ENERGY CONSUMPTION PREDICTION USING SMART METER DATA

DUAL DEGREE PROJECT THESIS

Submitted in partial fulfillment of the requirements

for the award of degree of

BACHELOR and MASTER OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

By

L.Harshavardhan Reddy
(EE14B088)

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS, CHENNAI

MAY-2019

THESIS CERTIFICATE

This is to certify that the project work entitled "**ENERGY CONSUMPTION PREDICTION USING SMART METER DATA"** submitted by **L.Harshavardhan Reddy,EE14B088**, to **Indian Institute of Technology Madras** in partial fulfillment of the requirements for the award of degree of **Master of Technology**, is a bona fide record of work carried out by him. The contents of this report, in full or part have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. K. Shanti Swarup

Professor, Project Guide

Dept. of ElectricalEngineering

IIT Madras, 600036

Place:Chennai

Date:

ACKNOWLEDGEMENTS

I take pleasure in thanking various people, who helped and supported me to carry out this work during execution of the project activities. I am thankful to those who influenced my thinking during the course of study.

I would like to express my sincere thanks to my project guide Dr.K Shanti Swarup, for his support and guidance. His suggestions had been the source of inspiration for me during the conduct of this project. I am also thankful to the lab facilities given to me in laboratories, which really helped me in shaping this project work. I would like to thank my friends Jadhav Pradeep, Srinivas Gunna for making this journey enjoyable.

Finally, most importantly, I wish to express my heartfelt thanks to my family and friends, for being my source of energy.

L.Harshavardhan Reddy

ABSTRACT

Key words: Energy consumption; Smart meter data; Short term forecasting; Neural networks;

The large automation of appliances and adoption of high energy consuming utilities in homes has the potential to exert a huge constrain on present electric infrastructure. Majority of the energy supply chain infrastructure across the world are incapable to handle large and concentrated energy demands. Therefore, electric energy suppliers are challenged to make accurate and granular forecasting of future electricity demand. Recent developments in Power sector has led to the development of smart meters with bi-directional communication capabilities between customers and utilities. The data stored by these meters are huge and pose challenges in extracting useful information from them. Accurate load forecasting helps the utility to schedule their resources in order to maintain demand supply balance. Short term forecasting using smart meter data is significantly important as it greatly affects the resource planning and control operations of the utility. Machine learning and data mining have been widely used by researchers for extensive intelligent analysis of data to recognize the normal patterns of behaviour.

Smart meters records the energy consumption values of a household periodically. In this project artificial neural networks were used to build the prediction model. The neural network recognizes the consumption pattern of the consumer. The neural network used in this project is a feed forward network with 1 hidden layer. The network is trained using 3 year smart meter data to recognize the pattern of consumption by the consumer. Later that pattern is used to make prediction. Later the network is tested against one year data to evaluate the performance. The correlation coefficient between predicted values and actual value show how close the predictions are and it also points at the potential of smart meter data in building a prediction model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOTATION

σ → Standard Deviation

z_j → Input to the node at layer j

y_j → Output of the node at layer j

E → Root mean square error

r→ Correlation coefficient

y→ Output of the network

t→ Target/Expected output

Chapter 1
Introduction

1.1 Introduction:

Electricity is essential for modern society as it powers everyday devices like computers, televisions, telephones. Most of our devices are powered by electricity. Our electricity consumption is rapidly increasing with increasing population and dependency on machines for most of our day to day activities. Modern power systems are also expected to incorporate electric vehicles and renewable sources. The consumption varies with time, at certain times there are peaks. Production of more energy will help this situation at peak demand. There is a certain gap between production and consumption. The existing power system has reached its limitations. There are more frequent blackouts and visibility is also poor. In most cases the fault in the grid cannot be detected unless the customers report or manual verification is done. Smart grid is proposed to improve the current situation of power grid.

1.2 Smart Grid:

Smart grid is a concept that covers wide spectrum of electric power engineering. Different stakeholders have different definition for smart grid to fit their own vision. For U.S. Department of Energy (DOE), a smart grid should be “intelligent, efficient, accommodating, motivating, opportunistic, quality-focused, environment friendly and resilient”. The European Smart Grid ETP defines the smart grids as “electricity network that can intelligently integrate the behavior and actions of all users connected to it- generators, consumers and those that do both- in order to efficiently deliver sustainable, economic and secure electricity supplies”.[3]

The common interpretation of smart grid is it is a combination of communication and power infrastructure. There is two way communication between utility and consumer. This interpretation gives the impression that having advanced metering infrastructure (AMI) makes grid smart. Though AMI is indispensable part of smart grids the intelligent use of information i.e. data analysis and decision making gives smartness to the grid. Intelligent use of information is the core of smart grids.[3] Use of this information can improve the performance of grid in areas of control, operation and protection of grid. One of the most basic application that can be implemented using data is a prediction model. Prediction itself means making a guess of future based on the historical data.

1.3 Forecasting:

Load forecasting has been an essential task throughout the development of modern power system. Long-term forecasting (a few months to years) aims to assist in power system infrastructure planning, while med-term (a few weeks to months) and short-term (less than an hour to a few weeks) can be essentially useful in system operations. Short-term load forecasting can provide an accurate means to predict future loads which would lead to precise planning and optimal distribution of resources. Over the past decades many models have been proposed to solve this problem. Neural networks were used to make the prediction model in this paper.

1.4 Objective:

The objective of this project can be written in 2 stages

I. Find the energy consumption pattern of the consumer.
II. Short term forecasting of the energy consumption based on the energy consumption values and pattern recognized.

The basis of this is that the consumers has a pattern in the usage of energy. In short the aim is to find that pattern and make the predictions.

1.5 Scope:

Making accurate prediction will enable the utilities to distribute their resources in most optimal and efficient way. In active market where the electricity price is determined by customer bid, the short term energy consumption forecasting can be used to determine bids and price signal. Good short term energy consumption forecasting can assist the operations of the grid.

1.6 Organization of the thesis:

This thesis is organized into 6 chapters:

Chapter 2 presents how the smart meter data form households is collected and how data from has to be read. This chapter also discusses about the functionalities of the smart meters and advanced metering infrastructure.

Chapter 3 presents how artificial neural networks are used to make prediction model of the energy consumed. This chapter also includes the specifications of the neural networks used in this paper.

Chapter 4 presents the steps involved in the simulations. The experiments involve many steps and this chapter briefly discusses the steps involved.

Chapter 5 presents the results of the simulations.

Chapter 6 concludes the thesis and discusses about future work

Chapter 2
Smart metering and Data aggregation

2.1 Introduction:

Utilities require proper knowledge of peak or off peak period, power usage pattern, higher frequency of usages information, two way communications between the meters at the consumer end and management system at the utilities end, etc. Smart metering through smart meters and smart grid can provide these abilities, and the consumers can reduce the electricity bills and the utilities can better manage the power supply. Therefore, the use of smart meter is increasing at a high rate. Implementation of smart metering and various algorithms has also the potential to perform power system fault detection, isolation, and restoration quickly with higher accuracy. [2]Below figure shows a sample meter.

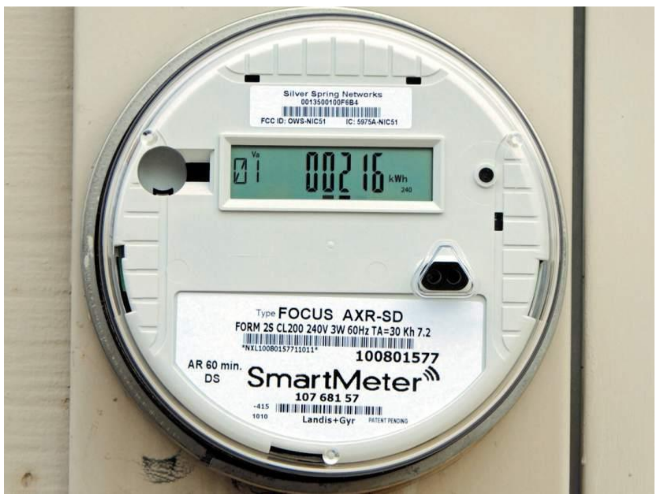

Figure 2.1**:** Above picture shows a sample smart meter[8]

2.2 Smart meters:

Smart meters are powerful tools which fundamentally change the operation of power grids. In addition to the functions of the traditional meters, smart meters can be used as sensors across the distribution grid. When an Advanced Metering Infrastructure (AMI) is in place, improves the visibility in the power grid and smart meter can measure and record actual power usage during day at a certain time interval. The readings data is then transmitted to data management system through secure network for further processing and storage. In addition, these sensors can be used by utilities to detect faults and send outage or restoration notifications. Use of this information allows the utilities to provide more reliable power grid. Utilities use this information for the following applications:

I. Faster outage detection, response and resolution by providing data to the field operations timely.

II. Keep customers better informed about status of the power grid. Utilities can communicate relevant information like cause of outage, estimated restoration time.

III. Improve resilience against disruptions, reduce potential outages and reduce frequency and duration of outages, thereby enhancing accuracy of the grid asset planning and management. [2]

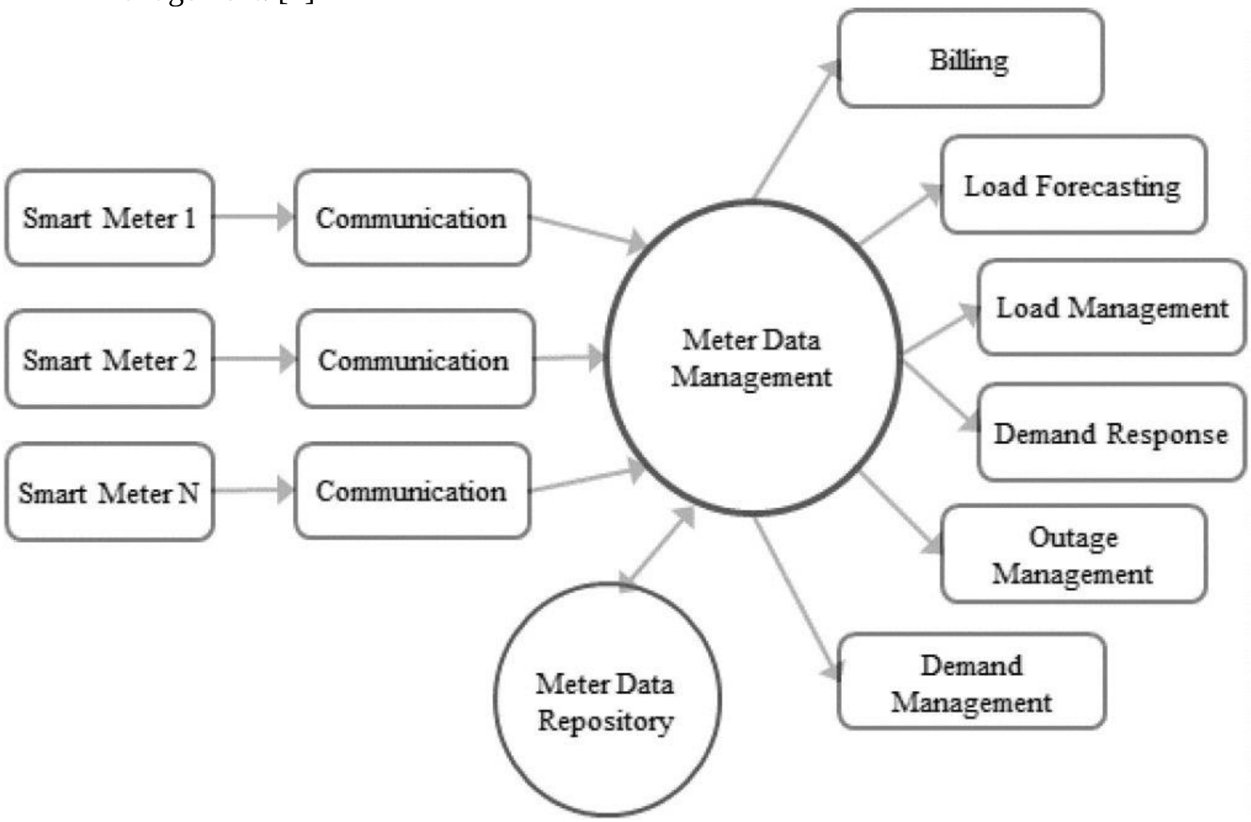

Figure 2.2: Figure illustrates how smart meters are connected [2]

The smart meter data used in the simulations are downloaded from AEMO website.

2.2.1 Background:

A basic meter records the total amount of energy consumed at a connection point from the initial energization of the meter. Periodic readings of basic meters are used to determine the energy used between two points of time.

$$E(\delta t) = E(t + \delta t) - E(t) \tag{2.1}$$

$E(\delta t) \rightarrow$ Energy consumed inδt time

$E(t + \delta t) \rightarrow$ Meter reading at time $t + \delta t$

$E(t) \rightarrow$ Meter reading at time t

Profiling is the process that converts readings from an accumulated energy meter reading into estimated energy consumption for each 30 minute trade interval period. Difference between two adjacent readings is used to find energy consumed in mean time. The energy consumption values are associated with time stamps. The Controlled Load Profile (CLP) is calculated from a group of approximately 200 sample meters installed by the local service provider for the profile area.

