NEURAL NETWORK APPLICATION
TO
SHORT TERM LOAD FORECAST

By

MOHAMAD FAIZ BIN MARZUKI

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfilment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

MAY 2012

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

NEURAL NETWORK APPLICATION TO SHORT TERM LOAD FORECAST

by

Mohamad Faiz bin Marzuki

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 2012

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.

____

Mohamad Faiz bin Marzuki

ABSTRACT

Power system planning and operation is an important part for power systems industry. By having a good planning and operation, the quality of power supplied will be improved ensuring both consumer and power provider getting their share equally. In this case, the most challenging part is the prediction of how much the load that will be used by the consumer for a short period of time. This prediction is called load forecasting. This will be very useful to every power system company as the trending for the load demand is different for each geographical location. There are different methods to do the load forecasting. One of the project involved MATLAB program for the short term load forecasting (STLF) using Artificial Neural Network (ANN) model. We are using Multilayer Perceptron (MLP) Neural Network architecture, it will improve the forecast value significantly by obtain a very small mean absolute percentage error (MAPE). By getting a smaller MAPE, it represents higher forecast accuracy of the model itself. The elements in this report contain of an introduction, problem statement, objectives, literature review and methodology which was used to solve the forecasting problems. The discussion of the obtained results will be looked further in this project.

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.

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

TABLE OF CONTENTS

ABSTRACT IV

ACKNOWLEDGEMENTS VI

LIST OF TABLES ix

LIST OF FIGURES x

LIST OF ABBREVIATION XII

CHAPTER 1 INTRODUCTION 1

1.1 Background of Study 1

1.2 Problem Statement 2

1.2.1 Problem Identification 2

1.2.2 Significance of Project 2

1.3 Project Objective and Scope 3

1.3.1 Main Objective 3

1.3.2 Project Scope 3

1.4 Project Relevancy 3

1.5 Project Feasibility 4

CHAPTER 2 LITERATURE REVIEW 5

2.1 Load Forecasting 5

2.2 Overview of Artificial Neural Network 6

2.3 Architecture of Artificial Neural Network 7

2.3.1 Single Layer Feed-forward Network 8

2.3.2 Multilayer Feed-forward Networks 8

2.3.3 Recurrent Network 9

2.4 Learning Paradigm 10

2.4.1 Unsupervised Learning 10

2.4.2 Supervised Learning 11

2.5 Learning Algorithm 12

2.5.1 Competitive Learning Algorithm 12

2.5.2 Hebbian Learning Algorithm 12

2.5.3 Back-propagation Learning Algorithm 13

2.6 Activation Function 13

2.6.1 Sigmoid Function 13

2.6.1.1 Hyperbolic Tangent Function 14

2.6.1.2 Logistic Function (Log-Sigmoid Transfer Function) 14
2.6.2 Lavenberg-Marquardt Back-Propagation Algorithm 15
2.6.3 Identity Function 15
2.7 Improving Generalization 15
2.7.1 Early Stopping 16
2.7.2 Regularization 17
2.8 Data Collected from GDC UTP 17
2.8.1 Data Analysis 17
CHAPTER 3 METHODOLOGY 21
3.1 Research Methodology 21
3.2 Flow Chart 22
3.2.1 UTP Load Data Gathering 23
3.2.2 Forecast Model Development 23
3.2.3 UTP Data Treatment 25
3.2.3.1 Data normalization 25
3.2.3.2 Data partitioning 25
3.2.3.3 Data Arrangement 25
3.2.4 Forecast Model 26
3.2.5 MAPE 26
3.2.6 User-friendly Interface 26
3.3 Project Duration and Milestones 26
3.4 Tools Required 28
CHAPTER 4 RESULTS & DISCUSSION 29
4.1 Results 29
4.2 Discussion 45
CHAPTER 5 CONCLUSION AND RECOMMENDATIONS 49
5.1 Conclusion 49
5.2 Recommendations 50
References 51
Appendix A UTP Academic calEndar 53
Appendix B Programming 57
Appendix C 77

LIST OF TABLES

Table 1 : Previous Forecast Models 23
Table 2 : Data Partitioning for Precious Forecast Model 23
Table 3 : New Forecast Models 24
Table 4 : Data Partitioning for New Forecast Models 24
Table 5 : Gantt Chart for Project Duration and Milestones 27
Table 6 : Simulations for Monday 33
Table 7 : Simulations for Tuesday 34
Table 8 : Simulations for Wednesday 34
Table 9 : Simulations for Thursday 35
Table 10 : Simulations for Friday 35
Table 11 : Simulations for Saturday 36
Table 12 : Simulations for Sunday 36
Table 13 : Average MAPE Each Day 37
Table 14 : Simulation for Monday 41
Table 15 : Simulation for Tuesday 41
Table 16 : Simulation for Thursday 42
Table 17 : Simulation for Friday 42
Table 18 : Simulation for Saturday 43
Table 19 :Simulation for Sunday 43
Table 20 :Average MAPE for Forecast With Adjustment 44
Table 21 : Last Semester Model Data Distribution 45
Table 22 : Curret Model Data Distribution 46
Table 23 : Different between Before and After 47

LIST OF FIGURES

Figure 1 : Nonlinear model of neuron [14] 6
Figure 2 : Feed-forward with a single layer of neurons [13] 8
Figure 3 : One hidden and one output layer of multilayer Feed-forward Network 9
Figure 4 :Recurrent Networks 9
Figure 5 : Unsupervised Learning Block Diagram 10
Figure 6 : Supervised Learning Block Diagram [13] 11
Figure 7 : Tan-Sigmoid Transfer Function 14
Figure 8 : Log-Sigmoid Transfer Function 14
Figure 9 : Linear Transfer Function 15
Figure 10 : Techniques on Improving Generalization 16
Figure 11 :Total Consumption of Electricity from 2006 until 2011 17
Figure 12 : Monday Group Electricity Consumption from 2006-2011 18
Figure 13 : Tuesday Group Electricity Consumption from 2006-2011 18
Figure 14 : Wednesday Group Electricity Consumption from 2006-2011 18
Figure 15 : Thursday Group Electricity Consumption from 2006-2011........ 19
Figure 16 : Friday Group Electricity Consumption from 2006-2011 19
Figure 17 : Saturday Group Electricity Consumption from 2006-2011 19
Figure 18 : Sunday Group Electricity Consumption from 2006-2011........ 20
Figure 19 : Project Methodology 22
Figure 20 : Monday Forecast Performance........ 29
Figure 21 : Tuesday Forecast Performance........ 30
Figure 22 : Wednesday Forecast Performance........ 30
Figure 23 : Thursday Forecast Performance 31
Figure 24 : Friday Forecast Performance........ 31
Figure 25 : Saturday Forecast Performance 32
Figure 26 : Sunday Forecast Performance 32
Figure 27 : Monday Load Forecast with Adjustment 37
Figure 28 : Tuesday Load Forecast with Adjustment 38
Figure 29 : Thursday Load Forecast with Adjustment........ 38
Figure 30 : Friday Load Forecast with Adjustment 39
Figure 31 : Saturday Load Forecast with Adjustment 39

Figure 32 : Sunday Load Forecast with Adjustment.. 40

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

Electricity in Universiti Teknologi Petronas is supplied by two sources. The main electricity source is supplied by the Gas District Cooling (GDC) while the backup supply is provided by Tenaga Nasional Berhad (TNB) which is the largest power utility company in Malaysia. To supply the needs of UTP electricity, GDC have two gas turbine generators which generates 4.2 MW for each turbine generator. The activities in UTP give a major impact to the electricity consumption throughout the years. As the population of students in UTP is not the same for each semester, the challenge for the GDC plant to balance the electricity supply according to the load used by the operations in UTP.

The aim for this project is to forecast the university electricity demand for a short term interval by using Artificial Neural Network (ANN). The usage of ANN is followed by few goals which are to help GDC for pre-planned scheduling; to determine the suitable time for the maintenance of the plant and giving the plant a reference or guidance throughout the years for the operation. This method will use the previous load demand data which acts as the input and the output will be the future load demand data. Then, the accuracy of the output will be determined by comparing with the actual data which gives us the mean absolute percentage error (MAPE).

1.2 Problem Statement

1.2.1 Problem Identification

Load forecasting is very important in planning for electricity resource management. As for that, conventional methods of load forecasting have been used in several companies in Malaysia. The relationship between load power and factors influencing load power is non-linear, which give difficulty to identify the non-linearity of them by using the conventional method [1]. The non-linearity factors were contributed by the change in the surrounding such as the weather and the temperature. In this project, the main subject is GDC UTP and the factors leads to the complexity of the electric demand are temperature and weather surrounding the subject.

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]. Because of that, a precise load forecasting need to be done to ensure the effect from the above factors will not affect the precision of our load forecast.

1.2.2 Significance of Project

In this project, artificial neural network is used in finding the short time load forecasting for the GDC load demand which usually varies from days to a weeks. The load demand varied by the effect from numerous factor which usually a seasonal effect [3]. By using ANN, GDC can improve their management by scheduling the maintenance for the plant operation. The usage of ANN is held for a short term load forecasting (STLF) so that all possible input will be consider to ensure that the error of the forecasted value will be as minimum as possible compared to the real value.

1.3 Project Objective and Scope

1.3.1 Main Objective

For every forecasting, the main objective is to get the lowest minimum average percentage error (MAPE). In this project, the subject which will be used is GDC UTP to find the lowest MAPE. Using Artificial Neural Network (ANN), the targeted MAPE will be used to measure the performance of the model used.

The sub objective of this project is listed as below:-

i. To outline the importance of load forecasting
ii. To outline the fundamental of Artificial Neural Network
iii. To construct a forecast load demand for GDC UTP with the suggested model
iv. To present the load forecasting models in user-friendly environment

1.3.2 Project Scope

This project will be started with a literature review which related to the load forecasting and the theory of ANN. Next, data gathering will be made from GDC UTP which then will be sorted and used as the input for the load forecasting model design. The forecasting design model will be created by using the MATLAB software. STLF is used to predict the forecasted data for one week ahead. Further testing will be carried out to design the load prediction model with the highest accuracy.

1.4 Project Relevancy

This project gives a prediction for the future electricity demand in UTP. This model will help to save the electricity generation by GDC from being wasted as the production will follow the predicted demand value for the day which has been predicted. Hence, this will ensure continuous and reliability of the supply to the consumer, UTP. Then, the SLTF also will allow an optimization of the load – generate ratio which will reduce the possibilities of the turbine generator to break down.

1.5 Project Feasibility

The project is a two semester project. In two semesters, there are three areas which are research, development and also improvement of the model itself. The objective is to reduce the MAPE compared to the previous project. The input will be gathered from GDC UTP and MATLAB will still be used as the tools to develop the ANN algorithm. Both system testing and implementation will be using this software. From this, it is assured that the project will be feasible to be carried out within the time given.

CHAPTER 2
LITERATURE REVIEW

2.1 Load Forecasting

Load forecasting is a very useful method in the last few decades. As the consumption of electricity nowadays rises, there have been several technique develop within this few decades to improving one another. From here, we can say that load forecast is needed in the industry especially power system industry. Load forecasting can be divided into three categories which are short term, medium term and long term [4]. Short term load forecasting (STLF) means the load forecasted usually ranges from one hour to a week [5]. This shows a potential for economic saving and providing a reliable operation of power system [6]. Medium term load forecasts (MTLF) usually related to some additional influences – demographic and economic influences [2]. Medium term load forecasting means the load forecasting usually from a week to a year [7]. Long term load forecasting (LTLF) give a forecast of more than a year [7] which is very useful for a longer duration of planning operations.

STLF is still an important tool in the day to day operation of utility system. Forecasting for Up to now, it is still the main focus for forecasting [8]. A good forecast model will surely held GDC to schedule generators operation either to start, stop and do the maintenance works [4]. It has been proved that the improvement of load forecast gives a positive impact on the cost of operation and the system security [9]. The main application of STLF function is to drive the scheduling functions which give us the most economic commitment for generation sources consistent with the reliability requirements, operational constraints and policies, physical, environmental and equipment limitation [5].

These forecast, integrated with the information about scheduled wheeling transaction, transmission avail-ability, spot market energy pricing are used to determine the best strategy for the utility resources [10]. If STLF is applied to the system security assessment problem, the information to detect many vulnerable situations in advance can be provided [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 Overview of Artificial Neural Network

Artificial neural network are mathematical tools originally inspired by the way the human brain processes information [12]. Neural networks are highly interconnected simple processing units designed in a way to model how the human brain performs particular tasks [13]. ANN traces previous load patterns and predicts (i.e. extrapolates) a load pattern using recent load data [11]. Processing elements in an ANN are also known as neurons. Each neuron can have multiple channel inputs while there can be only one output which is represented in Figure 1. Inputs to a neuron could be from external stimuli or from output of other neurons [14].

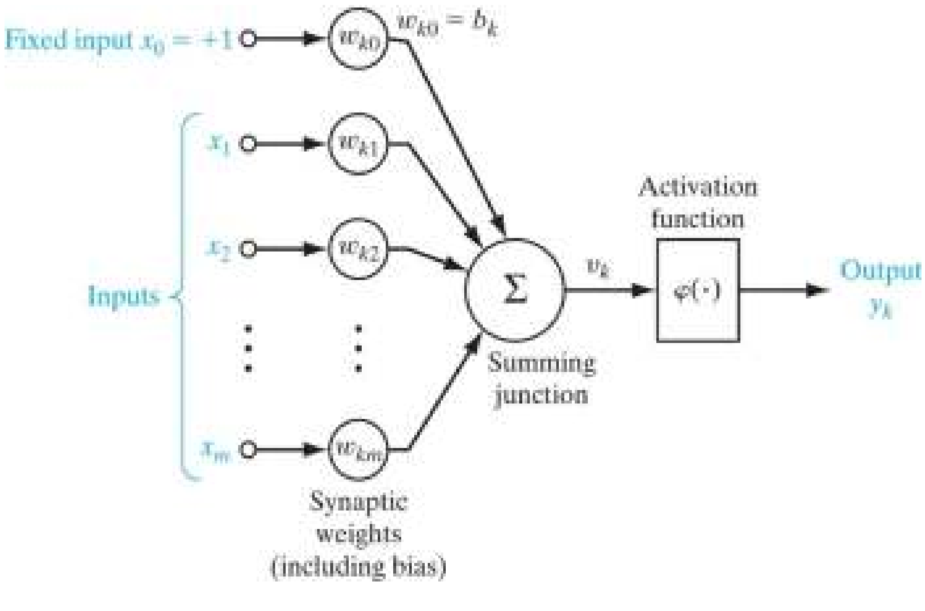

Figure 1 : Nonlinear model of neuron [14]

From the input, the data will pass through a transfer function which is linear, sigmoid or hyperbolic tangent before we get the output response from the neuron. ANN have been build with three basic building block which are Network Architecture (connection between neurons), Training or Learning (determine the weight on the connections) and Activation function. These basic building blocks will be explained in the next part of the report. The basic building blocks of the ANN are such as Network Architecture (connection between neurons). These basics are Training or Learning (determining weights on the connections) and Activation function.

