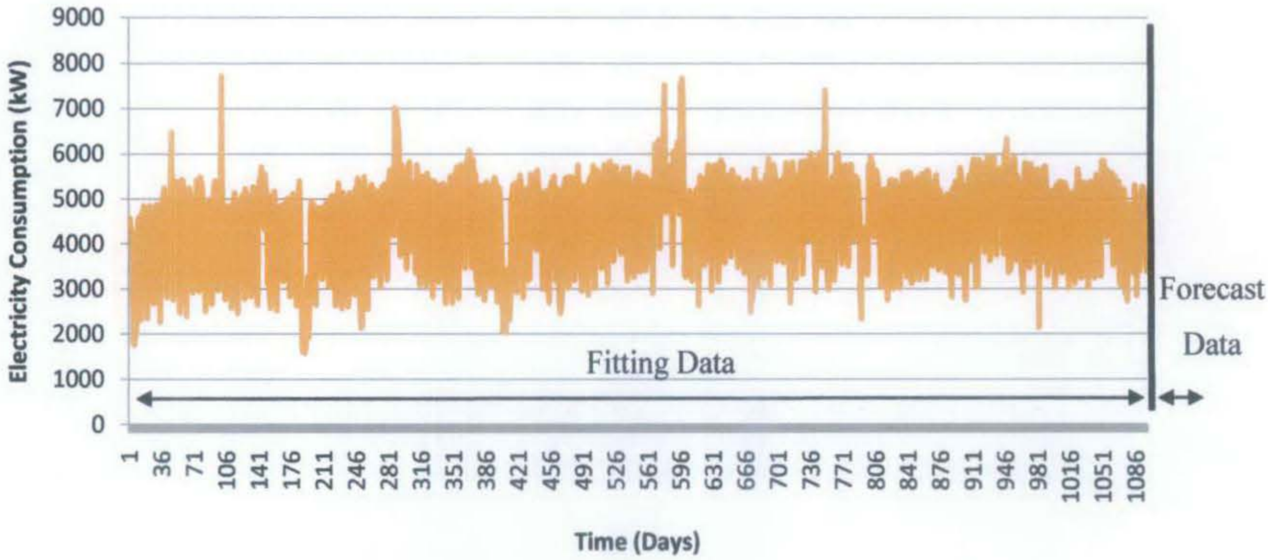

Figure 14 : Graph of Fitting Data and Forecast Data during Semester ON

Figure 14 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 will be used for training and testing purpose in ANN.

CHAPTER 3
METHODOLOGY

3.1 Research Methodology

In order to achieve the main objective of this project, the goals for the three sub objectives highlighted in the earlier part need to be accomplished. In outlining the importance of load forecasting, detailed review as well as brief research about the topic is focused on the selected papers which concentrate on the load forecasting itself. The issues relevancy between the selected papers and our project's objective need to be taken into account to ensure the credibility of this project.

For the second sub objective which is to outline the fundamental of ANN, literature reviews as well as brief research about the topic are carried out on several resources such as books, journals and also internet.

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

3.2 Flow Chart

The following flow chart explains the methodology in executing the project:

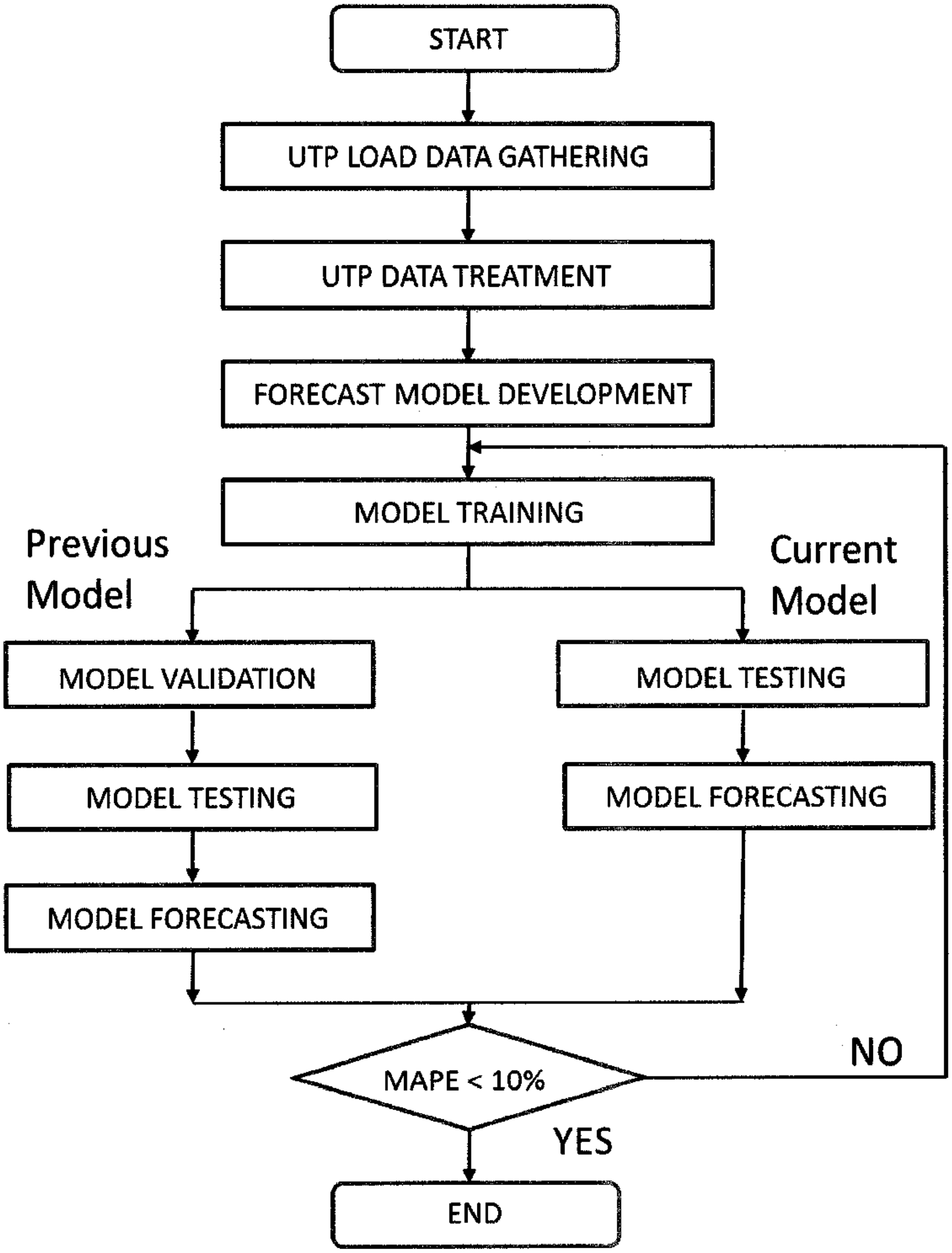

Figure 15 : Project Methodology

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 31stDecember 2010. The data have been categorized into Semester ON and Semester OFF based on UTP Academic Calendar [28].