2.2.2 Profile area:

Each network is assigned a profile name that is similar to the name of the network. For example where Energy Australia is Local network provider, the profile name assigned is ENERGYAUST. Similarly each area has different name. The data used in this project is smart meter data, the meter record the consumption of the energy for every 30 minutes. So for 1 day we get 48 data points. All these data points, profile name, area etc are published in a csv sheet. Each row contains relevant information (profile name, area) and 48 data points each corresponding to 30 minute period in a day. [15]

2.3 Summary:

This chapter discusses about smart meters and smart metering. Their functions and functional requirements. Chapter also discusses about the working principle of smart meter, advanced metering infrastructure and meter data management system.

The visibility of the power grid can be improved by implementation of smart meters and advanced metering infrastructure (AMI). The implementation of the infrastructure depends on the requirements of the utility. In some cases smart meters can also acts as the controller of the load that changes load based on the price signal. Among many functionalities the most important functions are power consumption record with time and bi-directional communication. The implementation of smart meters will improve visibility of the grid and then models that reduce demand supply gap can be developed. The implementation of smart meters and its advanced functionalities there is a potential for significant change in power grid.

Chapter 3
ANN approach to Prediction of Energy Consumption

3.1 Introduction to ANN:

Artificial neural networks are computing systems inspired from biological neural networks. The inventor of first neuro computer, Dr. Robert Hetch-Nielsen defines a neural network as "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs". [5] The neural networks itself is not an algorithm, but rather a framework. They are programmed to learn specific task from the examples. The algorithms enable the computers to find behaviors based on statistical data.

The neural networks are computer model of human brain. The human brain is composed of billions nerve cells called neurons. Each neuron is connected to thousands of other neurons by axons. Schematic of neuron can be seen in fig3.1.

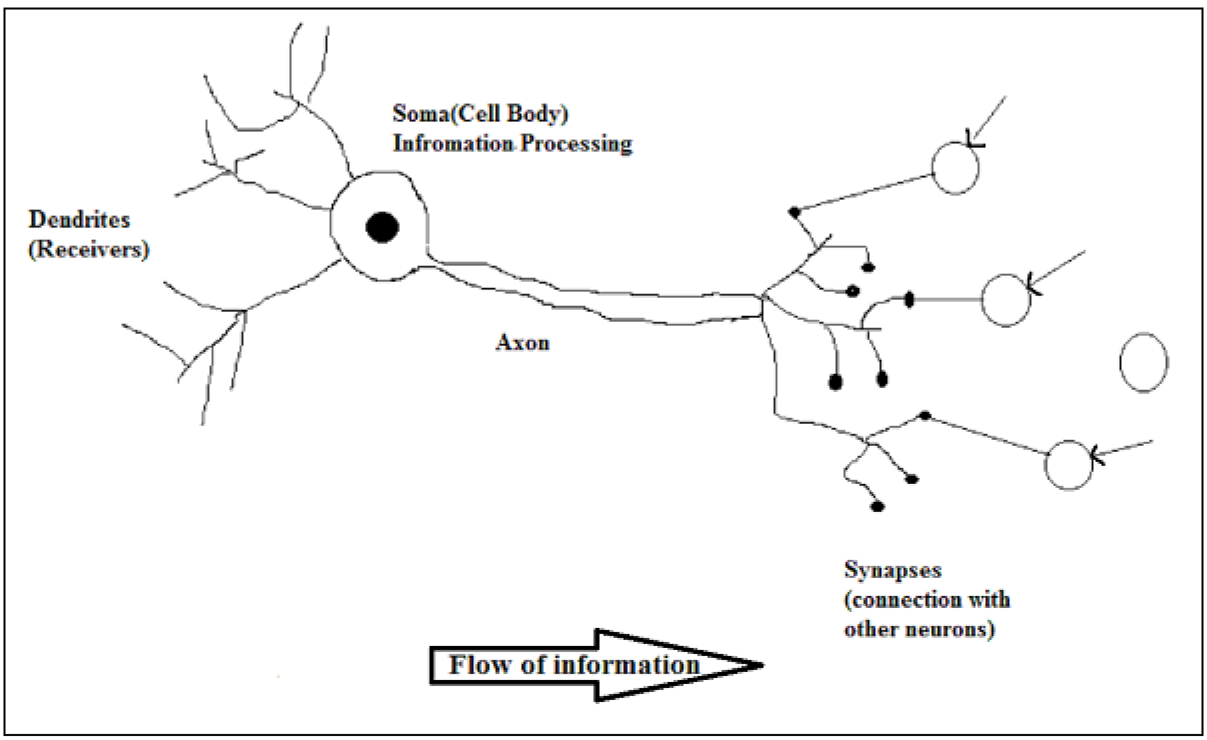

Figure 3.1: Schematic of a biological neuron [5]

Dendrites receive the electric impulse from external environment (either from other neurons or sensory organs), they are responsible for receiving information. Soma is the cell body and processes the information received from dendrites. Axon is just like the cable through which neurons send the output signal. Synapses is the connection between the axon and the dendrites of other neurons thereby transmitting output signal to other neurons. [5]

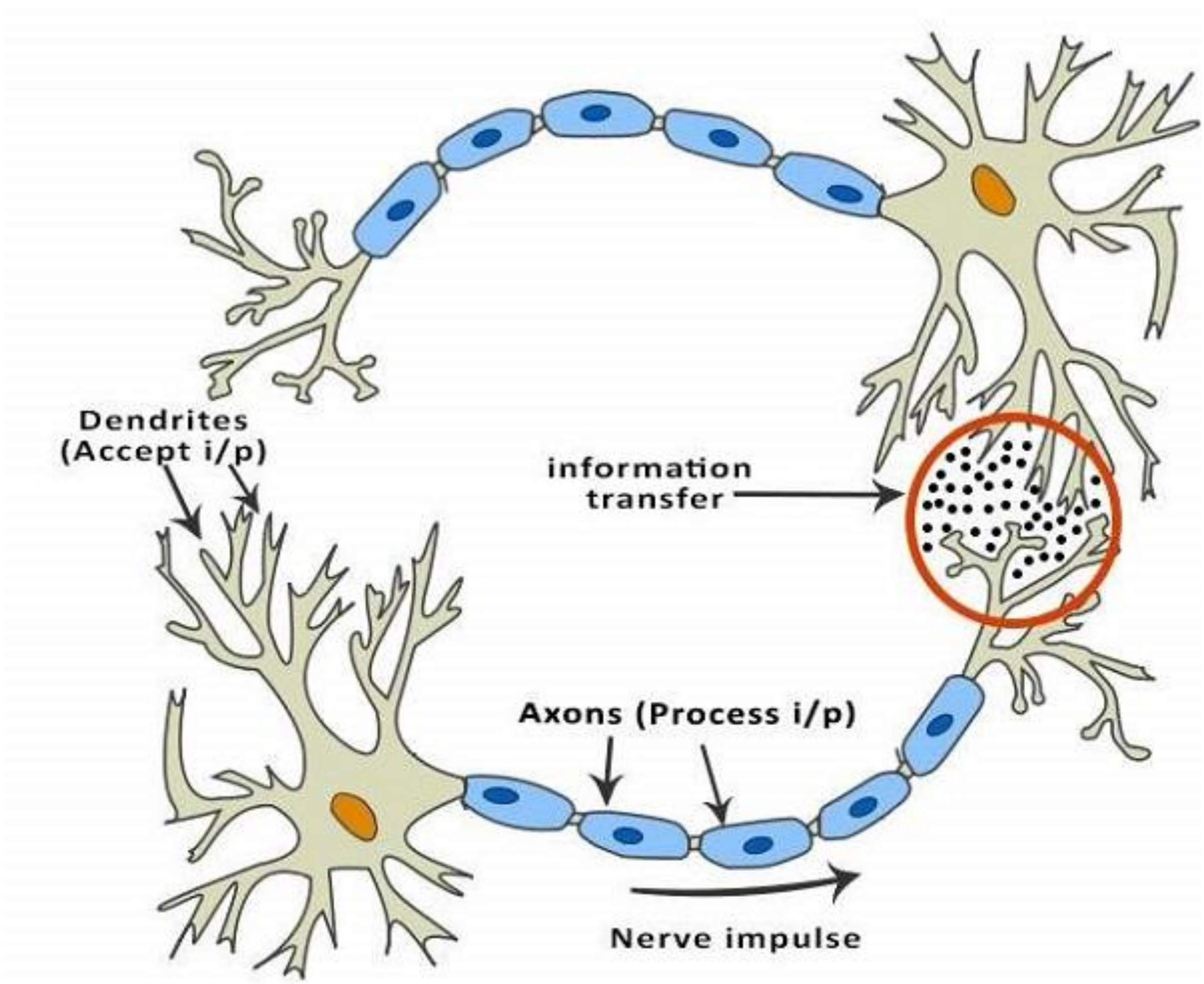

Figure 3.2**:** Information transfer between two neurons [5]

The input from sensory organs create electric impulses, which quickly travel through the network. A neuron can send the message to other neuron to handle the issue or does not send it forward.

The ANNs are composed of nodes, which try to imitate biological neurons of brain. Just like neurons nodes also have 4 processes:

1. Receive inputs from other nodes
2. Process inputs
3. Generate output signal based on input signal
4. Send output signal to other nodes

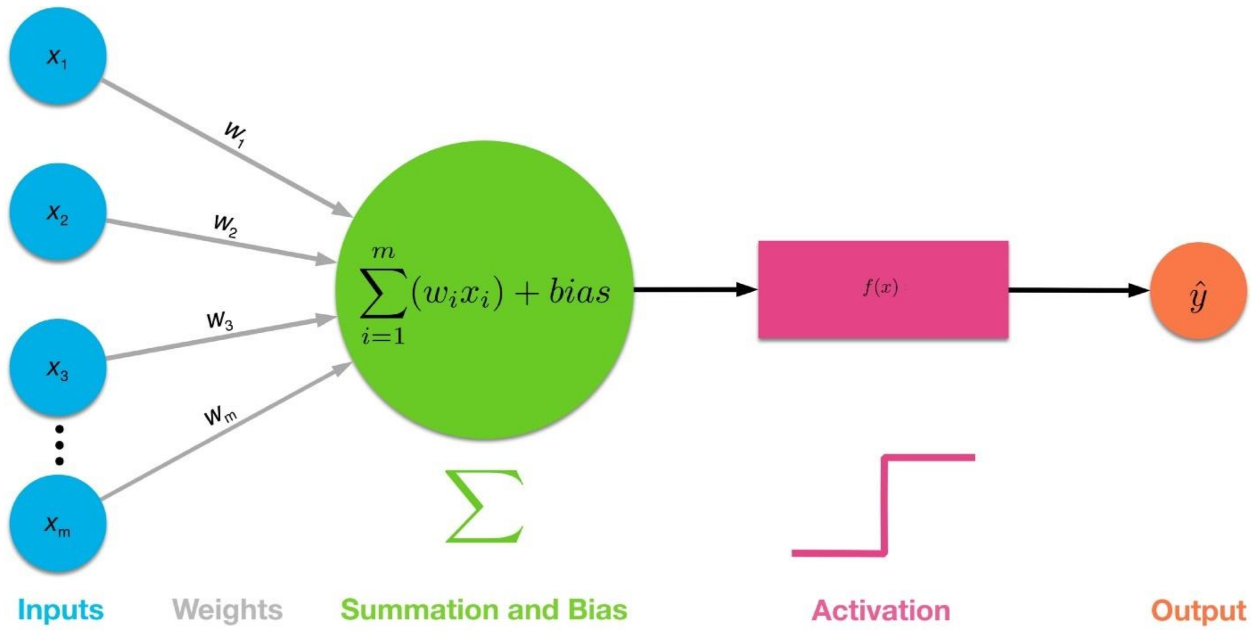

Figure 3.3: Schematic representation of Node [10]

The nodes are connected to one another by a connection link. Each connection link has a weight associated with it. The network learns from examples by changing the weights thereby achieving a good set of weights.

$$z_{in} = \mathrm{x1.\,w1} + \mathrm{x2.\,w2} + \ldots + \mathrm{xm.\,wm}(\text{total output})$$

$$\mathrm{Y{=}f}(z_{in})$$

Each node has an internal state called activation signal. The output of a node is the function of net input.

3.2 ANN Approach to build Prediction Model:

The processing of a neural network depends on 3 blocks:

1. Network topology
2. Activation function
3. Learning

3.2.1 Network topology:

Network topology is the arrangement of the network along with its nodes and connecting lines. The nodes and connection can be arranged in many ways. Feed forward networks with one hidden layer is used in this project.

Feed forward is a non-recurrent network having processing units in layers and all the nodes in a layer are connected with nodes of previous layer. There is no feedback loop, which means the signal can only flow in one direction, from input to output. An example of feed forward network is in fig 3.4.