2.3 Architecture of Artificial Neural Network

Artificial Neural Network Architecture is a word which used to describe the Neural Network configuration, with other attributes of the Neural Network such as the learning rule, activation function, learning and momentum factors [15].

Network with interconnections that do not form any loops are called feed-forward [16]. Recurrent or non feed-forward networks in which 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 [13]. Three layered, feed-forward system with back propagation is the most popular algorithm [18].

Generally, the network architecture can be defined into three main classes; single later feed-forward network, multilayer feed-forward network and recurrent network.

2.3.1 Single Layer Feed-forward Network

This network consist only a single layer of weighted interconnections. The input can be connected fully to the output units [19]. This network is strictly a feed-forward or acyclic type [13]. Figure 2 exhibit the feed forward with single later of neuron model.

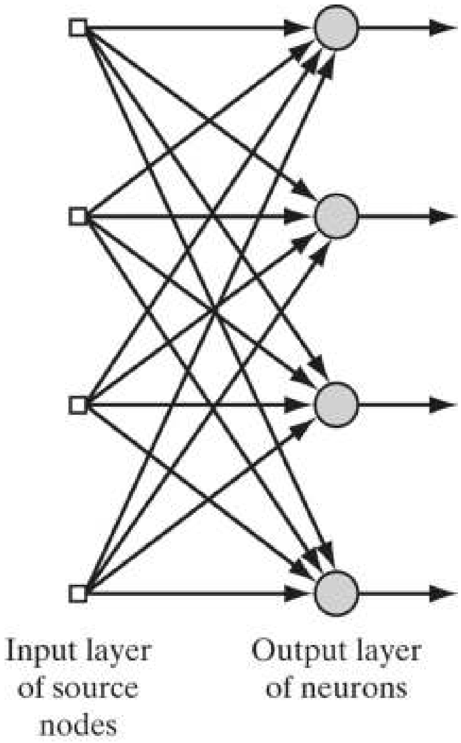

Figure 2 : Feed-forward with a single layer of neurons [13]

2.3.2 Multilayer Feed-forward Networks

This type of network 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 hidden neurons functioned as intervene between the external input and the network output in some useful manner [13]. More complicated problems can be solved by increasing the hidden layers in the network. The model for multilayer feed-forward network with one hidden and one output layer is shown in Figure 3.

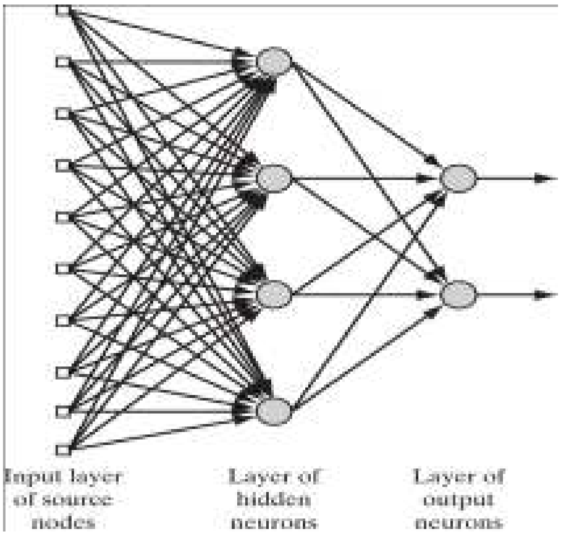

Figure 3 : One hidden and one output layer of multilayer Feed-forward Network

2.3.3 Recurrent Network

For this type of network, all units is connected to all other units and every unit is both input and an output [19]. Figure 4 explains the operation of the recurrent networks. The neurons have self-feedback loops which ensure the output is fed back becoming the input. The presence of feedback loops has a profound impact on the learning capability of the network and on its performance [13].

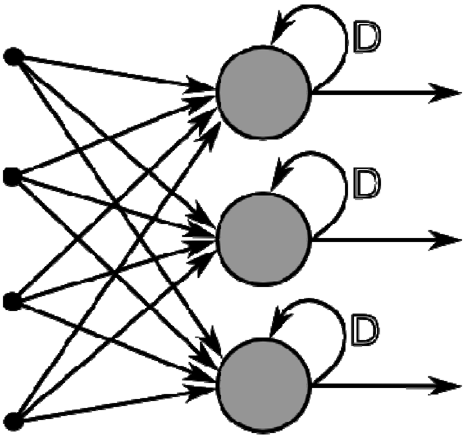

Figure 4 :Recurrent Networks

2.4 Learning Paradigm

Learning Paradigm refers to a model of the environment in which the neutral network operates [13]. Learning 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 [20]. The success of this project configuration dwells in the fact that it can learn from retrospective information in a supervised learning [17]. In general, there are two types of learning paradigms which is shown as follows:

2.4.1 Unsupervised Learning

Unsupervised training is defined as the training input vector which the target output is not known [19]. For this type of learning process, there will be no teacher to supervise the learning process as supervised learning. This will clearly explain in Figure 5. Modification of the weight by network will be done to ensure the most similar input vector is assigned to the same output unit. Provision is made for a task independent measure of the quality of representation that the network is required to learn because there is no teacher in this type of network. Free parameters of the network are optimized with respect to that measure [13]. The ability to form internal representations for encoding features of the input and creating new classes automatically will be develop once the network has become tuned to the statistical regularities of the input data [21].

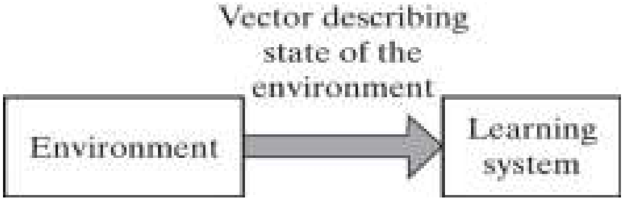

Figure 5 : Unsupervised Learning Block Diagram

2.4.2 Supervised Learning

Supervised learning is a process of providing the network with a series of inputs sample and comparing the output with the expected responses which the training continues until the network is able to provide the expected response [18]. Supervised learning also trained the network by using historical data derived from the system, which the relationship between input and output will be determined [17]. This can be compared to learning with a teacher. Teacher will have the knowledge of the environment that is represented by a set of input-output examples. The teacher refers to the neural network which now 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 [13]. Based on Figure 6, learning system will deliver its actual response with the objective is 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.

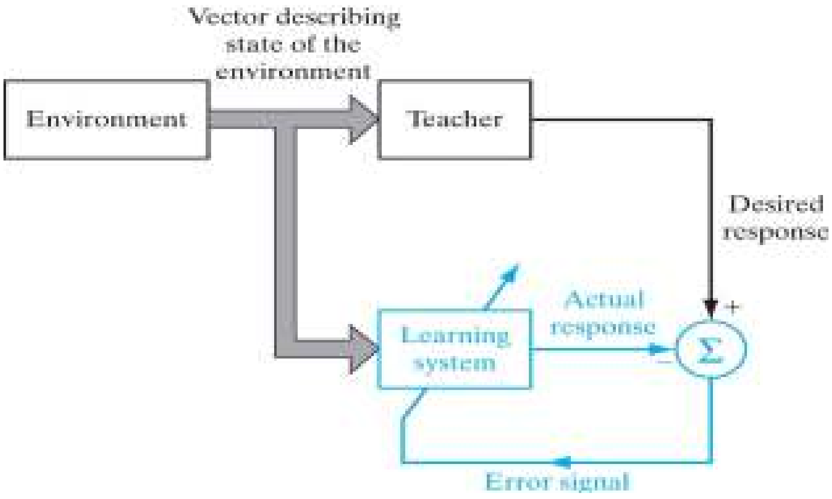

Figure 6 : Supervised Learning Block Diagram [13]

2.5 Learning Algorithm

The objective of learning algorithm is to achieve the expected output which also called training a network by modifying the weights in the connections between the networks layer [19]. 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 a way in which the adjustment to a synaptic weight of neuron is formulated. There are various learning rules that available such as:

2.5.1 Competitive Learning Algorithm

The output of the neural network neurons competes among themselves to become active. This feature that makes competitive learning highly suited to discover statically salient features that may be use used to clarify a set of input patter [13]. 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 [19].

2.5.2 Hebbian Learning Algorithm

This learning can be called correlational learning. The rules of this learning are:

i- If two neurons on either side of a synapse (interconnection) 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 [13]

This learning rule represents a purely feed-forward, unsupervised learning [19].

2.5.3 Back-propagation Learning 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 the target output, 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 [22]. The learning rate is proportion to error gradient by which the weight should be adjusted. Larger values can give a faster converge 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 [23].

2.6 Activation Function

Neuron's output response is calculated by the activation function [19]. Artificial Neural Network behaviour depends on both the weights and the input-output function (Activation Function) that is specified for the units [23]. To choose 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 [24]. A configuration consisting of a one hidden layer that uses a hyperbolic tangent sigmoid transfer function and the output 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]. The most common used functions is being described in the following sections:

2.6.1 Sigmoid Function

A sigmoid function which has s-shaped graph is the most common form of activation function used in ANN. It is defined as strictly increasing function that exhibits a grateful balance between linear and non-linear behaviour [14]. Two types of sigmoid functions which been used in ANN is defined:

2.6.1.1 Hyperbolic Tangent Function

For the corresponding form of a sigmoid function we may use the hyperbolic tangent function, defined by equation 1:

$$\varphi(v) = \tanh(v) \tag{1}$$

The function generates output between -1 and +1 as the neuron's net input goes from negative infinity to positive infinity. Figure 7 shows the tan sigmoid transfer function.

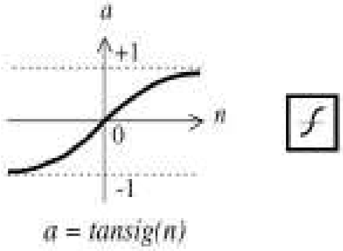

Figure 7 : Tan-Sigmoid Transfer Function

2.6.1.2 Logistic Function (Log-Sigmoid Transfer Function)

An example of the sigmoid function is the logistic function which is defined as equation 2:

$$\varphi(v) = \frac{1}{1+\exp{(-av)}} \tag{2}$$

Where a is the slope parameter of the sigmoid function. The sigmoid transfer function showed in Figure 8 takes the input that have value between plus and minus infinity, and squashes the output into the range 0 to 1.

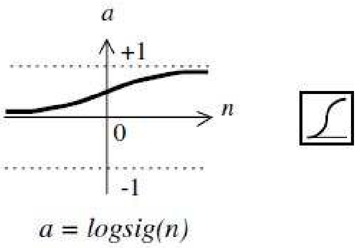

Figure 8 : Log-Sigmoid Transfer Function

2.6.2 Lavenberg-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] [27].

2.6.3 Identity Function

This is a linear transfer function where

$$f(x) = x \tag{3}$$

Linear transfer function which was shown in Figure 9 explain the relationship between input and output which are the same when the input is being fed into the network.

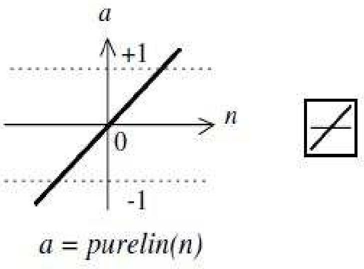

Figure 9 : Linear Transfer Function

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 explains shows us the techniques on improving generalization in the system.

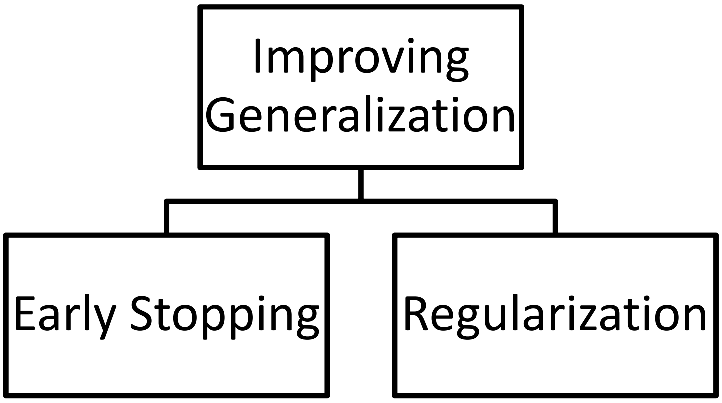

Figure 10 : Techniques on Improving Generalization

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 [28].

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 Data Collected from GDC UTP

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 2011. The data for all the data is presented in Figure 11.

