YAŞAR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER’S THESIS

ANN BASED ELECTRICITY CONSUMPTION FORECASTING IN YASAR UNIVERSITY

TUTKU ÇİMENDERE BUTEKİN

THESIS ADVISOR: ASST. PROF. DR. HACER ÖZTURA

ELECTRICAL AND ELECTRONICS ENGINEERING

PRESENTATION DATE: 15.01.2019

BORNOVA / İZMİR
JANUARY 2019

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Jury Members:	**Signature:**
Asst. Prof. Dr. Hacer ÖZTURA Yaşar University	....
Asst. Prof. Dr. Abdül BALIKÇI Dokuz Eylül University	....
Asst. Prof. Dr. Nalan ÖZKURT Yaşar University	....

Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

ABSTRACT

ANN BASED ELECTRICITY CONSUMPTION FORECASTING

OF YASAR UNIVERSITY

Butekin Çimendere, Tutku

MSc, Electrical Electronics Engineering

Advisor: Asst. Prof. Dr. Hacer Öztura

Jan 2019

Although accurate electricity demand forecasts are critical for daily operations planning, electricity demand forecasting is a nonlinear and complex problem. Forecasting can be classified as long, medium, and short-terms, including long-term forecast for new power plant planning, medium-term forecast for maintenance scheduling and inventory, and short-term forecast for daily operations. Accurate load forecasting is an important issue in the daily operation and control of a power system. A suitable short term load forecasting will enable a utility provider to plan the resources and also take control measures to balance the supply and demand of electricity. Due to the current high energy prices it is essential to find ways to take advantage of new energy resources and enable consumers to better understand their load curve. This understanding will help to improve customer flexibility and their ability to respond to price or other signals from the electricity market.

There are many statistical forecasting techniques applied to short term load forecasting, such as Stochastic Time series, Regression Analysis, Expert Systems, and Artificial Intelligence. This paper presents the Artificial Neural Networks (ANNs) for the Short-Term Load Forecasting (STLF).

In this study, the electric energy consumption of a University building in the Campus of Yasar University, which has 24000 m2-usage area and central heating and cooling systems, will be predicted by using ANN's. This building has been using since September 2014 and there are hourly recorded data of this building electricity consumption for last three years. We have a four different energy monitoring equipment to be recorded consumption at the lighting, electrical sockets & UPS, mechanical (HVAC) part, and environmental lighting. There are with classes, offices, and laboratories at the building. Since the non-linearity of the load is associated to the activity in the building, we have demonstrated that the best way to deal with it is using the work day schedule as day-type classifier. Moreover, we have evaluated a number of statistical methods and Artificial Intelligence methods to adjust the typical hourly consumption curve.

Keywords—electric load forecasting; artificial neural networks; short term electric load forecasting, energy consumption, measurement data

ÖZ

YAPAY SİNİR AĞLARI KULLANILARAK YAŞAR ÜNİVERSİTESİ

YÜK TAHMİNİ

Butekin Çimendere, Tutku

Yüksek Lisans Tezi, Elektrik Elektronik Mühendisliği
Danışman: Dr. Hacer Öztura
Ocak 2019

Günlük operasyonların planları için kesin elektrik tüketim tahmin işlemleri doğrusal olmayan ve kompleks problemlerdir. Tahmin etme uzun, orta ve kısa zamanlı olarak sınıflandırılabilinir. Uzun zamanlı tahminler yeni elektrik santrallerinin planlanması için, orta vadeli tahminler bakım planlaması için ve kısa vadeli tahminler de günlük operasyon planlaması için kullanılır. Doğru tüketim tahminleri günlük operasyonlar ve güç sistemlerinin kontrolü için önemldir. Elverişli bir kısa dönem yük tahmini kaynak planlama için ve ayrıca control ölçülerinin alınması ile ihtiyaç olan elektriğin dengeli sağlanmasına olanak sağlar. Yüksek enerji fiyatlandırmalarına bağlı olarak yeni enerji kaynaklarının avantajları ve tüketicilerin yük kavislerini anlamak konusunda önemlidir. Bu bilgiler tüketicinin elektrik piyasası içerisinde pazarlık ve diğer ekonomik göstergeler konusunda esnekliğini ve yeteneğini geliştirmek konusund yardımcı olur.

Kısa dönem yük tahmini yapmak üzerine geliştirilmiş birçok istatiksel tahmin yöntemi bulunmaktadır. Regresyon Analizi, Zaman Serileri, Uzman Sistemler, Bulanık Mantık ve Yapay Sinir Ağları bu yöntemlere örnektir. Bu tez içerisinde Kısa Dönem Yük Tahmini için Yapay Sinir Ağları yöntemi kullanılmıştır.

Bu çalışmada, Yaşar Üniversitesi kampüs binalarından Y-blok mekanik elektrik tüketimi tahmini, merkezi ısıtma ve soğutma sistemlerine sahiptir, Yapay Sinir Ağları kullanılarak yapılmıştır. Bu bina Eylül 2014 yılından beri kullanılmaktadır ve son 3 seneye ait saatlik elektrik tüketim verileri kaydedilmektedir. Aydınlatma elektrik tüketimleri, elektrik girişleri ve Upsler, mekanik tüketimler ve dış çevre aydınlatmaları gözetleme yapılarak kaydedilen verilerdir. Bina içerisinde sınıflar, ofisler ve laboratuvarlar yer almaktadır. Binanın elektrik tüketiminin doğrusal olmayışı bina içerisindeki aktivitelere bağlıdır. Bu sorun günlük çalışma programları ve gün tipi ayırımı ile üstesinden gelinebilmektedir. Buna ek olarak, tipik saatlik tüketim eğisi ayarlamak için çeşitli istatistiksel ve Yapay Ağ metotları geliştirilmiştir.

Anahtar Kelimeler: Yük tüketim tahmini, yapay sinir ağları, kısa dönem yük tüketim tahmini, enerji tüketimi, ölçüm verileri

ACKNOWLEDGEMENTS

First of all, I would like to thank to my thesis advisor Asst. Prof. Dr. Hacer OZTURA who assisted me to improve myself in artificial neural network, one of the leading prediction methods that will change the future. I also express my thanks to Research Asistant Aysegul KAHRAMAN who helped me to figure out the root of the artificial neural network which is difficult to understand. I am also grateful to Asst. Prof. Dr. Nalan OZKURT for her help in understanding, analyzing and improving related with the project. I owe them a lot of thanks due to their contributions concerning knowledge that I gained and support that they gave me to use in my life.

I need to emphasize that M.Sc did not only gave me theoric knowledge but also immaterial gains. Research Asistant Zeynep ERTEKIN is one of the most important moral gains that master degree programme gave me. She helped me both during courses and thesis processes through her recommendations and guidance. I would like to also thank to her in pursuading and encouraging me when I was about to give up this difficult thesis. I could not finish this thesis without her.

I am also grateful to my dear parents who encourged me to start my master degree programme and the motivations that gave me during all the process, Also many regards to my dear husband for his moral support in all processes and understanding me to be able to complete this difficult and important step for my life.

İzmir , 2018

Tutku CIMENDERE BUTEKIN

TEXT OF OATH

I declare and honestly confirm that my study, titled "Ann Based Electricity Consumption Forecasting of Yasar University" and presented as a Master's Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Tutku CIMENDERE BUTEKIN

Signature

....

TABLE OF CONTENTS

TEXT OF OATH .... *vii*

TABLE OF CONTENTS .... *viii*

LIST OF TABLES .... *xiv*

1. INTRODUCTION *1*

1.1 Outline Of Thesis **1**

1.2 Time Horizons of Load Forecasting **2**

1.3 Aim of Project **2**

2. LITERATURE REVIEW *3*

2.1 Literature Search **3**

2.2 Factors Affecting Load Forecast **4**

2.2.1 Temporal (CALENDER) Effects: 4
2.2.2 Economic Effects: 6
2.2.3 Random Events: 6
2.2.4 Meteorological Effects: 6

2.3 Load Forecasting Methods **8**

2.3.1 Regression Method 9
2.3.2 Time Series 11
2.3.3 Expert System 12
2.3.4 Fuzzy Logic 13

3. ARTIFICIAL NEURAL NETWORK ANALYSIS *14*

3.1 Properties Of Artificial Neural Network **14**

3.1.1 History OF ANN : 15

3.2 Biological Neurons Model **16**

3.2.1 Models of a Neuron 17

3.3 Activation Functions **17**

3.3.1 Sigmoid Function: 17
3.3.2 Tanh Function: 18
3.3.3 ReLu Function: 19

3.4 Artificial Neural Network Architectures **19**

3.4.1 Feedforward Neural Network 20
3.4.2 Recurrent Neural Network (RNN) 25
3.4.3 Convolutional Neural Network: 27

3.5 APPLICATIONS OF ARTIFICIAL NEURAL NETWORK **27**

3.6 NETWORK TRAINING **28**

3.6.1 Supervised Learning 28
3.6.2 Unsupervised Learning 28

3.6.3 Reinforcement Learning 28

4. EXPERIMENTAL WORKING STEPS *29*

4.1 Nonlinear Autogressive Neural Network With Exogeneous Inputs 29

4.2 Recording and Collecting Data 31

4.3 Data Normalization Process 46

4.3.1 Z-score Normalization 46

4.3.2 Decimal Scaling Normalization........ 47

4.3.3 Min-Max Normalization 47

4.4 Artificial Neural Networks Designing 48

4.5 Training And Testing The ANN........ 48

5. CONCLUSION........ *70*

6. FUTURE WORK *73*

7. REFERENCES........ *74*

LIST OF FIGURES

Figure 2.1 The Daily Load Consumption Of September 2016 5

Figure 2.2 The daily load consumption table of July 2016 5

Figure 2.3 Monthly Load Demand and Average Air Temperature Graphic of 2016 7

Figure 2.4 Monthly Load Demand and Average Air Temperature Graphic of 2017 8

Figure 2.5 Simple Linear Regression Line 10

Figure 2.6 Fuzzy Logic Diagram (Tanaka, 1993) 13

Figure 3.1 Biological Neuron Model (Halici, 2008) 16

Figure 3.2 Block Diagram of a Human Nervous System (Kumar, 2013) 16

Figure 3.3 Biological Neurons Mathematical Model (Johnson, 2018) 17

Figure 3.4 Sigmoid Function (Johnson, 2018) 18

Figure 3.5 The Tanh Function (Johnson, 2018) 18

Figure 3.6 Rectified Linear Unit (ReLU) Activation Function (Johnson, 2018) 19

Figure 3.7 A Single Layer Feed-Forward Neural Network (Sazli, 2006) 20

Figure 3.8 McCulloch and Pitt Neuron Model (Lagandula, 2018) 21

Figure 3.9 Perceptron Model of Minsky-Papert (Lagandula, 2018). 22

Figure 3.10 A Feedforward Neural Network Flow of Information (Brilliant, 2018) 23

Figure 3.11 Back Propagation Training Steps (Saurabh, 2018) 24

Figure 3.12 Recurrent Neural Network and Feed-Forward Neural Network Information Flow Difference (Donges, 2018) 26

Figure 3.13 Feedforward Neural Network One to One Map and RNN’s One to Many, Many to One and Many to Many Map (Donges, 2018) 26

Figure 3.14 Convolutional Neural Network Figure (Maladkar, 2018) 27

Figure 4.1 Series Parallel Architecture (open-loop) (Amani et al, 2011) 30

Figure 4.2 Parallel Architecture (Closed-Loop) (Amani et al, 2011) 30

Figure 4.3 The Load Consumption Figure Of January 2016 32

Figure 4.4 The Daily Load Consumption Of Yasar University January 2016................................32

Figure 4.5 The Load Consumption Figure Of February 2016..33

Figure 4.6 The Daily Load Consumption Figure Of February 2016..33

Figure 4.7 The Load Consumption Figure Of March 2016..34

Figure 4.8 The Daily Load Consumption Figure Of March 2016..34

Figure 4.9 The Load Consumption Figure Of April 2016..35

Figure 4.10 The Daily Load Consumption Figure Of April 2016...35

Figure 4.11 The Load Consumption Figure Of May 2016...36

Figure 4.12 The Daily Load Consumption Figure Of May 2016...36

Figure 4.13 The Load Consumption Figure Of June 2016..37

Figure 4.14 The Daily Load Consumption Figure Of June 2016...37

Figure 4.15 The Load Consumption Figure Of July 2016...38

Figure 4.16 The Daily Load Consumption Figure Of July 2016...38

Figure 4.17 The Load Consumption Figure Of August 2016..39

Figure 4.18 The Daily Load Consumption Figure Of August 2016...39

Figure 4.19 The Load Consumption Figure Of September 2016...40

Figure 4.20 The Daily Load Consumption Figure Of September 2016...40

Figure 4.21 The Load Consumption Figure Of October 2016..41

Figure 4.22 The Daily Load Consumption Figure Of October 2016...41

Figure 4.23 The Load Consumption Figure Of November 2016...42

Figure 4.24 The Daily Load Consumption Figure Of November 2016..42

Figure 4.25 The Load Consumption Figure Of December 2016...43

Figure 4.26 The Daily Load Consumption Figure Of December 2016..43

Figure 4.27 Comparision of Measures of Forecast Error..50

Figure 4.28 NARX Neural Network Model..50

Figure 4.29 Regression Line of The Remaining Data of Trainlm Function with 10 Hidden Layer Size...51

Figure 4.30 Regression Plots Of ANN During Training, Testing And Validation Of Trainlm Function With 10 Hidden Layer Size........ 52

Figure 4.31 Structure Of Back Propagation Neural Network In The Case Under Study........ 53

Figure 4.32 Regression Line of The Remaining Data of Trainlm Function with 15 Hidden Layer Size........ 53

Figure 4.33 Regression Plot For Training, Testing And Validation Of All Products ANN Model Of Trainlm Function With 15 Hidden Layer Size........ 54

Figure 4.34 Regression Line of The Remaining Data of Trainlm Function with 10 Hidden Layer Size........ 55

Figure 4.35 Regression Plot For Training, Testing And Validation Of All Products ANN Model Of Trainlm Function With 10 Hidden Layer Size........ 55

Figure 4.36 Regression Line Of The Remaining Data Of Trainbr Function With 15 Hidden Layer Size........ 56

Figure 4.37 Regression Plot For Training, Testing And Validation Of All Products ANN Model Of Trainlm Function With 15 Hidden Layer Size........ 57

Figure 4.38 Regression Line of The Remaining Data of Trainlm Function with 10 Hidden Layer Size........ 58

Figure 4.39 Regression plot for training, testing, and validation of ANN model with Trainlm Function and10 Hidden Layer Size........ 59

Figure 4.40 Open-Loop Architecture Network Model........ 59

Figure 4.41 Regression of NARX plot of Remaining Data (plotregression) of Trainlm Function with 15 Hidden Layer Size........ 60

Figure 4.42 Regression of NARX plot (plotregression) of Trainlm Function with 15 Hidden Layer Size........ 60

Figure 4.43 The Regression Line Of The Remaining Data of Trainbr Function with 10 Hidden Layer Size........ 61

Figure 4.44 The Regression Lines Of Training, Testing And Output Data Of Trainbr Function With 10 Hidden Layer Size........ 62

Figure 4.45 The Regression Line Of The Remaining Data of Trainbr Function with 15 Hidden Layer Size........ 63

Figure 4.46 The Regression Lines Of Training, Testing And Output Data Of Trainbr Function With 15 Hidden Layer Size 63

Figure 4.47 Closed-Loop(Series Architecture) Network Model 64

Figure 4.48 The Regression Line Of The Remaining Data Closed Loop Architecture 65

Figure 4.49 The Regression Line Of The Closed Loop Architecture 65

Figure 4.50 Comparison Between Normalized Actual And Normalized Forecasted Load 67

Figure 4.51 De-Normalized Power Output vs Actual Output 68

Figure 4.53 Multi-Step Prediction for 720 Hours 69

Figure 4.54 De-Normalized Power Output vs Actual Output for Multi-Step Prediction 69

LIST OF TABLES

Table 2.1 The Average Load Consumption and Air Temperature of 2016 in Izmir.......................... 6

Table 2.2 The Average Load Consumption and Air Temperature of 2017 in Izmir 7

Table 4.1 The Table of 08.01.2016 Input Data 44

Table 4.2 The Table of 08.01.2016 Target Data.. 45

Table 4.3 The Normalized Table of 08.01.2016 Input Data 48

Table 4.4 The Performance Results Of 2016 Data Inputs 66

Table 4.5 The Performance Results Of 2016 and 2017 Data Inputs 66

ABBREVIATIONS

ANN: Artificial Neural Network

AR: Auto Regressive

ARIMA: Auto Regressive Integrated Moving-Average

ARMA: Auto Regressive Moving-Average

BP: Back-Propagation

CNN: Convolution Neural Network

FESCO: Faisalabad Electric Supply Corporation

FF: Feed Forward

HVAC: Heating, Ventilation and Air Conditioning

KBES: Knowledge-Based Expert Systems

LM: Levenberg-Marquardt

LTLF: Long-Term Load Forecasting

MAPE: Mean Absolute Percentage Error

MLP: Multi-Layer Perceptron

MLR: Multiple Linear Regression

MTF: Medium-Term Load Forecasting

MSE: Mean Square Error

RNN: Recursive Neural Network

STLF: Short-term Load Forecasting

STS: Stochastic Time Series

TF: Transfer Function

1. INTRODUCTION

Electric power generation and distribution have high importance for electric power suppliers and consumers of electricity, such as commercial or residential buildings, agriculture and especially industrial buildings. It should be procured in high quality, uninterrupted even though the demand is high.