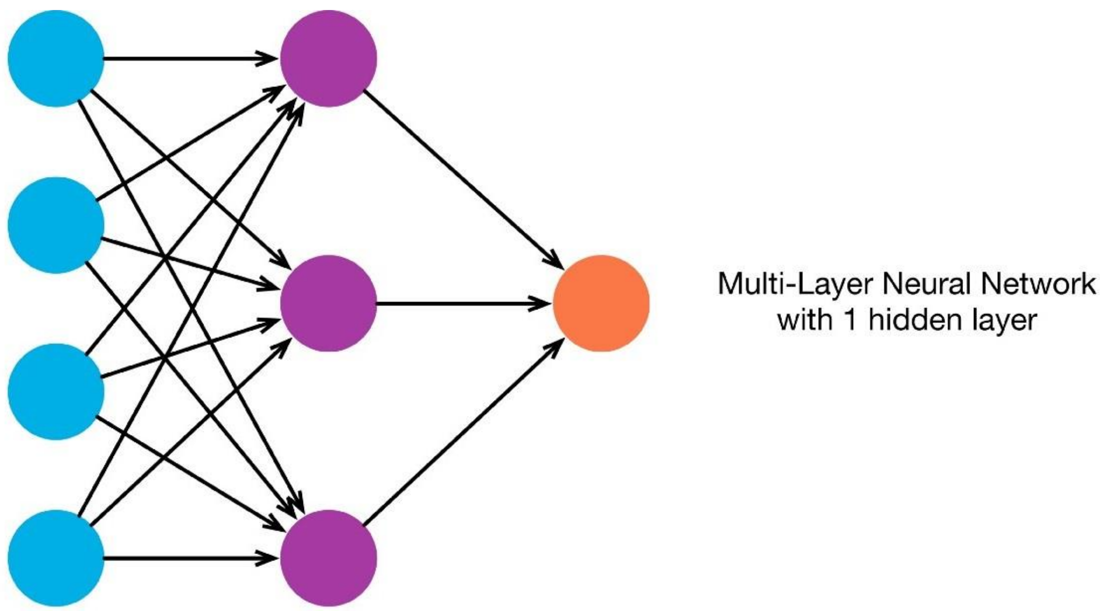

Figure 3.4: Example of a feed forward network [10]

Topology of the network used in this project is similar to the one from above figure. The network has 3 layers, first layer is the input layer that has 48*n nodes. The second layer is a hidden layer and contains 48*n nodes. The final layer has 1 node that output the predicted value.

3.2.2 Activation function:

This is the internal state of the node. The output of the node is a function of net input. An activation function can be anything that suits the operation. For the purpose of prediction linear neurons are used in this project.

Linear Activation Function:

In this case output is same as input.

$$f(x) = x$$

3.2.3 Learning:

Learning in neural networks is a method of modifying the weights of connection links between the neurons of specified network, to achieve a specific goal. The network changes the weights in accordance with the examples.

Delta learning rule:

Delta learning rule is used to train the network without hidden layers. The rule is simple, change the weights such that the root mean square error is minimum.

$$E = \sqrt{\frac{\sum(t-y)^2}{n}}$$

$$\frac{dE}{dw_i} = -x_i(t-y)$$

After each iteration

$$\Delta w_i \propto -\frac{dE}{dw_i}$$

Change in weight after each iteration is determined by the error derivative. Though the principle is same for networks with hidden units, the problem in applying these rules is that the error is not directly affected by hidden unit weights. So an algorithm called back propagation is developed to train networks with hidden layers.

Back propagation:

Back propagation is the algorithm that is used to train the networks with hidden layers. Delta rule cannot be applied to networks with hidden layers. The problem issue is how to learn the weights of the hidden layers. So instead of calculating change in error with weight, we calculate change in error with hidden unit value.

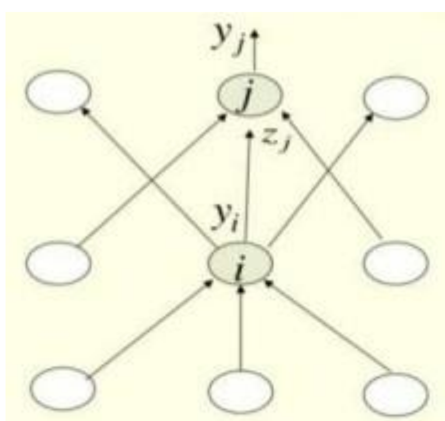

Figure 3.5**:** Network showing last 3 layers of the network [4]

As shown in above figure, say output layer is indicated with index j and hidden layer is indicated with index i then

y_j → Output of final layer

z_j → Input of final layer

y_i → Output of hidden layer

E → Root mean square error

E is defined as

$$E = \frac{\sum(t_j - y_j)^2}{2}$$

$$\frac{\partial E}{\partial y_i} = -(t_j - y_j) \qquad (3.1)$$

Using above equation the error derivatives of the final layer is determined.

z_j→ Total input to the final layer

For Linear neuron $z_j = y_j$

$$\frac{\partial E}{\partial y_j} = \frac{\partial E}{\partial z_j}$$

And $z_j = \sum w_{ij} * y_i$

Then

$$\frac{\partial E}{\partial y_i} = \sum \frac{\partial z_j}{\partial y_i} * \frac{\partial E}{\partial z_j}$$

And

$$\frac{\partial z_j}{\partial y_i} = w_{ij}$$

$$\Rightarrow \frac{\partial E}{\partial y_j} = \sum w_{ij} * \frac{\partial E}{\partial z_j} \tag{3.2}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial z_j}{\partial w_{ij}} * \frac{\partial E}{\partial z_j}$$

$$\Rightarrow \frac{\partial E}{\partial w_{ij}} = y_i * \frac{\partial E}{\partial z_j}$$

$w_{ij} \rightarrow$ Weight of the link connecting i and j

$$\Delta w_{ij} = (learning\ rate) * -\frac{\partial E}{\partial w_{ij}} \tag{3.3}$$

So implementation of back propagation involves 4 steps:

1. From inputs calculation of output (signal goes in forward direction).
2. Calculate the error at output layer using the equation (3.1).
3. Calculate the error derivative of previous layers using equation (3.2).
4. Change the weights according to the formula (3.3).

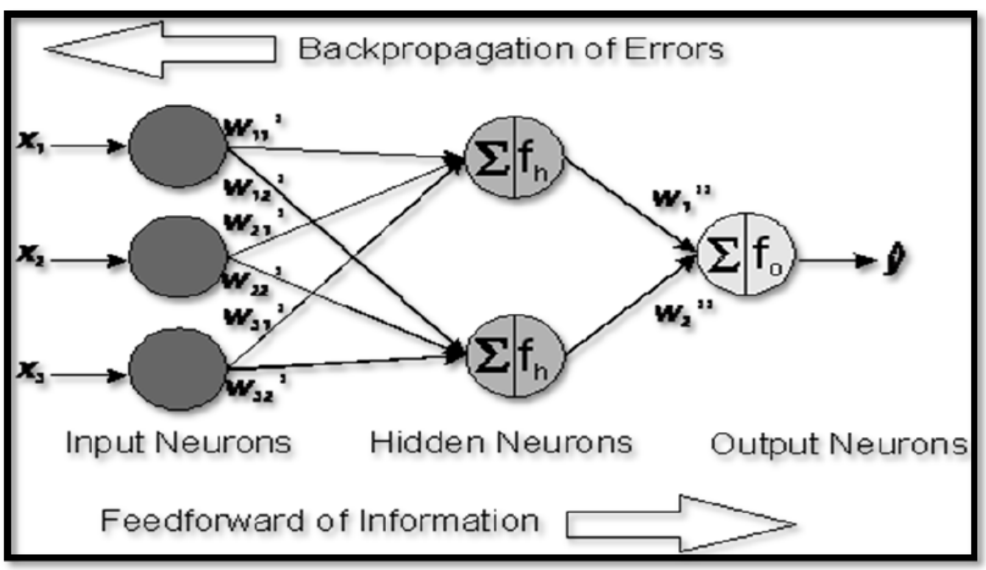

Figure 3.6: Figure shows how signal and error travels [9]

This is a feed forward network, signal flows from input to output. The error derivative of the final layer are determined and they travel back to the input layer, so this is called back propagation.

Appropriate learning rate has to be determined. Learning should not be too high or low. Below figure 3.7 illustrate the importance of determining learning rate. If learning is too low many iterations are needed to reach minimum error. If learning is too high the weights change rapidly

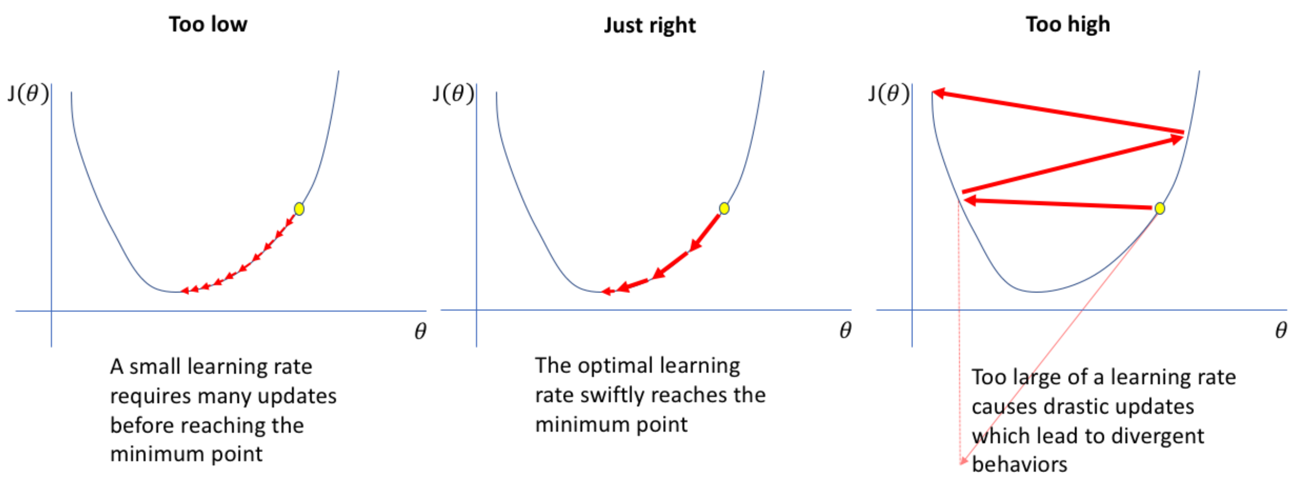

Figure 3.7: Shows how error changes with respect to learning rate [7]

3.3 Summary:

This chapter discusses about the neural networks, its components, algorithms. This also specifies the type of network used in this project and also derives the equation of how weights change with respect to error signal.

The neural networks were used to build the prediction model. The neural network contains 3 layers. The first layer consists of 48*n input nodes. The second layer is a layer of hidden units that contain 48*n nodes. The output layer consists of 1 node that outputs the predicted value. Back propagation algorithm is used to train the network.

Chapter 4
Methodology

Machine learning is widely used in many areas of research for deriving computational intelligence. It allows understanding of underlying behavior of complex systems. A neural network consists of three layers. The first layer consists of various features from dataset given as inputs. The second layer consists of some hidden layer and final layer consists of output. The main goal of applying neural network techniques here is to learn the behavior of the customer. The data .The processing of data involves few steps in completing the simulation.

The methodology involves 4 steps:

4.1 Data Cleaning:

The success of model is heavily dependent on data. Once data from smart meters are aggregated it arranged as a time series data. The data has to go through processing before being fed to the network. This process is called data cleaning. Data cleaning is a data mining process which focuses on identifying and correcting inaccurateness, incompleteness and inconsistencies in raw data. This is a preprocessing step. It involves few tasks like identifying missing or invalid values and detecting outliers.

The missing values can be identified by missing time stamp. Invalid points (in our case negative values) are identified by placing hard limits. The next step involves detecting outliers, mean and standard deviation (σ) are computed and all the values that do not lie within 3σ of mean are considered outliers. The assumption is that these outliers may correspond to peak energy consumption activities during holidays or special occasions. The following flow chart briefly describes data cleaning process.

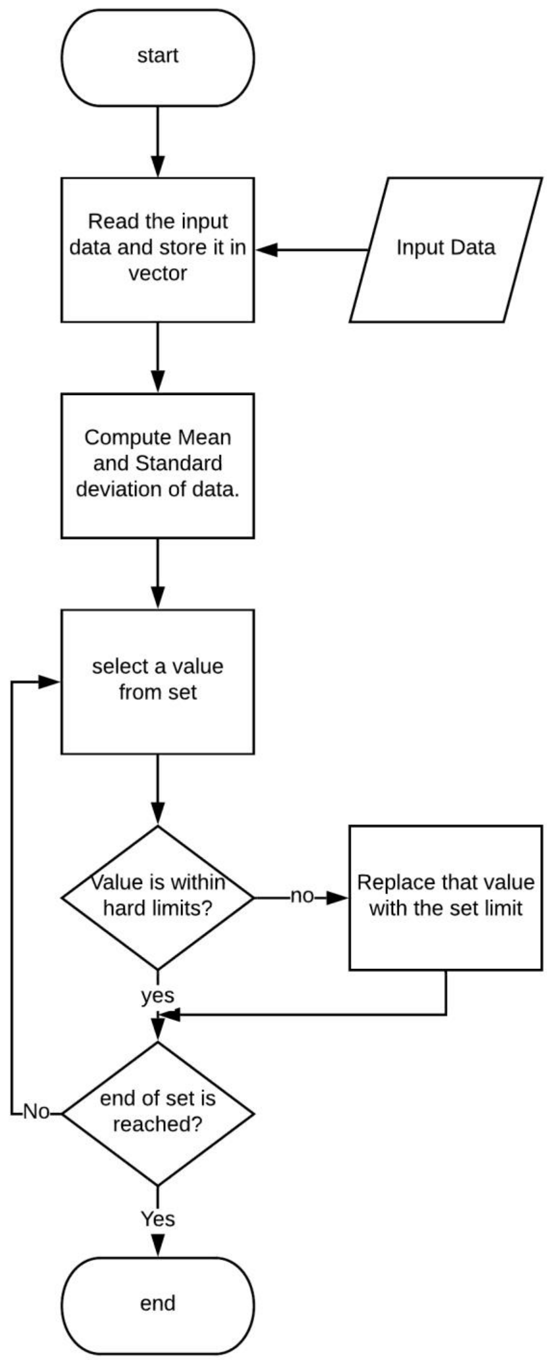

Figure 4.1: Flow chart of data cleaning process

4.2 Feature Selection:

One of the most important step is to select the features. Features are the values that are fed at the input layer of the network. The features are the energy consumption value of the past few days. As show in fig4.2the Energy consumed is predicted based on these values. The background knowledge of the type of load can help while selecting features. For example a commercial load that repeats pattern over week give best result when energy consumption values of past 7 days are considered as features.