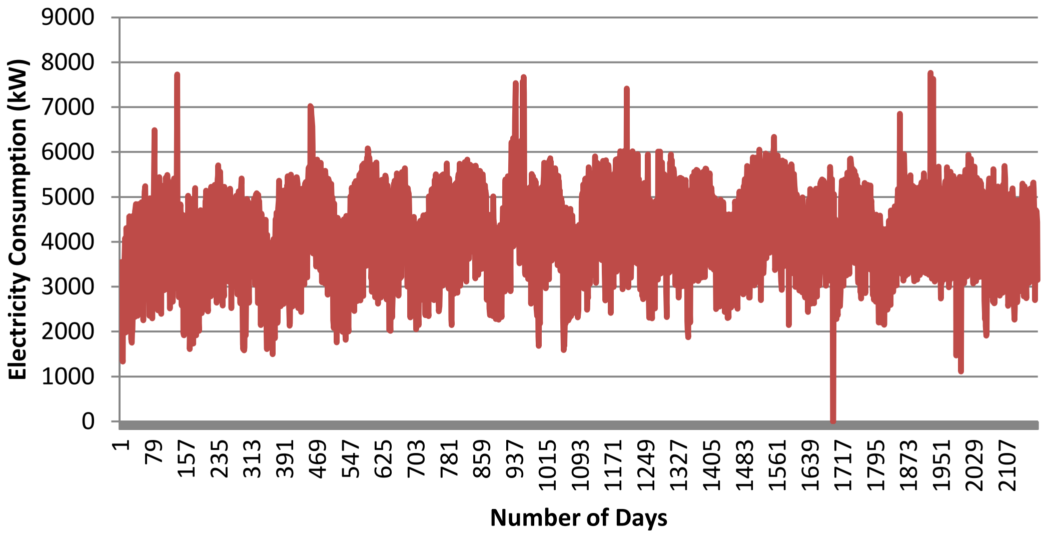

Figure 11 :Total Consumption of Electricity from 2006 until 2011

2.8.1 Data Analysis

The data gathered are based on daily interval data. The data have been categorized into seven groups of similar days which start from Monday up till Sunday. Figure 12 to 18 will show us the trending for each groups of the days.

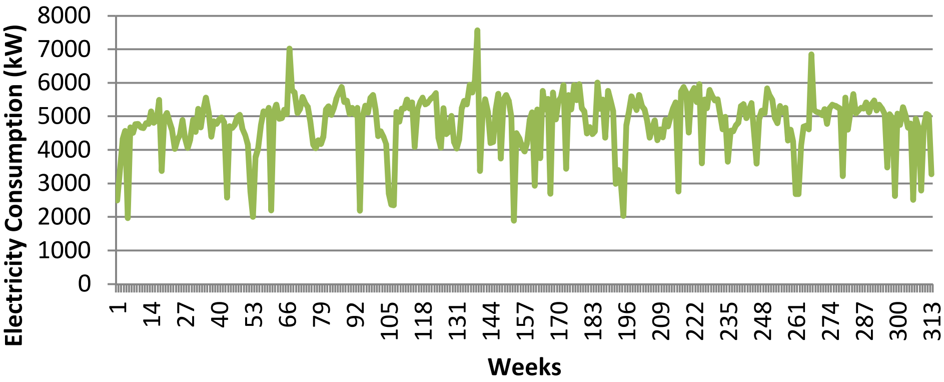

Figure 12 : Monday Group Electricity Consumption from 2006-2011

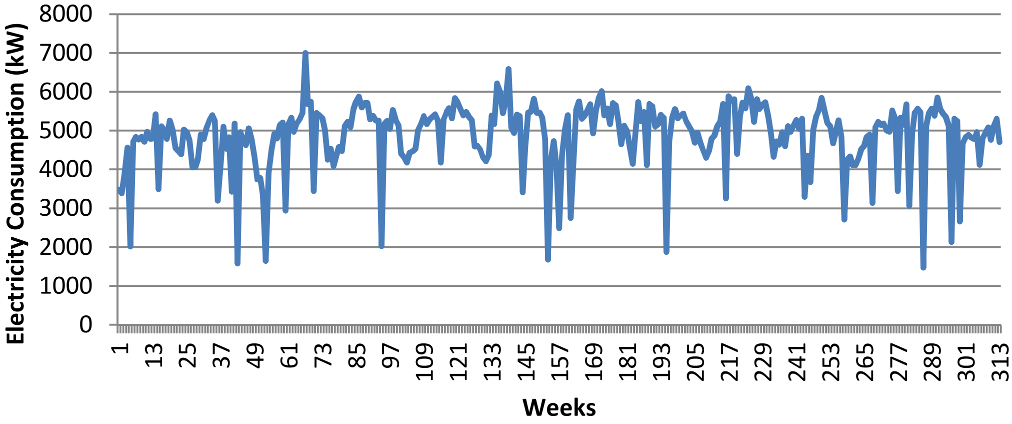

Figure 13 : Tuesday Group Electricity Consumption from 2006-2011

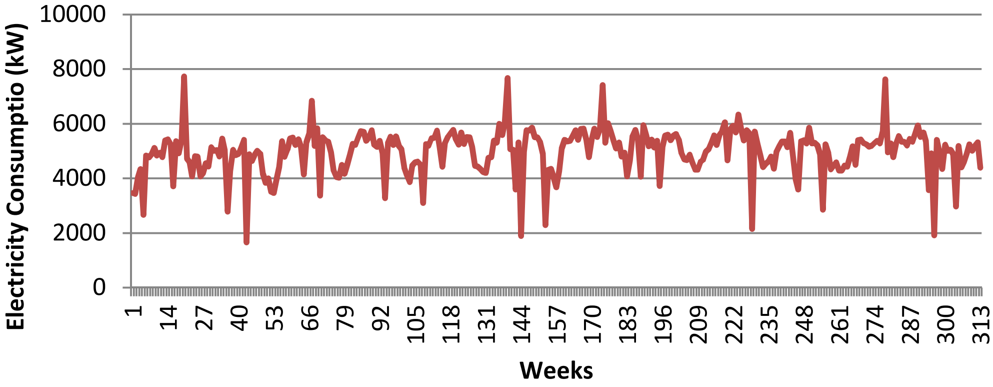

Figure 14 : Wednesday Group Electricity Consumption from 2006-2011

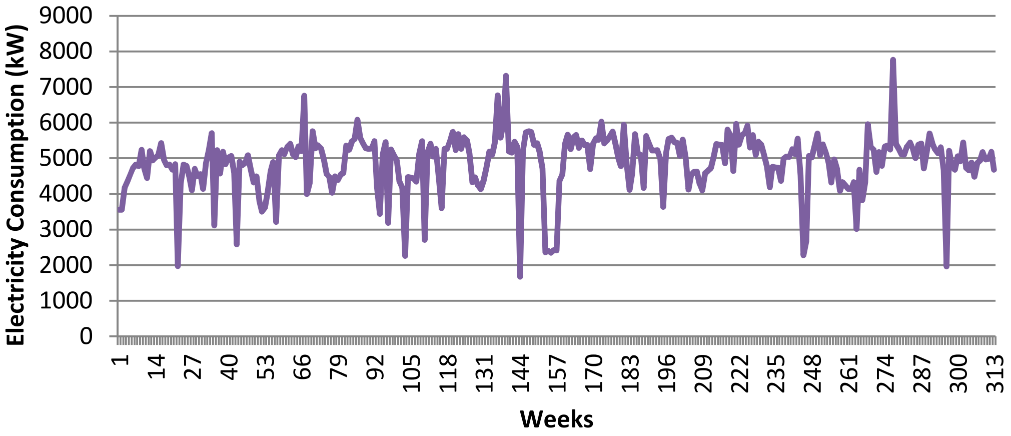

Figure 15 : Thursday Group Electricity Consumption from 2006-2011

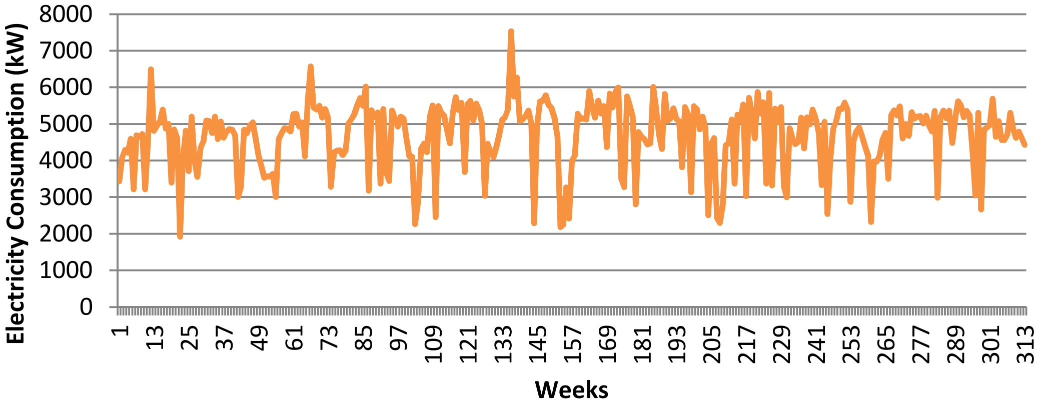

Figure 16 : Friday Group Electricity Consumption from 2006-2011

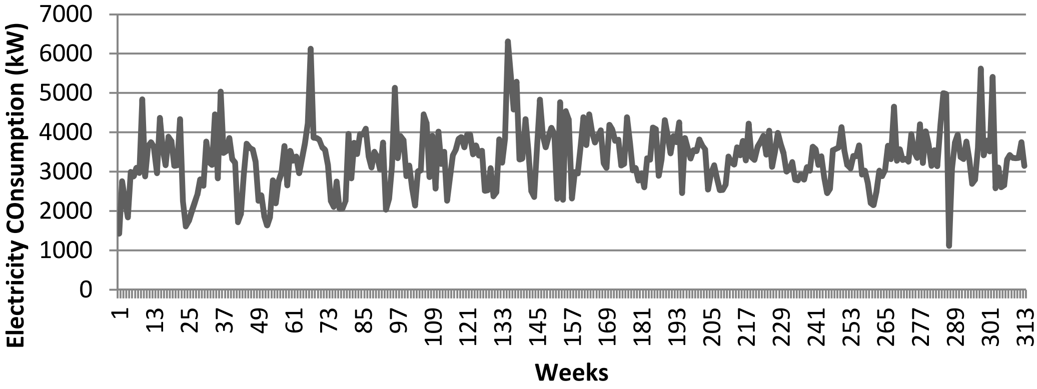

Figure 17 : Saturday Group Electricity Consumption from 2006-2011

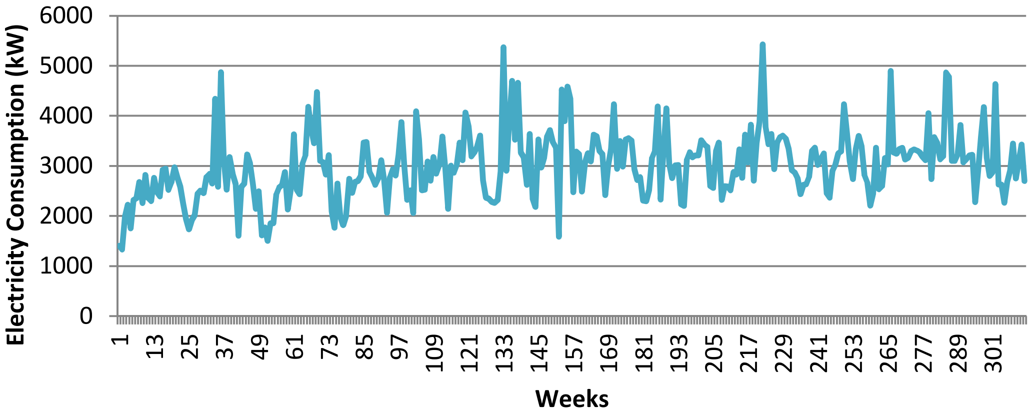

Figure 18 : Sunday Group Electricity Consumption from 2006-2011

From the group trending shown by Figure 12 up to Figure 18, electricity consumption in UTP has been analysed. The usage of the electricity varied from day to day but still is fixed in certain range for each group. This is because of the minimum and maximum operation value which the turbine can supply. With this trending, the model of the neural network programming can be determined. The trending also shows us that there are ‘outliers’ which shows us there are data. ‘Outliers’ is mean by the data which is out of the minimum or maximum range that have been set in the model. By this, each and every date of need to be keep track to make sure that the electricity consumption variation happens for a reason.

CHAPTER 3
METHODOLOGY

3.1 Research Methodology

In order to achieve the main objective of this project, the goals for the four 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 issue of 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 shown in Figure 19 explains the methodology in executing the project:

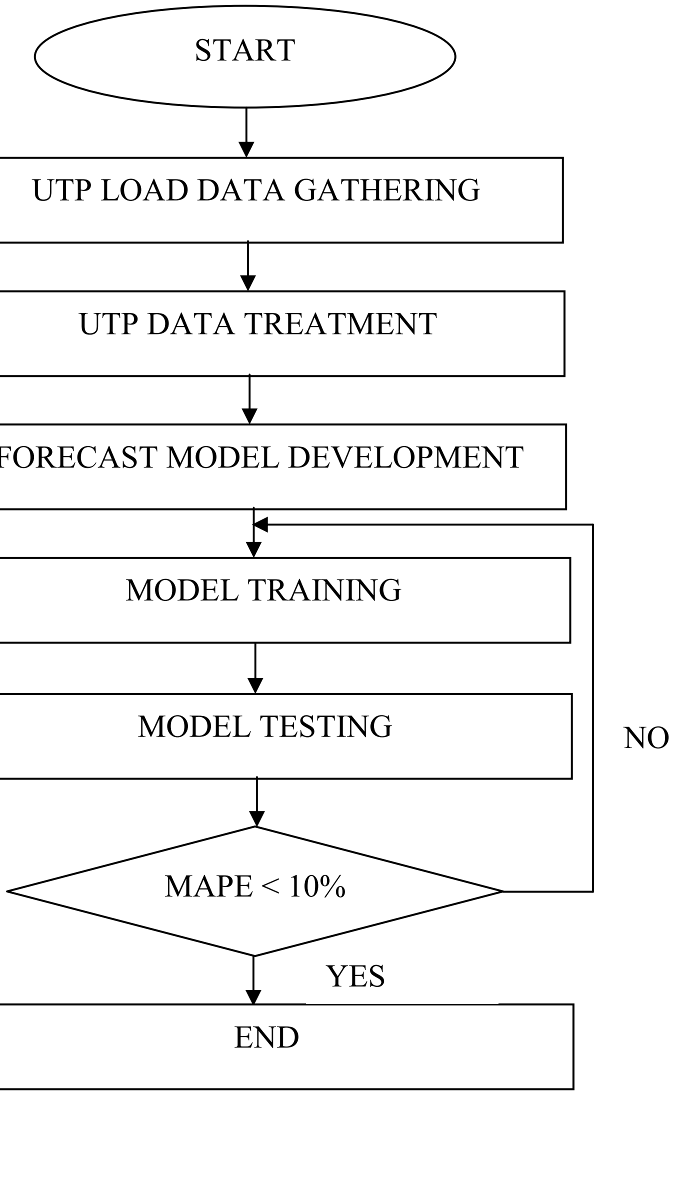

Figure 19 : Project Methodology

3.2.1 UTP Load Data Gathering

UTP electrical usages have been gathered from GDC UTP ranging from 1st January 2006 up till 31st December 2011. The data have been categorised into 7 groups of day which is Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday without categorizing the data into semester OFF or semester ON.