3.2.2 Forecast Model Development

The forecast models have been developed using the data gathered. Previously, there are two models that have been developed. The models are based on Semester ON and Semester OFF and the duration of load forecasting are fixed on 7 days ahead prediction. The summary of the two (2) models are as below:

Table 4 : Previous Forecast Models

Model 1	OFF	7 Days
Model 2	ON	7 Days

Table 5 : Data Partitioning for Previous Forecast Models

Data Partitioning	Training	Testing	Validation
Model 1	40%	30%	30%
Model 2	40%	30%	30%

Table 4 shows the two models developed previously. While table 5 shows data partitioning used for both previous models. New models will be developed based on the previous forecast models with some modifications on the data partitioning. The new models developed shown below:

Table 6 : Current Forecast Models

Model	Semester Type	Forecasting Duration
Model 1	OFF	7 Days
Model 2	ON	7 Days

Table 7 : Data Partitioning for Current Forecast Model

Data Partitioning	Training	Testing	Validation
Model 1	70%	30%	×
Model 2	70%	30%	×

Table 6 shows the new models developed with respect to the previous models with the objective to improve the existed design of the load forecasting models in terms of Minimum Average Percentage Error (MAPE). By getting smaller MAPE, the goal will be achieved. Meanwhile Table 7 showing data partition used for both new models.

The MATLAB coding script(s) used before have to be modified in order to fulfil the new data partitioning criteria. The two models coding script are included in Appendix B.

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 and data partitioning. The data treatment for the new models will undergo a significant change on the data partitioning compared to the previous model.

a) Data Normalization

The data used for the model development should be in the range of -1 to +1 since the transfer function used for input of the model is tansig. This is to ensure that the MATLAB can work well without any delay and etc. Hence the daily data has been normalized by dividing them with 10000. As the forecast load value has been obtained, the value then should be converted back by a multiplication of 10000.

b) Data Partitioning

The gathered data has been partitioned into two (2) partitions for the purpose of training and testing as follows:

i) Training data – 70%

ii) Testing data – 30%

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

c) Data Arrangement

The arrangement of data need to be done in order that the first day of the week starts on Monday and end on Sunday. This will provide ANN with organized information which helps the network to predict accurately.

3.2.4 *Model Training*

The training of the new models involves the first 70% of the gathered data. The data for each model is represented in the table as below:

Table 8 : Training Data for each model

Model	Semester Type	Training Data
Model 1	OFF	387
Model 2	ON	764

3.2.5 *Model Testing*

The testing data for the current models consists of the last 30%of the total data. The purpose of testing is to observe the efficiency of the developed models [28] and the numbers of testing data are as follows:

Table 9 : Testing Data for each model

Model	Semester Type	Training Data
Model 1	OFF	166
Model 2	ON	328

3.2.6 *Forecast Model*

The forecast model is developed to forecast seven days ahead. The range for the forecast data for this project will be selected within the fitting data as it will be compared to the actual data. Comparison between these two data will be then be used to calculate Mean Absolute Percentage Error (MAPE).

Table 10 : Actual & Forecast Data Range

Model	Semester Type	Actual Data	Forecast Data
Model 1	OFF	13th-19th December 2010	13th-19th December 2010
Model 2	ON	4th-10th October 2010	4th-10th October 2010

3.2.7 *MAPE*

MAPE or Mean Absolute Percentage Error is calculated once the forecasting load obtained. The program will compare the actual load with the forecast load and therefore error will be calculated based on the following formula:

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

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

3.3 User-friendly Features

A user-friendly feature is added to allow users to run MATLAB program of either Model 1 (Semester Off) or Model 2 (Semester On) in a single M-file. A selection screen will appear in the command window for users to choose the models. A counter has been added to make sure the users can run the program for the number of times that they want to. Average forecast demand and average MAPE will be calculated at the end of the program.