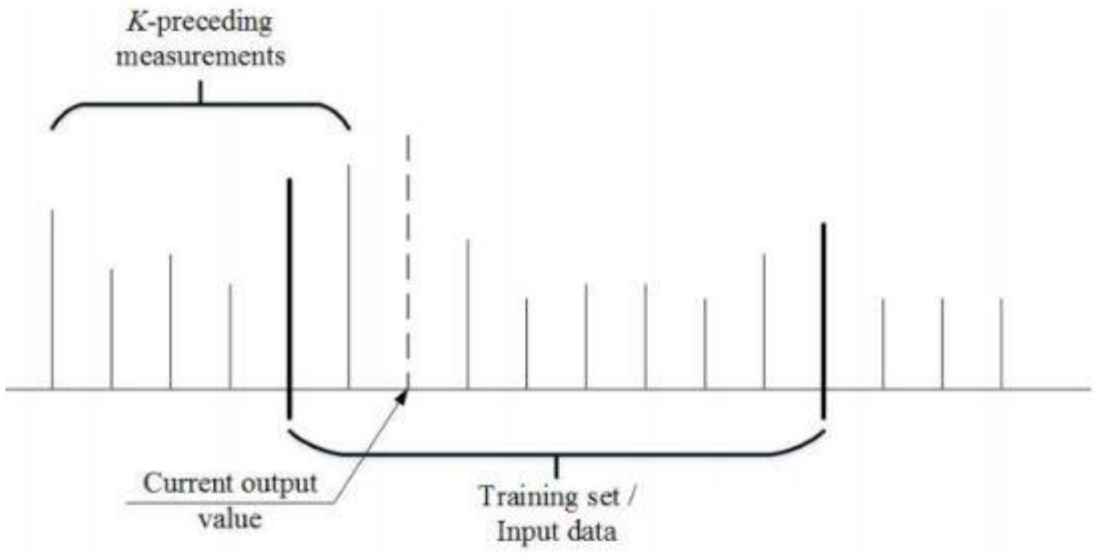

Figure 4.2: Time series of data [16]

4.3 Training of the Network:

After the features are selected, training set and validation set are prepared. The network is trained using 3 year training data. The network contains 1 hidden layer, as explained in section 3.2.1 and weights are initialized at random value. Back propagation algorithm is used to train networks with hidden layers. The network is fed with consecutive data points and error signal is generated. The weights change based on the error signal generated. After training a good set of weights are reached. Learning rate has to be determined in this stage, if learning rate is too high the values may exceed the system limits, so will not result in valid weights. Learning rate is determined using trial and error method.

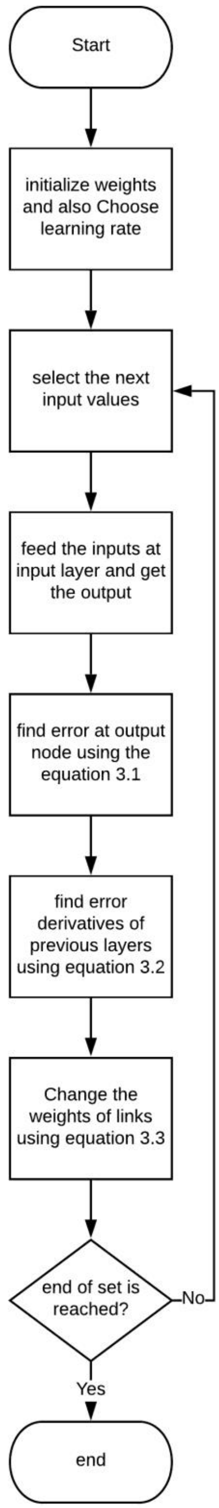

Figure 4.3: Flow chart of training process

4.4 Testing:

Once the network is trained, it has to be tested with validating set. Validating set is 1 year long. Root Mean Square Error serves as the deviation indicator between predicted value and the actual value.

$$RMSE = \sqrt{\frac{\sum(t-y)^2}{n}}$$

t is the actual value, y is the predicted value and n is the total number of instances.

Best prediction occurs when actual value is same as predicted value. So along with RMSE, correlation between actual value (t) and predicted value (y) can also be used to judge the prediction. Correlation coefficient is calculated using the following equation.

$$r = \frac{N\sum ty - (\sum y)(\sum t)}{\sqrt{(N\sum t^2 - (\sum t)^2)(N\sum y^2 - (\sum y)^2)}}$$

N →total number of samples

$\sum ty$ → Sum of the product of the actual value and predicted value

$\sum y$ → Sum of predicted values

$\sum t$ → Sum of actual values

$\sum t^2$ → Sum of the squares of actual values

$\sum y^2$ → Sum of the squares of predicted values

The following flow chart shows the steps involved in testing process.

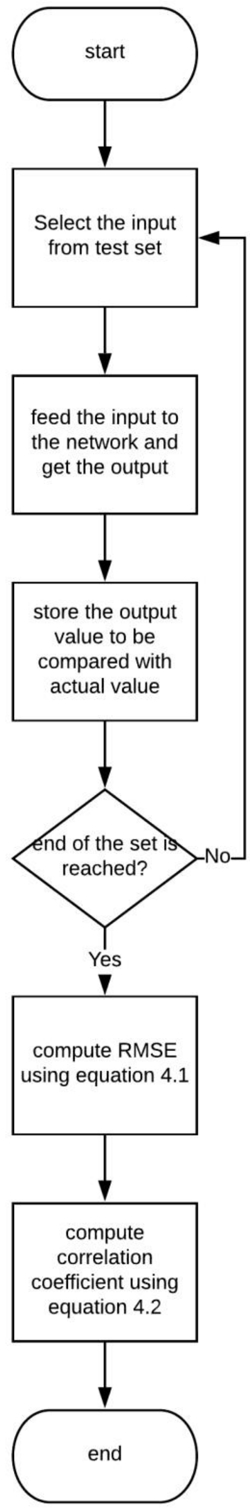

Figure 4.4: Flow chart of testing process

Chapter 5
Simulation results

As discussed in the chapter 4 the experiments involves 4 steps. The results are displayed in this chapter. Here predicted values are compared to the actual values. As mentioned earlier validating set contains the data of 1 year. As mentioned earlier smart meter records data for every 30 minute interval. So for a day we have 48 data points. For 1 year we have 17520 (48*365) points. The prediction is done based on the energy consumption values from previous days, so first few days will not have inputs and prediction cannot be made. So excluding those first few days there are 17520-features (number of data points from first few days) data points where predicted value and actual value can be compared.

As mentioned in chapter2 each profile area is given a codename based on the name of the local service provider. Below table gives details.

Table 5.1: Information about the load

Index	Profile Name	Profile area	Network/company name
1	CLOADNSWCE	COUNTRYENERGY	Essential Energy
2	CLOADNSWEA	ENERGYAUST	Ausgrid
3	CLOADNSWIE	INTEGRAL	Endeavour Energy
4	QLDEGXCL31	ENERGEX	Energex Limited
5	QLDEGXCL33	ENERGEX	Energex Limited
6	SACLOAD	UMPLP	SA Power Networks
7	NSLP	ACTEWAGL	Actew Distribution Ltd and Jemena Networks (ACT) P

7 loads are considered for the application. Each graph is divided into 3 each representing 4 months of the year. Simulations are done in the order mentioned in the table.

Simulation study 1:

This Simulation corresponds to load 1 as specified in table 5.1. The load forms a pattern over a week and considering 7 week values as features gives best results. Here no of features=48*7

Table 5.2: Information about simulation 1

	Actual Value	Predicted value
Mean	27.66	25.9
Standard Deviation	29.42	27.47

Table 5.3: Performance Indicators for simulation 1

Root Mean Square Error (RMSE)	6.4088
Correlation coefficient (r)	0.9789

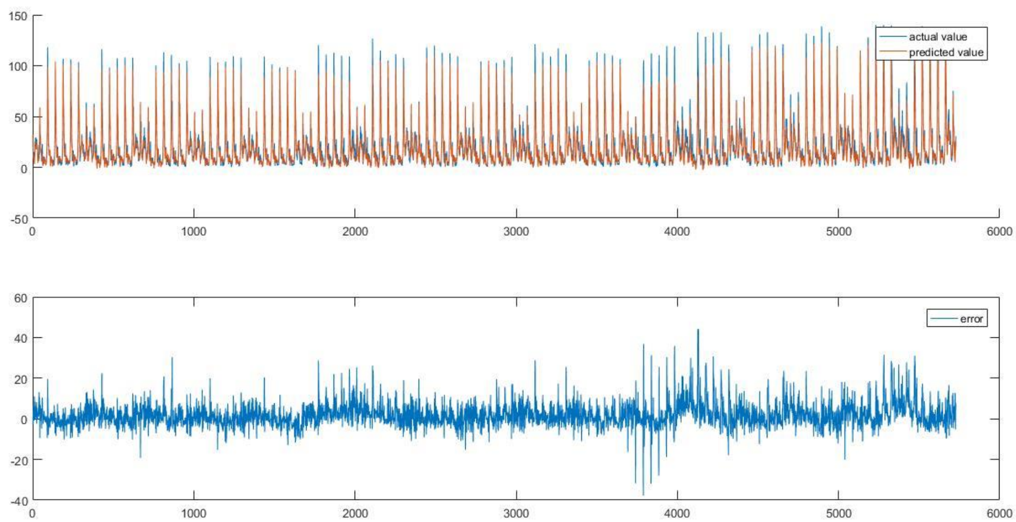

Figure 5.1: January- April of load 1

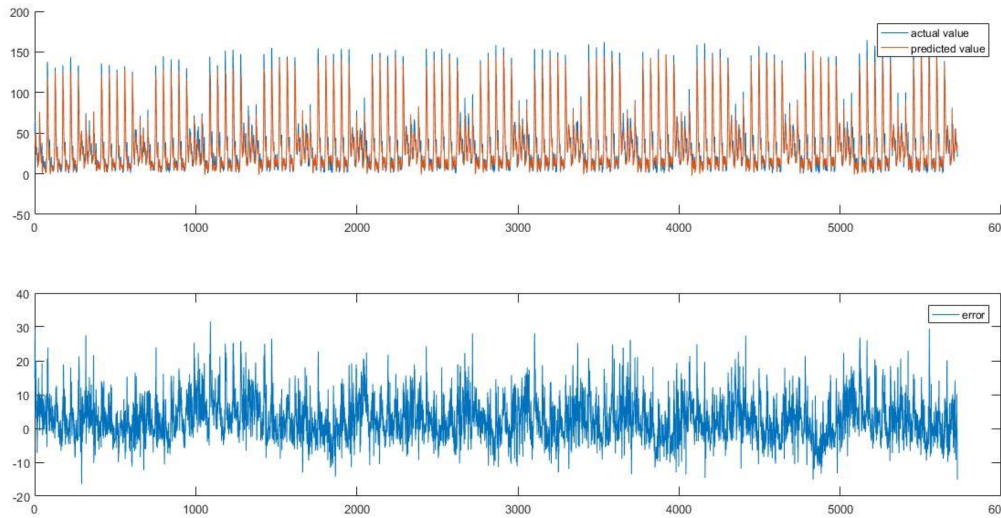

Figure 5.2: May- August of load 1

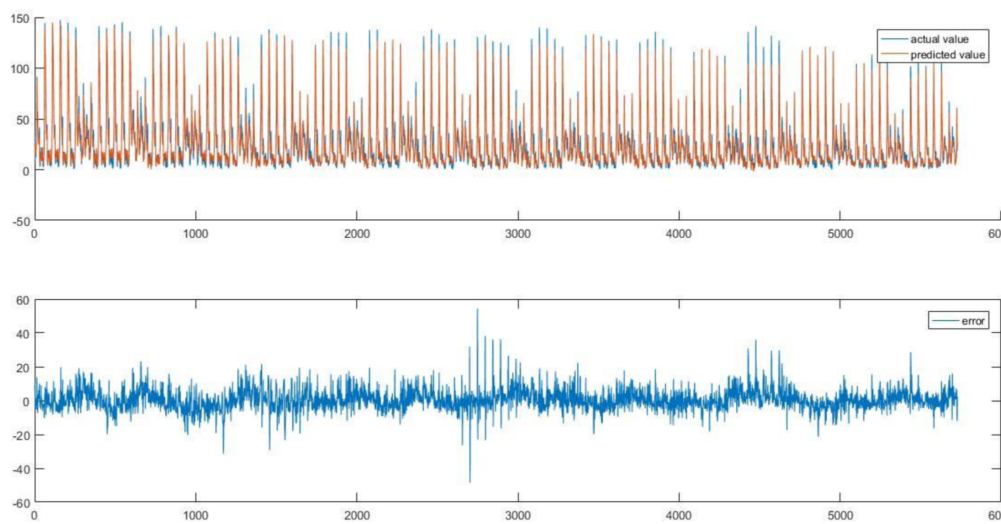

Figure 5.3: September- December of load 1

This is a load that repeats weekly so selection of 7 day data as features gives best output.