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 depicted in the tables 6 to 7:

Table 1 : Previous Forecast Models

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

Table 2 : Data Partitioning for Precious Forecast Model

Model	Training	Testing	Validation
Model 1	70 %	30 %	X
Model 2	40 %	30 %	30 %

Based on the previous model, the new models for the different group of days have been developing. The previous model have been develop to make sure that if the data is about the same or if the data are consider to be in range, it will use model 1 and if the data are consider to be out of range, it will use model 2. Based on this model, the model is been develop. The new forecast model is been shown below.

Table 3 : New Forecast Models

Model	Day	Forecasting Duration
Model 1	Monday	7 Weeks of Days
Model 2	Tuesday	7 Weeks of Days
Model 3	Wednesday	7 Weeks of Days
Model 4	Thursday	7 Weeks of Days
Model 5	Friday	7 Weeks of Days
Model 6	Saturday	7 Weeks of Days
Model 7	Sunday	7 Weeks of Days

Table 4 : Data Partitioning for New Forecast Models

Model	Training	Testing	Validation
Model 1	70 %	30 %	X
Model 2	70 %	30 %	X
Model 3	70 %	30 %	X
Model 4	70 %	30 %	X
Model 5	70 %	30 %	X
Model 6	70 %	30 %	X
Model 7	70 %	30 %	X

The data have been partitioned for 70 % training data and 30 % testing data. Based on the previous model, the method has produced a good result. The method has been tested on the new model and it also give a good result for the new grouping method. The grouping method have been choose to accommodate 7 weeks of days because UTP nowadays is running for a trimester which is now having 14 weeks of lecture days. Based on this method, we can forecast at which weeks we can do maintenance for GDC.

3.2.3 UTP Data Treatment

The data treatment will be done so that a stable model is created. The data consist of data normalization and data partitioning.

3.2.3.1 Data normalization

We are using transfer function of tansig as the input model hence; the data range development that we can use is only -1 and +1. The data been normalized by dividing the input by 10000. When we get the forecast value, the data will be converted back by multiplying it by 10000. The usage of this idea is to avoid lagging in data processing by MATLAB.

3.2.3.2 Data partitioning

70 % of the data will be use for training and 30 % will be use for testing. The data that we have been using before is only 192 data for each day that we forecasted. As the new data arrangement is added into the forecast and the data is updated until end of 2011, the usage of data increase to 274 data for each day.

3.2.3.3 Data Arrangement

The arrangement of the data is done so that the data recognized 7 weeks of interval for the data to forecast. This will ensure that for each half of the semester, the data will compare to another half of semester. This will provide the ANN with organized information which helps the system itself to predict accurately.

3.2.4 Forecast Model

The forecast model will develop so that the program will predict the load for one week ahead. The time zone for the forecasted data will be selected later.

3.2.5 MAPE

This error will be calculated once the forecasted data is obtained. The program will use the formula below to check the MAPE for the program. If the error is above 10%, the program will redo the steps above.

$$Absolute\ Error = \frac{|Forecast\ Load - Actual\ Load|}{Actual\ Load} \times 100\% \tag{4}$$

3.2.6 User-friendly Interface

When the system is stable and the forecasted value's MAPE is below the selected error, a user-friendly interface will be added to ensure easiness to use the Artificial Neural Network.

3.3 Project Duration and Milestones

To monitor the progress of this project, a Gantt chart consists of two semester duration had been constructed. For the progress of Final Year Project 1, the progress is been displayed in Table 3 and for Final Year Project 2, the progress is displayed in Table 5.

Table 5 : Gantt Chart for Project Duration and Milestones

ACTIVITIES	FINAL YEAR PROJECT 1														FINAL YEAR PROJECT 2													
	WEEK NO.														WEEK NO.													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Choose topic	■																											
Study the topic		■	■																									
Literature Review				■	■																							
Extended Proposal																												
Run and analyze the simulation based on example						■	■																					
Gather data from GDC UTP								■	■																			
Study and analyze data										■	■	■																
Draft Report																												
Final Report																												
Design and develop forecast model													■	■	■	■	■	■										
Forecast & Simulation																			■	■	■	■	■	■	■			
Results and Conclusion																										■	■	
Final Presentation																												■
Dissertation																■	■	■	■	■	■	■	■	■	■	■	■	■
Final Report (Hard bound)																												

3.4 Tools Required

The tools that is required for developing the model is MATLAB R2010a. MATLAB is an ideal tool for working with ANN as it is proven to be highly efficient in performing the calculations of vectors and matrices. Neural Network Toolbox which is included in MATLAB will provide useful tools for working with ANN.

To create a user friendly interface, the tools that we require is Microsoft Visual Basic 2010 Express. This programming software provides us with a programming language which can be transformed into an interface for the easement of the user.

CHAPTER 4
RESULTS & DISCUSSION

4.1 Results

In the experiment, data from semester January 2012 as the input of the forecast model and then the output will be compared with the data from semester Mei 2012. The usage of the data from each the 2012 semester is because we want to increase the data input for our training. As we know, the data provided for last semester forecast is not enough creating the forecast model to get high value of MAPE which is out of our target range. The result of the forecast for each day will be shown in Figure 20 to Figure 26:-