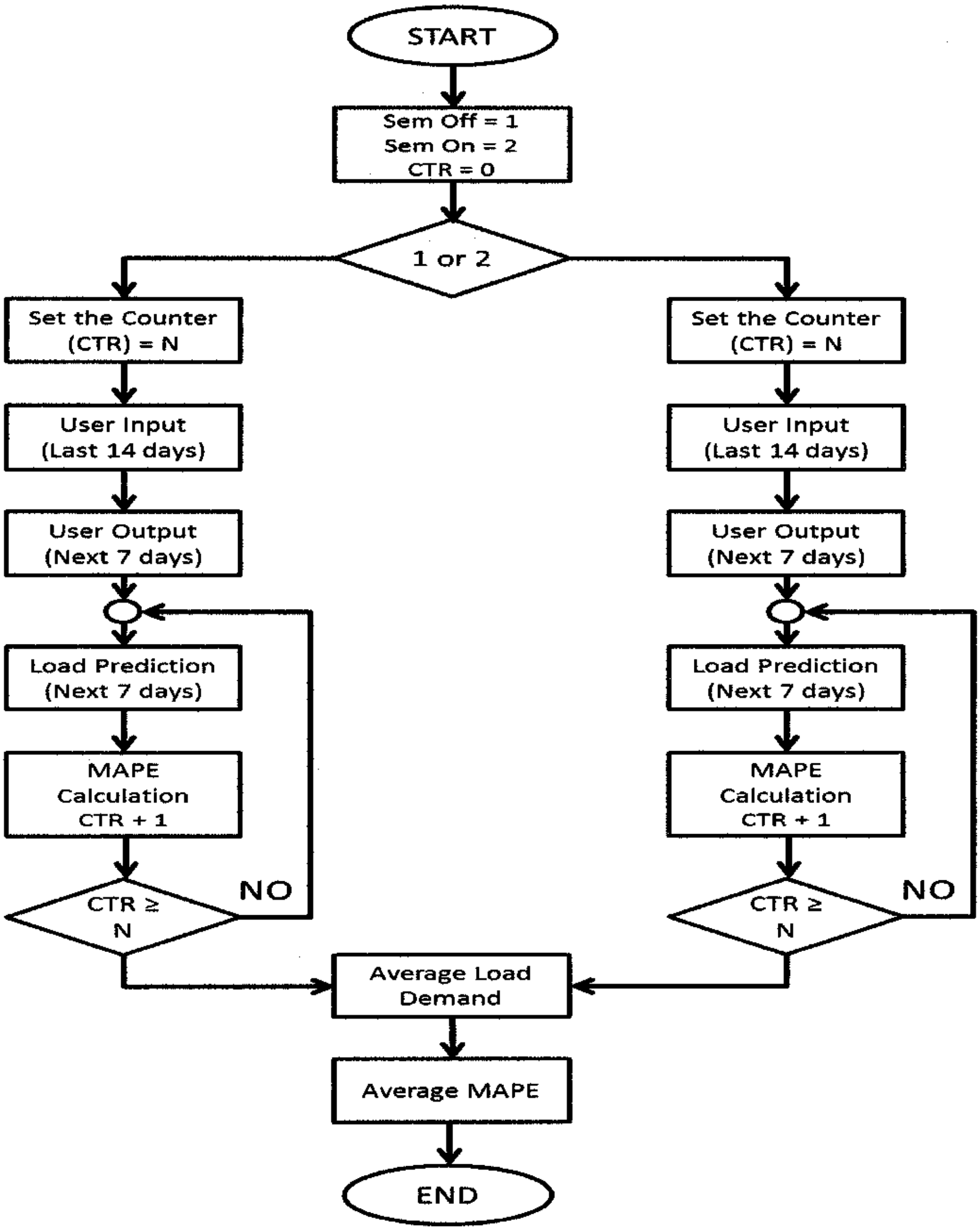

Figure 16 : User-friendly Features Flow Chart

3.4 Project Duration

In order to effectively monitor the progress of this project, a Gantt chart consists of one year duration had been constructed. See Appendix C.

3.5 Tool Required

The MATLAB R2010a software is used as the main tool for the new load forecasting developments. MATLAB is an ideal tool for working with ANN since it is highly efficient in performing vector and matrix calculations. Neural Network Toolbox which is included in the MATLAB software also will provide useful tools for ANN development.

CHAPTER 4
RESULTS & DISCUSSION

4.1 Results

The forecast models for 2010 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 17 and Table 11:

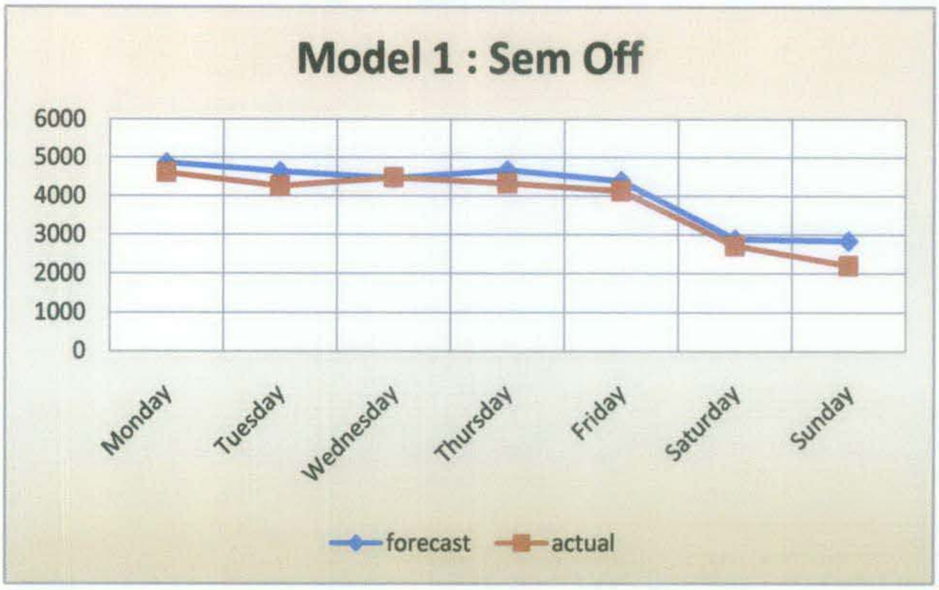

Figure 17 :Forecast and Actual Load Demand for Model 1 (13th -19th December 2010)

Table 11 : Comparison between Actual and Forecast Load Demand for Model 1

Day	Load Demand (kW)		MAPE
	Forecast	Actual	
Monday	4866.9	4604	7.75
Tuesday	4634.1	4252	
Wednesday	4459.6	4472	
Thursday	4670.1	4328	
Friday	4387.8	4128	
Saturday	2886.4	2716	
Sunday	2844.2	2204	

Based on the value tabulated, MAPE obtained for Model 1 is 7.75. While for Model 2, the comparison between Actual and Forecast Load Data are shown in Figure 18 and Table 12:

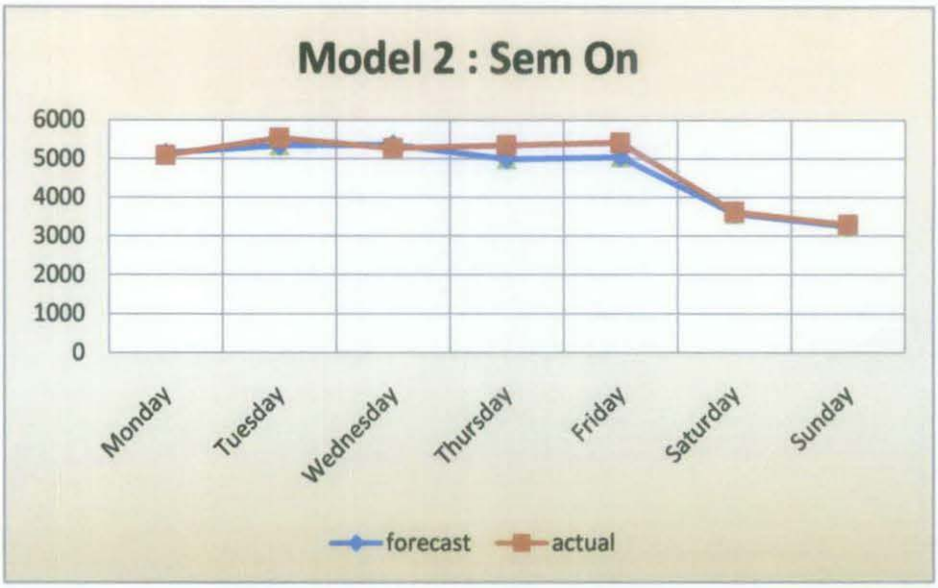

Figure 18 : Forecast and Actual Load Demand for Model 2 (4th -10th October 2010)