Load forecasting is very important for the energy management of electrical power systems and used for scheduling of generating capacity, determining budget and planning for energy transactions (Afkhami-Rohani, Khotanzad, Maratukulam, 1998). In the near future, the importance of load forecasting will be directly proportionate to increase of energy requirement.

1.1 OUTLINE OF THESIS

This thesis has 6 main chapters which is summarized as under:

Chapter 1 is the introduction part of the thesis, it starts with the explanation of the importance of load forecasting and includes outline of the thesis, time horizons of load forecasting and the aim of the project.

The artificial neural network has been using for load forecasting since in the late 1980s. However, the beginning of Neurocomputing is accepted as the article of McCulloch and Pitts wrote in 1943. In 1949, the book named *The Organization of Behaviour* was written by Hebb proposed the idea about specific learning law for the synapses of neurons. After Hebb's book, Rosenblatt introduced the simple neural network now called Perceptrons. In 1986, The Back-Propagation Learning algorithm for Multi-Layer Perceptrons was re-discovered and in 1990's the sub-field of Radial Basis Function Network was discovered. The history of load forecasting is explained completely in Chapter 2.

Also, the factors affect the load forecasting performance such as calendar, economic, random and the meterological effects are detailed. Moreover, the informations about load forecasting methods such as regression method, time series, expert systems, fuzzy logic and our analysis type artificial neural network anaylsis are given.

Chapter 3, firstly gives the information about our ANN's type NARX. The NARX is a recurrent dynamic neural network which has feedback connections that include several layers of the network. NARX network can be classified as two by architecture types such as open loop and closed loop. In open loop, series-parallel architecture, the future value of output is calculated with the present and past values of inputs and true past values of the output. However, in the closed-loop architecture, the calculation is performed with the present and the past values of input and the past predicted values of the time series outputs. In this thesis, both architecture types are performed for the prediction about to show the effect on the performance. Then, the experimental working steps which are recording and collecting data, normalization process, designing and the testing results respectivly explained.

The research and the results are summarized in the Chapter 4, the Conclusion Part and the future work recommendations are given in Chapter 5.

1.2 TIME HORIZONS OF LOAD FORECASTING

Load forecasting can be categorized by time periods which planning to estimate and can be divided into three categorize: short-term forecasting, medium forecasts, and long-term forecasting.

- Short-term forecasting (STLF) is used to determine one hour to one-week load and gives important load data for managing day-to-day operations. Also, power generation unit's activation times are planed by using STLF. Many countries such as Turkey, electricity distribution companies, which generate and distribute electricity, are privatized. Therefore, STLF helps to determine the pricing of electricity for selling or buying in the market (Yasin, 2015).

- Medium-term forecasts are one week to a year and help to plan to schedule required power.

- The last forecasting type is long-forecasting is longer than a year. Also used for prediction of future needs (Feinberg, 2005; Gupta:2017).

1.3 AIM OF PROJECT

There are many statistical forecasting techniques applied to short-term load forecastings, such as the Stochastic Time series, Regression Analysis, Expert Systems, and Artificial Intelligence. This paper presents the Artificial Neural Networks (ANNs) for the Short-Term Load Forecasting (STLF).

The aim of this thesis is the electric energy consumption of campus building of Yaşar University will be predicted by using Yaşar University Y block mechanical panel load values of 2016 and 2017. Artificial neural network method and Matlab programming have been used during the project. This building has been using since September 2014 and lightning, electrical sockets and Ups, mechanical (HVAC) part, and environmental lighting consumptions have been recorded.

2. LITERATURE REVIEW

2.1 LITERATURE SEARCH

Load forecasting is more accurate than traditional methods such as regression method and used since the early 1960s. However, the importance of method is to understand in the middle of the 1980s. Before the discovery of the backpropagation training method in 1974, it was impossible to train more than two layers. Multilayer training allowed understanding the linear or nonlinear relationships between input and output variables. (Highley, 1993)

Penya, Borges, and Fernandez focused very special case of STFL such as non-residential buildings STFL, which means schools, universities, public buildings etc. All common point is they all have stationary, seasonal and regular electricity consumptions. Their Project has shown that STLF by using historical data gives sufficient prediction accuracy. (Penya, 2001)

Kiartzis, Bakirtzis, and Petridis worked on one-year data from Greek interconnected power system which has a peak load of about 4,7 GW and is supplied by the Public Power Corporation. Daily high and low-temperature measurements and load data were used for prediction. The root-mean-square error is used for measuring the performance of training. Using additional weather information such as humidity, cloud cover, rainfall etc and a load of holidays are expected to increase of accuracy of the forecast. (Kiartzis, 1995)

Elgahry, Othman, Taha, and Hasanien tested the performance of the method for short-term load forecast by using the actual hourly load data in many kinds of research using ANN. They used two variables; the historical load data and time in 24 hours. The Matlab Levenberg Marquardt optimization training Function was used. The comparisons and subsequent discussions show the efficiency of the proposed method and its superiority over other load forecasting techniques. (Elgarhy, 2017)

In Aggrwal and Kumar's article, ANN methodology and Multiple Linear Regression technique are used to predict daily usage of customers. The research has shown that the prediction is affected by weather conditions, special days and especially training dataset selection. (Kumar, 2013)

Özden and Öztürk compared the performance between ANN and time series method to forecast energy demand in an industrial area in Turkey. Their model used energy consumption on the previous day and temperature as input parameters. Depending on their results, the result of time series regression value R is 0,93901 and the same R-value in the ANN method is 0,9859. According to these results, they reach a conclusion that the ANN method has more accurate result than time series method because of time series method needs more input values for better prediction. (Özden, 2018)

In the article 'Daily Electric Load Forecasting: Case of Thailand' Dilhani and Jeenanunta researched the importance of temperature data for analyzing STLF with ANN model using thirty minutes load data from Electricity Generating Authority of Thailand (EGAT). The training data set divided into four, which are with and without temperature. Also, only one pair dataset weekday and weekend load demand, the other three training data sets consist of only weekdays without holidays. Results show training with temperature data have better accuracy of forecasting using ANN. (Dilhani, 2016)

The article named as "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks" written by Rahman and his co-workers proposed two recurrent neural network models to forecast electricity consumptions of commercial and residential buildings. The presented models have limitations such as equipment or estimated weather changes s would likely have decreased accuracy. (Rahman, 2018)

Saatwong and Suwankawin present a short-term load forecasting in the building by using the SARIMA model which was obtained by the Box-Jenkins methodology. The actual data which are taken from Electrical Engineering Building at Chulalongkorn University and used for evaluating the results. The % 20 Mean Absolute Percentage Error and can be said that the models are successful to predict the peak-load instant. (Saatwong, 2016)

Papadopoulos and his co-workers worked about the real and reactive power consumption of nine university campuses of Democritus University Thrace, Greece. Load shape factors are used for investigating the energy performance of campuses. While the load level decreases, the load becomes more flexible and diverse. (Papadopoulos, 2016)

2.2 FACTORS AFFECTING LOAD FORECAST

In the literature research, it has been seen that there are several factors affect the performance of the load forecast and the artificial neural Networks performance. The main factors can be classified as follows: (Cavallaro, 2005)

2.2.1 TEMPORAL (CALENDER) EFFECTS:

Seasonal differences such as winter and summer which affect using of air conditioning, hours of daylight for using lightning, day and night hours, weekday /weekend, holiday or not is important for consumption of electricity.

In our case, the date university is started and ended, weekdays; student numbers which take the class is also changing the load demand. Spring semester in 2016 is started at Jan 30 and ended at May 08.

Student number takes 0 at spring/summer breaks and holidays such as weekends, New Year 's Day, Ramadan Feast and The Feast of Sacrifice.

The Feast of Sacrifice holiday was at 12 September to 15 September and 10 September and 11 September were weekends in 2016. The daily load consumption of September 2016 shown in Figure 2.1 which is given below:

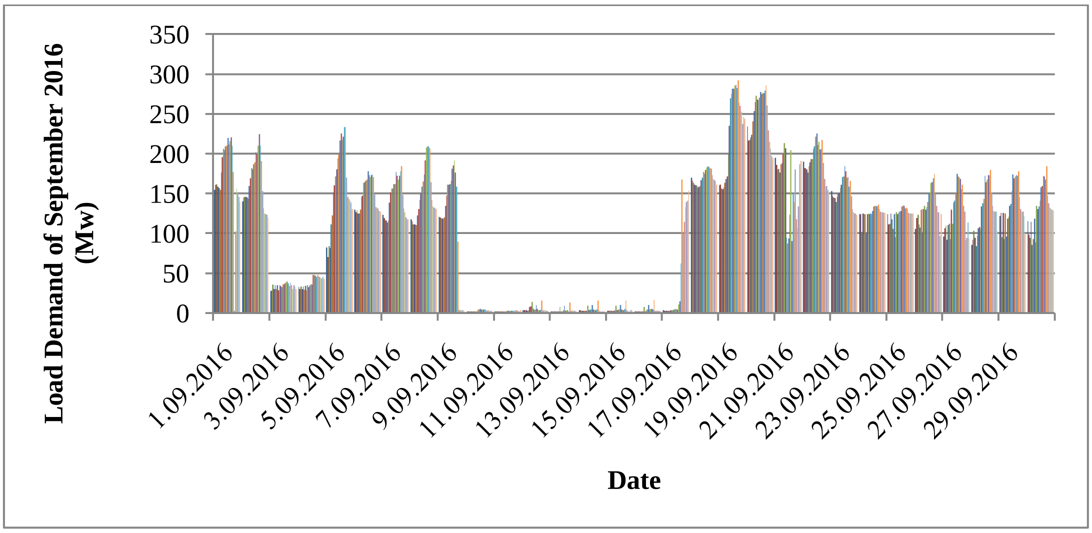

Figure 2.1 The Daily Load Consumption Of September 2016

Ramadan Feast in 2016 was started at 5 July and ended at 7 July. However, the holiday was extended by the government to exhilarate to tourism and ended on 10 July. The relationship between holiday and consumption can be seen clearly in the table of daily load consumption figure of July 2016 is given in Figure 2.2 which is given below:

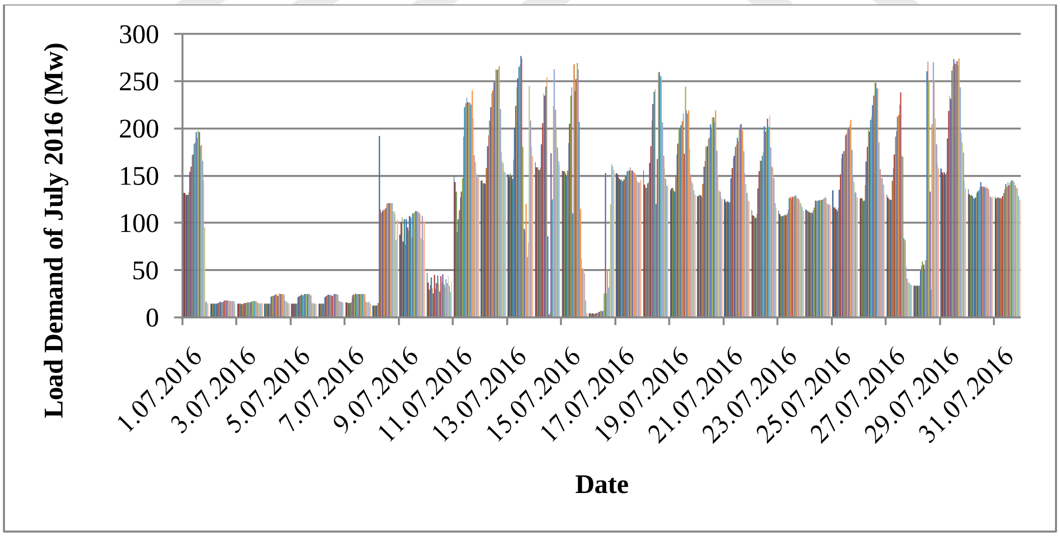

Figure 2.2 The daily load consumption table of July 2016

2.2.2 ECONOMIC EFFECTS:

Economic trends (recession or expansion), new industrial sites, electricity price changes, demand-side management affect load demand.

2.2.3 RANDOM EVENTS:

Sporting events or conference at the university, popular television shows, start-up or shutdown operations of industrial plants requiring a lot of power (iron and steelworks, etc.) are named random events.

2.2.4 METEOROLOGICAL EFFECTS:

Air conditioning is the most powerful factor which affects the forecast. Weather, rainfall or snow, maximum and minimum temperatures, humidity levels, sunshine duration, cloud amount can be considered as meteorological effects.

Switching on and off heating and air conditioning equipment has quite impact on consuming electricity.

The Table 2.1 and The Figure 2.3 are given below shows the average demand load and temperature of İzmir in 2016 as monthly. In the 2016 August, which average temperature is 30 degree, the load consuming has 110.844,89 kWh which is the highest level than other months.

Table 2.1 The Average Load Consumption and Air Temperature of 2016 in Izmir

Months	Average demand load (kW)	Average air temperature (°C)
1	25,09	8,7
2	16,59	9,5
3	20,53	11,6
4	37,71	15,8
5	46,45	20,7
6	81,83	25,5
7	88,51	28
8	110,85	30
9	77,74	23,6
10	55,75	18,7
11	14,73	14
12	37,30	10,4

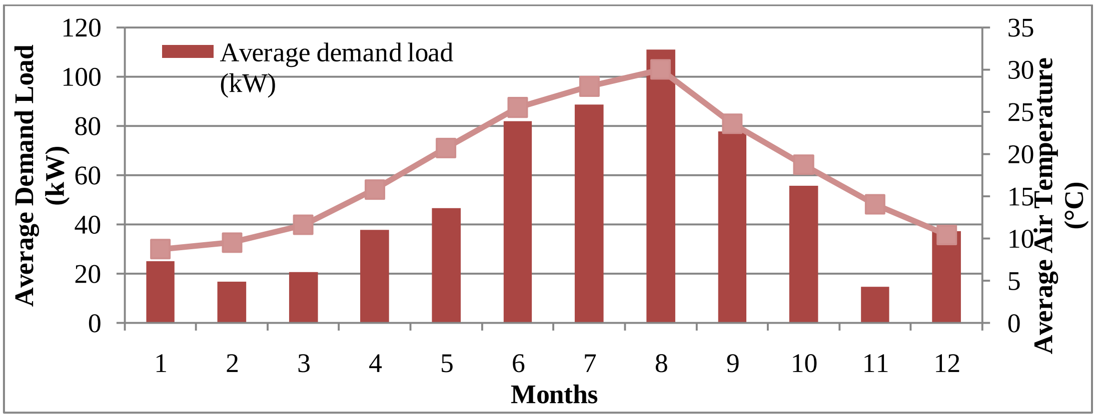

Figure 2.3 Monthly Load Demand and Average Air Temperature Graphic of 2016

The below Table 2.2 is the table of consumption load and measured temperature values of 2017. The higher consumptions are at July and August which temperature values were highest values.

Table 2.2 The Average Load Consumption and Air Temperature of 2017 in Izmir

Months	Average demand load (kW)	Average air temperature (°C)
1	34,72	6,2
2	29,47	10,6
3	16,42	13,9
4	11,25	17,2
5	61,71	23
6	70,12	28,4
7	119,92	30,4
8	123,88	29,6
9	88,69	25,5
10	44,38	18,9
11	17,35	12,8
12	29,73	9,7

The highest temperature values of the 2016 are can be seen clearly in Figure 2.4:

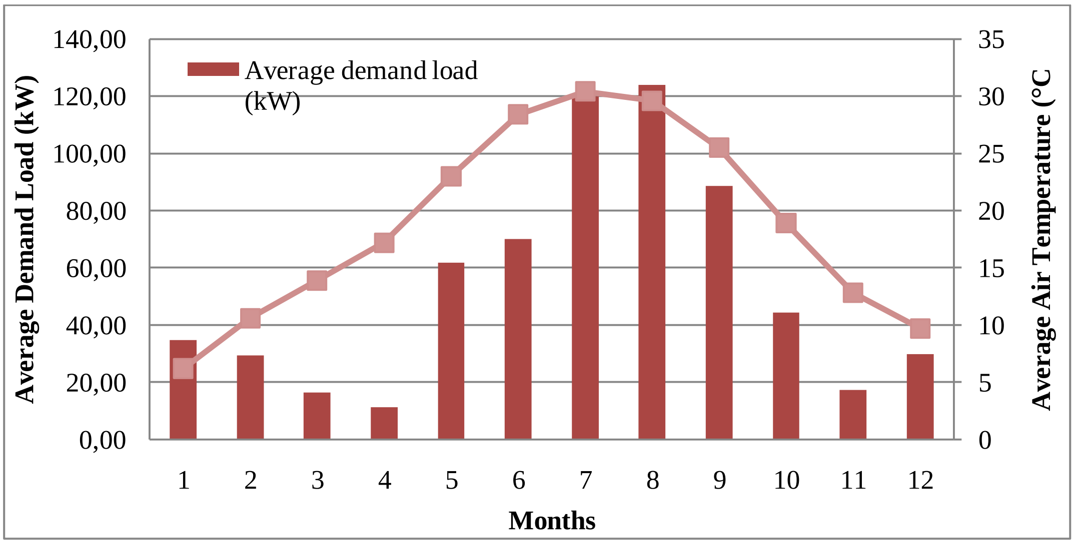

Figure 2.4 Monthly Load Demand and Average Air Temperature Graphic of 2017

Both 2016 and 2017 graphics present the relationship between energy consumption and temperature of air clearly, the increase of the air temperature causes increasing of the load consumption.