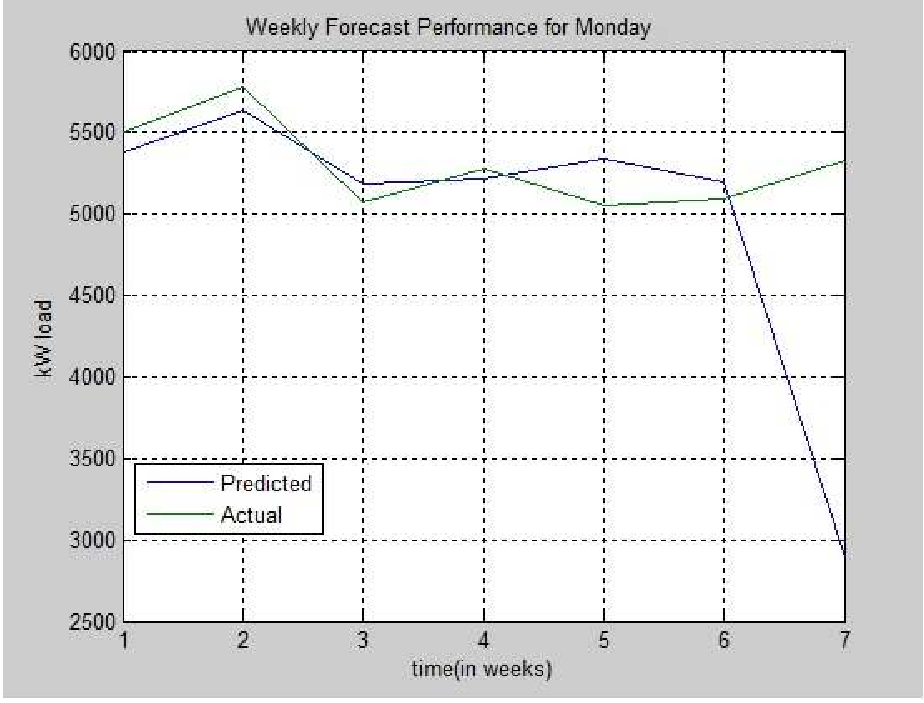

Figure 20 : Monday Forecast Performance

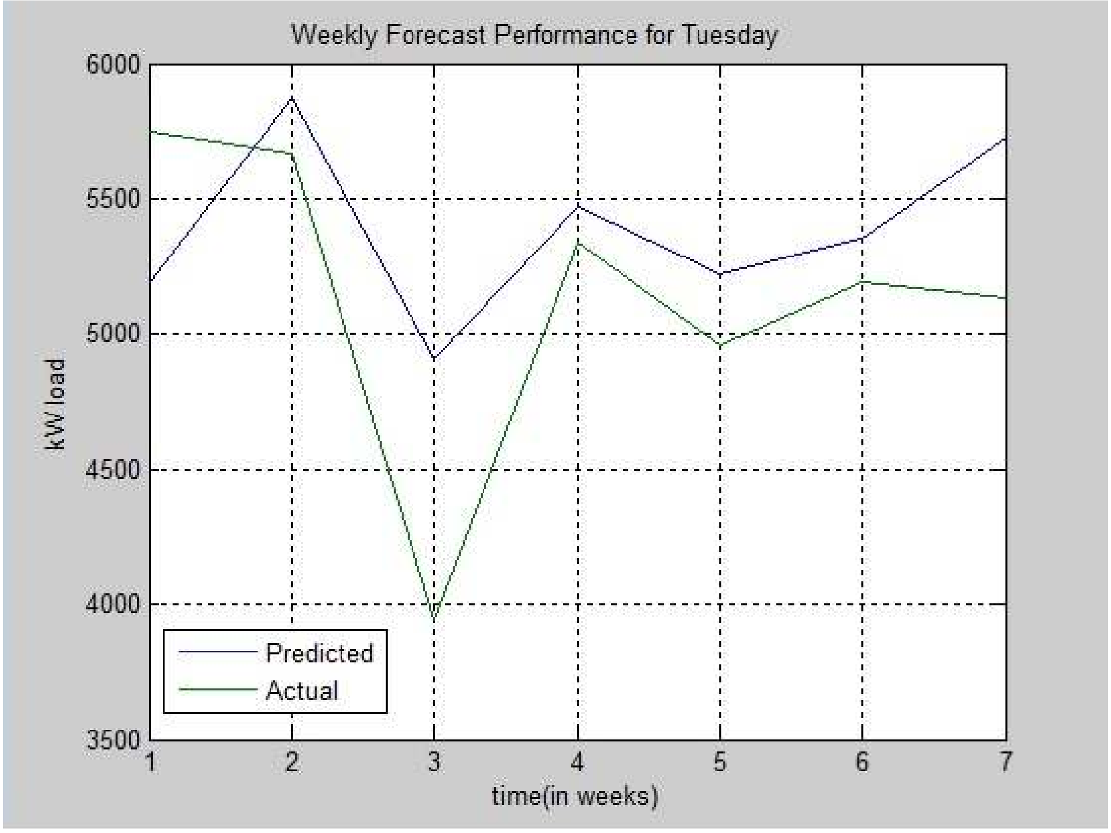

Figure 21 : Tuesday Forecast Performance

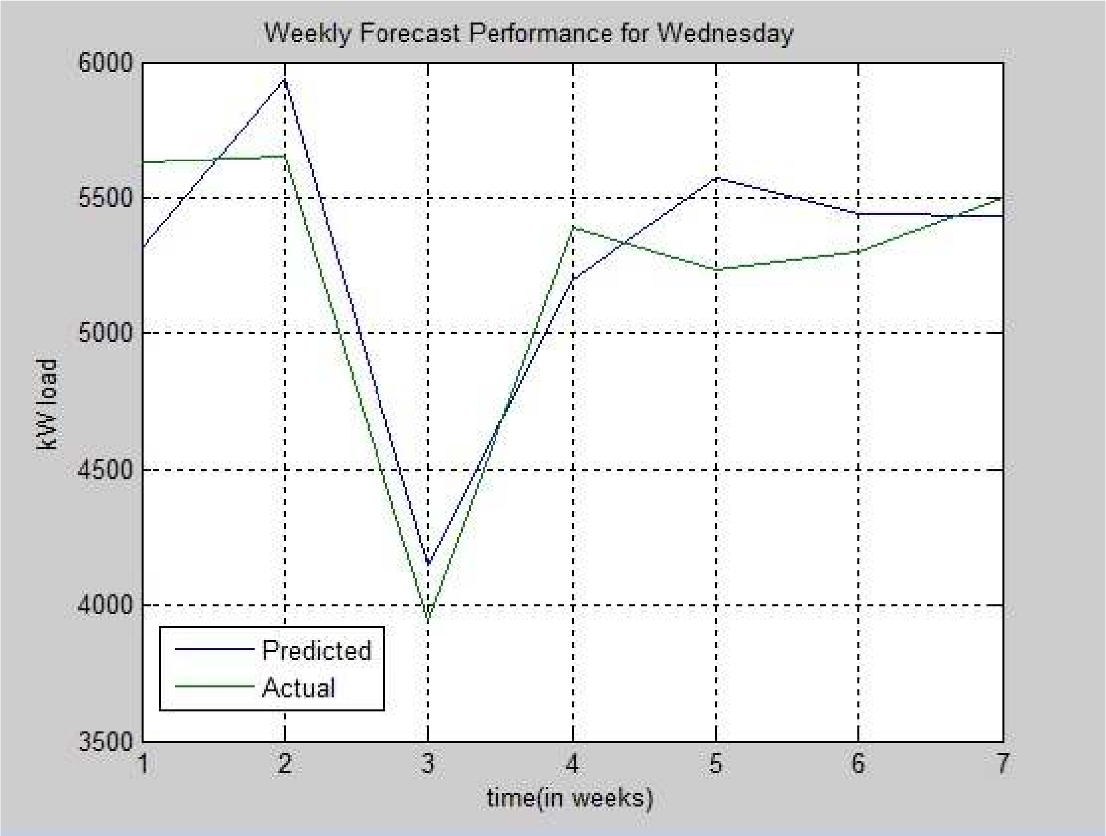

Figure 22 : Wednesday Forecast Performance

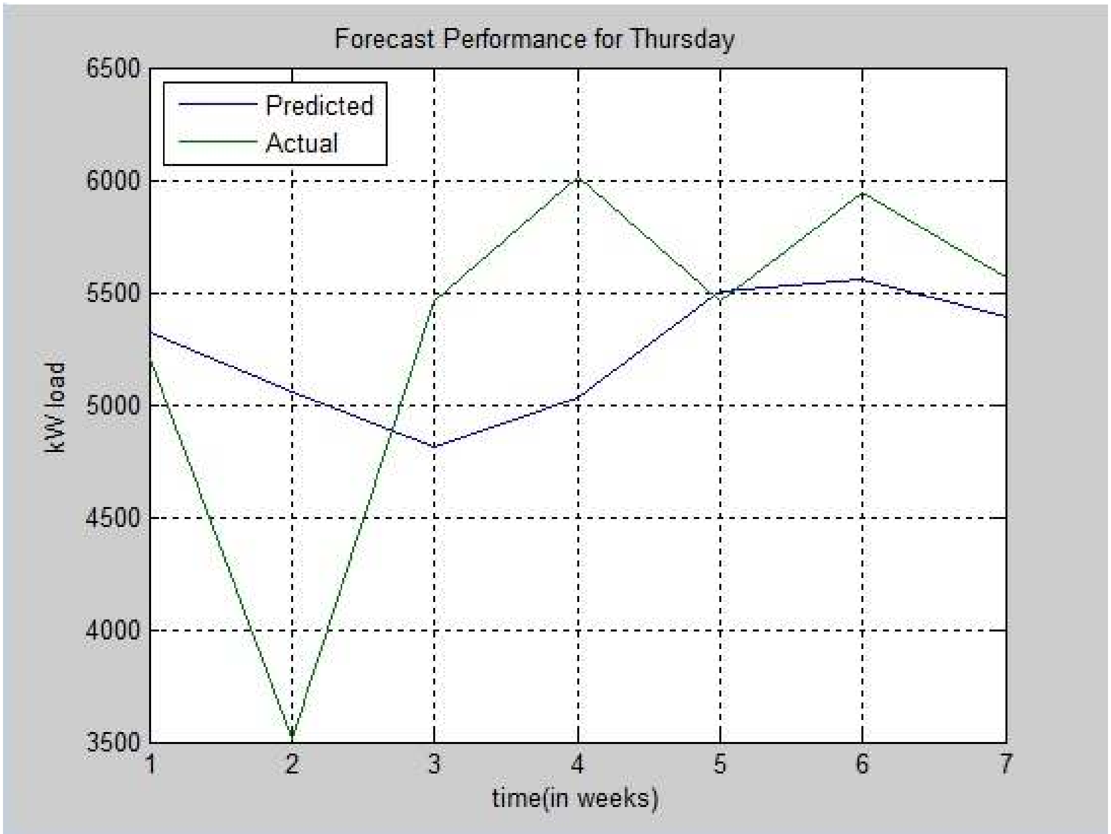

Figure 23 : Thursday Forecast Performance

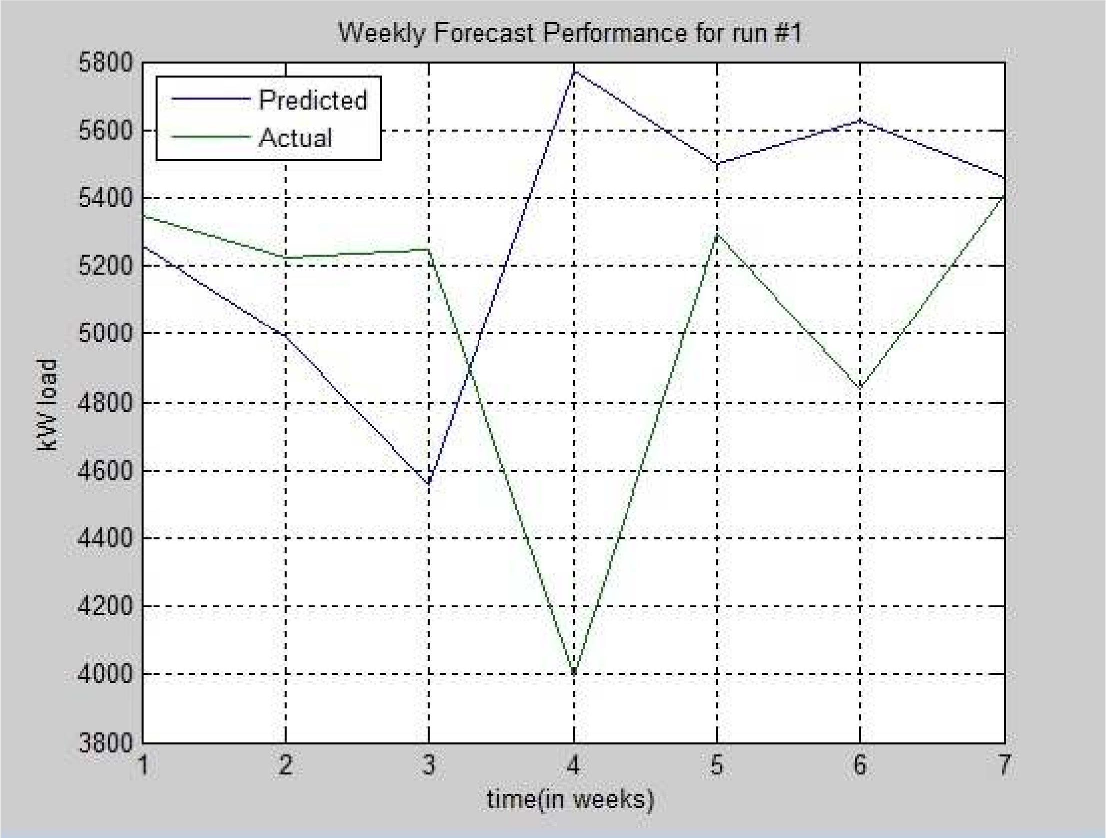

Figure 24 : Friday Forecast Performance

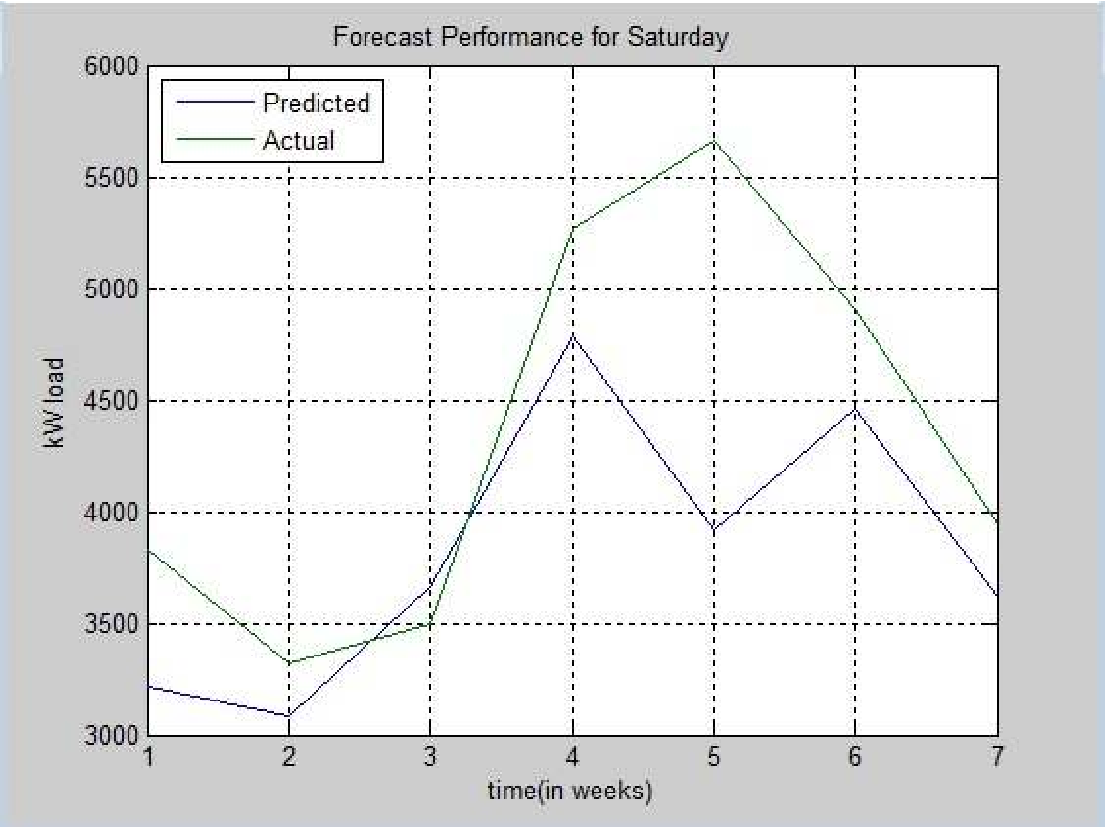

Figure 25 : Saturday Forecast Performance

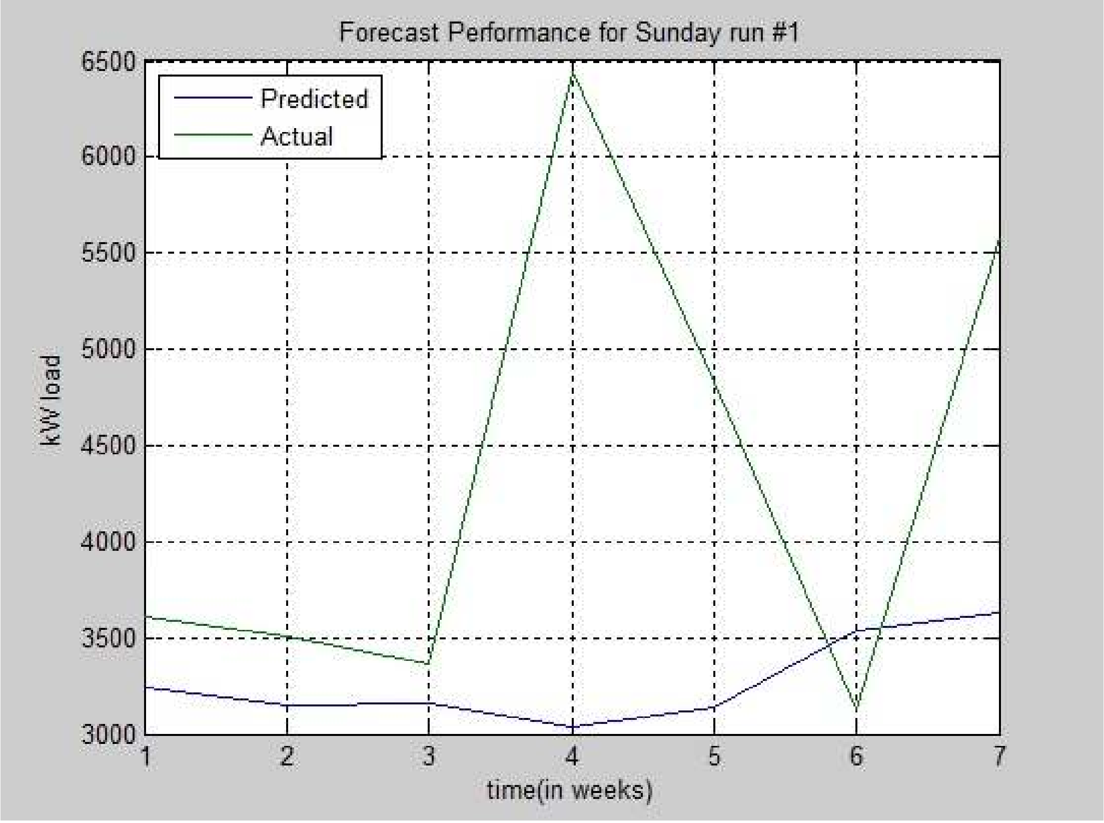

Figure 26 : Sunday Forecast Performance

Each day forecast has been run for 10 times. The reason the data have been simulated for 10 times is for us to get the average MAPE that been projected by our forecasting model.

Table 6 : Simulations for Monday

Simulation	MAPE Value (%)
1	11.996
2	12.417
3	9.866
4	7.337
5	7.863
6	7.503
7	7.895
8	12.689
9	9.537
10	9.313
Average MAPE value (%)	9.642

Table 7 : Simulations for Tuesday

Simulation	MAPE Value (%)
1	9.236
2	8.007
3	6.236
4	7.028
5	9.042
6	12.298
7	8.670
8	8.323
9	11.515
10	10.011
Average MAPE value (%)	9.037

Table 8 : Simulations for Wednesday

Simulation	MAPE Value (%)
1	7.738
2	5.101
3	8.332
4	4.364
5	4.947
6	7.642
7	7.309
8	7.616
9	6.335
10	6.350
Average MAPE value (%)	6.573

Table 9 : Simulations for Thursday

Simulation	MAPE Value (%)
1	10.415
2	10.586
3	10.430
4	10.761
5	10.526
6	10.366
7	10.582
8	10.718
9	10.693
10	10.665
Average MAPE value (%)	10.570

Table 10 : Simulations for Friday

Simulation	MAPE Value (%)
1	10.801
2	10.196
3	10.226
4	10.151
5	9.663
6	10.780
7	10.118
8	10.526
9	10.944
10	10.671
Average MAPE value (%)	10.4079

Table 11 : Simulations for Saturday

Simulation	MAPE Value (%)
1	13.426
2	14.106
3	13.025
4	14.168
5	12.856
6	15.115
7	13.340
8	12.985
9	14.329
10	13.851
Average MAPE value (%)	13.72

Table 12 : Simulations for Sunday

Simulation	MAPE Value (%)
1	23.75
2	22.797
3	28.738
4	22.503
5	16.797
6	18.277
7	19.657
8	20.795
9	29.167
10	26.890
Average MAPE value (%)	22.93

Table 13 : Average MAPE Each Day

Day	MAPE Value (%)
Monday	9.642
Tuesday	9.037
Wednesday	6.573
Thursday	10.570
Friday	10.408
Saturday	13.720
Sunday	22.930

The average MAPE has been taken as the table 13 above. The lowest MAPE that have been projected is in Wednesday and the highest is in Sunday. The result before have told us that the lowest MAPE is on Thursday and the highest is still on Sunday. From here, some improvement has been made to ensure that the MAPE value will be lower from what have been predicted on table 13. The data have been sorted back and each of the single date which has represented a holiday in the data has been normalized. The result of the new input data is shown in Figure 27.