Table 12 : Comparison between Actual and Forecast Load Demand for Model 2

Day	Load Demand (kW)		MAPE
	Forecast	Actual	
Monday	5263.8	5092	3.704
Tuesday	5393.1	5540	
Wednesday	5514.6	5276	
Thursday	5250.6	5352	
Friday	5190.5	5412	
Saturday	3522.1	3616	
Sunday	3018.4	3288	

The MAPE value obtained is 3.704. In order to obtain more accurate results, both models has been run 20 times and the average value of MAPE is taken. The current results are compared with the previous model results [25]. The results are shown in Table 13:

Table 13 : Simulation for both Model 1 and Model 2 [25]

Simulation	Current Model		Previous Model	
	Model 1	Model 2	Model 1	Model 2
1	7.7412	3.0679	7.1546	3.7042
2	8.3883	3.7578	5.2386	3.5971
3	7.6327	4.4304	3.6287	4.1158
4	9.3478	3.4776	5.5743	3.0855
5	8.4831	3.0238	7.7631	2.4734
6	8.9779	3.0249	9.4669	4.889
7	9.3897	3.5168	9.3826	4.3525
8	8.366	2.8402	9.8419	3.4071
9	8.2384	4.2686	9.6389	4.4519
10	9.3372	2.8569	4.4725	3.5399
11	7.5861	3.509	6.5133	3.1771
12	8.6058	3.8518	4.9058	4.8466
13	7.1219	3.1243	4.9813	5.3818
14	9.2791	3.7831	9.2332	3.3560
15	6.9947	3.383	5.5859	2.5488
16	8.2273	3.1297	6.6534	4.6263
17	8.593	3.1087	5.9580	3.6896
18	8.8798	3.9178	5.8613	5.2790
19	8.3784	3.2183	5.4148	4.8582
20	7.7514	3.5693	7.1163	2.9513
Average MAPE Value (%)	**8.37**	**3.44**	**6.72**	**3.92**

The forecast model developed for both Semester Off (Previous Model 1) and Semester On (Current Model 2) has been tested using the 2011 data to assess the system capability to forecast new data. Based on the results for 2010 data, it is proven that Previous Model 1 is capable in handling data during Semester Off while Current Model 2 is suitable for Semester On data. Thus, these forecast models are simulated to determine the result of one week forecast load for 2011 data. Semester Off will be using Previous Model 1 and for Semester Off, another approach which is Current Model 2 will be applied. The comparison of the forecast load and the actual load demand for Semester Off are shown in Figure 19 and Table 14:

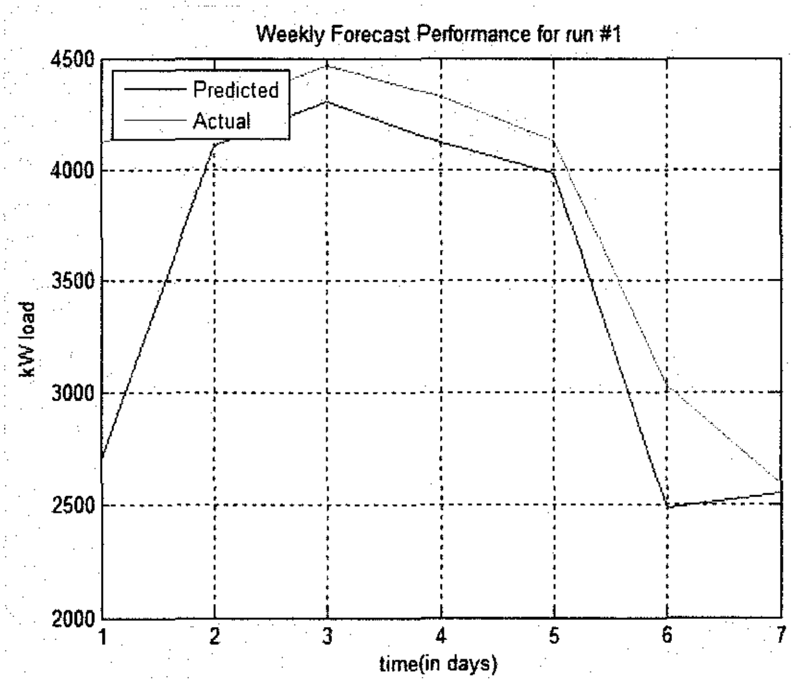

Figure 19 : Forecast and Actual Load Demand using Previous Model 1 for Semester Off (10th – 16th January 2011)

Table 14 : Comparison between Actual and Forecast Load Demand using Previous Model 1 for Semester Off (10^{th} – 16^{th} January 2011)

Day(Semester Off)	Load Demand (kW)		MAPE (%)
	Forecast	Actual	
Monday	2703.1	4124	9.931
Tuesday	4105.9	4272	
Wednesday	4308.1	4468	
Thursday	4126.5	4328	
Friday	3986.9	4128	
Saturday	2482.8	3024	
Sunday	2551.3	2596	

Based on Table 14 and Figure 18, the value of MAPE is less than 10% which is 9.931. The evaluation between the forecast load and the actual load demand for Semester On are shown in Figure 20 and Table 15:

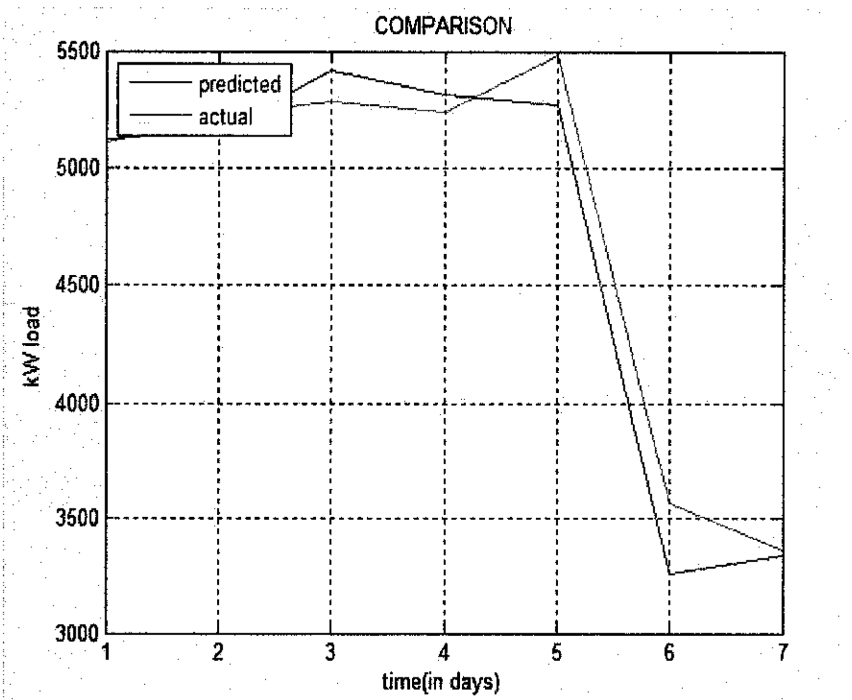

Figure 20 : Forecast and Actual Load Demand using Current Load Demand for Semester On (28^{th} February – 6^{th} March 2011)