Simulation study 2:

This Simulation corresponds to load 2 as specified in table 5.1. The load seems to repeat itself on daily basis. so here no of features=48*2

Table 5.4: Information about simulation 2

	Actual Value	Predicted value
Mean	30.54	30.8
Standard Deviation	37.67	36.58

Table 5.5: Performance Indicators for simulation 2

Root Mean Square Error (RMSE)	6.0852
Correlation coefficient (r)	0.987

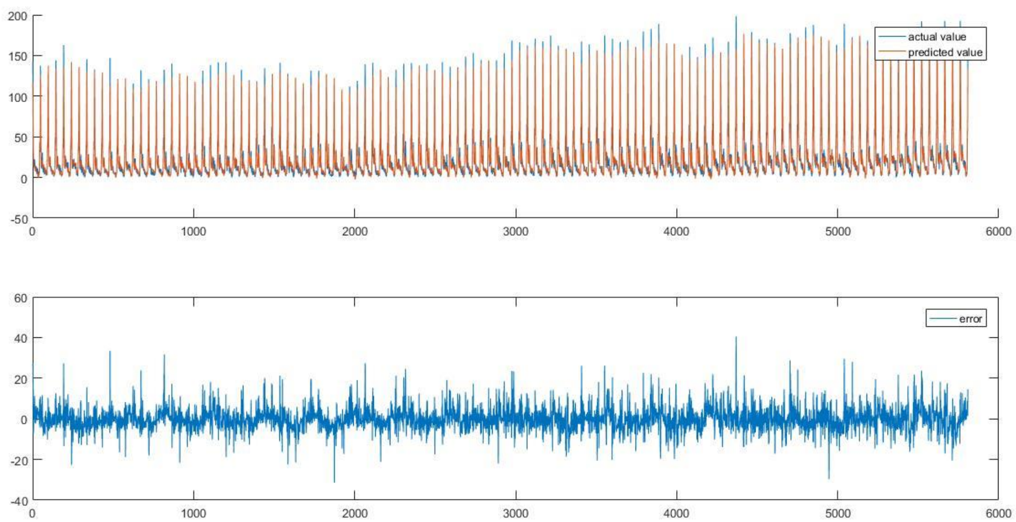

Figure 5.4: January- April of load 2

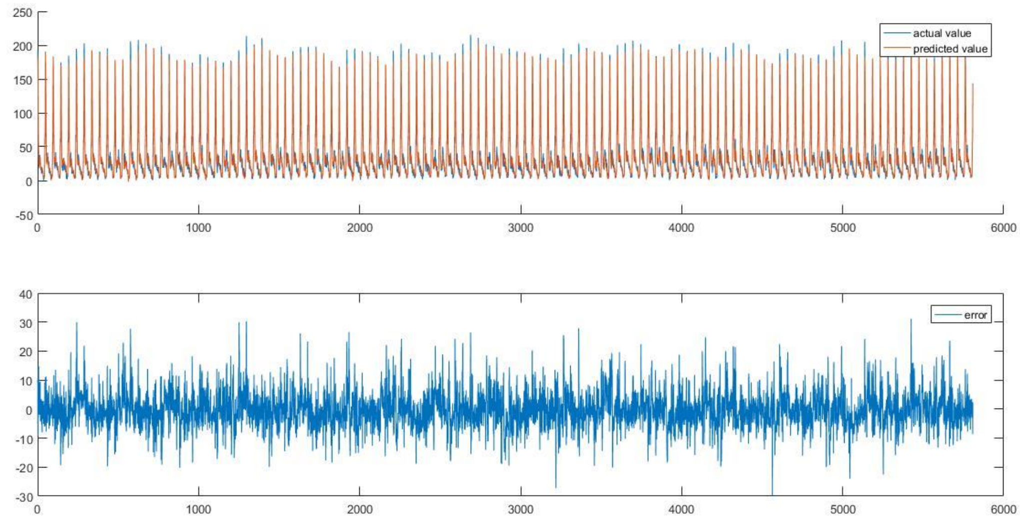

Figure 5.5: May- August of load 2

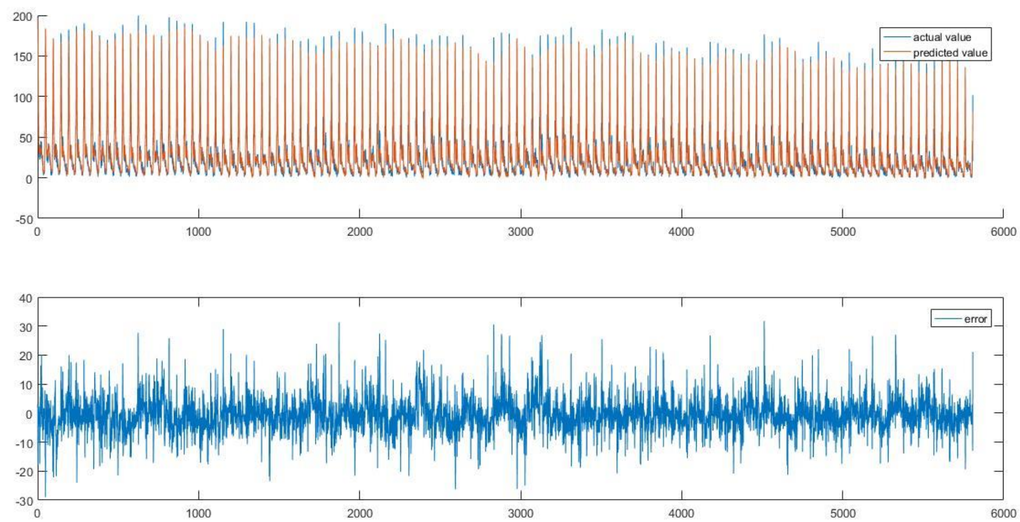

Figure 5.6: September- December of load 2

This load has a pattern over a day. So considering 2 day load gives best results.

Simulation study 3:

This Simulation corresponds to load 3 as specified in table 5.1. There are many undetected peaks, which are not present in the training set. These values contribute to the increase in error Here no of features=48*2. As more features are includes the error increases.

Table 5.6: Information about simulation 3

	Actual Value	Predicted value
Mean	56.39	56.49
Standard Deviation	23.68	22.11

Table 5.7: Performance Indicators for simulation 3

Root Mean Square Error (RMSE)	9.7876
Correlation coefficient (r)	0.919

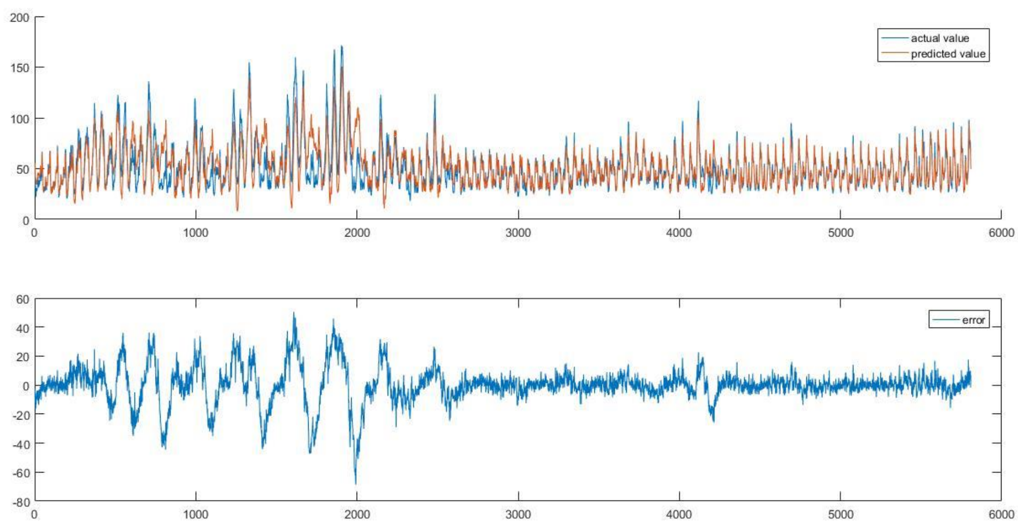

Figure 5.7: January- April of load 3

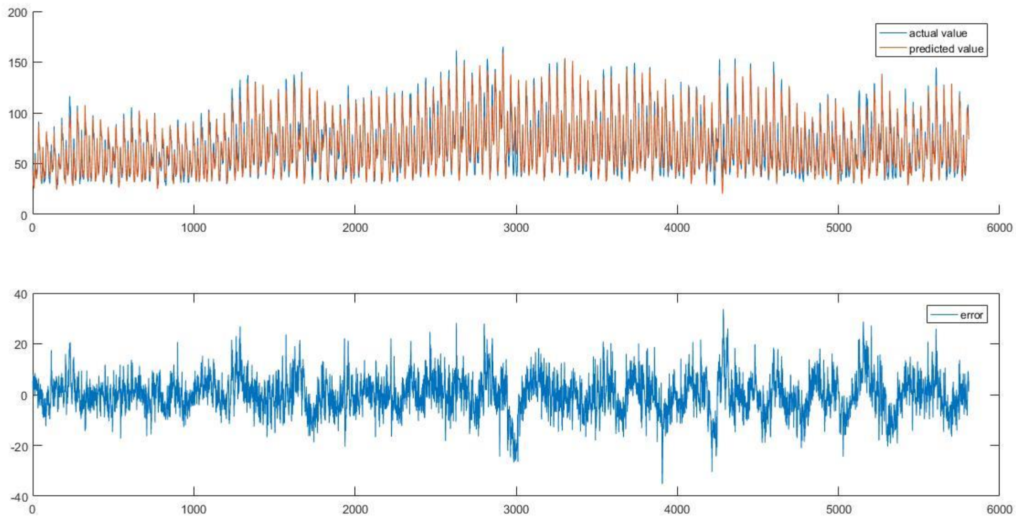

Figure 5.8: May- August of load 3

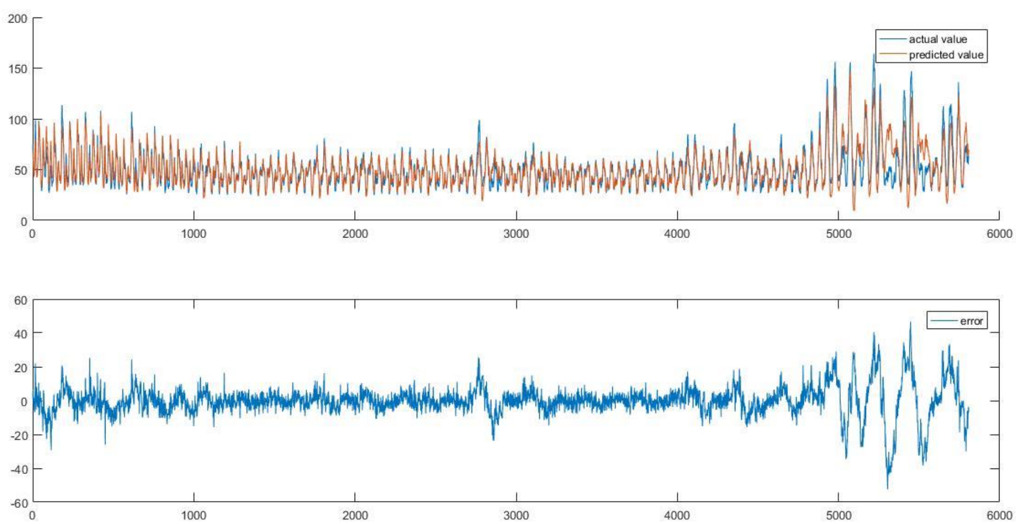

Figure 5.9: September- December of load 3

This load as observed contains many peaks in January and December. These peaks are do not appear in the training set, the training set carries the information of average increase of load in mid-year. The more the features are the more the peaks influence the outcome and the worse the performance becomes. Selecting 48*2 values as features gives the best result.