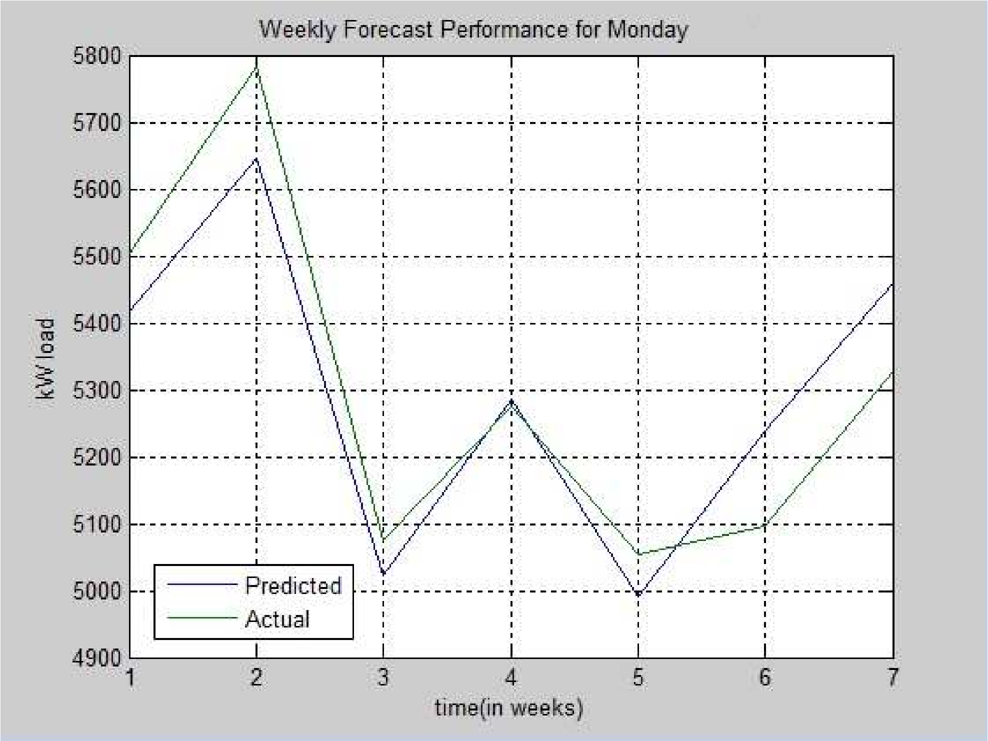

Figure 27 : Monday Load Forecast with Adjustment

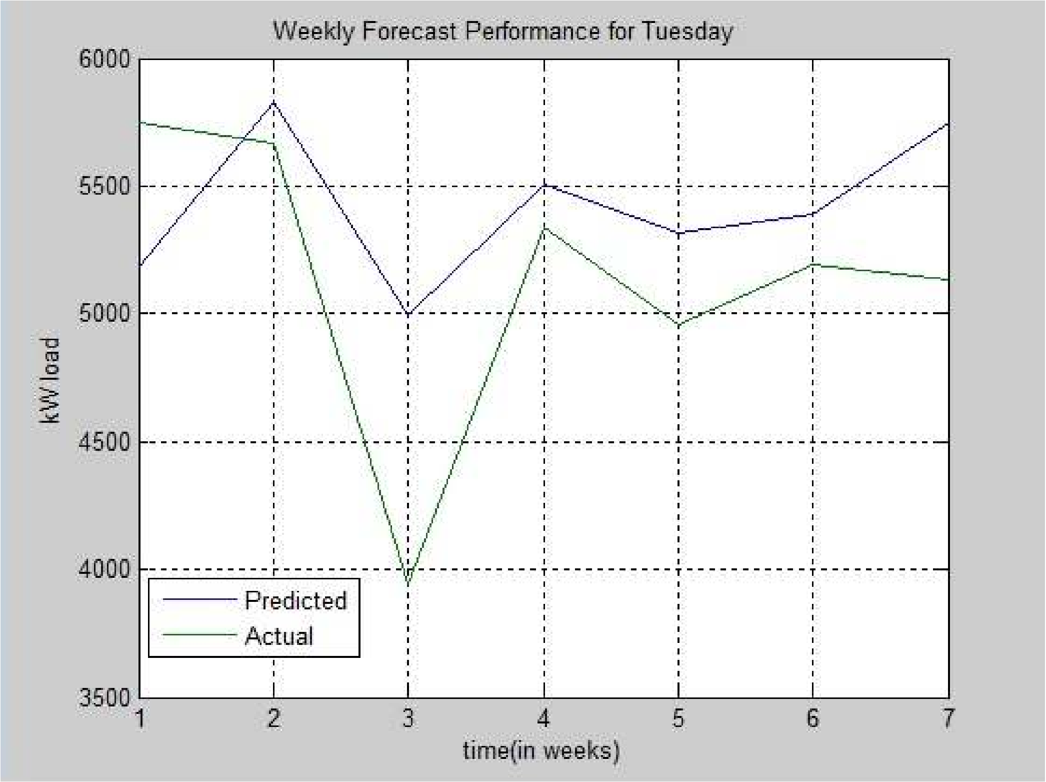

Figure 28 : Tuesday Load Forecast with Adjustment

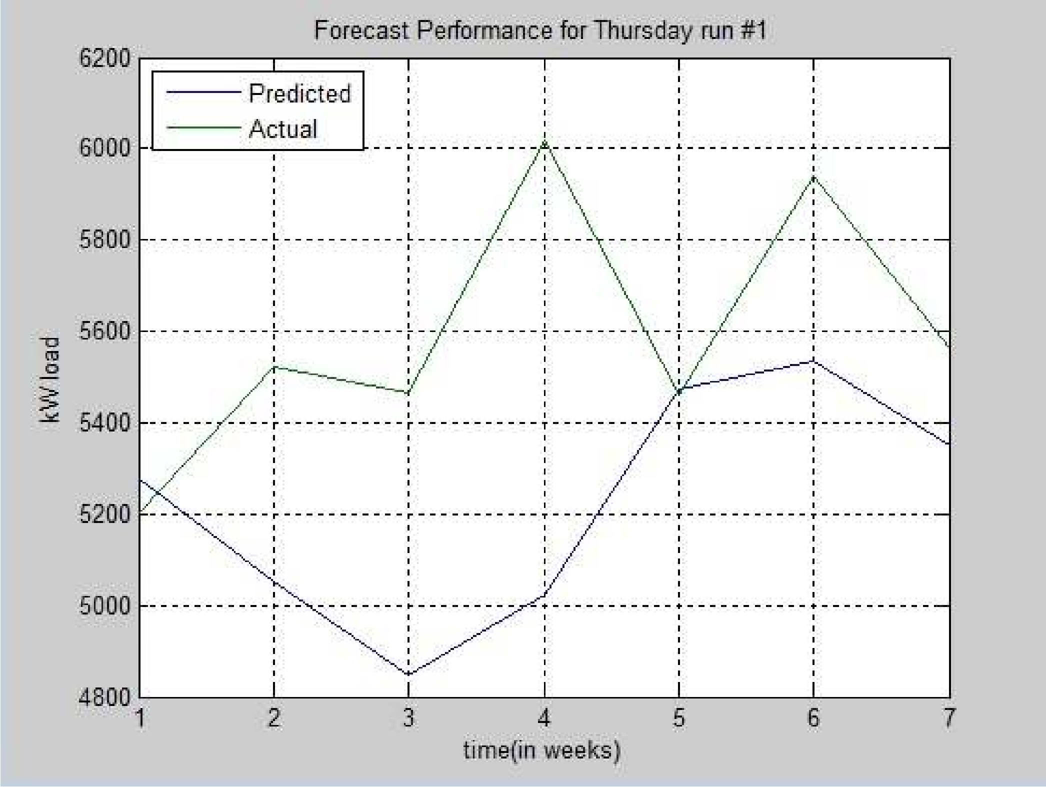

Figure 29 : Thursday Load Forecast with Adjustment

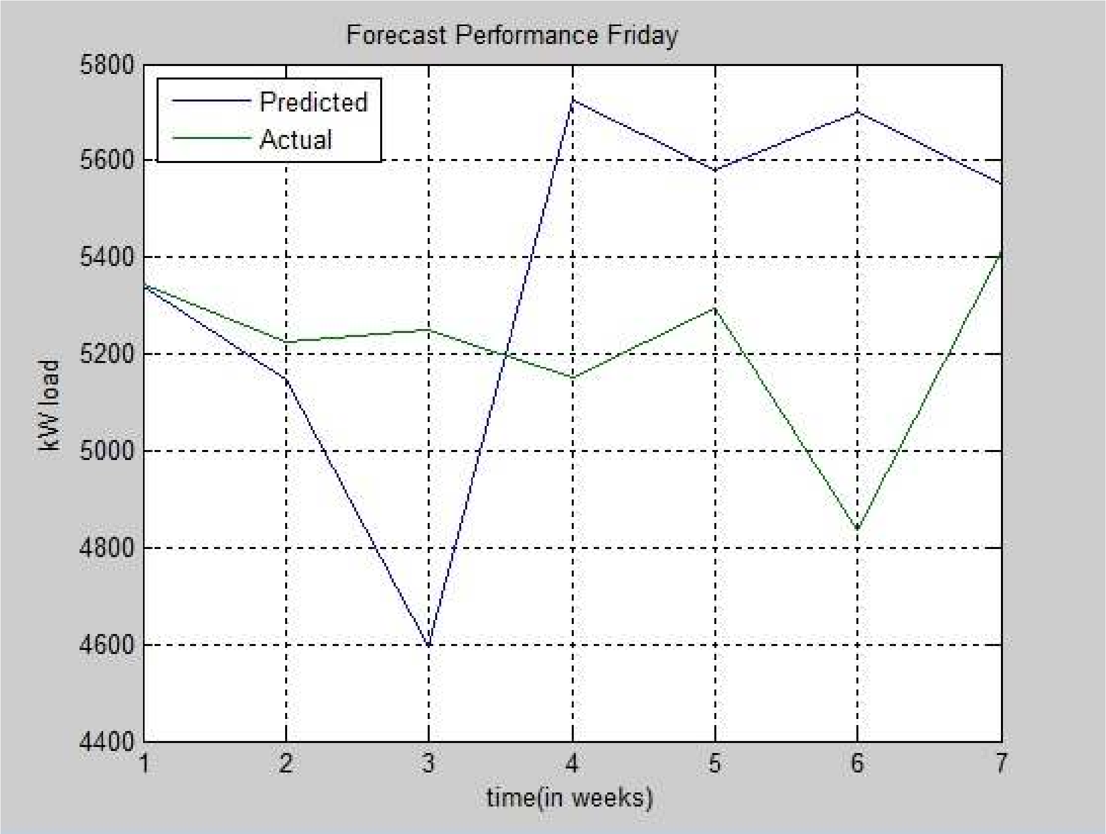

Figure 30 : Friday Load Forecast with Adjustment

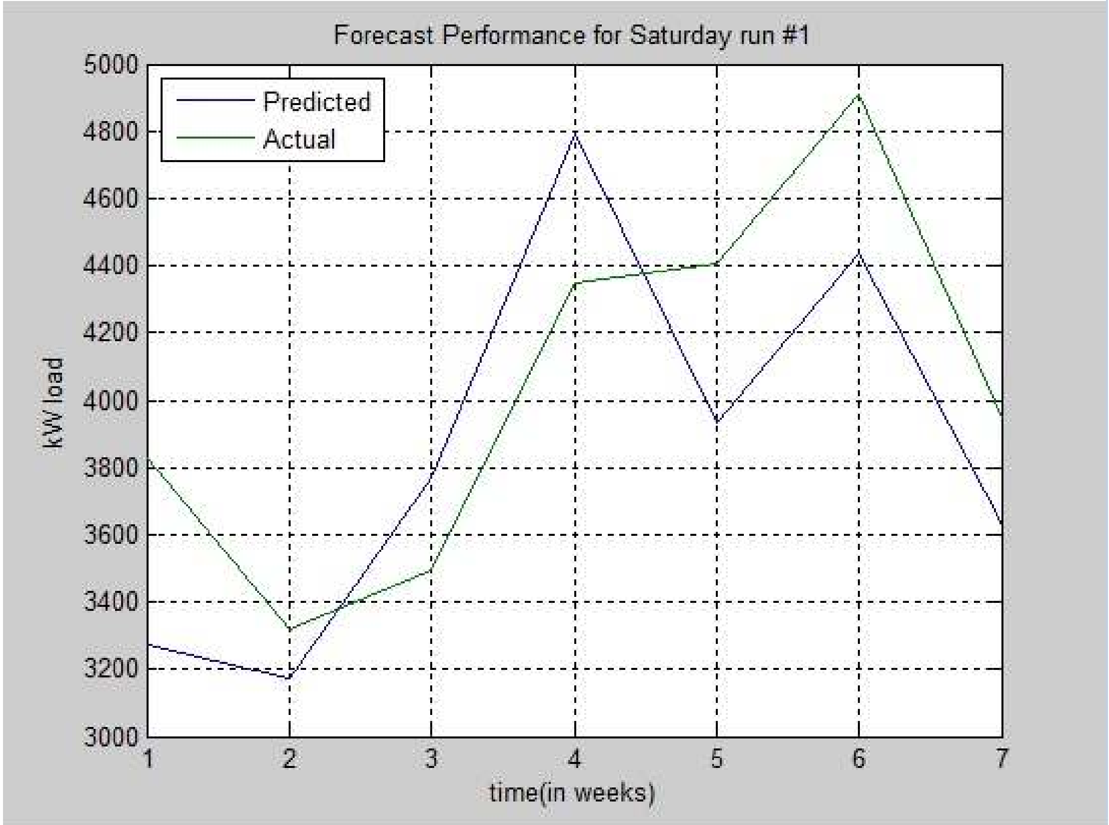

Figure 31 : Saturday Load Forecast with Adjustment

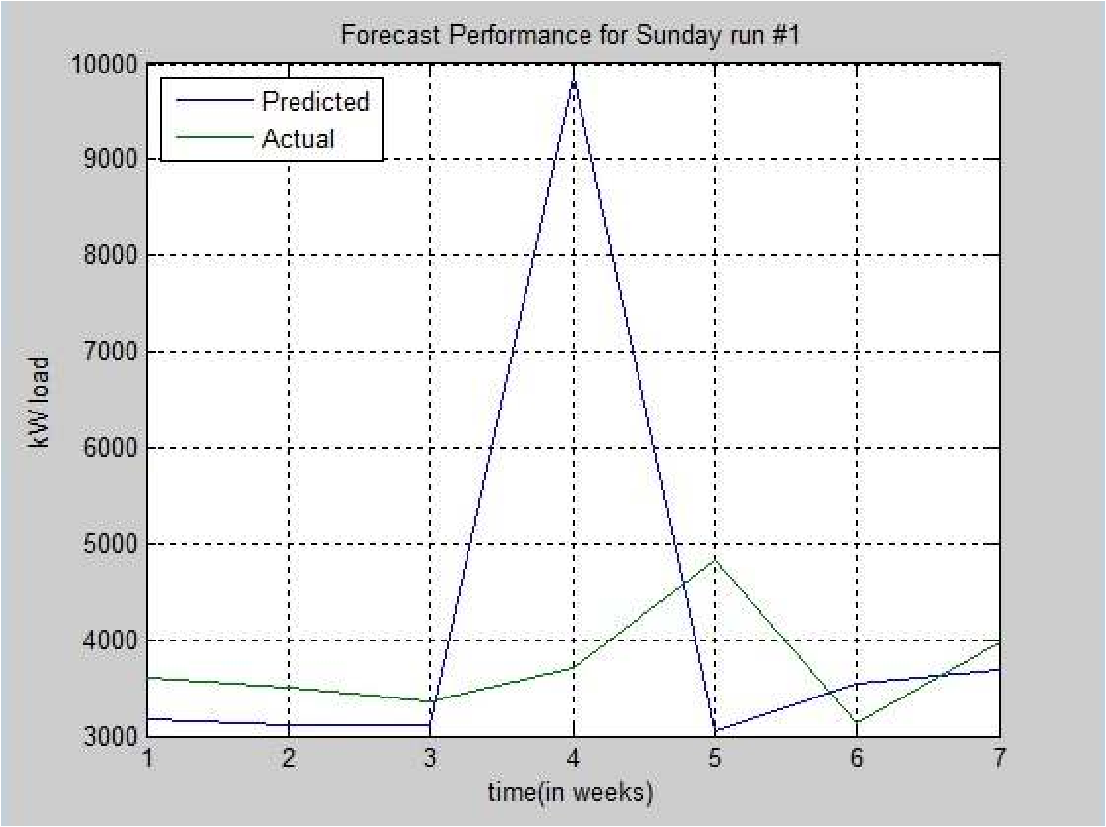

Figure 32 : Sunday Load Forecast with Adjustment

Same as without adjustment, each and every group of the day have been simulated for 10 times and the MAPE of the forecast is shown below according to the day group. For Wednesday group, as the result have been under 10% it does not need any further adjustment. So, Wednesday is excluded from the adjustment simulation:-

Table 14 : Simulation for Monday

Simulation	MAPE Value (%)
1	2.781
2	5.498
3	2.541
4	4.114
5	3.666
6	7.117
7	2.339
8	1.745
9	6.183
10	5.242
Average MAPE value (%)	4.120

Table 15 : Simulation for Tuesday

Simulation	MAPE Value (%)
1	10.003
2	8.935
3	9.346
4	8.052
5	9.600
6	8.898
7	7.698
8	8.272
9	8.625
10	8.784
Average MAPE value (%)	8.820

Table 16 : Simulation for Thursday

Simulation	MAPE Value (%)
1	9.06
2	9.257
3	9.318
4	9.070
5	9.180
6	9.229
7	9.511
8	9.142
9	9.248
10	9.295
Average MAPE value (%)	9.230

Table 17 : Simulation for Friday

Simulation	MAPE Value (%)
1	6.691
2	7.345
3	6.872
4	7.655
5	6.492
6	6.406
7	6.333
8	6.768
9	6.752
10	6.711
Average MAPE value (%)	6.833

Table 18 : Simulation for Saturday

Simulation	MAPE Value (%)
1	9.268
2	8.323
3	9.411
4	8.975
5	9.250
6	9.060
7	8.603
8	9.112
9	8.256
10	8.764
Average MAPE value (%)	8.91

Table 19 :Simulation for Sunday

Simulation	MAPE Value (%)
1	15.51
2	15.47
3	12.55
4	13.09
5	16.58
6	45.60
7	44.12
8	20.51
9	27.28
10	23.77
Average MAPE value (%)	22.45

Table 20 :Average MAPE for Forecast With Adjustment

Day	MAPE Value (%)
Monday	4.120
Tuesday	8.820
Wednesday	-
Thursday	9.23
Friday	6.833
Saturday	8.91
Sunday	22.45

4.2 Discussion

From the simulation, we know that only Thursday achieve the MAPE lower than the targeted MAPE. The reasons of the high MAPE is because between the input data, there are some holidays and the input data for Saturday and Sunday is highly unstable because of the load depends on the activities that have been done in the weekends in UTP. Moreover, the data that have been used is only from the previous semester ON data from 2006 up until 2011. Because of that, we have a limited data to be projected.

Table 21 : Last Semester Model Data Distribution

Data Partitioning	New Model (Without Validation)			
	Monday		Tuesday	
	%	Data	%	Data
Training	70	136	70	136
Validation	×	×	×	×
Testing	30	56	30	56
TOTAL	100	192	100	192

Data Partitioning	Wednesday		Thursday	
	%	Data	%	Data
Training	70	136	70	136
Validation	×	×	×	×
Testing	30	56	30	56
TOTAL	100	192	100	192

Data Partitioning	Friday		Saturday	
	%	Data	%	Data
Training	70	136	70	124
Validation	×	×	×	×
Testing	30	56	30	54
TOTAL	100	192	100	178

Data Partitioning	Sunday	
	%	Data
Training	70	124
Validation	×	×
Testing	30	54
TOTAL	100	178

Table 22 : Curret Model Data Distribution

Data Partitioning	New Model (Without Validation)			
	Monday		Tuesday	
	%	Data	%	Data
Training	70	166	70	166
Validation	×	×	×	×
Testing	30	68	30	68
TOTAL	100	234	100	234

Data Partitioning	New Model (Without Validation)			
	Wednesday		Thursday	
	%	Data	%	Data
Training	70	166	70	166
Validation	×	×	×	×
Testing	30	68	30	68
TOTAL	100	234	100	234

Data Partitioning	New Model (Without Validation)			
	Friday		Saturday	
	%	Data	%	Data
Training	70	166	70	166
Validation	×	×	×	×
Testing	30	68	30	68
TOTAL	100	234	100	234

Data Partitioning	Sunday	
	%	Data
Training	70	166
Validation	×	×
Testing	30	68
TOTAL	100	234

From this, the data which have been used for training is larger, which improves the training for neural network itself. As we know, huge data gives a bigger distribution for the neural network to train and produce better. Even though the data supplied for training and testing is larger for the new model, the data is consider to be small, comparing to the weekly model.

The MAPE of both of the data before modification and after modification is shown below. The data modification is used to adjust the out-layered data from the reading. The out-layered data comes when there are holidays in UTP during the weekdays. Because of that, the reading of the weekdays will be affected. For weekends, the activities that happen during the weekend affect the load data hence making the data to be in range.

Table 23 : Different between Before and After

Days	**MAPE Before Adjustment (%)**	**MAPE After Adjustment (%)**
Monday	9.642	4.120
Tuesday	9.037	8.820
Wednesday	6.573	-
Thursday	10.570	9.23
Friday	10.408	6.833
Saturday	13.720	8.91
Sunday	22.930	22.45

From table 23, every group of the day shows an improvement in MAPE except for Sunday. The fluctuation in Sunday data from UTP does not maintain for every week creating high MAPE. But, the adjustment is made to fit the purpose of our forecasting only and for real life this adjustment should not be done.

The fluctuation of the data can only be seen when there are holidays. The fluctuation is the effect of the number of student which is studying in UTP itself. Larger student number will create much more fluctuation on Sunday as students activities on weekends are hard to predict. Below is the data for student number in UTP back from January 2006 until the recent semester, May 2012. Based on Appendix C, the number of students which are staying in UTP either foundation or undergraduate is considered to be in the same range. But, when trimester is fully implemented in UTP which is from January 2011, we may see that the load for September semester decrease but still maintain. In May 2012, the student's number is the highest among all of the semester.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Electric Forecasting gives an important role to GDC to ensure that the amount supplied can be enough to power up the area of UTP. As GDC UTP system is an islanded generation system, there is a need for the GDC to ensure that the power that have been used in UTP is optimized either from the generation or the load. Therefore, the forecasting will help GDC to produce the exact need of the UTP electricity demand