2.3 LOAD FORECAST METHODS

Since load forecasting has become prominent, many methods are proposed in the literature. Studies about demand forecasting can be classified into two basic groups such as; quantitive and qualitative methods. (Singh, 2013)

Delphi method and Curve fitting method can be named as qualitative methods. This method comes from experience and instincts of experts. On the contrary, quantitative methods are based on mathematical models, often use historical data. These types of forecasting methods are objective. Regression analysis, exponential smoothing and the Box-Jenkins approach can be examples. ("Forecasting Fundamentals", 2018)

The basic methods for load forecasting are listed below:

2.3.1 REGRESSION METHOD

Regression method is a statistical method for studying and estimating the relationships between two quantitative variables which are dependent or response variable, x. denoted y, and independent or predictor variable.

The main idea of simple linear regression is to understand how the typical value of the dependent variable changes when one of the independent variables is changed while the other independent variables are held fixed.

Regression analysis is largely used for forecasting and prediction problems, because this method have substantial overlap with the field of machine learning. One of the other usages is to understand which among of the independent variables are related to dependent variables.

Regression models can be linear or nonlinear. In linear regression model assumes that the relationships between independent and dependent variables are straight line relationships while a nonlinear model assumes the lines are curved.

In the literature, many techniques are designed for regression analyses. Most commonly used are simple linear and multiple linear regression that explained below:

2.3.1.1 Simple Linear Regression:

Linear regression method concerns two-dimensional sample points with one independent variable and one dependent variable and predicts dependent variable values as a linear Function of independent variables. The linear Function has the following form:

$$y = \alpha + \beta x, \tag{1}$$

The Formula can be described as a line with slope β and a y-intercept α. y is a predicted or expected value of the outcome while x is the predictor.

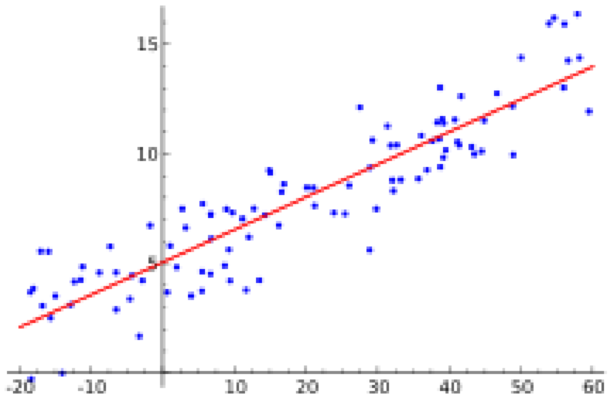

Figure 2.5 Simple Linear Regression Line , Source:http://www.wikizeroo.net/index.

In the simple linear method, regression line touches some points, but not others. The distance from the other points to the line is called "error terms". In all predictions by using the regression method have error terms because of unexpected factors in reality. The best procedure for the finding of the best line is "least square method". In the least square method, the differences between all points and line are calculated and adding the values after squaring each difference.

2.3.1.2 Multiple Regression Method:

Multiple regression method is an improved version of the simple linear method and uses to estimate the relationship between two or more independent variables and a single continuous dependent variable. The Function of the method with k predictor values is $x_1, x_2, \ldots, x_k$, a response y_i and $\beta_0,\ \beta_1, \ldots, k$ is can be written as follows:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k \qquad (2)$$

To estimate the coefficients β are the values that square errors sum for the sample. The values fit in the equation are denoted $\hat{y}_i$ and the residuals ei are equal to $y_i - \hat{y}_i$, the difference between the observed and fitted values. The sum of the residuals is equal to zero. The variance σ^2 may be estimated by $s^2 = \frac{\sum e_i^2}{n-p-1}$, also known as the mean-squared error (or MSE).

The estimate of the standard error s is the square root of the MSE.

2.3.2 TIME SERIES

A time series can be expressed as a sequential set of data points, measured over different times. It is mathematically defined as a set of vectors $x(t) = 0,1,2$ ... where t represents the time and $x(t)$ is the random variable. The measurements are collected during an event and arranged in chronological order. (Adhikari, 2013)

A time series are affected four general components, such as Trend, Cyclical, Seasonal and Irregular. The trend is a long-term movement in a time series. Time series can increase, decrease or stagnate. The cyclical variation in a time series is the medium-term changes in time series caused by a repeat in cycles. Seasonal variations are fluctuations within a year during the season like weather and climate conditions. Irregular or random variables are caused by unexpected influences that not repeat in time and not regular. The unexpected influences can be war, strike, earthquake or flood.

In time series forecasting, past values data are collected and analyzed to advance a proper mathematical model for determining future events. There are two largely used of linear time series models in literature Autoregressive (AR) and Moving Average (MA) models. The combination of AR and MA is Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) is expended. ARIMA model is based on Box-Jenkins principle and known as Box-Jenkins models.

In the autoregressive (AR) model, the future value of the variable is assumed to be a linear combination of past values and random error and can be expressed as:

$$
\begin{aligned}
y_t &= c + \sum_{i=1}^{p} \varphi_i \ y_{t-i} + \varepsilon_i \\
&= c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \ldots\ldots + \varphi_p y_{t-p} + \varepsilon_t
\end{aligned}
\tag{3}
$$

Where y_i is the actual value and ε_i is random value at the time period, c is a constant and $\varphi_i \ (i = 1,3,\ldots,p)$ are model parameters. P is the order of the model.

The moving average (MA) models use past errors as the explanatory datas and the expression of MA (q) is given below:

$$
\begin{aligned}
y_t &= \mu + \sum_{j=1}^{q} \theta_i \ \epsilon_{t-j} + \varepsilon_i \\
&= \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots\ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t
\end{aligned}
\tag{4}
$$

The difference AR (p) and MA (q) are, MA (q)'s autocorrelation Function cuts off sharply while the AR (p) series decline exponentially.

The combination of autoregressive (AR) and moving average (MA) models is more effective and more useful, known as the ARMA model which is represented as;

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i \; y_{t-i} + \sum_{j=1}^{q} \theta_i \; \epsilon_{t-j} \tag{5}$$

$y_t = actual\ value$

$\varepsilon_t = random\ error\ at\ time\ period\ t$

$\varphi_i = model\ parameters$

$c = a\ constant$

2.3.3 EXPERT SYSTEM

An expert system is the computer applications for solving complex problems with the components of the knowledge base, interference engine, and user interface. The knowledge base has high-quality knowledge and exhibits intelligence. Interference engine applies the rules repedetly to the facts that are known from earlier rule application, adds new knowledge into the knowledge base and resolves rules when multiple rules are optional. User interface connects the user and expert system to each other.

The expert system has less production cost and less error rate than human and offers great speed which reduces the amount of work. Besides all these advantages, it also has some limitations such as;

- The ability of the expert system is limited by the technology,
- Expert system development is required significant time,
- The computer resources and large systems are costly.

Expert systems can be applied for various purposes. In the medical domain can be used in diagnosis systems, in process control systems it can help to control a physical process. Also, it can use for designing to camera lenses and automobiles. (“Artificial Intelligence - Expert Systems", 2018).

2.3.4 FUZZY LOGIC

In general terms, fuzzy logic is concerned with the formal principles of approximate reasoning, with precise reasoning viewed as a limiting case. (Zadeh, 1996)

Fuzzy logic was invented by Dr. Lotfi Zadeh at the University of California at Berkeley in 1965. The fuzzy set theory is the basis of the fuzzy logic. In a traditional set, a given element is completely in the set or not in the set at all. However, the fuzzy sets can be partially in the set. (Tanaka, 1993)

Fuzzy logic is useful because it gives not accurate but acceptable reasoning. It can control the machines and consumer products and also helps to deal with uncertainty problems or cases.

It consists of four basic parts as shown;

- Fuzzification Module: The transformation the crips numbers into fuzzy sets and splitting the input signal into five steps are dealing in this part. The steps are LP for large positive, MP is medium positive, S is small, medium negative MN and LN is used for large negative.
- Knowledge Base: Experts provide the IF-THEN rules and knowledge base stores.
- Inference Engine: It simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules.
- Defuzzification Module: The part transforms the fuzzy set into crisp values.

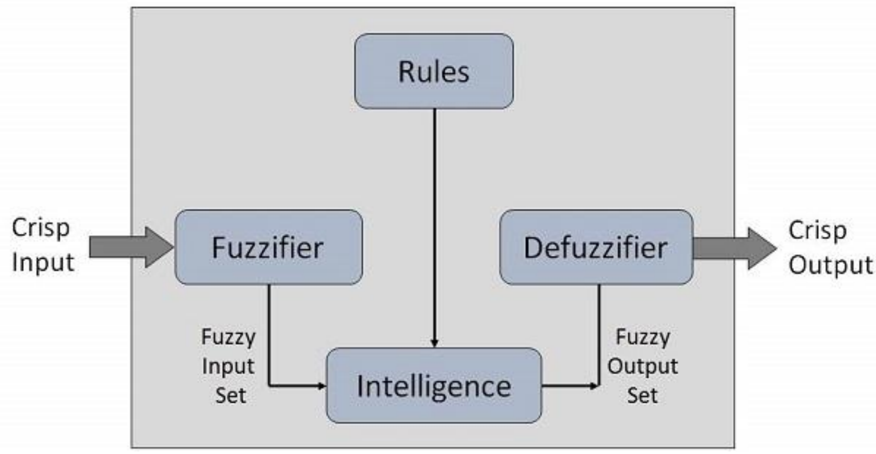

Figure 2.6 Fuzzy Logic Diagram (Tanaka, 1993)

3. ARTIFICIAL NEURAL NETWORK ANALYSIS

An artificial neural network(ANN) is a mathematical model which is used to predict demand load by taking model on human brains such as; the faculty of Learning, adaptation, tolerance to inaccurate information etc (Moghram and Rahman, 1989)

ANN model's basic processing components are billion of neurons which are using for data transfering like human brains.

Using an artificial neural network has many advantages which are listed below;

- Information can be stored on the entire network, not only on the database; the disappearance of a few information is not affecting the performance.

- ANN based model methods can work with incomplete knowledge. The loss of performance is depending on the missing information.

- Artificial Neural Networks can learn and make decisions by using historical data. Also does not need to be reprogrammed.

- More than one task can be performed by using ANN because it has numerical strength. (Mijwel, 2018)

These types of methods also have some disadvantages such as;

- ANN uses probing solutions which give different solutions for each time.

- The network structure is not determined for ANN. The best structure should be achieved through trial and error.

- It is required more historical data for the more accurate solution.

3.1 Properties of Artificial Neural Network

- Nonlinearity: Nonlinearity is a highly important property, especially the underlying physical mechanism responsible for the generation of the input signal is nonlinear.
- Input-output mapping: Artificial neural network has no fixed algorithm, therefore the algorithm and weights should be arranged by working through training samples or task examples.
- Adaptivity: An artificial neural network calibrates the weights for changing in problems. For this reason, it can be adapted for every change in the problem.
- Fault Tolerance: Artificial neural network has parallel distributed structure, and the information in the network is distributed. Losing some of connections or neurons does not affect the accuracy of the network.

- ❖ VLSI Implementability: Artificial neural network can be realized with very large scale integration(VSLI) which increases the ability of data processing.

3.1.1 HISTORY OF ANN :

First thought of human brain neurons how work was written by neurophysiologist Warren McCulloch and mathematician Walter Pitts in 1943 and they modeled a first and simple neural network using the electrical circuit. (Roberts, 2009)

Donald Hebb was pointed out the neural pathways are getting stronger each time they are used like human brains learn in *The Organization of Behavior*, in 1946.

In IBM research laboratories, Nathanial Rochester tried to simulate a hypothetical neural network. However, his research was failed.

The first neuro-computer was founded by Frank Rosenblatt and Charles Wightman during 1957 and 1958. Rosenblatt is known as the developer of Neuro-computing and writer of the book on Neurocomputing which is named as *Principles of Neurodynamics* (Yadav and Kumar, 2015).

In 1962, Widrow & Hoff developed a new type of neural network processing element named ADALINE and MADALINE, Multiple ADAptive LINear Elements. ADALINE can recognize the binary patterns for reading streaming bits from a phone line and predict the next bit. Besides that MADALINE can eliminate echoes on phone lines by using adaptive filters and it was the first neural network application in the real world.

Minsky and Papert conducted a campaign denigrate neural network and published a book named as Perceptrons which gave the idea of neural network researches were led to the dead end.

The multilayered network was used in 1975.

In 1982, John Hopfield developed an approach of using bidirectional lines instead of only one-way connected neurons.

In 1986, the basic idea of back propagation Networks was considered because it distributes pattern recognition errors throughout the network.

The first open conference on Neural Networks, the IEEE International Conference of Neural Network was held in 1987, in San Diego.

3.2 BIOLOGICAL NEURONS MODEL

An artificial neural network is a new form of modeling which inspired by brains/biological models. A human brain consist of approximately 10^{11} neurons and each neuron has a spherical cell body called soma. Soma generates the signals and the signals are transmitted to other neurons through an extension on the cell body which called axon or nerve fibers. The dendrites are responsible for receiving signals generating by other neuronsSynapses are the fundamental units that intercede the interactions between neurons. Chemical synapses are the most common kind of synapses. (Halici, 2008)

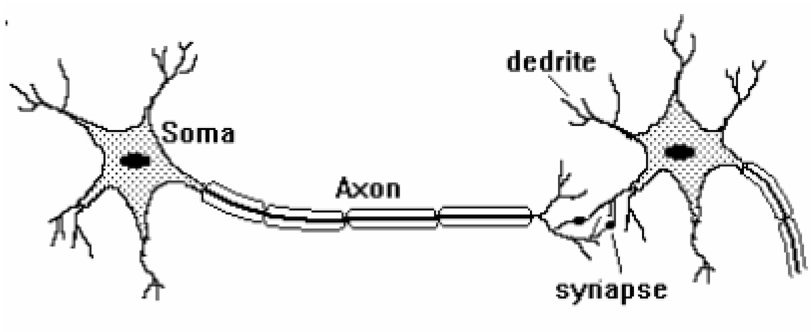

Figure 3.1 Biological Neuron Model (Halici, 2008)

The first principle of information coding and representation in the brain is redundancy, which means each piece of the data is processed by a set of neurons for not losing all the data in the case of the brain gets damaged.

The human nervous system has three-stage such as receptors, neural net, and the effectors. The receptors convert stimuli from the human body or the environment into electrical impulses which transmit data to the brain (neural net). The last stage effectors convert those electrical impulses into system outputs.

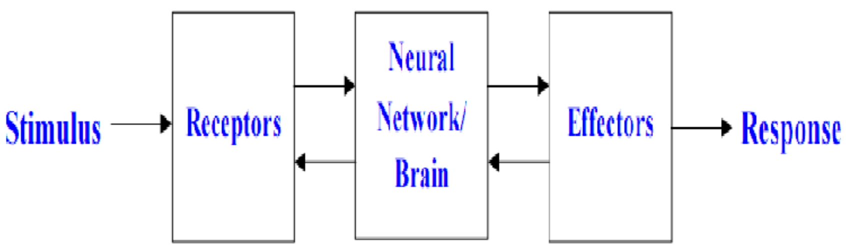

Figure 3.2 Block Diagram of a Human Nervous System (Kumar, 2013)

3.2.1 Models of a Neuron

In the computational model of a neuron, the signals that travel along the axons (x_o) interact with dendrites ($x_o w_0$) of the other neuron based on the synaptic strength at the synapse (w_0). The synaptic strengths are the weights that can learn and control the strength of the influence. The dendrites carry the signal to the cell body to the summation of data. If the sum is above a certain threshold, the neuron can fire and the spike will be sent along its axon. The timing of the spike does not matter, still, the frequency of the firing transmit the data. Firing rate of the neuron with an ***activation Function f,*** which represents the frequency of the spikes along the axon.

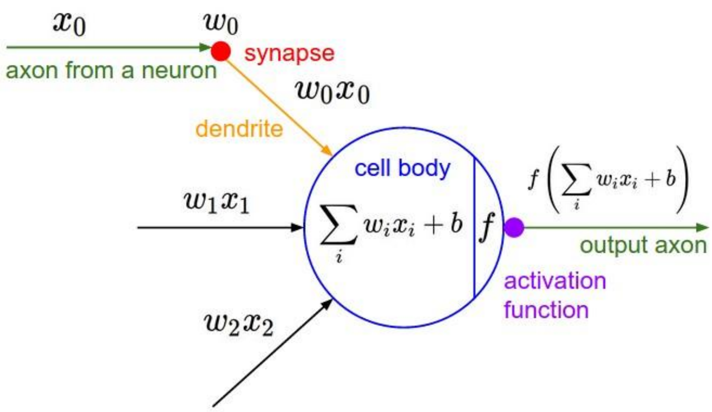

Figure 3.3 Biological Neurons Mathematical Model (Johnson, 2018)

Some of the activation Functions given below:

3.3 ACTIVATION FUNCTIONS

Every activation Function takes a single input and performs a specific mathematical operation on it. There are several activation Functions which are used frequently in practice:

3.3.1 SIGMOID FUNCTION:

The sigmoid Function is the most common form of activation Function used in the artificial neural network and it receives the real-valued number and squashes it into the range between 0 and 1. It is an increasing Function which exhibits a balance between linear and nonlinear behavior. The mathematical expression of the Function is;

$$\psi(v) = \frac{1}{1 + e^{-av}} \tag{5}$$

Where a is the slope parameter of the sigmoid Function.