Table 15 : Comparison between Actual and Forecast Load Demand using Current Model 2 for Semester On (28th February – 6th March 2011)

Day(Semester On)	Load Demand (kW)		MAPE (%)
	Forecast	Actual	
Monday	5125.3	5112	2.4781
Tuesday	5156.9	5224	
Wednesday	5413.6	5288	
Thursday	5315.7	5244	
Friday	5268.0	5480	
Saturday	3259.9	3568	
Sunday	3337.0	3364	

Based on Table 15 and Figure 19, the value of MAPE obtained is very small which is 2.4781.

4.2 Discussion

The simulation results of the new models are compared to the previous model's results and it is tabulated as below:

Table 16 : Comparison between Model 1 and Model 2

	Current Model		Previous Model	
	Model 1	Model 2	Model 1	Model 2
Average MAPE Value (%)	**8.37**	**3.44**	**6.72**	**3.92**

Table 16 shows the comparison of the results obtained for previous model (With Validation) and the current model (Without Validation) respectively. From the results, it is clear that Model 2 suit most for the new model (Without Validation) as compared to Model 1. Different settings between Previous Model and New Model are the main factor that leads to this result. The settings that been used are tabulated in Table 17 as shown below:

Table 17 : Setting Comparison between Previous Model and Current Model

Data Partitioning	Previous Model (With Validation)				Current Model (Without Validation)			
	Model 1		Model 2		Model 1		Model 2	
	%	Data	%	Data	%	Data	%	Data
Training	40	221	40	437	70	387	70	764
Validation	30	166	30	328	0	0	0	0
Testing	30	166	30	327	30	166	30	328
TOTAL	100	553	100	1092	100	553	100	1092

For previous model (With Validation), data partitioning involves training, validation and testing while for the new model, only training and testing has been included. The purpose of validation is to guide the training of the particular model. During Semester Off, due to the fact that there are many university events conducted during this semester which lead to non-steady state load during this period, hence new Model 1 will not be able to predict accurately due to the absence of validation part. Besides that, because of the lower number of total data, this may affect the accuracy of the model itself. For the new Model 2, the efficiency has been improved and represents a high degree of accuracy in the load forecasting. Factors that may contribute to this good accuracy are such as high total number of data and also the steady state load data. During semester on, the data mostly are in steady state as the usage of electricity is quite the same in each day.

Forecast Models for 2011 data have shown positive achievement especially for Semester On Model. For Semester Off 2011, the MAPE generated randomly for this simulation is 9.931, while for Semester On 2011, the MAPE value obtained is 2.4781. Both good results show the ability of the network to predict the current load demand data although the 2011 data has dissimilar trend compared to 2010 load demand data. The results show the ability of the proposed models to predict the current load demand below 10% of MAPE even though the load demand trend data is different from the historical data. The factor that may lead to this unalike trend will be the development of new buildings or facilities that consume electricity which affect the overall electricity usage.

CHAPTER 5
CONCLUSIONS & RECOMMENDATIONS

5.1 Conclusions

Electricity forecasting plays a key role in GDC system operations as the principal driving element for all daily and weekly operations scheduling 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, during the demand is less than 4.2MW, the other generator can be shut down for maintenance works. Besides that, electricity forecasting will help GDC to produce exact amount of electricity generated at the exact time.

Consequently, it leads to the optimization in power generation and reduce the power wastage during the generation. Load forecasting accuracy significantly impacts the amount power generated in operational planning of the GDC. 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 8.37 % for Model 1 and 3.44 % 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 feed forward neural network.

5.2 Recommendations

From the result obtained, it is believe there are still rooms for improvement. For Model 1, improvement will need to be done as the value obtained is still higher than the one in previous model. 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. The effect on electricity forecasting accuracy of weather parameters such as temperature, cloud cover and wind chill factor will be also be investigated in the future work.

In addition, the main 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]Tomonobu Senjyu, Hitoshi Takara, Katsumi Uezato and Toshihisa Funabashi, "One-Hour-Ahead Load Forecasting Using Neural Network", *IEEE Transactions on Power System*, vol. 17, pp. 113, 1, 2002

[2] Wenjin Dai and Ping Wang, "Application of Pattern Recognition and Artificial Neural Network to Load Forecasting in Electric Power System",*icnc*, vol. 1, pp. 381-385, Third International Conference on Natural (ICNC 2007), 2007

[3] Mohd Fatimie Irzaq bin Khamis, Zuhairi bin Baharudin, Nor Hisham bin Hamid, Mohd Faris bin Abdullah and Siti Sarah Md Yunus, "Electricity Forecasting for Small Scale Power System using Fuzzy Logic", *IPEC 2010*, pp. 1040, 2010

[4] G. Gross and F. D. Galiana. "Short-term load forecasting", *Proceedings of the IEEE*, vol. 75, pp. 1558-1573, 1987.

[5] N. Amjady,. "Short-Term Bus Load Forecasting of Power Systems by a New Hybrid Method", *Power Systems, IEEE Transactions on*, vol. 22, pp. 333-341, 2007.

[6] C. Chatfiled, "Time Series Forecasting. ", Chapman and Hall, 2000.

[7] Heiko Hahn, Silja Meyer-Nieberg and Stefan Pickl. "Electric load forecasting methods: Tools for decision making ", *European Journal of Operational Research*, vol. 199, pp. 902-907, 2009.

[8] Gonzalez-Romera, Jaramillo-Moran, Carmona-Fernandez. "Monthly electric energy demand forcasting based on trend extraction. " *IEEE Transactions on Power Systems* 21 (4), 1946-1953.

[9] Pang Qingle, Zhang Min, "Very Short Term Load Forecasting Based on Neural Network and Rough Set" icicta, vol.3, pp.1132-1135, *2010 International Conference on Intelligent Computation Technology and Automation*, 2010.