Because of the optimization, the wastage of energy will surely be reduce and hopefully it can be reduced to zero. 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 purpose design of the model does not meet the target that we have set. Only the Sunday MAPE value meets the expectation. There are many things that need to be done to ensure that the forecasted value will be as close as possible to the real value of the load demand.

5.2 Recommendations

Forecast model for Sunday need to be done back to ensure that the forecast result can get lower MAPE. Furthermore, as the data that have been used is combining all of the days from each group, the data will fluctuate more as UTP have a season of OFF Semester and ON Semester before. Both of this loads differs from each other. So, for the next forecast model, there should be a different data for training and testing.

REFERENCES

[1] Hitoshi Takara, Katsumi Uezato and Toshihisa Funabashi Tomonobu Senjyu, "One-Hour-Ahead Load Forecasting Using Neural Network," *IEEE Transactions on Power System*, vol. 17, no. 1, p. 113, 2002.

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

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

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

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

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

[7] C. Chatfiled, *Time Series Forecasting.*: Chapman and Hall, 2010.

[8] Jaramillo-Moran, Carmona-Fernandez Gonzalez-Romera, "Monthly electric energy demand forcasting based on trend extraction," *IEEE Transactions on Power Systems*, vol. 21, no. 4, 1946-1953.

[9] W. Lee, S. Rasmiddattaa, J.Lia and R. Ross K. Methaprayoon, "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.

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

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

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

[13] S. Haykin, *Neural Networks : A Comprehensive Foundation*. New Jersey:

Prentice-Hall, 1999.

[14] J.H Park K.Y. Lee, "Short-Term Load Forecasting using an Artificial Neural Network," *IEEE Transactions on Power Systems*, vol. 7, no. 1, February 1992.

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

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

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

[18] L.A. Sanabria, T.S. Dillion B.R. Szkuta, "Electricity price short term forecasting using artificial neural networks," *IEEE Transactions Power System*, vol. 14, no. 3, 1999.

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

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

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

[22] Smith M., *Neural Networks for Statistical Modeling*. New York: Van Nostrand Reinhold, 1993.

[23] L.A. Sanabria, T.S. Dillion B.R. Szkuta, "Electricity price short term forecasting using artificial neural networks," *IEEE Trans. Power Syst*, vol. 14, no. 3, pp. 851-857, 1999.

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

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

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

[27] M. K. Soni L. M. Saini, "Artificial neural network based peak load forecasting using Levennberg-Marquardt and quasi-Newton method," *IEE Proc.-Gener. Transm. Distrib. 149*, no. 5, pp. 278-584, 2002.

[28] Chi-Leung Hui, "Artificial Neural Networks - Application," *InTech*, 2011.

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

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 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

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

APPENDIX B

PROGRAMMING

```
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 day to forecast (1 until 7 according to
Monday to Sunday): ');
ulang = input ('How many times ??: ');


if selection==1

fprintf('\n');
fprintf('||||| Monday Selection Model |||||\n');
fprintf('\n');
%interface for Monday Model
ain3 = zeros(7,2);
aout3 = zeros(7,1);
weeks1 = {'14','13','12','11','10','9','8'};
weeks2 = {'7','6','5','4','3','2','1'};
weeks3 = {'7','6','5','4','3','2','1'};

fprintf('------------------------------------------------------------
-------------\n');
fprintf('Please Enter 14 days of Monday before historical load
demand (kW)\n');
fprintf('------------------------------------------------------------
-------------\n');

for m = 1:2
for n = 1:7
if m==1
fprintf('Please enter the value for the past %s weeks',weeks1{n})
else
fprintf('Please enter the value for the past %s weeks',weeks2{n})
end
ain3(n,m) = input (': ');
end
end

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

```
for n = 1:7
fprintf ('Please enter the value for the next %s weeks',weeks3{n})
aout3(n) = input (': ');
end

load mondayload;

for i = 1:ulang

p=mondaytrdat';
pt=mondaytrtgdat';
ts=mondaytsdat';
tst=mondaytstgdat';

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=[1:1:82];
figure('Name','Training Data for Monday','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data Monday 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=[1:1:35];
```

```
figure('Name','Testing Data for Monday','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Testing Data Monday 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 Monday run #%d',i);
fprintf('\n-----------------------------------------\n');
pload=pload*10000;
totalp = totalp + pload;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, pload (n));
end
fprintf('\n----------------------------------------\n');
fprintf('Actual Load Demand (kW) for Monday run #%d',i);
fprintf('\n----------------------------------------\n');
aload=(aout)*10000;
totala = totala + aload;
for n = 1:7
fprintf ('%s: %.0f\n',weeks3{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 weeks)')
ylabel('kW load')
title(['Weekly Forecast Performance for Monday run #',num2str(i)])
legend('Predicted','Actual',2)
grid on
end


fprintf('\n------------------------------------------\n');
fprintf('Average Forecast Load Demand (kW) for Monday\n');
fprintf('--------------------------------------------\n');
avgp = totalp/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avgp (n));
end