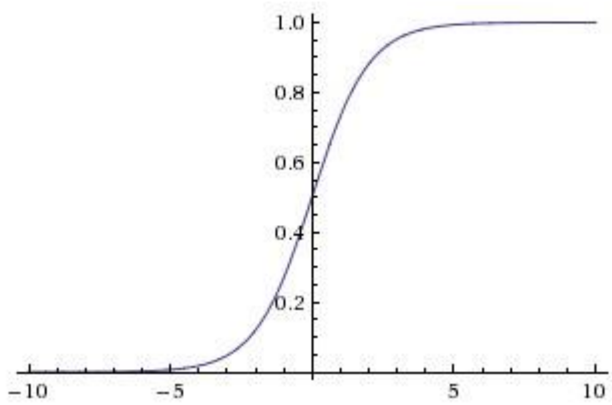

Figure 3.4 Sigmoid Function (Johnson, 2018)

3.3.2 TANH FUNCTION:

The Function squashed a real-valued number to the range [-1 1]. It is similar to the sigmoid Function, but unlikely tanh Functions outputs are zero centered. Therefore, tanh Function is preferred to the sigmoid Function in practice. The Formula of the Function is given below:

$$\tanh(x) = 2\sigma(2x) - 1 \quad (6)$$

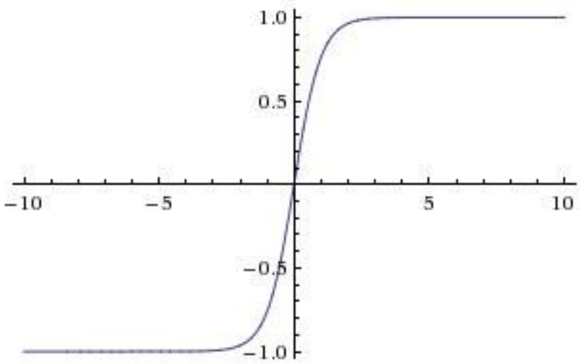

Figure 3.5 The Tanh Function (Johnson, 2018)

3.3.3 RELU FUNCTION:

The Rectified Linear Unit is thresholded at zero. It has more simple operations than sigmoid/ tanh neurons and computes the function as $f(x) = \max(0, x)$.

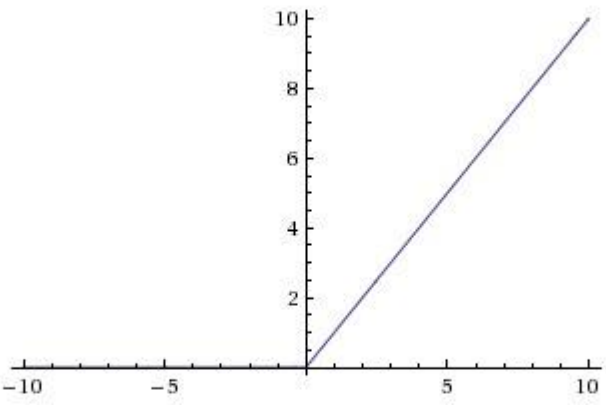

Figure 3.6 Rectified Linear Unit (ReLU) Activation Function (Johnson, 2018)

3.4 ARTIFICIAL NEURAL NETWORK ARCHITECTURES

Generally, the architectures of artificial neural network have three parts, named layers which are known as:

- Input Layer: The information/data, signals or measurements from the external environment are received in the input layer. These inputs renormalized by using activation Function limit values. The normalization process makes the results in better numerical performance.
- Hidden Layer: The information coming from the input layer is extracted patterns which associate with the system being analyzed. Most of the internal processing from a network is performed by the hidden layer.
- Output Layer: This layer is the last layer of the architecture of artificial neural network which data are processed. Output layer performs and presents the final network output which results from processing performed in the hidden layer. (Silva, 2017)

The main architectures can be divided as below considering the neuron disposition, how they are interconnected and the layers are composed:

3.4.1 FEEDFORWARD NEURAL NETWORK

A feed-forward network consisting of an input layer, one or more layers of neurons, hidden layers, an output layer.

There is no feedback from outputs of the neurons towards the input. (Sazli,2006) In the case of feed forward neural network, input data are fed the network and transformed into outputs.

Feed-forward neural Networks cannot predict future data in consideration of having no memory.

Continuous neurons, frequently with sigmoidal activation, are used in the context of backpropagation.

3.4.1.1 Single Layer Feedforward Network

Single layer feedforward consists of just one input layer and a single neural layer which is called an output layer.

The input signal flows input layer to the output layer in unidirectional. Pattern classification and linear filtering problems usually use these network.

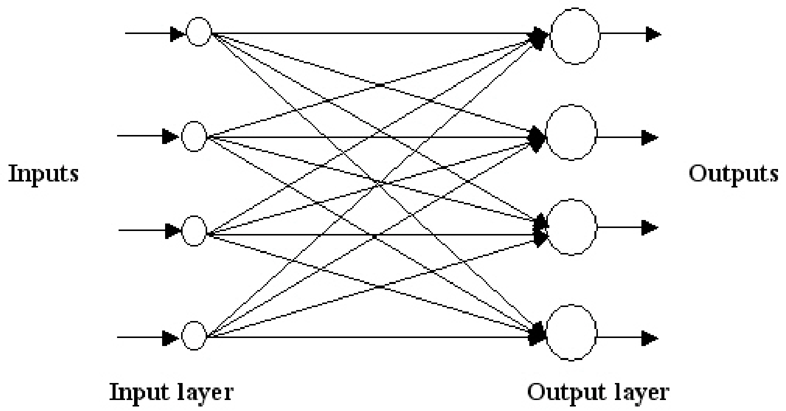

Figure 3.7 A Single Layer Feed-Forward Neural Network (Sazli, 2006)

Feedforward architecture has three main network types, which can learn algorithms used in their training process are listed below:

3.4.1.1.1 Thresholding Logic Unit

The most fundamental unit of the neural network is called as artificial neuron or perceptron which was founded by McCulloch and Pitt in 1943, by assimilating of a human brains neurons work system (Lagandula, 2018). In McCulloch and Pitt's computational neuron system, it is divided into 2 parts. The first part, g takes an input, like dendrite in the neuron, and collects them together. The second part f makes a decision based on the collection coming from g part.

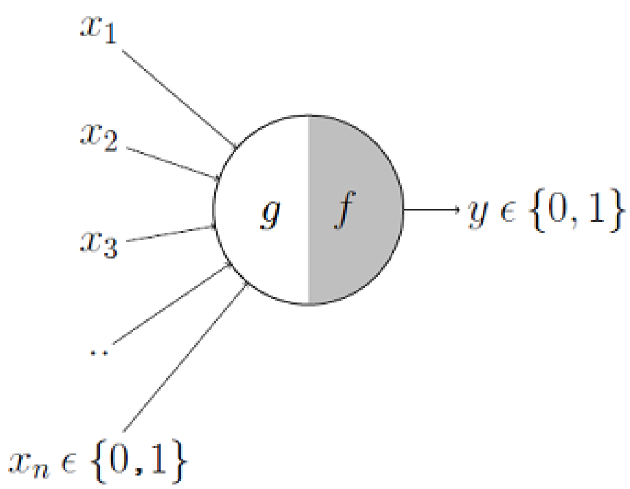

Figure 3.8 McCulloch and Pitt Neuron Model (Lagandula, 2018)

The input can be *excitatory* or *inhibitory* which inhibitory inputs have the maximum and pressor effect on the decision. Excitatory inputs cannot make the neuron fire and it needed a companion with them to fire a neuron. The formal equation is shown as;

$$g(x_1, x_2, \ldots, x_n) = g(x) = \sum_{i=1}^{n} x_i \tag{7}$$
$$y = f(g(x)) = 1 \; if \; g(x) \geq \theta$$
$$= 0 \; if \; g(x) < \theta$$

In the equation g(x) is collect inputs, theta is called *thresholding parameter.*

3.4.1.1.2 Perceptron

Frank Rosenblatt, an American psychologist, proposed a classical perceptron model in 1958 and the model is refined by Minsky and Papert in 1969 is the more computational model than McCulloch and Pitts (M-P) neuron. M-P neurons have limitations such as working with Boolean values; the perceptron model is overcoming with numerical weights and working real numbers which makes it more useful and generalized.

Perceptron is an algorithm for supervised Learning and binary classifiers which can detect to whether or not an input belongs to some class. It is based on a linear predictor Function combining a set of weights with the characteristic vector. Perceptron separates the input into two halves, negative and positive. All the inputs produce output 1 in the positive half side while all inputs produce 0 lines on the negative side.

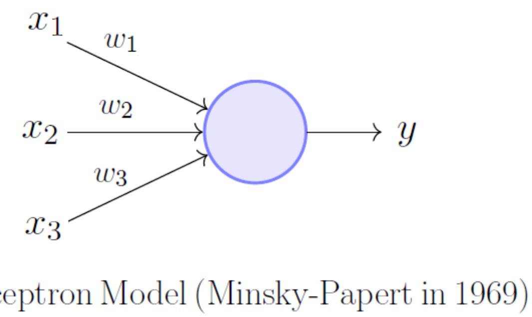

Figure 3.9 Perceptron Model of Minsky-Papert (Lagandula, 2018).

3.4.1.1.3 Adaline

Adaline is a single-layer artificial neural network which is based on McCulloch-Pitts neuron. It was developed by Prof. Bernard Widrow and his student Ted Hoff in 1960. ADALINE, which is Adaptive Linear Network, is not very different from perceptron except using a linear Function instead of the threshold function. (Berberler, 2017)

The Widrow-Hoff rule or the least-square algorithm gives accurate solution than perceptron model which can be sensitive to noise.

3.4.1.2 Multiple-Layer Feedforward Architecture

Multiple-layer feed work architecture has multiple layers are composed of one or more hidden layers.

The figure shows that a feedforward network with multiple layers has one input layer, called Layer 0, with 3 sample signals, two hidden layers which are Layer One and Layer Two, and the one last layer as the output layer. Also, it shows the input flow direction as the input layer to the output layer in a single direction.

The figure also shows that the amount of the neurons creating the first hidden layer is usually different from the signals creating the input layer of the network. The problems complexity affects the number of hidden layers and its neuron numbers.

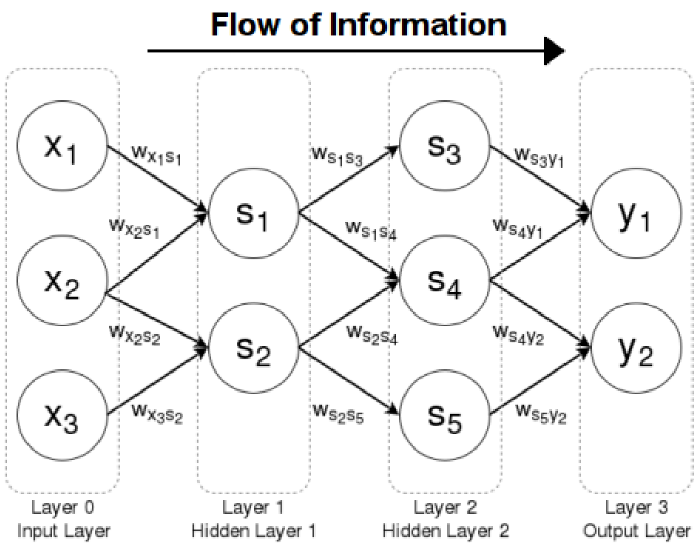

Figure 3.10 A Feedforward Neural Network Flow of Information (Brilliant, 2018)

The Function of a multilayer feedforward network calculates is given below: (Leshno, 1991)

$$f(x) = \sum_{j=1}^{k} B_j * \psi(w_j * x - \theta_j) \qquad (8)$$

k = the number of the processing unit;
w_j = the set of weights, one for each pair of connected units;
ψ = an activation Function $\psi: R \to R,$ same for each processing unit;
θ_J = the set of threshold values, one for each processing unit

Multilayer Perceptron (MLP), Radial Basis Function (RBF) and Backpropagation Algorithm (BP) are the examples of multi-layer feedforward architectures. These types are based on generalized delta rule and the competitive delta rule.

3.4.1.2.1 Multilayer Perceptron (MLP)

MLP is the most popular architecture in ANN. A set of interconnected layers of artificial neurons, inputs, hidden and output layers interconnect to organize the MLP. Input layer provides a neural group with data; the neurons in the first layer propagate the weighted data and select random bias through hidden layers. The output response is provided at the node by using the transfer Function, after the net sum at a hidden node is determined.

MLP has two important characteristics are its non-linear processing element which has a non-linear activation Function and its massive interconnectivity. (Memarian and Balasundram, 2012)

3.4.1.2.2 Radial Basis Function (RBF)

RBF is another popular architecture used in ANN. The general architecture of RBF is similar to MLP by having three layers as input, hidden and output. The similarity of inputs to examples from the training set is measured and RBF performs classification by using this measurement. One of the examples from the training set which called as the prototype is stored by RBF neurons. While a new input will be classified, each neuron computes the Euclidean distance between the input and its prototype. (Radial Basis Function Network Tutorial, 2013)

3.4.1.2.3 Backpropagation Algorithm

While designing a neural network, weight values are selected as randomly and the error is calculated based on selected weight values. Consequently, the error can exceed the tolerance values of the algorithm and then until the error calculation is acceptable the selected weight values should be changed.

The backpropagation algorithm looks for the minimum value of the error Function in weight space using a technique called delta rule or gradient descent. The diagram of the algorithm is given in the Figure 3.11:

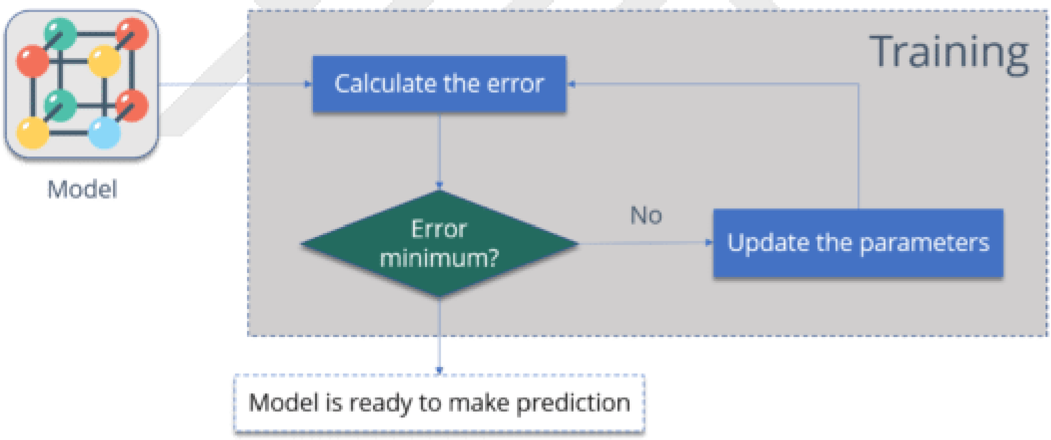

Figure 3.11 Back Propagation Training Steps (Saurabh, 2018)

The summarize of the steps:

- Calculating the error: The distance between actual output and model output.
- An error is minimum : Check the error value is acceptable or not
- Update the parameters: If the error is not acceptable, updates the parameters(weights and biases) and again check the error. Repeat the process until the error becomes minimum.

- Model is ready to make the prediction: If the error gets minimum, the model can be fed by inputs and ready to produce the output.

The learning algorithm has two stages such as propagation and weight update:

3.4.1.2.3.1 Propagation

Each propagation involves the following steps:

- Propagation forward through the network to generate outputs
- Calculation the error term
- To calculate the difference between the targeted and actual output values (deltas), using the training pattern target propagation of the output activations back through the network.

3.4.1.2.3.2 Weight Update

For each weight following steps must be followed:

- The gradient of the weight is calculated by multiplying the weight's output delta and input activation
- A percentage of the weight's gradient is subtracted from the weight.

3.4.2 RECURRENT NEURAL NETWORK (RNN)

The connection between nodes has a direction associated with them. Recurrent Neural Networks were developed in the 1980s by John Hopfield.

The basic differences between feed-forward network and recurrent neural network are that feedback loop is connected to their past decisions. It is said to the recurrent network have memory and adding this memory is the help to use the information in the sequence itself.

RNN's are able to remember important issues about inputs, which helps to be precise in prediction. In another saying is, RNN is a Function f_W with inputs x_t and previous state h_{t-1}. The new state will be h_t.

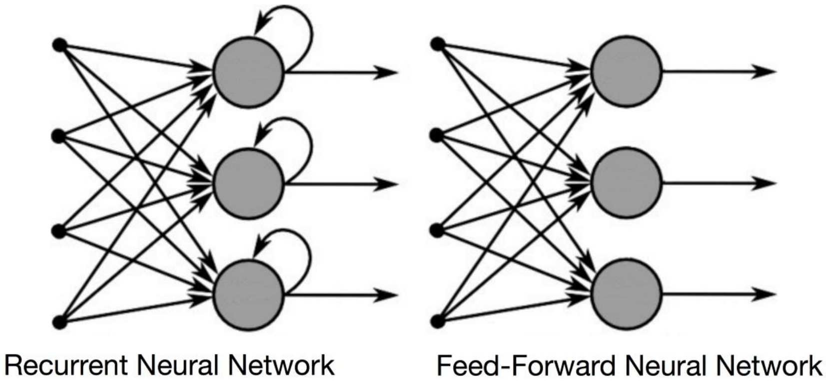

Figure 3.12 Recurrent Neural Network and Feed-Forward Neural Network Information Flow Difference (Donges, 2018)

The figure presents the difference in the information flow between the recurrent neural network and the feed-forward neural network. The feed-forward neural network has no memory of the input and never touches a node twice exact opposite of recurrent neural network.