[10] K. Methaprayoon, W. Lee, S. Rasmiddattaa, J.Lia and R. Ross, "Multistage Artifical Neural Network Short Term Load Forecasting Engine With Front-End Weather Forecast," *IEEE Industrial Applications*, vol. 43, no. 6, pp. 1410-1416, 2007.

[11] Park D.C, El-Sharkawi M.A, Marks R.J II, Atlas L.E and Damborg M.J. "Electric Load Forecasting Using An Artificial Neural Network" *IEEE Transactions on Power System,* Vol. 6, No. 2, pp. 442-449, 1991.

[12] Hippert H.S, Pedreira C.E and Souza R.C. "Neural Networks for Short-Term Load Forecasting : A Review and Evaluation" *IEEE Transactions On Power Systems,* Vol. 16, No. 1, pp. 44-55, 2001.

[14] S. Haykin, Neural Networks : A Comprehensive Foundation, Prentice-Hall, New Jersey, 1999.

[15] Sapeluk A.T, Ozveren C.S and Birch A.P. "Short Term Electric Load Forecast Using Artificial Neural Networks", *Electrotechnical Conference, 1994. Proceedings., 7th Mediterranean,* pp. 905, 1994.

[16] 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.

[17] L.B. Almeida, in: E. Fiesler, R. Beale (Eds.), Multilayer Perceptrons, Handbook of Neural Computation, Oxford University Press, 1997.

[18] SN Sivanandam, S Sumathi and SN Deepa, Introduction to Neural Networks using MATLAB 6.0, McGraw-Hill, New Delhi, 2006.

[19] Mendel, J.M., and R.W. McLaren, 1970. "Reinforcement-learning control and pattern recognition systems," in Adaptive, Learning and Pattern Recognition Systems: Theory and Applications, vol. 66, J.M. Mendel and K.S. Fu, eds., pp. 287-318, New York: Academic Press.

[20] Becker, S., 1991. "Unsupervised learning procedures for neural networks," *International Journal of Neural Systems*, vol. 2, pp. 17-33.

[21] Smith M. "Neural Networks for Statistical Modeling". Van Nostrand Reinhold. New York 1993.

[22] B.R. Szkuta, L.A. Sanabria, T.S. Dillion, Electricity price short term forecasting using artificial neural networks, IEEE Trans. Power Syst. 14 (3) (1999) 851-857.

[23] Dr. S. K. Dass"Neural Network and Fuzzy Logic". Shree Publishers & Distributors, New Delhi, 2006.

[24] G. Zhang, B. E. Pattuwo, and M. Y. Hu, "Forecasting with artificial neural network: The state of the art" Int. J. Forecast., vol. 14, pp. 35-62, 1998.

[25] Salwa Binti Solahuddin, "Electricity Load Forecasting For the Small Scale Power System Using Artificial Neural Network", Universiti Teknologi PETRONAS, May 2011

[26] Chi-Leung Hui, "Artificial Neural Networks - Application", InTech, 2011

[27] L. M. Saini, M. K. Soni, Artificial neural network based peak load forecasting using Levennberg-Marquardt and quasi-Newton method, IEE Proc.-Gener. Transm. Distrib. 149 (5) (2002) 578-584.

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

APPENDIX A

UTP ACADEMIC CALENDAR

JAN 2006 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	14-Jan-06	22-Jan-06
Registration of Existing Students	1 day	22-Jan-06	
Lecture	7	23-Jan-06	10-Mar-06
Mid-Semester Break	1	11-Mar-06	19-Mar-06
Lecture	7	20-Mar-06	5-May-06
Study Week	1	6-May-06	14-May-06
Examination Week	3	15-May-06	2-Jun-06
End of Semester Break	7	3-Jun-06	23-Jul-06

JULY 2006 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	15-Jul-06	23-Jul-06
Registration of Existing Students	1 day	23-Jul-06	
Lecture	7	24-Jul-06	8-Sep-06
Mid-Semester Break	1	9-Sep-06	17-Sep-06
Lecture	7	18-Sep-06	3-Nov-06
Study Week	1	4-Nov-06	12-Nov-06
Examination Week	3	13-Nov-06	1-Dec-06
End of Semester Break	7	2-Dec-06	21-Jan-07

JAN 2007 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	13-Jan-07	21-Jan-07
Registration of Existing Students	1 day	21-Jan-07	
Lecture	7	22-Jan-07	9-Mar-07
Mid-Semester Break	1	10-Mar-07	18-Mar-07
Lecture	7	19-Mar-07	4-May-07
Study Week	1	5-May-07	13-May-07
Examination Week	3	14-May-07	1-Jun-07
End of Semester Break	7	2-Jun-07	22-Jul-07

JULY 2007 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	14-Jul-07	22-Jul-07
Registration of Existing Students	1 day	22-Jul-07	
Lecture	7	23-Jul-07	7-Sep-07
Mid-Semester Break	1	8-Sep-07	16-Sep-07
Lecture	7	17-Sep-07	2-Nov-07
Study Week	1	3-Nov-07	11-Nov-07
Examination Week	3	12-Nov-07	30-Nov-07
End of Semester Break	7	1-Dec-07	20-Jan-08

JAN 2008 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	12-Jan-08	20-Jan-08
Registration of Existing Students	1 day	20-Jan-08	
Lecture	7	21-Jan-08	7-Mar-08
Mid-Semester Break	1	8-Mar-08	16-Mar-08
Lecture	7	17-Mar-08	2-May-08
Study Week	1	3-May-08	11-May-08
Examination Week	3	12-May-08	30-May-08
End of Semester Break	7	31-May-08	20-Jul-08

JULY 2008 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	12-Jul-08	20-Jul-08
Registration of Existing Students	1 day	20-Jul-08	
Lecture	7	21-Jul-08	26-Sep-08
Mid-Semester Break	1	27-Sep-08	7-Oct-08
Lecture	7	8-Oct-08	31-Oct-08
Study Week	1	1-Nov-08	9-Nov-08
Examination Week	3	10-Nov-08	28-Nov-08
End of Semester Break	7	29-Nov-08	18-Jan-09

JAN 2009 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	10-Jan-09	18-Jan-09
Registration of Existing Students	1 day	18-Jan-09	
Lecture	7	19-Jan-09	20-Mar-09
Mid-Semester Break	1	21-Mar-09	29-Mar-09
Lecture	7	30-Mar-09	1-May-09
Study Week	1	2-May-09	10-May-09
Examination Week	3	11-May-09	29-May-09
End of Semester Break	7	30-May-09	19-Jul-09