Simulation study 4:

This Simulation corresponds to load 4 as specified in table 5.1. The algorithm gives good predictions in this case. The load has a well-defined pattern in terms of usage. Here no of features=48*2

Table 5.8: Information about simulation 4

	Actual Value	Predicted value
Mean	28.68	27.27
Standard Deviation	40.98	38.36

Table 5.9: Performance Indicators for simulation 4

Root Mean Square Error (RMSE)	6.0814
Correlation coefficient (r)	0.9911

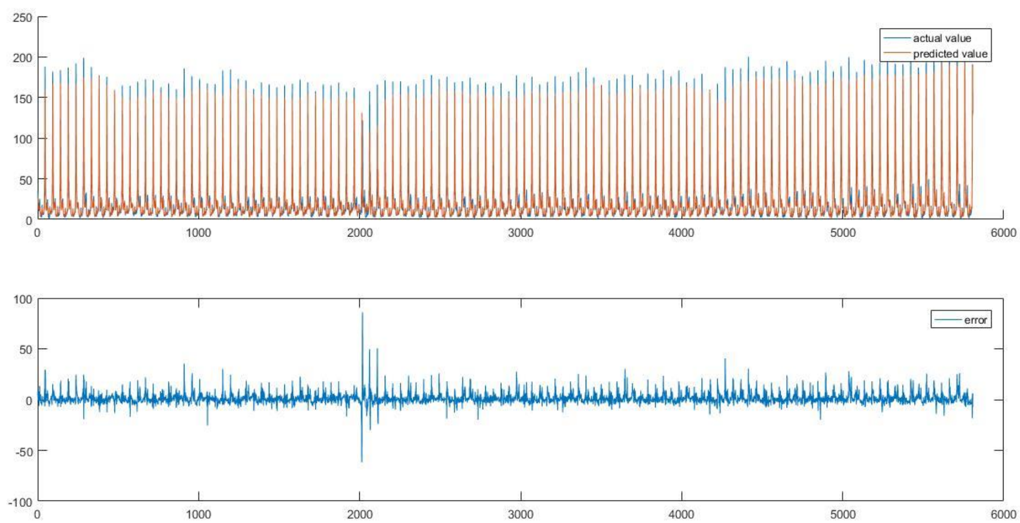

Figure 5.10: January- April of load 4

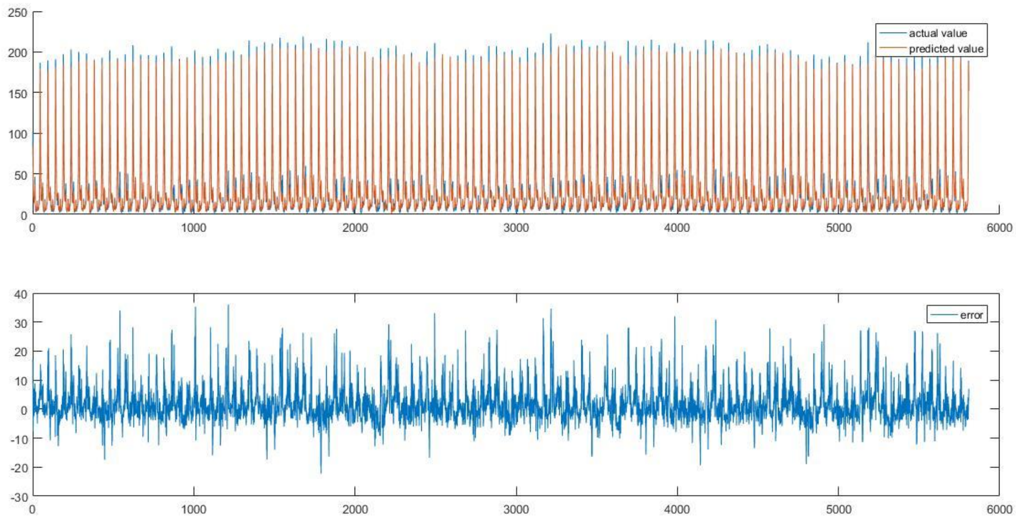

Figure 5.11: May- August of load 4

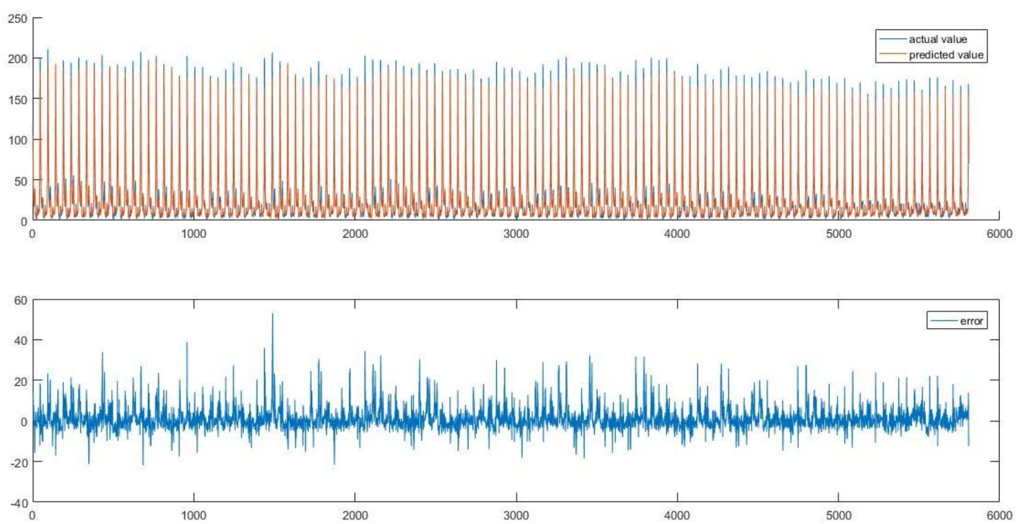

Figure 5.12: September- December of load 4

This load repeats itself daily, selecting 48*2 consumption values as features gives best result.

Simulation study 5:

This Simulation corresponds to load 5 as specified in table 5.1. Similar to load 3 this load also contains many unrecognized peaks which has most contribution to the error. Here no of features=48*2

Table 5.10: Information about simulation 5

	Actual Value	Predicted value
Mean	27.55	26.9
Standard Deviation	14.88	13.6

Table 5.11: Performance Indicators for simulation 5

Root Mean Square Error (RMSE)	5.4734
Correlation coefficient (r)	0.9339

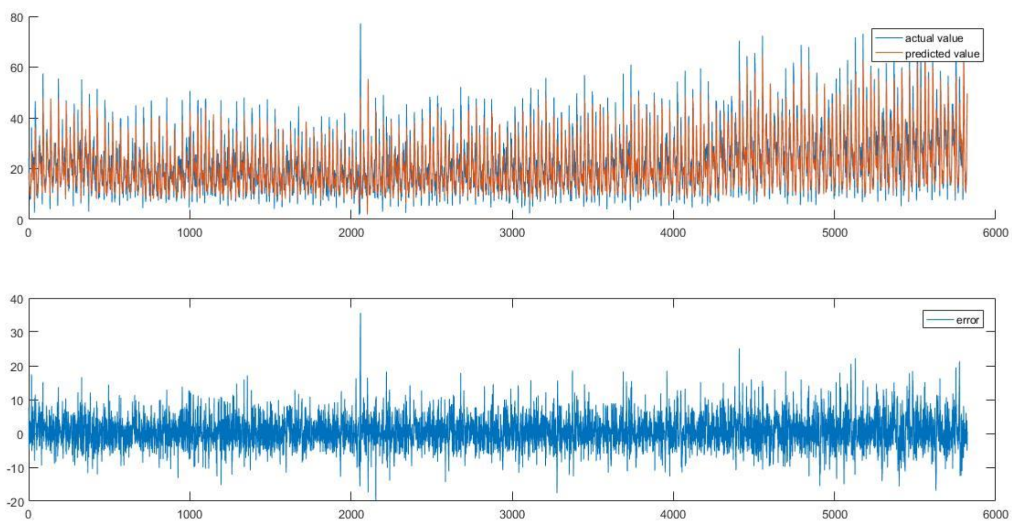

Figure 5.13: January- April of load 5

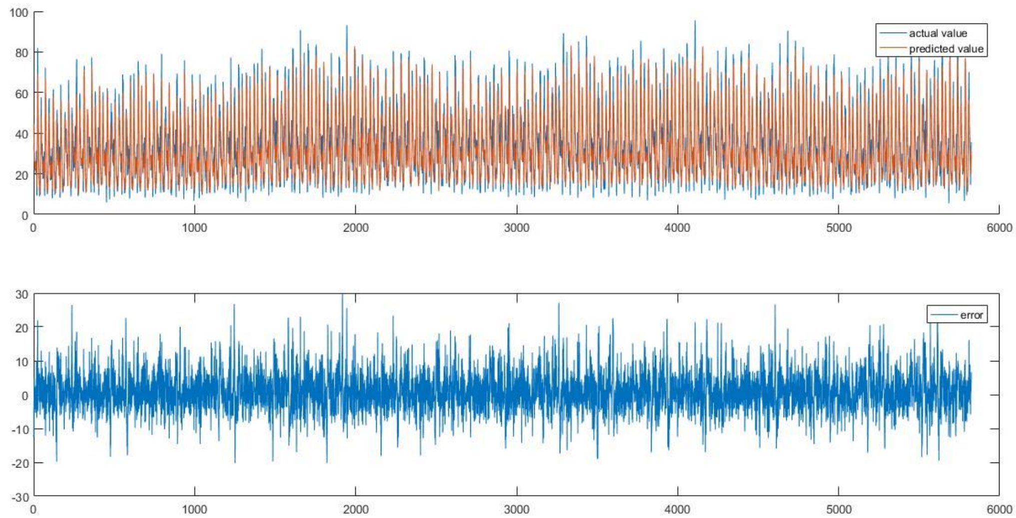

Figure 5.14: May- August of load 5

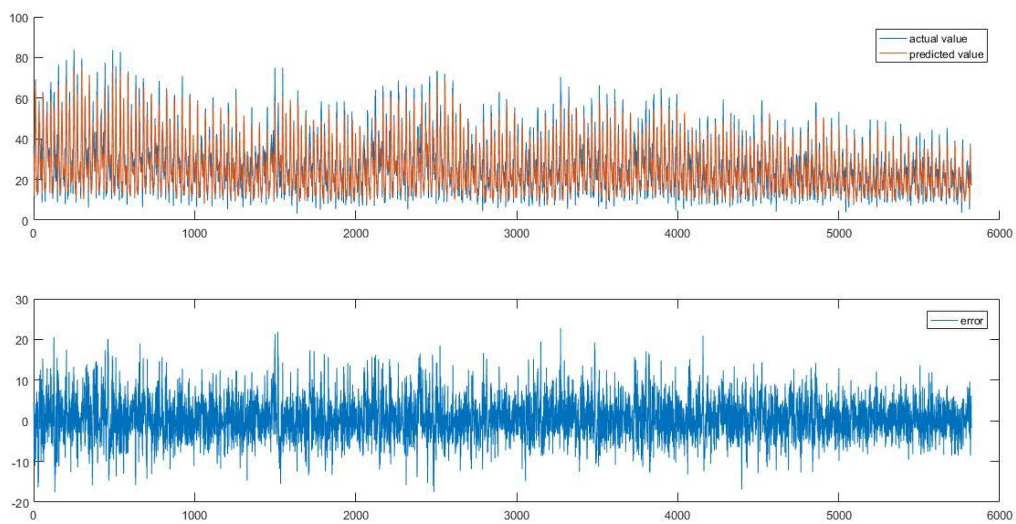

Figure 5.15: September- December of load 5

In this case selecting 48*2 values as features gives best results.