Feed-Forward Neural Network map one input to one output. However, RNN can map one to many, many-to-one(speech recognition) and many-to-many(translations between languages).

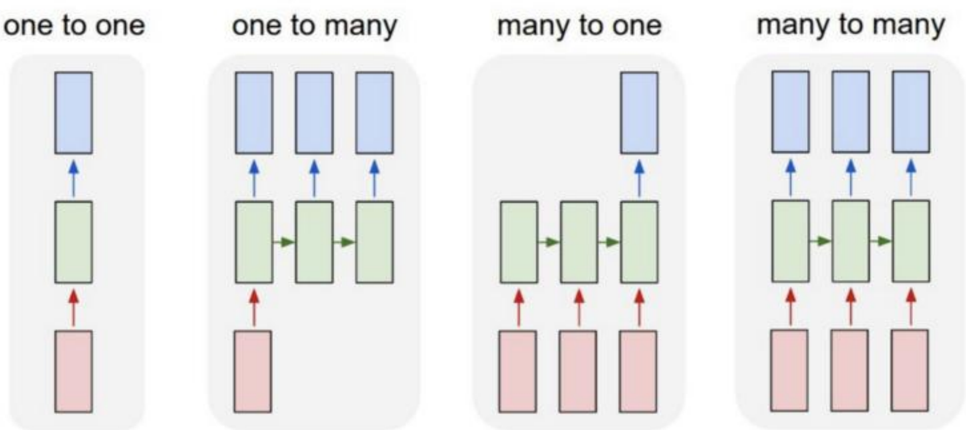

Figure 3.13 Feedforward Neural Network One to One Map and RNN's One to Many, Many to One and Many to Many Map (Donges, 2018)

3.4.3 CONVOLUTIONAL NEURAL NETWORK:

The main idea of the convolutional neural network are similar to feed-forward neural Networks, the neurons have learnable wights and biases. ConvNet is mostly applied in signal processing, image, and classification techniques.

ConvNet extracts features from the input in batchwise like a filter which helps the network have a memory about the input data like images in parts and can estimate the conversion of the input. After operations, input data like images can be classified by using pixel value changes and detecting the edges.

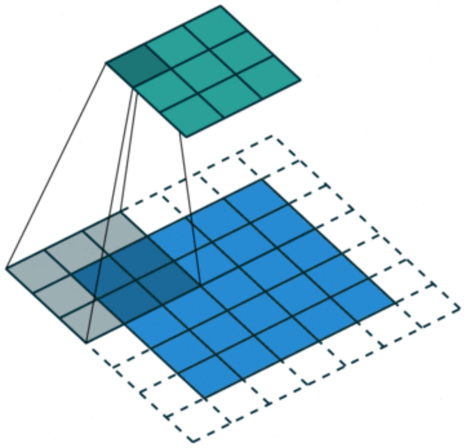

Figure 3.14 Convolutional Neural Network Figure (Maladkar, 2018)

3.5 APPLICATIONS OF ARTIFICIAL NEURAL NETWORK

Neural network algorithms can be applied an extensive spectrum of data-intensive applications such as mainly listed below;

- ❖ Fault Diagnosis: ANN is a network can learn and remember the past Network can learn the systems work properly and can detect the fault by severalizing the difference between true or false.
- ❖ System Planning: By using ANN's features for prediction, system needs can be calculated and used for budgets, transmission designs, generation, and distribution planning.
- ❖ Medical Areas: Detection of cancer cells, analysis of signals of EEG or ECG, the design of prosthesis and optimizations of expense in hospitals can be determined by using ANN.

- ❖ Communication Fields: Image processing, translations of languages, conversion of speech to handwrite etc. can be shown as examples.

3.6 NETWORK TRAINING

One of the most important features of the artificial neural network is the ability to learn like the human brain. It is known that the human brain's neural structure is changed, increasing or decreasing the strength of its synaptic connections depending on their activity while during the process. The more relevant information means more easy to recall, because of more relevant information have stronger connections than less relevant information. (Jocobson, 2014)

Learning algorithms are useful for difficult problems like facial recognition and there are different algorithms that can be used, each with their own advantages and disadvantages. The three major Learning algorithms are explained below:

3.6.1 SUPERVISED LEARNING

The Learning algorithm the desired output for the network is provided with the input while the training the network. By providing with both an output and input pair allows the calculate an error based on its target output and actual output. It can use the error for correcting the weights.

There are many different supervised learning algorithms but the most popular one is backpropagation.

3.6.2 UNSUPERVISED LEARNING

In supervised Learning, the neural network finds some kind of pattern with only given a set of inputs. The system during the process arranges the weight values of the neural net inside a certain range depending on given input values. The aim is to group similar weight values together in certain areas of the value range. This Learning type can be used effectively in pattern classification purposes.

3.6.3 REINFORCEMENT LEARING

Reinforcement Learning appears similar to supervised learning by having feedback. However, instead of having a target output, a reward is given based on how well the system performed. The aim of this Learning is to maximize the reward the system receives through trial and error.

4. EXPERIMENTAL WORKING STEPS

In this project, the prediction of load forecast by using the artificial neural network is processed. Matlab programming and NARX method are used.

4.1 NONLINEAR AUTOGRESSIVE NEURAL NETWORK WITH EXOGENEOUS INPUTS

The load consumption data are time series, the NARX neural network is a good predictor of time series which is used in this research study. NARX is a nonlinear generalization of the Autoregressive Exogeneous (ARX), which has a linear black-box system.

The NARX is a recurrent dynamic neural network. It has feedback connections and utilizes the memory ability using the past values of predicted time series. The next value of the dependent output signal y(t) is regressed over the latest inputs of the NARX, which represents as n_x, and the dependent output signal n_y. A mathematical description of the NARX model is summarized below while f is a nonlinear Function: (Amani et al, 2011)

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y), x(t-1), x(t-2), \dots, x(t - x_n)) \quad (9)$$

The NARX network has three layers namely the input layer, hidden layer, and output layer. The input layer consists of the current and previous inputs and outputs. Hidden layer is fed by the input layer and consists of one or more neurons which are the mapping of an affine weighted combination of the values from the input layer. The last layer is output layer has affine combination values of from the hidden layer.

There are two different architectures of NARX neural network model, open-loop (series-parallel) and close-loop named also parallel architecture given by the equations respectively;

$$\hat{y}\,(t+1) = F\begin{pmatrix} y(t), y(t-1), \dots, y(t-n_y), x(t+1) \\ x(t), x(t-1), \dots, x(t-n_x) \end{pmatrix} \quad (10)$$

$$\hat{y}\,(t+1) = F\begin{pmatrix} \hat{y}(t),\ \hat{y}(t-1), \dots, \hat{y}(t-n_y),\ x(t+1) \\ x(t),\ x(t-1), \dots,\ x(t-n_x) \end{pmatrix} \quad (11)$$

F(.) = the Function of neural network;

$\hat{y}\,(t+1)$ = the output of the NARX at the time t for the time t+1 (the predicted value of y for the time t+1);

$\hat{y}(t), \hat{y}(t-1), \ldots, \hat{y}(t-n_y)$ = the past outputs of the NARX;

$y(t), y(t-1), \ldots, y(t-n_y)$ = the true past values of the time series (desired output values);

$x(t), x(t-1), \ldots, x(t-n_x)$ = the inputs of the NARX;

n_x = the number of input delays;

n_y = the number of output delays.

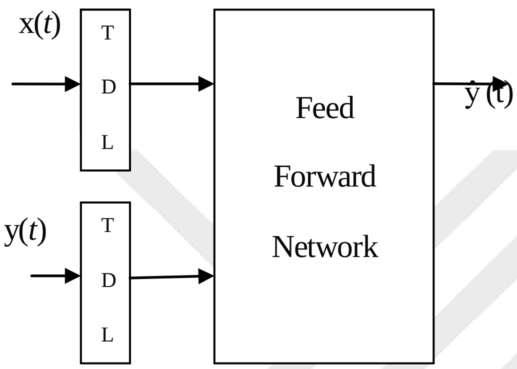

Figure 4.1 Series Parallel Architecture (open-loop) (Amani et al, 2011)

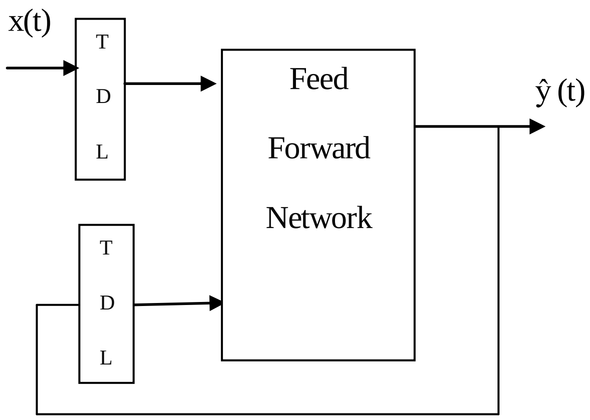

Figure 4.2 Parallel Architecture (Closed-Loop) (Amani et al, 2011)

The future value of the time series y(t-1) is predicted from the present and past values of input x(t) in the series-parallel architecture(open-loop). However, in the parallel architecture, the past predicted value of the time series ŷ(t), the present and past values of input x(t) are used to predict to the future values of the time series y(t-1). (Boussaada et al, 2018)

Open-loop architecture has some advantages. The first one is that the usage of true past values as inputs of the feedforward network. Also, after the training phase, the general algorithms for Multi-Layer Perceptron network can be used and the NARX neural network type can be changed to parallel architecture for the multi-step-ahead prediction.

In order to build the network, some of the steps are given below:

1. Recording and collecting data
2. Data Normalization Process
3. Artificial Neural Networks Designing
4. Training and Testing the ANN

4.2 RECORDING AND COLLECTING DATA

The first step of the build the network of prediction is determining which data are needed. Secondly, the determined data is needed to record, after recording process data are should collect.

For this project, student number of the associated school year, the hourly atmospheric temperature, and electric demand load as hourly and holidays of the year are needed and collected.

First of all, electric demand loads are recording since 2016 at Yaşar University buildings. In this thesis, firstly 2016 load are collected and used for analysis. The consumed electrical loads were tabulated as daily and hourly for understanding how the consumption patterns are changing.

It was seen in the table analysis, weekends, and holidays and off hours affect the consuming. In these figures, the effects of holidays like weekends and religious days and atmospheric temperature are clearly understandable.

The Figure 4.3 and Figure 4.4 The Load Consumptions Of January 2016 are given below:

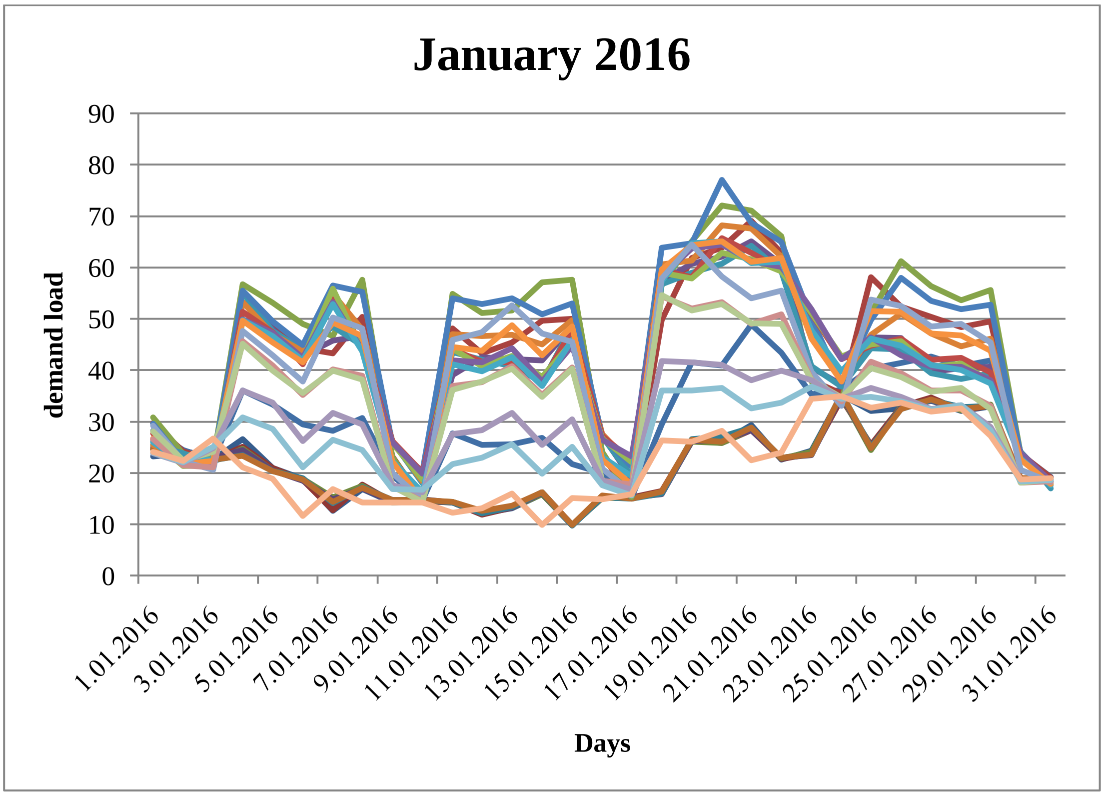

Figure 4.3 The Load Consumption Figure Of January 2016

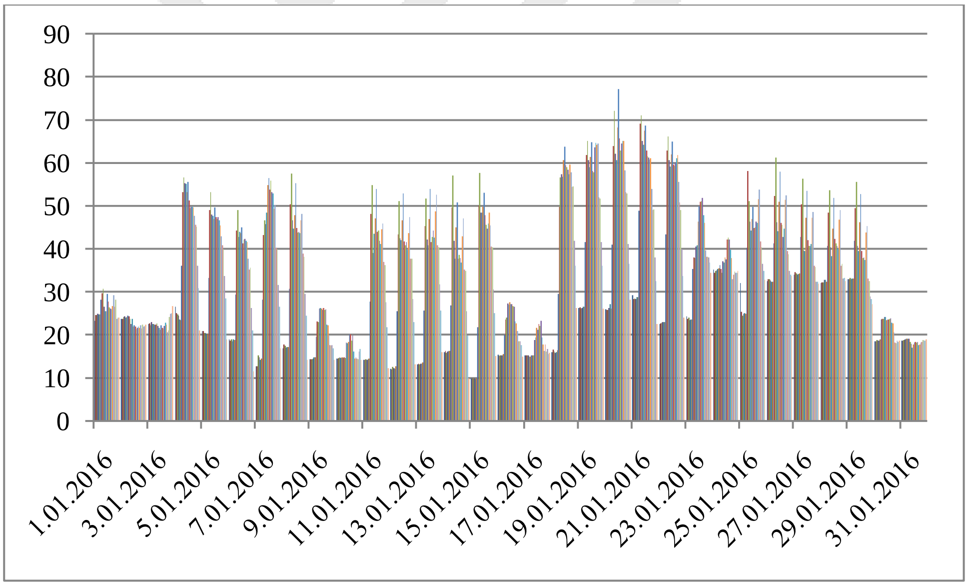

Figure 4.4 The Daily Load Consumption Of Yasar University January 2016

The figures are the load consumption of Yasar University January 2016. The semester of 2015-2016 had begun on 28 Sep 2015 and ended at 8 Jan 2016. After 8 January and the final exam's last day was February 12. The school was on a semester break at between 12 February to 22 February, the, decreasing on student number in the school causes a decreasing on a load demand line.