JULY 2009 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	11-Jul-09	19-Jul-09
Registration of Existing Students	1 day	19-Jul-09	
Lecture	7	20-Jul-09	18-Sep-09
Mid-Semester Break	1	19-Sep-09	29-Sep-09
Lecture	7	30-Sep-09	30-Oct-09
Study Week	1	31-Oct-09	8-Nov-09
Examination Week	3	9-Nov-09	27-Nov-09
End of Semester Break	7	28-Nov-09	17-Jan-10

JAN 2010 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	16-Jan-10	24-Jan-10
Registration of Existing Students	1 day	24-Jan-10	
Lecture	7	25-Jan-10	12-Mar-10
Mid-Semester Break	1	13-Mar-10	21-Mar-10
Lecture	7	22-Mar-10	7-May-10
Study Week	1	8-May-10	16-May-10
Examination Week	3	17-May-10	4-Jun-10
End of Semester Break	7	5-Jun-10	25-Jul-10

JULY 2010 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	17-Jul-10	25-Jul-10
Registration of Existing Students	1 day	25-Jul-10	
Lecture	7	26-Jul-10	3-Sep-10
Mid-Semester Break	1	4-Sep-10	14-Sep-10
Lecture	7	15-Sep-10	5-Nov-10
Study Week	1	6-Nov-10	14-Nov-10
Examination Week	3	15-Nov-10	3-Dec-10
End of Semester Break	7	4-Dec-10	23-Jan-11

JAN 2011 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	1	17-Jan-11	23-Jan-11
Registration of Existing Students	1 day	23-Jan-11	
Lecture	7	24-Jan-11	9-Mar-11
Mid-Semester Break	4 days	10-Mar-11	13-Mar-11
Lecture	7	14-Mar-11	29-Apr-11
Study Week	5 days	30-Apr-11	4-May-11
Examination Week	11 days	5-May-11	15-May-11
End of Semester Break	1	16-May-11	22-May-11

MAY 2011 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Registration of New Students	4 days	19-May-11	22-May-11
Registration of Existing Students	1 day	22-May-11	
Lecture	7	23-May-11	6-July-11
Mid-Semester Break	4 days	7-July-11	10-July-11
Lecture	7	11-July-11	26-Aug-11
Study Week	9 days	27-Aug-11	4-Sep-11
Examination Week	11 days	5-Sep-11	15-Sep-11
End of Semester Break	10 days	16-Sep-11	25-Sep-11

SEPTEMBER 2011 SEMESTER			
PARTICULAR	NO. OF WEEKS	DATE	
		START	ENDS
Lecture	7	26-Sep-11	9-Nov-11
Mid-Semester Break	4 days	10-Nov-11	13-Nov-11
Lecture	7	14-Nov-11	30-Dec-11
Study Week	5 days	31-Dec-11	4-Jan-12
Examination Week	11 days	5-Jan-12	15-Jan-12
End of Semester Break	7	16-Jan-12	22-Jan-12