Simulation study 6:

This Simulation corresponds to load 6 as specified in table 5.1. The algorithm gives best results in this case. Here no of features=48*2

Table 5.12: Information about simulation 6

	Actual Value	Predicted value
Mean	22.07	22.89
Standard Deviation	34.41	34.47

Table 5.13: Performance Indicators for simulation 6

Root Mean Square Error (RMSE)	4.4192
Correlation coefficient®	0.993

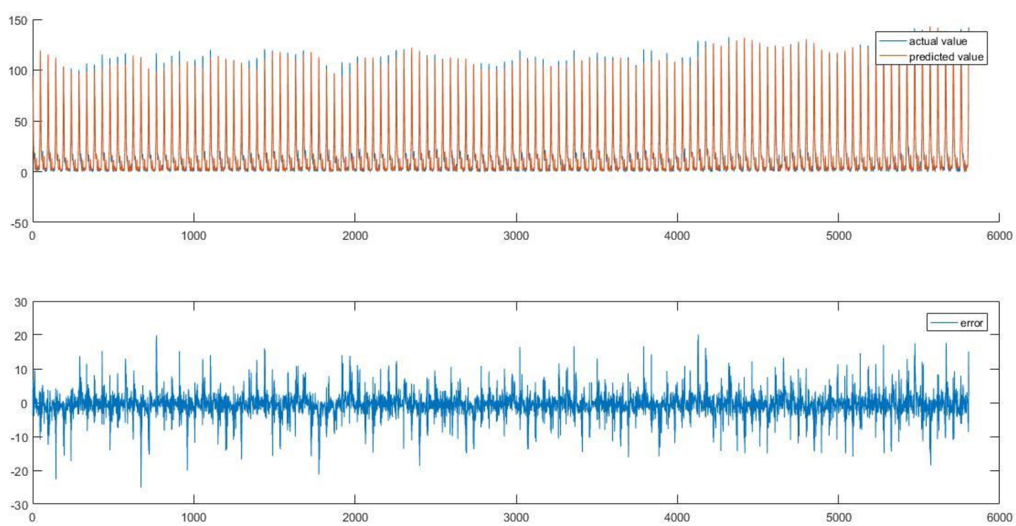

Figure 5.16: January- April of load 6

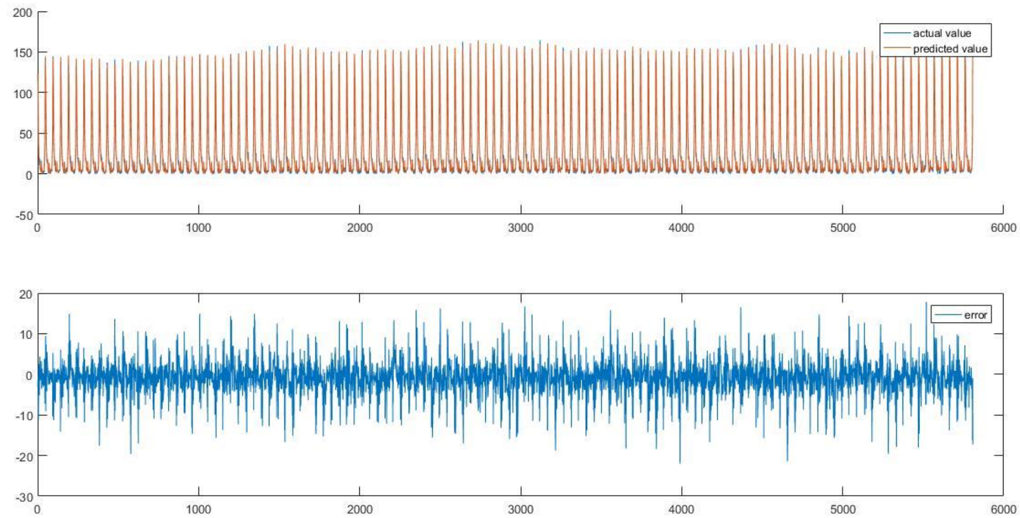

Figure 5.17: May- August of load 6

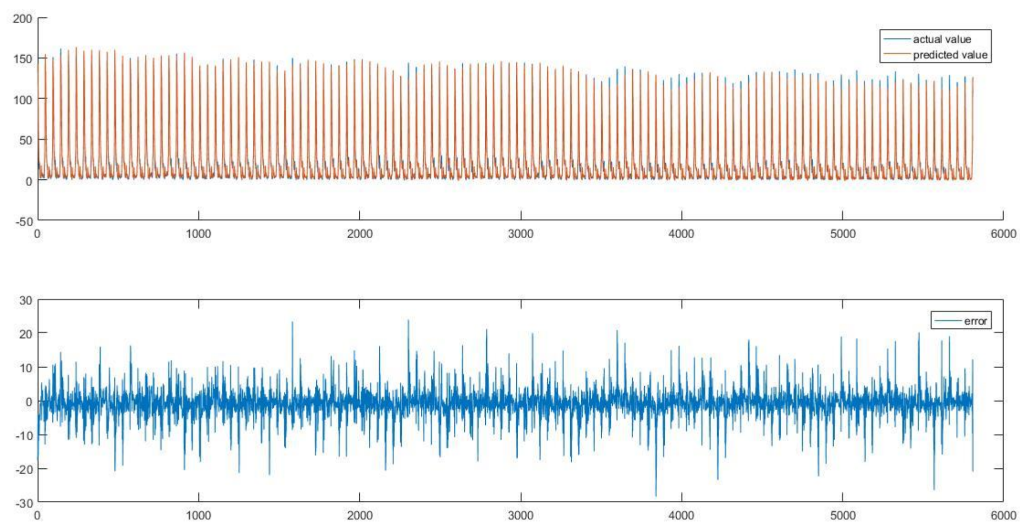

Figure 5.18: September- December of load 5

Mostly pattern repeat daily selecting 48*2 values as features gives best results.

Simulation study 7:

This Simulation corresponds to load 7 as specified in table 5.1. This load is different from others, as other loads are controlled this load corresponds to a larger area and hence has higher values when compared with others. Here no of features=48*1.

Table 5.14: Information about simulation 7

	Actual Value	Predicted value
Mean	78520	78160
Standard Deviation	36270	35540

Table 5.15: Performance Indicators for simulation 7

Root Mean Square Error (RMSE)	6641
Correlation coefficient (r)	0.9832

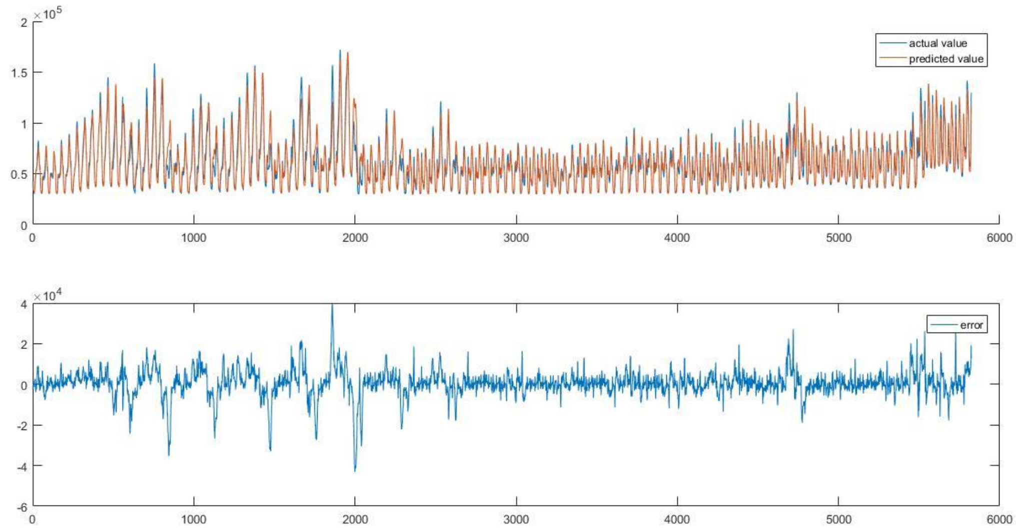

Figure 5.19: January- April of load 7

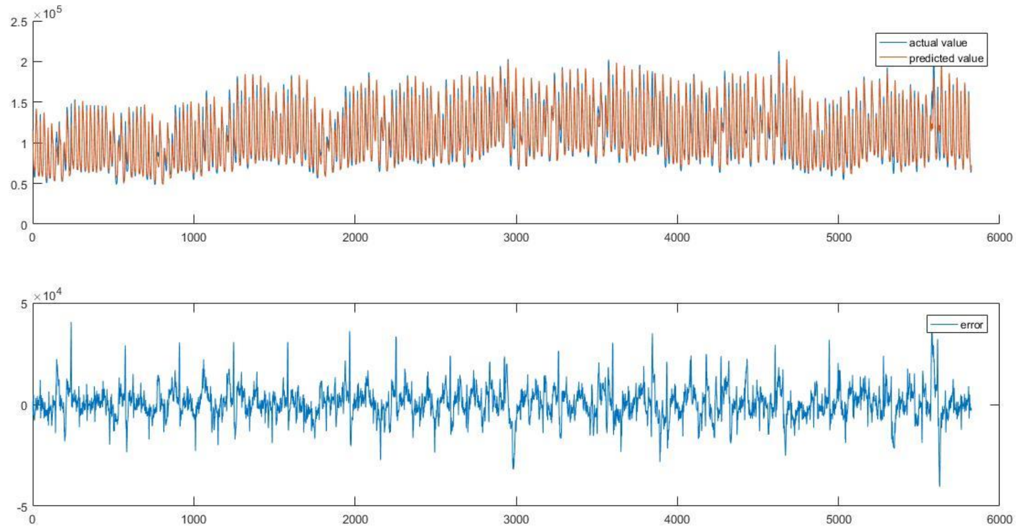

Figure 5.20: May- August of load 7

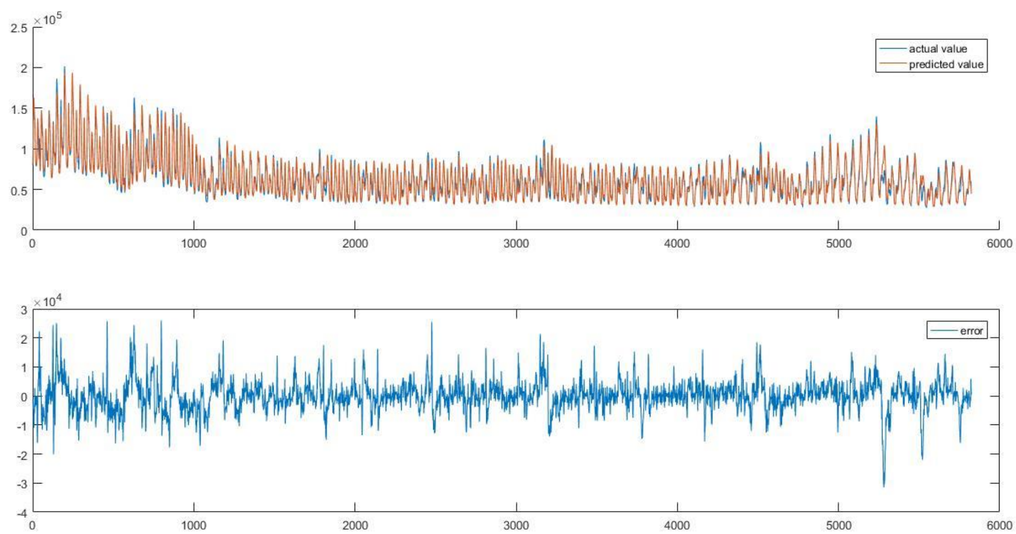

Figure 5.21: September- December of load 7

This is a large scale load, compared to others, considering 48 values as features gives the best results here.

Simulation Comparisons:

The simulations can be compared with one another. Correlation coefficient serves as a good indicator of performance as it does not vary a drastically as RMSE. Below table compares performance of the network on different types of load.

Table 5.16: Simulation results comparison

Load	Correlation coefficient
Load 1	0.9789
Load 2	0.987
Load 3	0.919
Load 4	0.9911
Load 5	0.9339
Load 6	0.993
Load 7	0.9832

It can be observed that performance of algorithm on load 3 and load 5 is not great, the main reason for this is that there are many unrecognized peaks that were not present in training set. Though the network was able to recognize the regular pattern these new peaks contributed in the systematic increase in error. The network performs better when trained with cleaned data rather than raw data.

Chapter 6
Conclusion and Future works

This research demonstrates that the neural networks can be used as a computational intelligence tool for short term forecasting. Real world historical energy consumption data from various periods of time were analyzed to build the energy consumption profile of the consumer and then the profile is used to predict the energy consumption.

The model uses smart meter data, based on results it can be concluded that there is high potential in predicting the short term load. It can be seen that the predicted values and actual values are highly correlated. The main advantage of this model is that input values selected makes the model robust to sudden changes in weather. The selection of inputs significantly affect the accuracy of the forecast. Background knowledge of the load can be helpful in the selection of features.

There is great potential to build various model that are based on smart meter data. As smart meters are becoming popular in the developing countries, there are great opportunities to explore this data in order to handle the demand side of power sector.

Several improvements can be made to the model. Learning is determined by trial and error method, other methods to determine learning rate can be explored. Also input layer and hidden layers can be better tuned to make better predictions. The model can be improved to deal with individual households rather than areas. Those predicted values can be used to place bids in active markets. This paper established has how smart meter data can be useful to make predictions. Similarly other computational intelligence tools can be used to extract intelligence and compare them with neural network prediction model.

Appendix A

This section contains the codes used in running the simulations. The code have 3 parts each performing one task specified in the chapter 4. The codes perform 4 operations data reading, data cleaning, training the network, testing the network. All four steps are mentioned in below sections.