The load consumption Figures 4.5 and Figure 4.6 of February 2016 are given below:

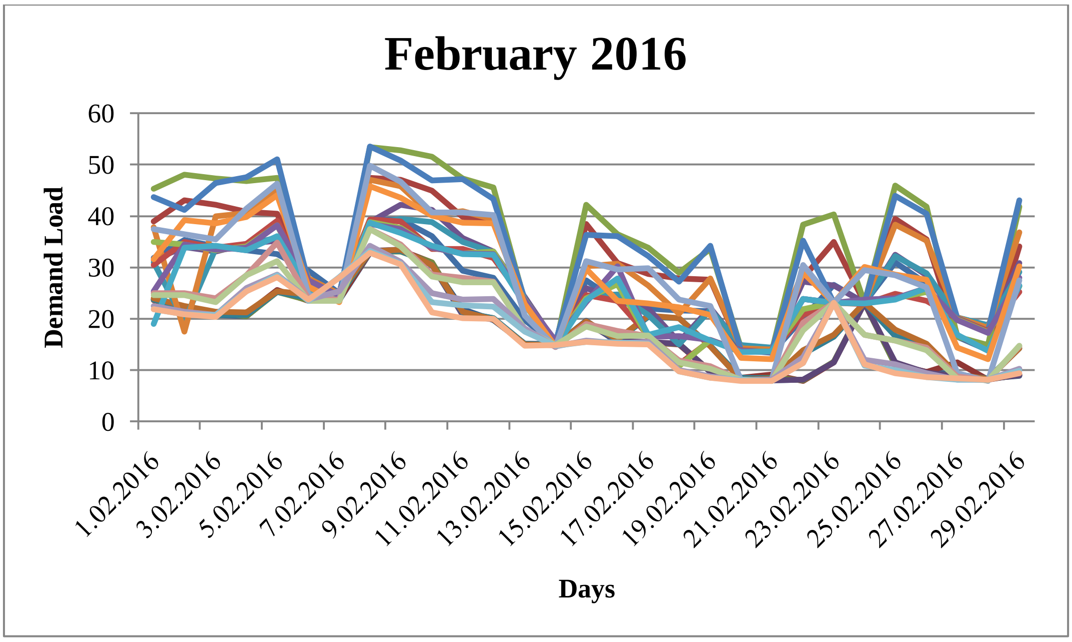

Figure 4.5 The Load Consumption Figure Of February 2016

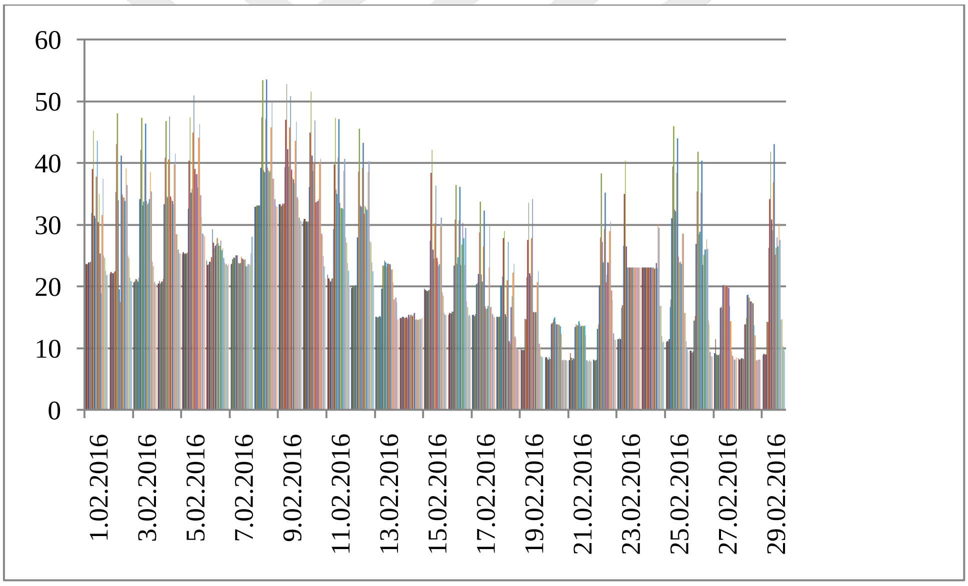

Figure 4.6 The Daily Load Consumption Figure Of February 2016

As can be seen from graphics, the load consumption of the university is bottom out at weekends like 6 and 7 February, 13 and 14 February, 20 and 21 February and the last 27 and 28 February 2016.

The load consumption Figure 4.7 and Figure 4.8 of March 2016 are given below:

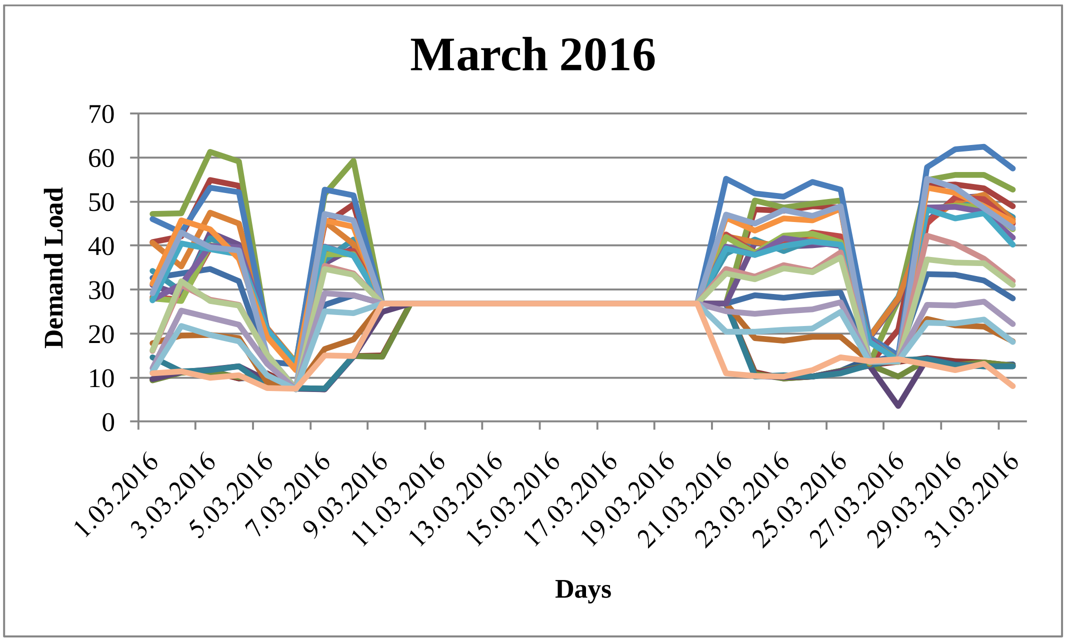

Figure 4.7 The Load Consumption Figure Of March 2016

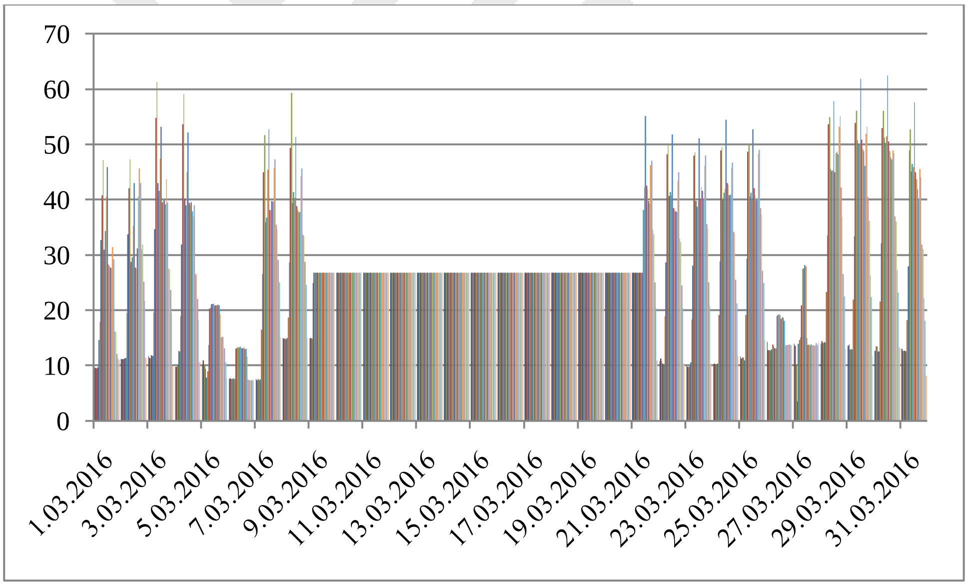

Figure 4.8 The Daily Load Consumption Figure Of March 2016

Load consumption records have same values which are 26,84 kWh on 9 March 2016 03:00 until 21 March 2016 10:00 am. This fault can be caused by the recorder.

The load consumption Figure 4.9 and 4.10 of April 2016 are given below:

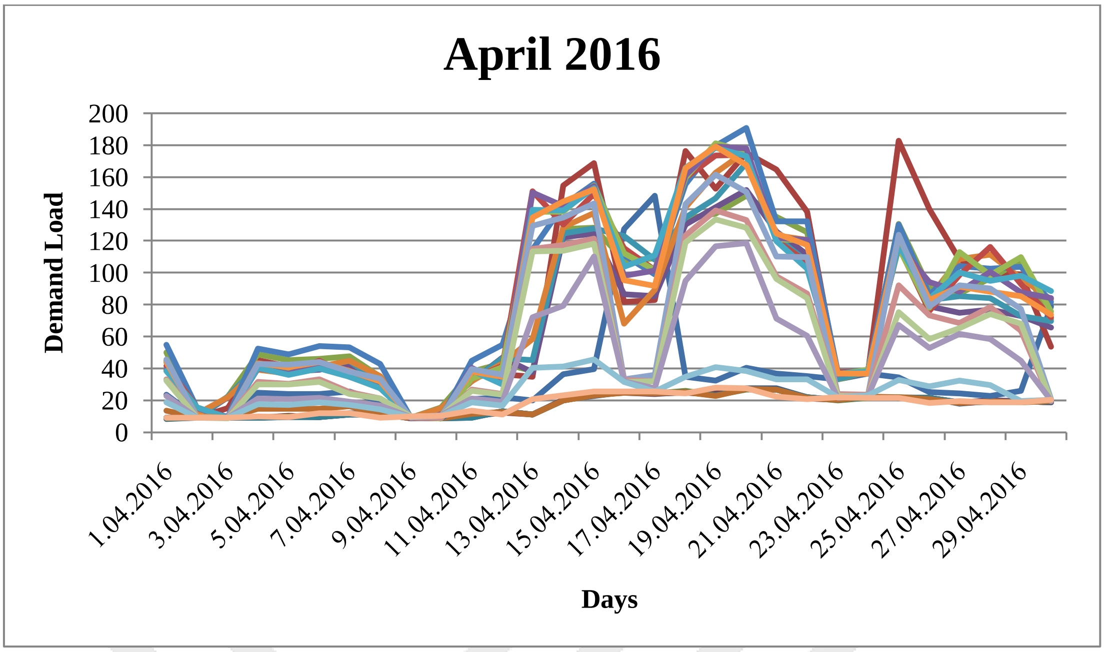

Figure 4.9 The Load Consumption Figure Of April 2016

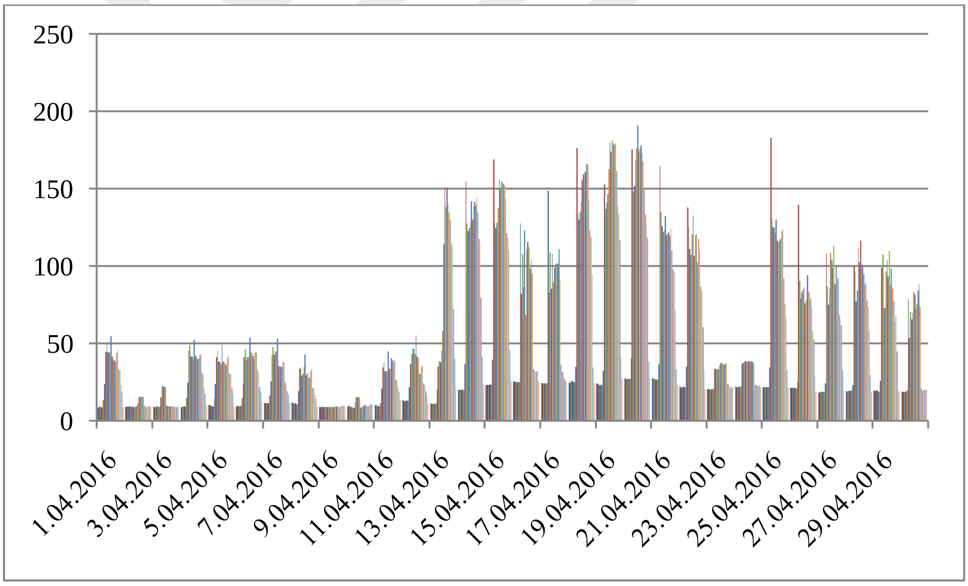

Figure 4.10 The Daily Load Consumption Figure Of April 2016

As can be seen from the demand load graphics, 23 April 2016 load demand is lower than the other days because of the day is important for Turkey. Every 23rd April is celebrated in Turkey as International Children's Day and it is a national holiday. The graph is showing the importance of the holiday for forecasting.

The load consumption Figure 4.11 and Figure 4.12 of May 2016 are given below:

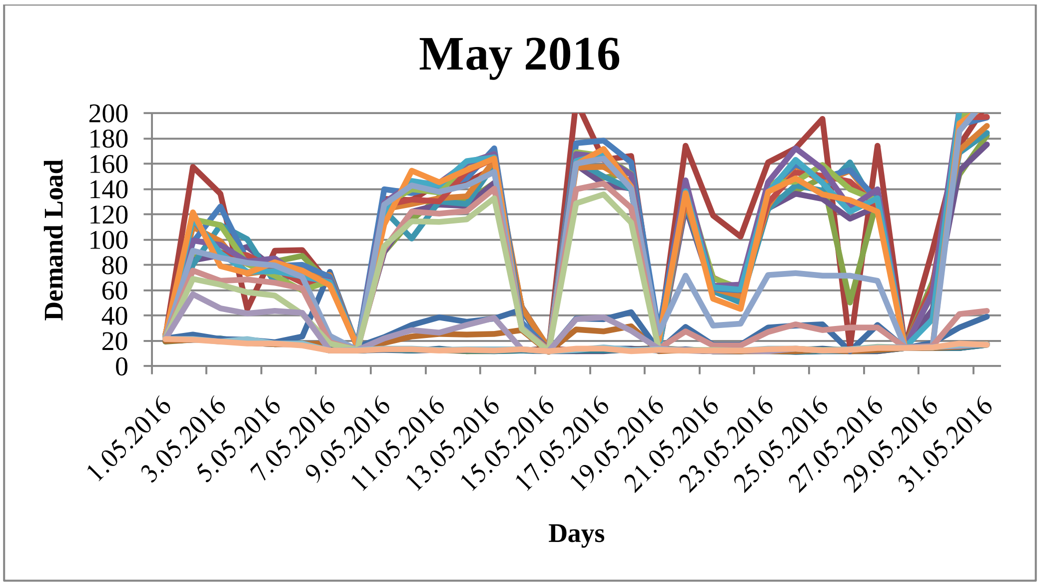

Figure 4.11 The Load Consumption Figure Of May 2016

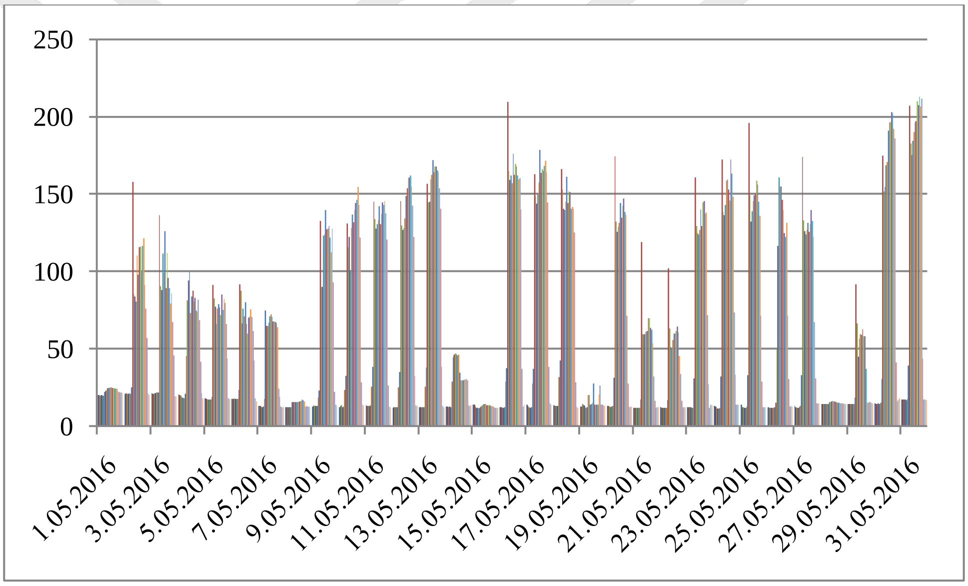

Figure 4.12 The Daily Load Consumption Figure Of May 2016

The May Day and the May 19 The Commemoration of Ataturk, Youth and Sports Day are the national holidays in May and the load consumption line is affected from the holidays.

Load consumption line of 19th May is similar to weekends like 28th May and 29th May.