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

```
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avga (n));
end

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

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

x=0;

elseif selection==2

fprintf('\n');
fprintf('||||| Tuesday Selection Model |||||\n');
fprintf('\n');
%interface for Tuesday Model
ain3 = zeros(7,2);
aout3 = zeros(7,1);
weeks1 = {'14','13','12','11','10','9','8'};
weeks2 = {'7','6','5','4','3','2','1'};
weeks3 = {'7','6','5','4','3','2','1'};

fprintf('----------------------------------------------------------
-------\n');
fprintf('Please Enter 14 days of Tuesday before historical load
demand (kW)\n');
fprintf('----------------------------------------------------------
-------\n');

for m = 1:2
for n = 1:7
if m==1
fprintf('Please enter the value for the past %s weeks',weeks1{n})
else
fprintf('Please enter the value for the past %s weeks',weeks2{n})
end
ain3(n,m) = input (': ');
end
end

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

for n = 1:7
fprintf ('Please enter the value for the next %s weeks',weeks3{n})
aout3(n) = input (': ');
end

load tuesdayload;

for i = 1:ulang

p=tuetrdat';
pt=tuetrtgdat';
ts=tuetsdat';
tst=tuetstgdat';
```

```
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=[1:1:83];
figure('Name','Training Tuesday On','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Tuesday 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=[1:1:34];
figure('Name','Testing Data for Tuesday','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Testing Data Tuesday 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 Tuesday run #%d',i);
fprintf('\n------------------------------------------\n');
pload=pload*10000;
totalp = totalp + pload;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, pload (n));
end
fprintf('\n-----------------------------------------\n');
fprintf('Actual Load Demand (kW) for Tuesday run #%d',i);
fprintf('\n-----------------------------------------\n');
aload=(aout)*10000;
totala = totala + aload;
for n = 1:7
fprintf ('%s: %.0f\n',weeks3{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 weeks)')
ylabel('kW load')
title(['Weekly Forecast Performance for Tuesday run #',num2str(i)])
legend('Predicted','Actual',2)
grid on
end


fprintf('\n-------------------------------\n');
fprintf('Average Forecast Load Demand (kW)\n');
fprintf('---------------------------------\n');
avgp = totalp/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avgp (n));
end



fprintf('-------------------------------\n');
fprintf('Average Actual Load Demand (kW)\n');
fprintf('-------------------------------\n');
avga = totala/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avga (n));
end

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

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

x=0;

elseif selection==3
```

```
fprintf('\n');
fprintf('||||| Wednesday Selection Model |||||\n');
fprintf('\n');
%interface for Wednesday Model
ain3 = zeros(7,2);
aout3 = zeros(7,1);
weeks1 = {'14','13','12','11','10','9','8'};
weeks2 = {'7','6','5','4','3','2','1'};
weeks3 = {'7','6','5','4','3','2','1'};

fprintf('-----------------------------------------------------------
---------\n');
fprintf('Please Enter 14 days of Wednesday before historical load
demand (kW)\n');
fprintf('-----------------------------------------------------------
---------\n');

for m = 1:2
for n = 1:7
if m==1
fprintf('Please enter the value for the past %s weeks',weeks1{n})
else
fprintf('Please enter the value for the past %s weeks',weeks2{n})
end
ain3(n,m) = input (': ');
end
end

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

for n = 1:7
fprintf ('Please enter the value for the next %s weeks',weeks3{n})
aout3(n) = input (': ');
end

load wednesdayload;

for i = 1:ulang

p=wedtrdat';
pt=wedtrtgdat';
ts=wedtsdat';
tst=wedtstgdat';

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=[1:1:83];
figure('Name','Training Wednesday On','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Wednesday 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=[1:1:35];
figure('Name','Testing Data for Wednesday','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Testing Data Wednesday 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 Wednesday run #%d',i);
fprintf('\n--------------------------------------------\n');
pload=pload*10000;
totalp = totalp + pload;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, pload (n));
end
fprintf('\n-------------------------------------------\n');
fprintf('Actual Load Demand (kW) for Wednesday run #%d',i);
fprintf('\n-------------------------------------------\n');
aload=(aout)*10000;
totala = totala + aload;
for n = 1:7
fprintf ('%s: %.0f\n',weeks3{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 weeks)')
ylabel('kW load')
title(['Weekly Forecast Performance for Wednesday run
#',num2str(i)])
legend('Predicted','Actual',2)
grid on
end


fprintf('\n-------------------------------\n');
fprintf('Average Forecast Load Demand (kW)\n');
fprintf('---------------------------------\n');
avgp = totalp/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avgp (n));
end


fprintf('-------------------------------\n');
fprintf('Average Actual Load Demand (kW)\n');
fprintf('-------------------------------\n');
avga = totala/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avga (n));
end

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

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

x=0;

elseif selection==4

fprintf('\n');
fprintf('||||| Thursday Selection Model |||||\n');
fprintf('\n');
%interface for Thursday Model
ain3 = zeros(7,2);
aout3 = zeros(7,1);
weeks1 = {'14','13','12','11','10','9','8'};
weeks2 = {'7','6','5','4','3','2','1'};
weeks3 = {'7','6','5','4','3','2','1'};

fprintf('-----------------------------------------------------------
--------\n');
fprintf('Please Enter 14 days of Thursday before historical load
demand (kW)\n');
```

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

for m = 1:2
for n = 1:7
if m==1
fprintf('Please enter the value for the past %s weeks',weeks1{n})
else
fprintf('Please enter the value for the past %s weeks',weeks2{n})
end
ain3(n,m) = input (': ');
end
end

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

for n = 1:7
fprintf ('Please enter the value for the next %s weeks',weeks3{n})
aout3(n) = input (': ');
end

load thursdayload;

for i = 1:ulang

p=thurtrdat';
pt=thurtrtgdat';
ts=thurtsdat';
tst=thurtstgdat';

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=[1:1:83];
```

```
figure('Name','Training Data for Thursday','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Thursday 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=[1:1:36];
figure('Name','Testing Data Thursday','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Thursday 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',weeks3{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',weeks3{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 weeks)')
ylabel('kW load')
title(['Forecast Performance for Thursday run #',num2str(i)])
legend('Predicted','Actual',2)
```

```
grid on
end


fprintf('\n-------------------------------\n');
fprintf('Average Forecast Load Demand (kW)\n');
fprintf('---------------------------------\n');
avgp = totalp/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avgp (n));
end


fprintf('-------------------------------\n');
fprintf('Average Actual Load Demand (kW)\n');
fprintf('-------------------------------\n');
avga = totala/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avga (n));
end

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

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

x=0;

elseif selection==5

fprintf('\n');
fprintf('||||| Friday Selection Model |||||\n');
fprintf('\n');
%interface for Friday Model
ain3 = zeros(7,2);
aout3 = zeros(7,1);
weeks1 = {'14','13','12','11','10','9','8'};
weeks2 = {'7','6','5','4','3','2','1'};
weeks3 = {'7','6','5','4','3','2','1'};

fprintf('-----------------------------------------------------------
------\n');
fprintf('Please Enter 14 days of Friday before historical load
demand (kW)\n');
fprintf('-----------------------------------------------------------
------\n');

for m = 1:2
for n = 1:7
if m==1
fprintf('Please enter the value for the past %s weeks',weeks1{n})
else
fprintf('Please enter the value for the past %s weeks',weeks2{n})
end
ain3(n,m) = input (': ');
end
end

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

for n = 1:7
fprintf ('Please enter the value for the next %s weeks',weeks3{n})
aout3(n) = input (': ');
end

load fridaydata;

for i = 1:ulang

p=fritrdat';
pt=fritrtgdat';
ts=fritsdat';
tst=fritstgdat';

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=[1:1:83];
figure('Name','Training Data for Friday','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Friday 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=[1:1:35];
figure('Name','Testing Data for Friday','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Friday 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',weeks3{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',weeks3{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 weeks)')
ylabel('kW load')
title(['Forecast Performance Friday 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/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avgp (n));
end


fprintf('-------------------------------\n');
fprintf('Average Actual Load Demand (kW)\n');
fprintf('-------------------------------\n');
avga = totala/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avga (n));
end

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

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

x=0;

elseif selection==6

fprintf('\n');
fprintf('||||| Saturday Selection Model |||||\n');
fprintf('\n');
%interface for Saturday Model
ain3 = zeros(7,2);
aout3 = zeros(7,1);
weeks1 = {'14','13','12','11','10','9','8'};
weeks2 = {'7','6','5','4','3','2','1'};
weeks3 = {'7','6','5','4','3','2','1'};

fprintf('-----------------------------------------------------------
-------------\n');
fprintf('Please Enter 14 days of Saturday before historical load
demand (kW)\n');
fprintf('-----------------------------------------------------------
-------------\n');

for m = 1:2
for n = 1:7
if m==1
fprintf('Please enter the value for the past %s weeks',weeks1{n})
else
fprintf('Please enter the value for the past %s weeks',weeks2{n})
end
ain3(n,m) = input (': ');
end
end

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

for n = 1:7
fprintf ('Please enter the value for the next %s weeks',weeks3{n})
aout3(n) = input (': ');
end

load saturdaydata;

for i = 1:ulang

p=sattrdat';
pt=sattrtgdat';
```

```
ts=sattsdat';
tst=sattstgdat';

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=[1:1:79];
figure('Name','Training Data for Saturday','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Saturday 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=[1:1:34];
figure('Name','Testing Data for Saturday','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Saturday 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',weeks3{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',weeks3{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 weeks)')
ylabel('kW load')
title(['Forecast Performance for Saturday run #',num2str(i)])
legend('Predicted','Actual',2)
grid on
end


fprintf('\n-------------------------------\n');
fprintf('Average Forecast Load Demand (kW)\n');
fprintf('---------------------------------\n');
avgp = totalp/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avgp (n));
end


fprintf('-------------------------------\n');
fprintf('Average Actual Load Demand (kW)\n');
fprintf('-------------------------------\n');
avga = totala/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avga (n));
end

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

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

x=0;
```

```
elseif selection==7

fprintf('\n');
fprintf('||||| Sunday Selection Model |||||\n');
fprintf('\n');
%interface for Sunday Model
ain3 = zeros(7,2);
aout3 = zeros(7,1);
weeks1 = {'14','13','12','11','10','9','8'};
weeks2 = {'7','6','5','4','3','2','1'};
weeks3 = {'7','6','5','4','3','2','1'};

fprintf('-----------------------------------------------------------
-------------\n');
fprintf('Please Enter 14 days of Monday before historical load
demand (kW)\n');
fprintf('-----------------------------------------------------------
-------------\n');

for m = 1:2
for n = 1:7
if m==1
fprintf('Please enter the value for the past %s weeks',weeks1{n})
else
fprintf('Please enter the value for the past %s weeks',weeks2{n})
end
ain3(n,m) = input (': ');
end
end

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

for n = 1:7
fprintf ('Please enter the value for the next %s weeks',weeks3{n})
aout3(n) = input (': ');
end

load sundayload;

for i = 1:ulang

p=suntrdat';
pt=suntrtgdat';
ts=suntsdat';
tst=suntstgdat';

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=[1:1:79];
figure('Name','Training Data for Sunday','NumberTitle','off')
plot(day,test1,day,pt)
xlabel('time(in weeks)')
ylabel('Normalization value kW-load')
title(['Training Data for Sunday 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=[1:1:34];
figure('Name','Testing Data for Sunday','NumberTitle','off')
plot(day,test2,day,tst)
xlabel('time(in days)')
ylabel('Normalization value kW-load')
title(['Testing Data for Sunday 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',weeks3{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',weeks3{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 weeks)')
ylabel('kW load')
title(['Forecast Performance for Sunday run #',num2str(i)])
legend('Predicted','Actual',2)
grid on
end


fprintf('\n-------------------------------\n');
fprintf('Average Forecast Load Demand (kW)\n');
fprintf('---------------------------------\n');
avgp = totalp/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avgp (n));
end


fprintf('-------------------------------\n');
fprintf('Average Actual Load Demand (kW)\n');
fprintf('-------------------------------\n');
avga = totala/ulang;
for n = 1:7
fprintf ('%s: %.3f\n',weeks3{n}, avga (n));
end

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

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

x=0;


end
```

APPENDIX C

Remark	Sem	Year	Undergraduate + Foundation					Going Internship					Sem Break				
			Female	%	Male	%	Total	Female	%	Male	%	Total	Female	%	Male	%	Total
UG+F	**JAN**	**2005**	1820	41.85%	2529	58.15%	**4349**	203	48.80%	213	51.20%	**416**					
UG+F	**JUL**	**2005**	1950	40.73%	2838	59.27%	**4788**	198	49.87%	199	50.13%	**397**					
UG+F	**JAN**	**2006**	2015	40.14%	3005	59.86%	**5020**	185	40.22%	275	59.78%	**460**					
UG+F	**JUL**	**2006**	2109	39.63%	3213	60.37%	**5322**	155	39.74%	235	60.26%	**390**					
UG+F	**JAN**	**2007**	2068	38.12%	3357	61.88%	**5425**	219	39.39%	337	60.61%	**556**					
UG+F	**JUL**	**2007**	2130	38.03%	3471	61.97%	**5601**	188	41.50%	265	58.50%	**453**					
UG+F	**JAN**	**2008**	2117	37.88%	3472	62.12%	**5589**	245	37.35%	411	62.65%	**656**					
UG+F	**JUL**	**2008**	2099	37.15%	3551	62.85%	**5650**	208	41.35%	295	58.65%	**503**					
UG+F	**JAN**	**2009**	2078	36.55%	3608	63.45%	**5686**	216	37.18%	365	62.82%	**581**					
F	**MAY**	**2009**	148	29.78%	349	70.22%	**497**	0		0		**0**					
UG	**JUL**	**2009**	1880	37.04%	3196	62.96%	**5076**	300	46.37%	347	53.63%	**647**					
F	**SEP**	**2009**	162	29.83%	381	70.17%	**543**	0		0		**0**					
UG+F	**JAN**	**2010**	1988	35.62%	3593	64.38%	**5581**	247	37.54%	411	62.46%	**658**					
F	**MAY**	**2010**	217	31.68%	468	68.32%	**685**	0		0		**0**					
UG	**JUL**	**2010**	1641	35.63%	2965	64.37%	**4606**	196	36.43%	342	63.57%	**538**					
F	**SEP**	**2010**	201	30.69%	454	69.31%	**655**	0		0		**0**					
UG+F	**JAN**	**2011**	1712	33.06%	3467	66.94%	**5179**	226	34.66%	426	65.34%	**652**					
UG+F	**MAY**	**2011**	1653	32.34%	3458	67.66%	**5111**	289	37.05%	491	62.95%	**780**					
UG+F	**SEP**	**2011**	1047	30.46%	2390	69.54%	**3437**	217	47.28%	370	80.61%	**459**	601	30.59%	1364	69.41%	**1965**
UG+F	**JAN**	**2012**	1560	31.93%	3326	68.07%	**4886**	157	33.76%	308	66.24%	**465**	102	22.77%	346	77.23%	**448**