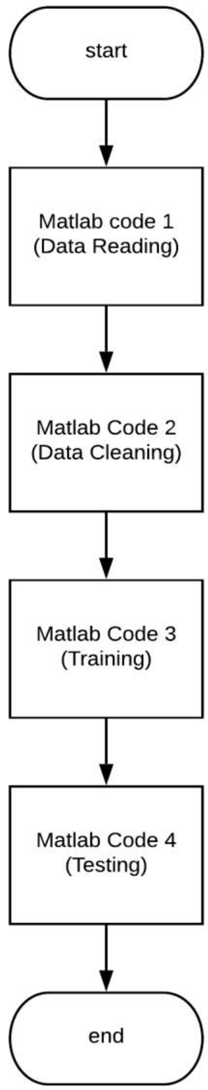

Figure A.1: Flow chart describing the entire process

Matlab Code 1:

This code below reads the data from data files. In data file the data is organized in such a way that each row conations data corresponding to one day i.e. 48 values and number of columns represent total number of days being considered. The data is stored in a vector.

```
clearall;
closeall;
clc;
M = csvread('C:\Users\student\Desktop\trail2.csv'); %reading
training set file%
B = transpose(M);
V = B(:); %V vector holds the training set values%
M = csvread('C:\Users\student\Desktop\trail3.csv'); %reading
test set file%
B = transpose(M);
       V1 = B(:); %vector V1 holds the values of test set%
```

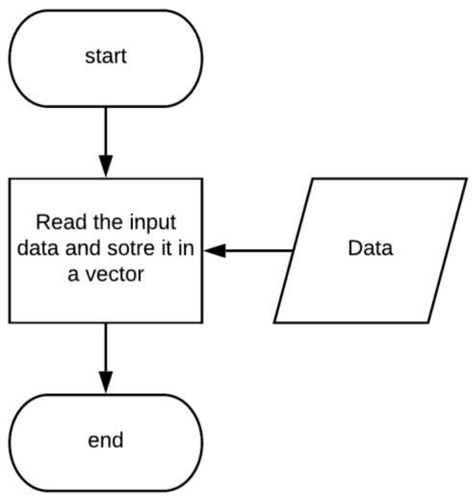

Figure A.2: Flow chart of data reading process

Matlab Code 2:

This part of code performs the action of data cleaning. This step involves detecting outliers. As mentioned in chapter 4, all values outside set hard limits are considered outliers. The outliers once detected are replaced by the limits. Below is the code and flow representing the code.

```
i=1;
c1=0;%c1 keeps count of the outliers%
mean=sum(V)/length(V);%calculates mean of the training data%
sd=std(V); %calculates the standard deviation of training set%
whilei<length(V)
if V(i)>mean+3*sd%value more than mean+3*sd is considered
outlier%
V(i)=mean+3*sd;
        c1=c1+1; %c1 is updated if a outlier is detected%
end
if V(i)<mean-3*sd%value outside range is considered outlier%
V(i)=mean-3*sd;%outlier value is changed to the stay within the
limits%
        c1=c1+1; %c1 is updated if a outlier is detected
end
i=i+1;
end
%after this step all outliers are detected and modified
```

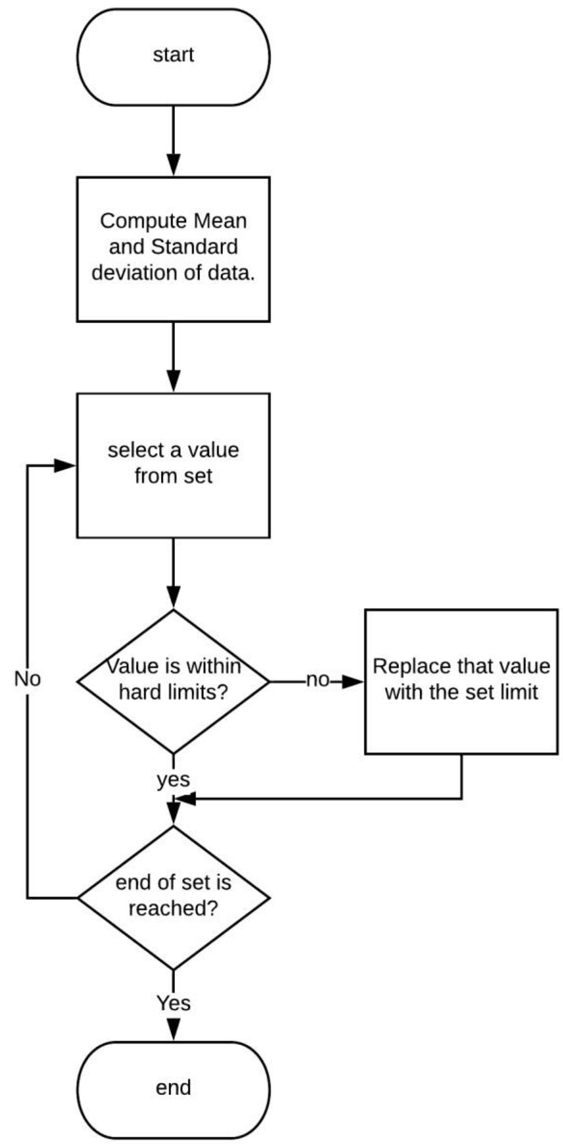

Figure A.3: Flow chart of data cleaning process

Matlab code 3:

This step involves training the network. Back propagation algorithm steps are implemented in this step. The learning rate and features are also determined in this part of code. Weights are initialized at random value and inputs are feed to the network. Weights change based on the error between expected output and actual output of the network. After this step good set of weights are reached.

```
learning_rate=1/1000000;
%{
learning is determined by trial and error
Too large value will not give any result as values exceed the
limit
Too small will result in no much change from initialization
%}
features=48*7;
%features are based on the background knowledge of the load
no_hidden_layers=1;
%no of hidden layers can be adjusted here
W=zeros(features,features,no_hidden_layers)+0.001;
%weights are initialized
final_layer_weights=zeros(1,features)+0.01;
layers=zeros(features,no_hidden_layers+1);
error=zeros(features,no_hidden_layers+1);
%error matrix includes error derivative values for all layers.
j=features;
i=1;
A=V;
while j<length(V) %loop runs till end of data set is reached
layers(:,1)=V(i:j); %input values
    k=1;
while k<=no_hidden_layers
        layers(:,k+1)=W(:,:,k)*layers(:,k); %the signal
propagates forward
        k=k+1;
end
    y=final_layer_weights*layers(:,no_hidden_layers+1); %output
of final layer
    t=V(j+1); %target value
err=y-t; %error signal
error(:,no_hidden_layers+1)=err*transpose(final_layer_weights);
%error derivatives of final layers are determined.
    k=no_hidden_layers;
while k>=1
        error(:,k)=transpose(W(:,:,k))*error(:,k+1); %error
propagates back to the input layer
```

```
        k=k-1;
end
    k=1;
%till this step error derivatives are determined
while k<=no_hidden_layers
        W(:,:,k)=W(:,:,k)-
learning_rate*transpose(layers(:,k)*transpose(error(:,k+1)));
%weights are changed from input layer
        k=k+1;
end
    final_layer_weights=final_layer_weights-
learning_rate*err*transpose(layers(:,no_hidden_layers+1));
% weights of the final layer are changed
i=i+1;
    j=j+1;
end
```

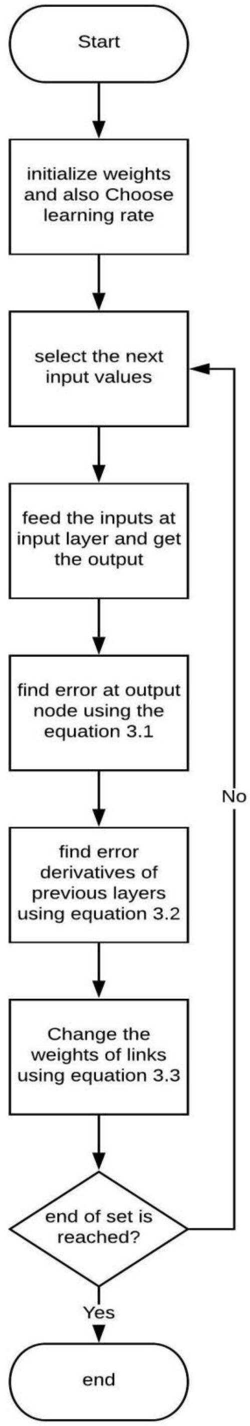

Figure A.4: Flow chart of training process

Matlab code 4:

Once the network is trained this part of the code test the network using test set. Similar to the previous step inputs are feed to the network and output of the network is compared with expected output to judge the accuracy of the prediction model. RMSE and correlation coefficient are used as parameters to judge the prediction model.

```
j=features;
i=1;
merr=0; %merr holds the value of summation of squared error
A=V1;
while j<length(V1) %loop runs till end of validation set is
reached
layers(:,1)=V1(i:j); %input values
    k=1;
while k<=no_hidden_layers
        layers(:,k+1)=W(:,:,k)*layers(:,k);
        k=k+1;
end
    y=final_layer_weights*layers(:,no_hidden_layers+1); %output
value is calculated
    t=V1(j+1); %t holds the target value
A(j+1)=y; %the predicted value are stored in vector A
merr=merr+(t-y)*(t-y);%calculation of root mean square error%
i=i+1;
    j=j+1;
end
disp(sqrt(merr/(length(V1)-features))); %displays root mean
squared error
plot(V1(features+1:end)); %plots actual values
holdon;
plot(A(features+1:end)); %plots predicted values
legend({'actual value','predicted
value'},'Location','northeast')
figure(2);
plot(V1(features+1:end)-A(features+1:end)); %plots the error
y1=A(features+1:end);% y1 holds the predicted values
y2=V1(features+1:end);% y2 holds the actual values
xy=y1.*y2;% hold of the product of the actual and predicted
values
N=length(y1);% total number of samples
r=((N*sum(xy))-(sum(y1)*sum(y2)))/sqrt((N*sum(y1.*y1)-
sum(y1)*sum(y1))*(N*sum(y2.*y2)-sum(y2)*sum(y2)));% r holds the
value of the Pearson correlation coefficient
%correlation coefficient is a good factor that shows how similar
2
```

```
%variables are.
```

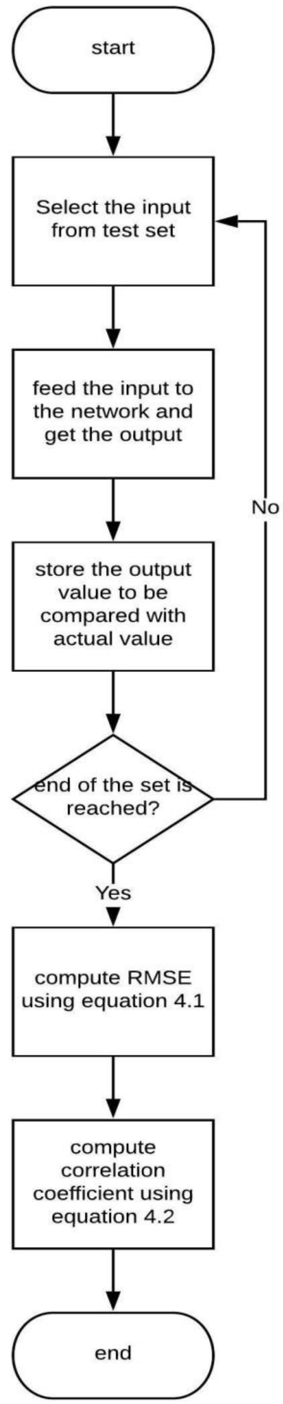

Figure A.5: Flow chart of testing process

References

1. Cristina Heghedus, AntorweepChakravorty andChunmingRong. 2018. “Energy Load Forecasting Using Deep Learning”. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8403442

2. Gouri R. Barai, Sridhar Krishnan and BalaVenkatesh. 2015. “Smart Metering and Functionalities of Smart Meters in Smart Grid - A Review”. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7379940

3. Harley, R.G. and Liang, Jiaqi. 2011. “Computational Intelligence in Smart Grids”.

4. https://stats.stackexchange.com/questions/267486/backpropagation-algorithm-in-neural-networks-nn-with-logistic-activation-funct

5. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm

6. JahnaviMahanta. 2017. “Introduction to Neural Networks, Advantages and Applications”. Last modified Jul 9, 2017. https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207

7. Jermy Jordan. 2018. “Setting of learning rate of your neural network”. Last modified Mar 1, 2018 https://www.jeremyjordan.me/nn-learning-rate/

8. Joseph Ditzler. 2016. “Pacific Power moving to smart meters”. Last modified Apr 13, 2016. https://www.bendbulletin.com/business/4205064-151/pacific-power-moving-to-smart-meters

9. Mekic, Ensar&Mekic, Emina&Ilgun, Erkan.2014. “Application of Ann in Australian Credit Card Approval”. European Researcher. 69. 334-342. https://www.researchgate.net/figure/Feed-forward-back-propagation-network-architecture_fig1_266951890

10. Nahua Kang. 2017. “Multi-level Neural Networkwith sigmoid function- Deep learning for Rookies (2)”. Last modified jun 27, 2017. https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f

11. ShubSaxena. 2017. “Artificial Neural Networks (Basics)| Introduction to Neural Networks”. Last modified Oct 26, 2017 https://becominghuman.ai/artificial-neuron-networks-basics-introduction-to-neural-networks-3082f1dcca8c

12. S.M. Sulaiman, P. ArunaJeyanthy and D. Devaraj. 2016. “Artificial Neural Network based Day Ahead Load Forecasting using Smart Meter Data”. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7516422

13. Stefan Hosein and Patrick Hosein. 2017. “Load Forecasting using Deep Neural Networks”. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8085971

14. Ujjwalkarn. 2016. “A Quick Introduction to Neural Networks”. Last modified Aug 9, 2016. https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/

15. Understanding load profiles. Last modified Aug 2013. https://www.aemo.com.au/-/media/Files/PDF/MTMA1772v0055UnderstandingLoadProfilesPublishedfrom-MSATS.pdf

16. Vitaly Ford, AmbareenSiraj and William Eberle. 2014. “Smart Grid Energy Fraud Detection using Artificial Neural Networks”. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7011557

17. Weicong Kong, Zhao Yang Dong, YouweiJia, David J. Hill, Yan Xu, Yuan Zhang. 2018. “Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network” https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8039509