The load consumption Figure 4.13 and Figure 4.14 of Jun 2016 are given below:

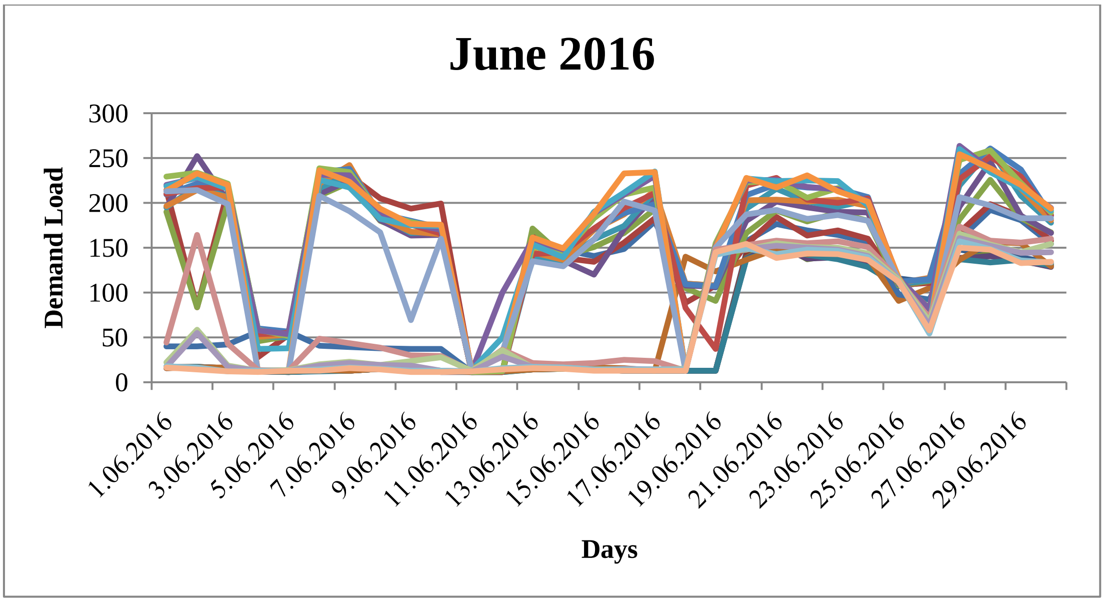

Figure 4.13 The Load Consumption Figure Of June 2016

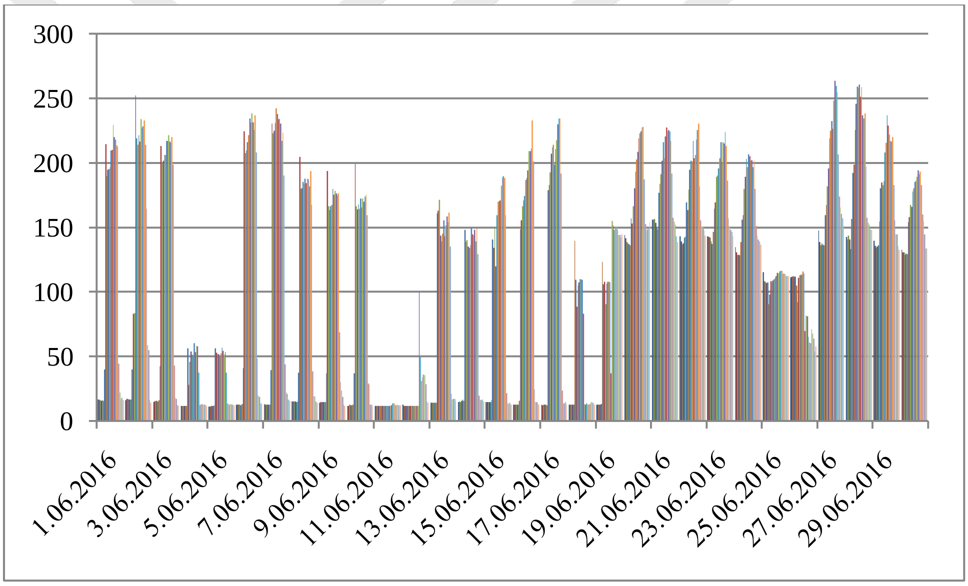

Figure 4.14 The Daily Load Consumption Figure Of June 2016

In the graph of demand load and days, the drop is seen at June 11 caused by the ending of the final exams. Also, the last days of June 2016, the summer school had begun and the student number was increased who caused to rise in load consumption.

The load consumption Figure 4.15 and Figure 4.16 of July 2016 are given below:

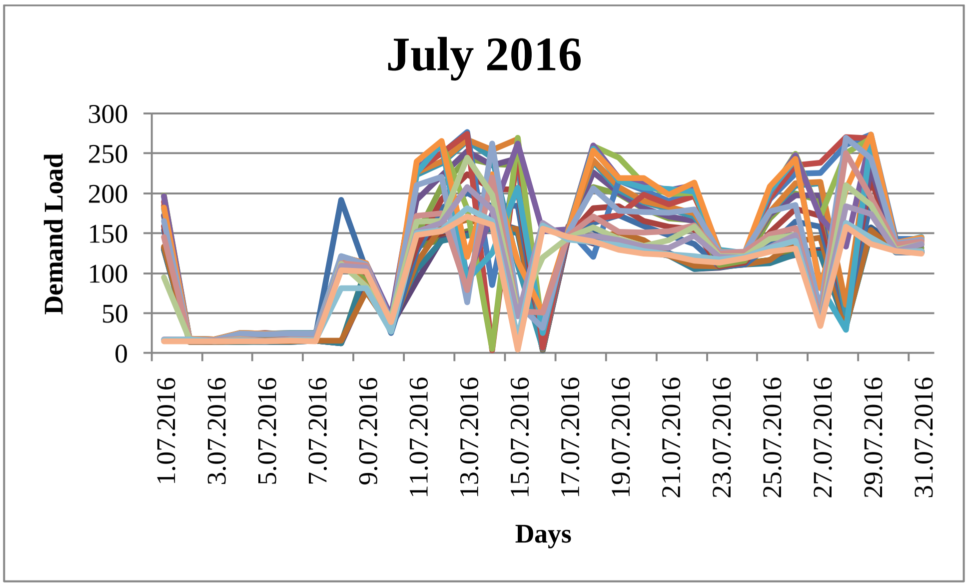

Figure 4.15 The Load Consumption Figure Of July 2016

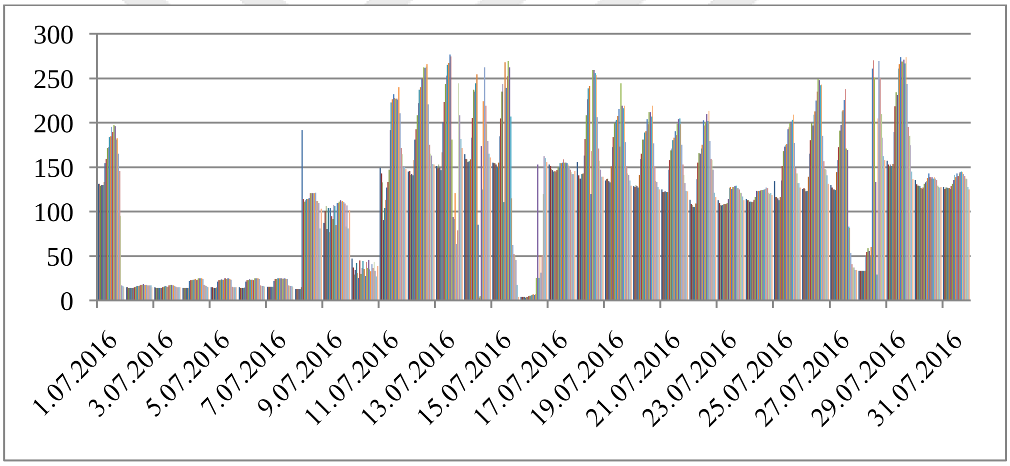

Figure 4.16 The Daily Load Consumption Figure Of July 2016

The Ramadan Bairam was started at 4 July and ended at 7 July. However, 2 July and 7 July was the weekend, the holiday of the Bairam was combined with weekends. The decreasing of the load demand on the table depends on the holiday.

The load consumption Figure 4.17 and Figure 4.18 of August 2016 are given below:

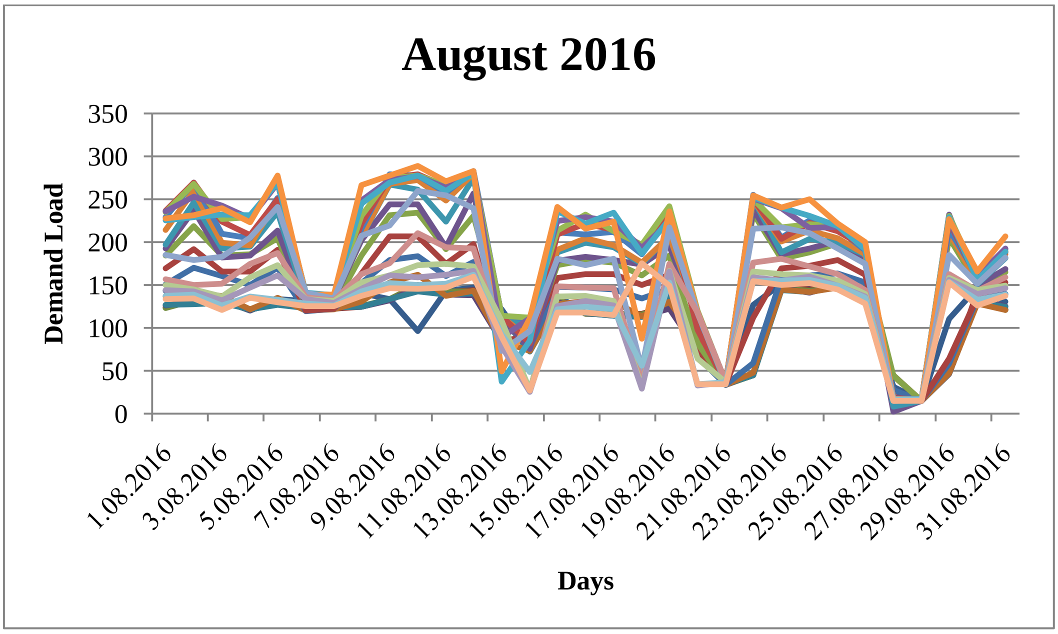

Figure 4.17 The Load Consumption Figure Of August 2016

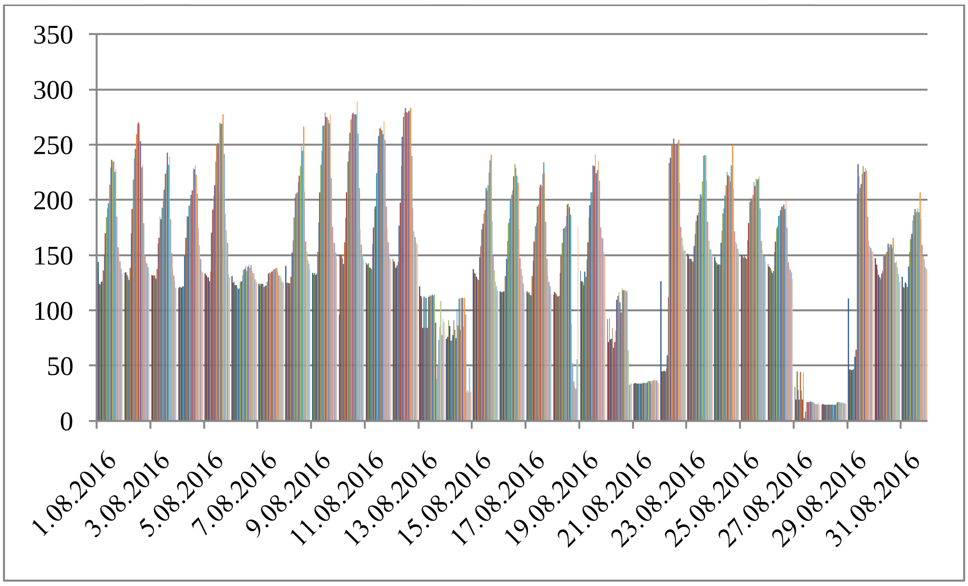

Figure 4.18 The Daily Load Consumption Figure Of August 2016

The higher temperature values and summer school student numbers gave rise to higher load consumption at Agust 2016.

The load consumption Figure 4.19 and Figure 4.20 of September 2016 are given below:

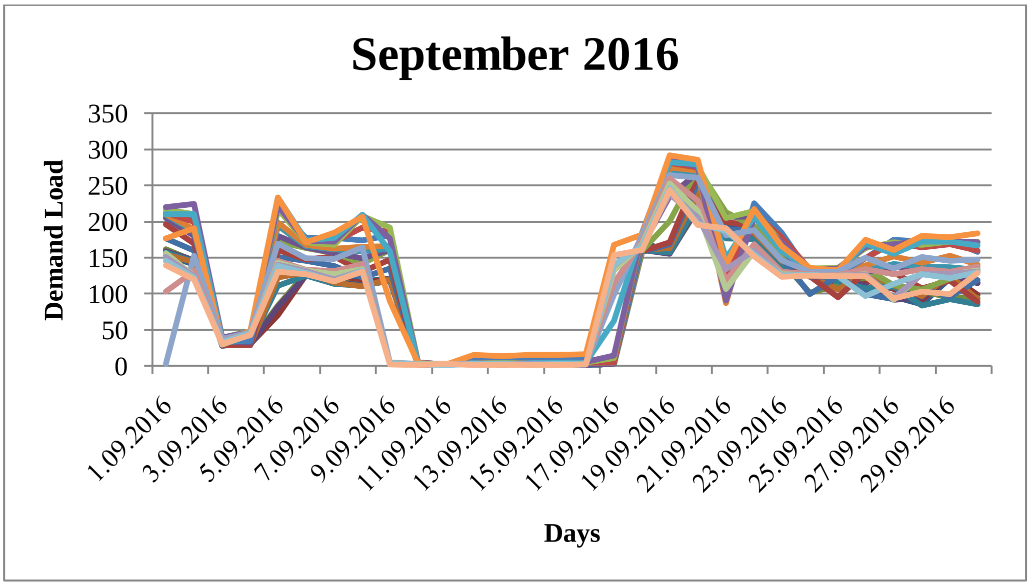

Figure 4.19 The Load Consumption Figure Of September 2016

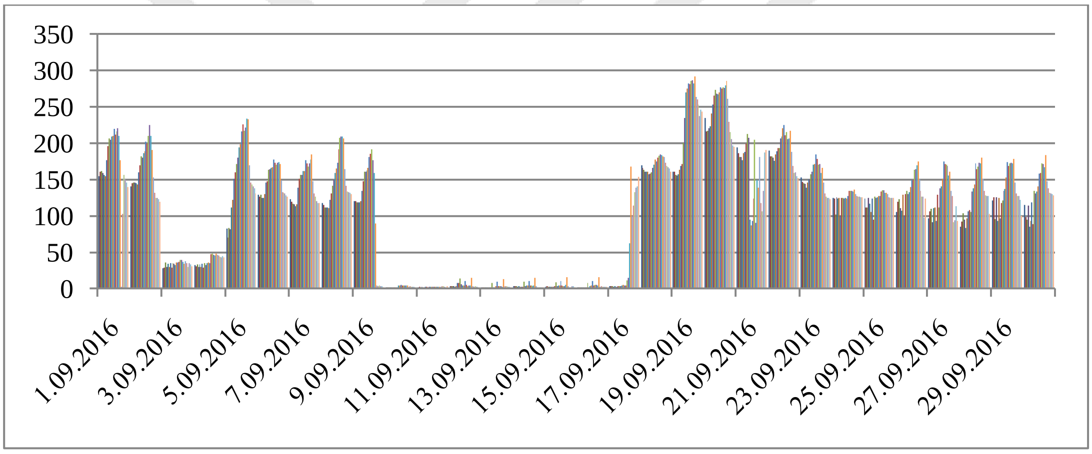

Figure 4.20 The Daily Load Consumption Figure Of September 2016

The Sacrifice Bairam was started at 12 September and ended at 15 September. However, 16 September and 17 September was the weekend, the holiday of the Bairam was combined with weekends. The decreasing of the load demand on the table depends on the holiday.

The load consumption Figure 4.21 and Figure 4.22 of October 2016 are given below:

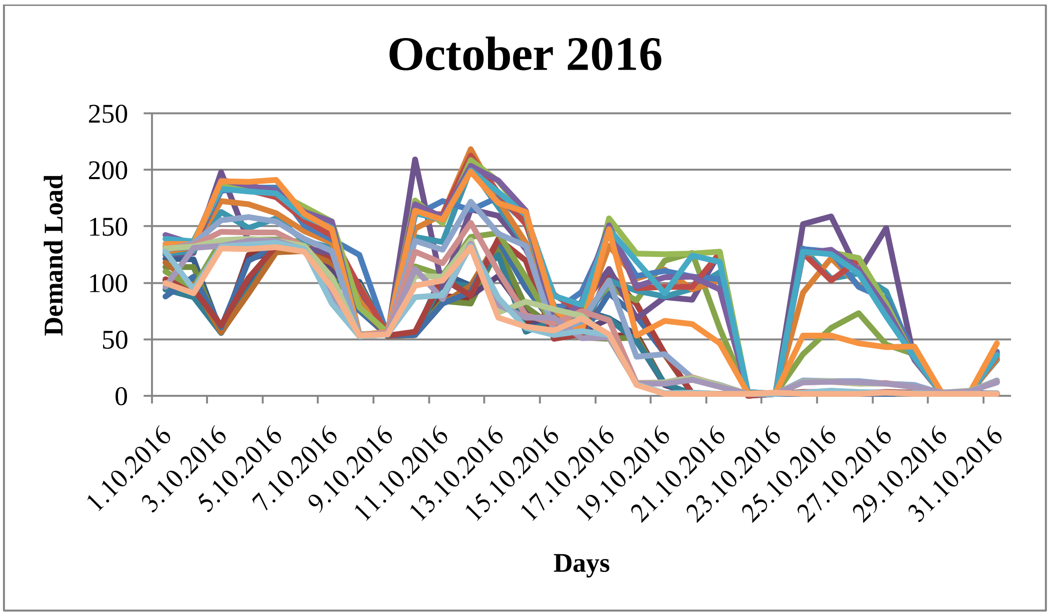

Figure 4.21 The Load Consumption Figure Of October 2016

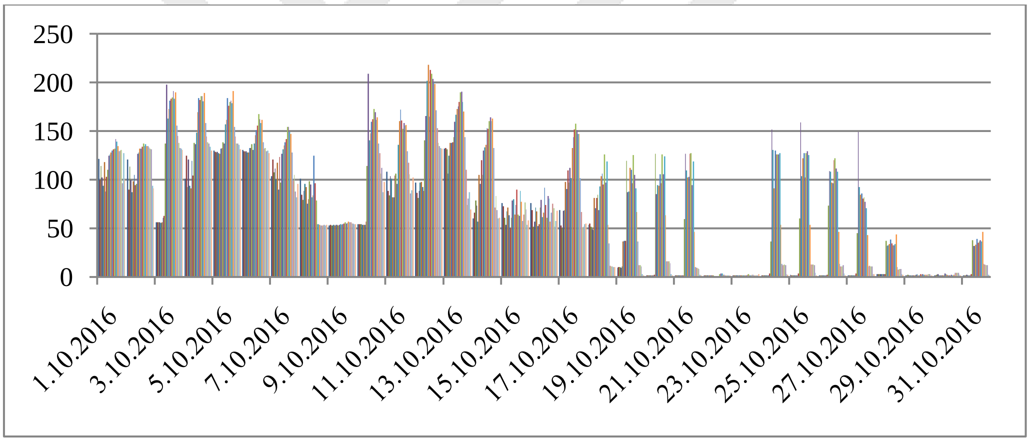

Figure 4.22 The Daily Load Consumption Figure Of October 2016

In Turkey, The Republic Day is an important day to celebrate. Every 29 October, the day is celebrated all over the country in Turkey. Also, Republic Day eve is a half-day holiday for preparing for the day.