APPENDIX B

MATLAB CODING FOR MODEL 1& MODEL 2

```
echo off;
clear;
clc;

totalp = zeros(1,7);
totala = zeros(1,7);
totalMAPE = 0;

fprintf('Welcome to UTP Load Forecasting Model 2011\n');

selection = input ('Which type (1 for Sem OFF or 2 for Sem ON): ');
tanya = input ('How many times ??: ');


if selection==1

fprintf('\n');
fprintf('||||| Semester Off Model |||||\n');
fprintf('\n');
%interface for user to input Ain & Aout (SEMESTER OFF)
aino3 = zeros(7,2);
aouto3 = zeros(7,1);
dayz =
{'Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunda
y'};

fprintf('------------------------------------------------------------
-\n');
fprintf('Semester OFF) Enter 14 days before historical load demand
(kW)\n');
fprintf('------------------------------------------------------------
-\n');

for m = 1:2

for n = 1:7
if m==1
             fprintf('Enter value for last 2 week''s %s',dayz{n})
else
             fprintf('Enter value for last week''s %s',dayz{n})
end
         aino3(n,m) = input (': ');
end
end

fprintf('----------------------\n');
fprintf('Enter actual value (kW)\n');
```

```
fprintf('-----------------------\n');

for n = 1:7
    fprintf ('Enter value for next week''s %s',dayz{n})
    aouto3(n) = input (': ');
end

load utploaddatasemoff40_30_30;

for i = 1:tanya

p=trdats';
pt=trtgdats';
VV.P=vals';
VV.T=valtgs';
ts=tsdats';
tst=tstgdats';

fprintf('\n');
net=newff((minmax(p)),[3 1],{'tansig''purelin'},'trainlm');
if i<2
gensim(net)
else
end
pause(2);

fprintf('ANN Configuration in progress\n');

net.trainParam.showWindow = true;
net.trainParam.epochs = 1000;
net.trainParam.goal = 0.00001;
net.trainparam.show=1;
%============Start training the
model.%Please%wait.================================
fprintf('============Start training the model. Please
Wait.================================\n');

net=train(net,p,pt,[],[],VV);

pause(2);

test1=sim(net,p);
fprintf('TRAINING DATA Graph Plot.\n');
pause(2);
day=[1:1:221];
figure('Name','Training Data Semester Off','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title(['Training Data Semester Off for run #',num2str(i)])
legend('Training','Actual',1)
grid on
error_test1=(sum(abs(test1-pt))/size(pt,2))*100;
%fprintf('Press Any Key to Continue\n');
pause(2);
test2=sim(net,ts);
fprintf('--------------------\n');
fprintf('TESTING MODE Starts.\n');
fprintf('--------------------\n');
fprintf('TESTING DATA Graph Plot.\n');
```

```
pause(2);
error=(sum(abs(test2-tst))/size(tst,2))*100;
day=[1:1:166];
figure('Name','Testing Data Semester Off','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title(['Testing Data Semester Off for run #',num2str(i)])
legend('Testing','Actual',2)
grid on
fprintf('------------------------\n');
fprintf('FINAL RESULTS for run #%d\n',i);
fprintf('------------------------\n');
pause (2);
ain=(aino3*0.0001)';
aout=(aouto3*0.0001)';
pload=sim(net,ain);
fprintf('Forecast Load Demand (kW)\n');
pload=pload*10000;
totalp=totalp + pload;
for n = 1:7
    fprintf ('%s : %.3f\n',dayz{n}, pload (n));
end
fprintf('\n--------------------------------\n');
fprintf('Actual Load Demand (kW) for run #%d',i);
fprintf('\n--------------------------------\n');
aload=(aout)*10000;
totala=totala + aload;
for n = 1:7
    fprintf ('%s : %.0f\n',dayz{n}, aload (n));
end

%fprintf('Press Any Key to Calculate the MAPE\n');
pause(2);
MAPE=(sum(abs(pload-aload))/sum(abs(aload)))*100;
fprintf('\n---------------------------\n');
fprintf('MAPE (Percentage) for run #%d',i);
fprintf('\n---------------------------\n');
fprintf('\n>>>> %.3f\n', MAPE);
totalMAPE= totalMAPE + MAPE;
day=[1:1:7];
plot(day,pload,day,aload)
xlabel('time(in days)')
ylabel('kW load')
title(['Weekly Forecast Performance for run #',num2str(i)])
legend('Predicted','Actual',2)
grid on
end

fprintf('\n--------------------------------\n');
fprintf('Average Forecast Load Demand (kW)\n');
fprintf('---------------------------------\n');
avgp = totalp/tanya;
for n = 1:7
    fprintf ('%s: %.3f\n',dayz{n}, avgp (n));
end

fprintf('--------------------------------\n');
fprintf('Average Actual Load Demand (kW)\n');
fprintf('--------------------------------\n');
avga = totala/tanya;
```

```
for n = 1:7
    fprintf ('%s: %.3f\n',dayz{n}, avga (n));
end

avgMAPE = totalMAPE/tanya;
fprintf ('\n\nAverage MAPE (Percentage): %3f\n', avgMAPE);
fprintf('\n\nPress Any Key to end the program\n');
pause


%SELECTION NO 2

elseif selection==2

fprintf('\n');
fprintf('||||| Semester On Model |||||\n');
fprintf('\n');
%interface for user to input Ain & Aout (SEMESTER ON)
ain3 = zeros(7,2);
aout3 = zeros(7,1);
dayz =
{'Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunda
y'};

fprintf('------------------------------------------------------------
--\n');
fprintf('(Semester ON) Enter 14 days before historical load demand
(kW)\n');
fprintf('------------------------------------------------------------
--\n');

for m = 1:2
for n = 1:7
if m==1
             fprintf('Enter value for last 2 week''s %s',dayz{n})
else
             fprintf('Enter value for last week''s %s',dayz{n})
end
    ain3(n,m) = input (': ');
end
end

fprintf('-----------------------\n');
fprintf('Enter actual value (kW)\n');
fprintf('-----------------------\n');

for n = 1:7
    fprintf ('Enter value for next week''s %s',dayz{n})
    aout3(n) = input (': ');
end

load utploaddatasemon70_30;

for i = 1:tanya

p=trdat70';
pt=trtgdat70';
ts=tsdat30';
tst=tstgdat30';
```

```
fprintf('\n');
net=newff((minmax(p)),[9 11 1],{'tansig''tansig''purelin'
},'trainlm'); %asal 9
if i<2
gensim(net)
else
end
pause (2);

fprintf('ANN Configuration in progress\n');

net.trainParam.showWindow = true;
net.trainParam.epochs = 1000;  %asal 100
net.trainParam.goal = 1e-4;    %asal 1e-3
net.trainparam.show=1;         %asal 1
%============Start training the
model.%Please%wait.===============================
fprintf('============Start training the model. Please
Wait.===============================\n');

net=train(net,p,pt,[],[]);

pause(2);

test1=sim(net,p);
fprintf('TRAINING DATA Graph Plot.\n');
pause (2);
day=[1:1:764];
figure('Name','Training Data Semester On','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title(['Training Data Semester On for run #',num2str(i)])
legend('Training','Actual',1)
grid on
error_test1=(sum(abs(test1-pt))/size(pt,2))*100;
%fprintf('Press Any Key to Continue\n');
pause (2);
test2=sim(net,ts);
fprintf('--------------------\n');
fprintf('TESTING MODE Starts.\n');
fprintf('--------------------\n');
fprintf('TESTING DATA Graph Plot.\n');
pause (2);
error=(sum(abs(test2-tst))/size(tst,2))*100;
day=[1:1:328];
figure('Name','Testing Data Semester On','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title(['Training Data Semester On for run #',num2str(i)])
legend('Testing','Actual',2)
grid on
fprintf('--------------\n');
fprintf('FINAL RESULTS.\n');
fprintf('--------------\n');
pause (2);
ain=(ain3*0.0001)';
aout=(aout3*0.0001)';
```

```
pload=sim(net,ain);
fprintf('\n------------------------------------\n');
fprintf('Forecast Load Demand (kW)for run #%d',i);
fprintf('\n------------------------------------\n');
pload=pload*10000;
totalp = totalp + pload;
for n = 1:7
    fprintf ('%s: %.3f\n',dayz{n}, pload (n));
end
fprintf('\n----------------------------------\n');
fprintf('Actual Load Demand (kW) for run #%d',i);
fprintf('\n----------------------------------\n');
aload=(aout)*10000;
totala = totala + aload;
for n = 1:7
    fprintf ('%s: %.0f\n',dayz{n}, aload (n));
end
%fprintf('Press Any Key to Calculate the MAPE\n');
pause (2);
MAPE=(sum(abs(pload-aload))/sum(abs(aload)))*100;
fprintf('\n----------------------------\n');
fprintf('MAPE (Percentage) for run #%d',i);
fprintf('\n----------------------------\n');
fprintf('\n>>>> %.3f\n', MAPE);
totalMAPE = totalMAPE + MAPE;
day=[1:1:7];
plot(day,pload,day,aload)
xlabel('time(in days)')
ylabel('kW load')
title(['Weekly Forecast Performance for run #',num2str(i)])
legend('Predicted','Actual',2)
grid on
end


fprintf('\n--------------------------------\n');
fprintf('Average Forecast Load Demand (kW)\n');
fprintf('----------------------------------\n');
avgp = totalp/tanya;
for n = 1:7
    fprintf ('%s: %.3f\n',dayz{n}, avgp (n));
end


fprintf('--------------------------------\n');
fprintf('Average Actual Load Demand (kW)\n');
fprintf('--------------------------------\n');
avga = totala/tanya;
for n = 1:7
    fprintf ('%s: %.3f\n',dayz{n}, avga (n));
end

avgMAPE = totalMAPE/tanya;
fprintf ('\n\nAverage MAPE (Percentage): %3f\n', avgMAPE);

fprintf('\n\nPress Any Key to end the program\n');
pause

x=0;
end
```

APPENDIX C

GANTT CHART