The load consumption Figure 4.23 and Figure 4.24 of November 2016 are given below:

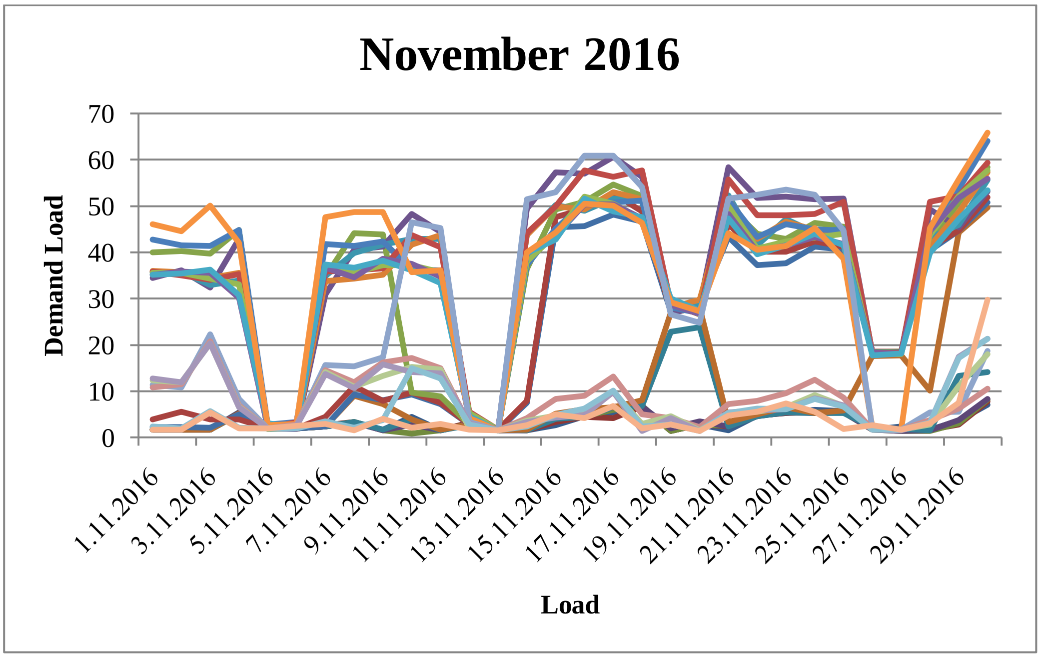

Figure 4.23 The Load Consumption Figure Of November 2016

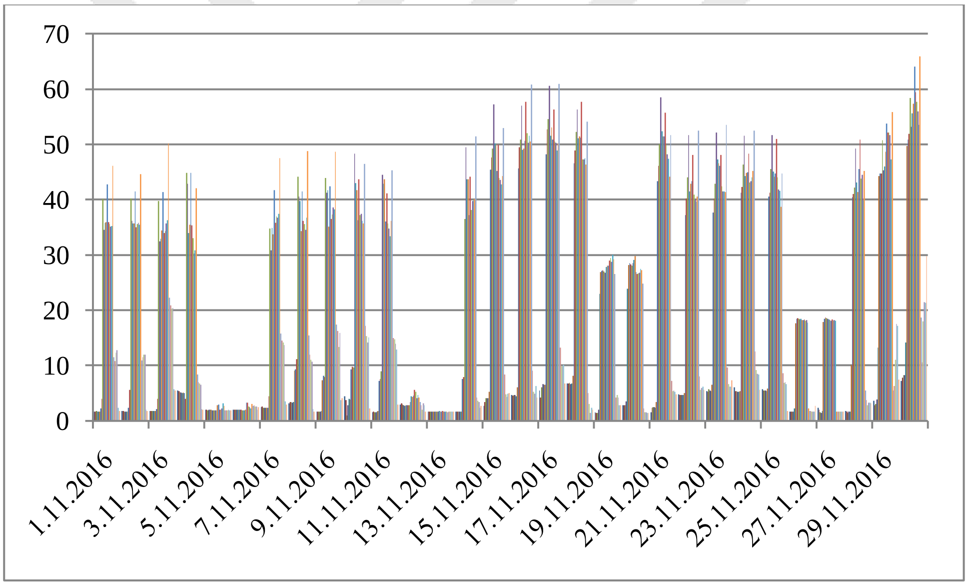

Figure 4.24 The Daily Load Consumption Figure Of November 2016

The load consumption Figure 4.25 and Figure 4.26 of December 2016 are given below:

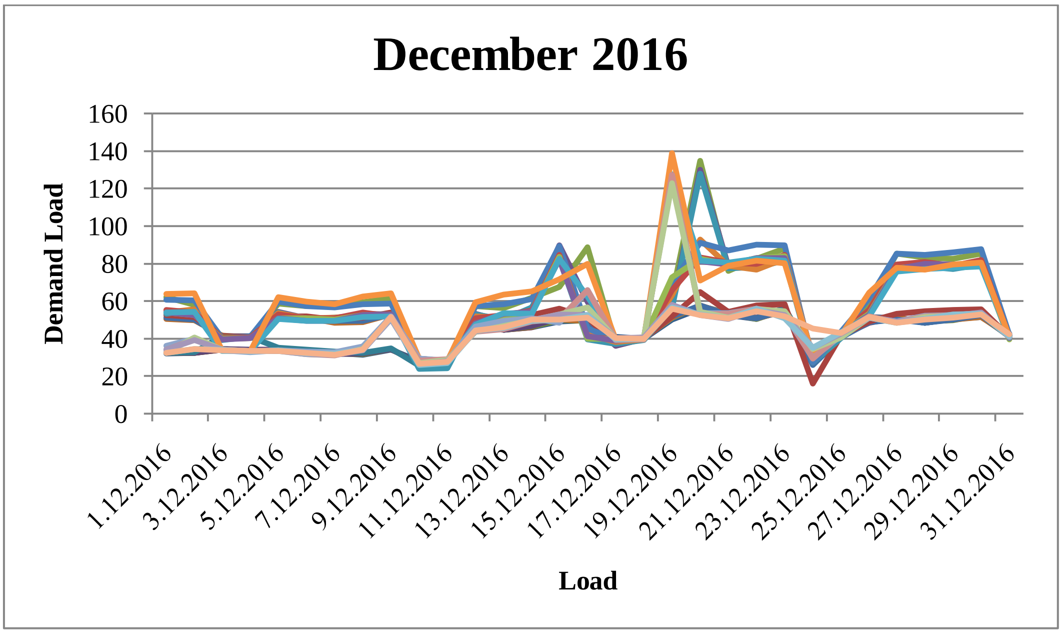

Figure 4.25 The Load Consumption Figure Of December 2016

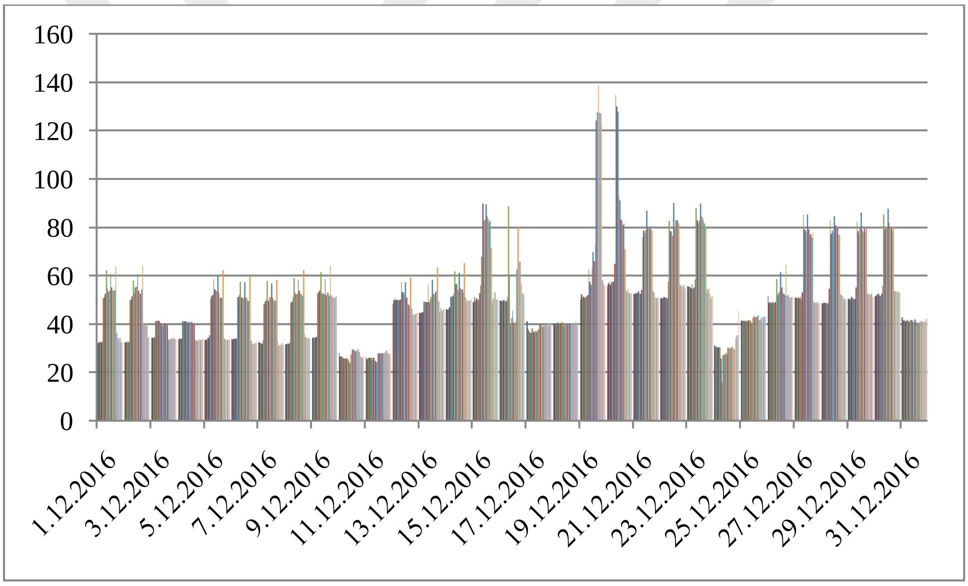

Figure 4.26 The Daily Load Consumption Figure Of December 2016

At 13th December school term is over for the student of English preparation class and the 26th December general school term is over. The line is dropped because of the student number decreasing.

By analyzing the tables of hourly and monthly documented load demand, the important features of the forecast are understood. The load demand behavior, like 24 hours earlier than the forecast time or last week same hour load demand, is the most import feature which affects the forecast.

The second important feature is the weather data. Some data like seasons and the temperature of the day change the using of air conditioner and heater.

The other important feature is the student number which takes the class during the analyzing semester. It affects the load demand, because of using air containing tools, lightning etc.

By taking these features into consideration, the table of 2016 is prepared for load forecasting as input data. The input data table is started with the date of 8 January, because load demand information of 2015 is not recorded for use, and ended at 31 December. The first day of the table 4.1 is shown below:

Table 4.1 The Table of 08.01.2016 Input Data

StartingDate	EndingDate	Temperat	StudentNumber	PrevWeekSameHourLoad	PrevDaySameHourLoad	Prev24Hour	IsWorkingDay	Day	Month
8.01.2016 00:00	8.01.2016 01:00	9,6	0	23,23	12,64	36,59375	1	8	1
8.01.2016 01:00	8.01.2016 02:00	8,6	0	24,55	12,75	36,76791667	1	8	1
8.01.2016 02:00	8.01.2016 03:00	8,7	0	24,57	15,19	36,97666667	1	8	1
8.01.2016 03:00	8.01.2016 04:00	8,9	0	24,97	14,83	37,07208333	1	8	1
8.01.2016 04:00	8.01.2016 05:00	8,8	0	24,73	14,17	37,16083333	1	8	1
8.01.2016 05:00	8.01.2016 06:00	9,3	0	24,74	14,42	37,28791667	1	8	1
8.01.2016 06:00	8.01.2016 07:00	8,3	0	28,22	28,19	37,4	1	8	1
8.01.2016 07:00	8.01.2016 08:00	7,2	0	29,66	43,29	37,50666667	1	8	1
8.01.2016 08:00	8.01.2016 09:00	7,4	0	30,79	46,65	37,80333333	1	8	1
8.01.2016 09:00	8.01.2016 10:00	10,2	0	26,58	45,77	38,25791667	1	8	1
8.01.2016 10:00	8.01.2016 11:00	12,4	0	25,52	48,51	38,29208333	1	8	1
8.01.2016 11:00	8.01.2016 12:00	12,9	0	25,48	54,84	38,13416667	1	8	1
8.01.2016 12:00	8.01.2016 13:00	14,0	0	29,52	56,43	37,8425	1	8	1
8.01.2016 13:00	8.01.2016 14:00	14,2	0	27,71	53,77	37,79208333	1	8	1
8.01.2016 14:00	8.01.2016 15:00	14,6	0	26,33	55,91	37,41791667	1	8	1
8.01.2016 15:00	8.01.2016 16:00	14,0	0	26,05	53,23	36,91583333	1	8	1
8.01.2016 16:00	8.01.2016 17:00	12,8	0	25,97	52,92	36,52791667	1	8	1
8.01.2016 17:00	8.01.2016 18:00	11,6	0	26,74	49,27	36,14416667	1	8	1
8.01.2016 18:00	8.01.2016 19:00	10,7	0	29,25	50,23	36,03458333	1	8	1
8.01.2016 19:00	8.01.2016 20:00	8,9	0	26,6	40,21	35,94541667	1	8	1
8.01.2016 20:00	8.01.2016 21:00	8,3	0	28,12	39,97	35,89333333	1	8	1
8.01.2016 21:00	8.01.2016 22:00	7,8	0	23,79	31,65	35,82	1	8	1
8.01.2016 22:00	8.01.2016 23:00	7,4	0	23,98	26,49	35,72958333	1	8	1
8.01.2016 23:00	9.01.2016 00:00	7,1	0	24,07	16,92	35,645	1	8	1

The table has titles that starting and ending date for declaring the date. Temperature data is hourly recorded, student number seems zero because of the school at spring break.

Artificial Neural Networks have the ability to learn from previous data. In the table, some hour load of the previous week and the previous day is used. Also, the average load of 2-previous 24 hours is calculated by using this data.

The day information is also used such as the working day or holiday. Furthermore, the day and the mounting number are used.

The artificial neural network has three layers as mentioned above; the input layer, hidden layer, and output layer. Output layer needs to target data to compare actual output and model output for calculating the error/accuracy of the network. In this thesis, target data are the actual load demand of the day. The below the first day's target data are shown in the table 4.2 :

Table 4.2 The Table of 08.01.2016 Target Data

StartingDate	EndingDate	LoadDemand
8.01.2016 00:00	8.01.2016 01:00	16,82
8.01.2016 01:00	8.01.2016 02:00	17,76
8.01.2016 02:00	8.01.2016 03:00	17,48
8.01.2016 03:00	8.01.2016 04:00	16,96
8.01.2016 04:00	8.01.2016 05:00	17,22
8.01.2016 05:00	8.01.2016 06:00	17,11
8.01.2016 06:00	8.01.2016 07:00	30,75
8.01.2016 07:00	8.01.2016 08:00	50,41
8.01.2016 08:00	8.01.2016 09:00	57,56
8.01.2016 09:00	8.01.2016 10:00	46,59
8.01.2016 10:00	8.01.2016 11:00	44,72
8.01.2016 11:00	8.01.2016 12:00	47,84
8.01.2016 12:00	8.01.2016 13:00	55,22
8.01.2016 13:00	8.01.2016 14:00	44,79
8.01.2016 14:00	8.01.2016 15:00	43,86
8.01.2016 15:00	8.01.2016 16:00	43,92
8.01.2016 16:00	8.01.2016 17:00	43,71
8.01.2016 17:00	8.01.2016 18:00	46,64
8.01.2016 18:00	8.01.2016 19:00	48,09
8.01.2016 19:00	8.01.2016 20:00	38,96
8.01.2016 20:00	8.01.2016 21:00	38,21
8.01.2016 21:00	8.01.2016 22:00	29,48
8.01.2016 22:00	8.01.2016 23:00	24,46
8.01.2016 23:00	9.01.2016 00:00	14,23

The Table 4.2 are prepared based on the consumption load hourly recorded data from the year of 2016.

4.3 DATA NORMALIZATION PROCESS

The second process of the analysis is the data normalization of the input and target tables. Normalization is a scaling or mapping technique which is used to scale up or scale down the range of data. There are so many normalization methods like Min-Max normalization, and Decimal scaling normalization.

4.3.1 Z-SCORE NORMALIZATION

Z-scores is also known as standardized scores that have been given a common standard. The standard is mean of zero and a standard deviation of 1. The standard deviation's Formula is; (Abdi, 2010)

$$\hat{S} = \sqrt{\frac{\sum(Y_n - M)^2}{N - 1}} \tag{11}$$

$\hat{S}$ = Standart Deviation

Y_n = A set of N scores

M = Mean

In order to normalize a set of scores using standard deviations, each score needs to divide by the standard deviation. The final Formula is;

$$Z_n = \frac{Y_n - M}{\hat{S}} \tag{12}$$

Z_n = The Mean of the Vector

4.3.2 DECIMAL SCALING NORMALIZATION

Decimal scaling normalization method normalizes by moving the decimal point of values of feature X. Maximum absolute value of X affects the number of decimal points moved. A normalized value v_n is corresponding to v is obtained using below formula in the equation 12 where c is the smallest integer such that max ($|v_n|$) $< 1.$ (Radivojac, 2005)

$$v_n = \frac{v}{10^c} \tag{12}$$

4.3.3 MIN-MAX NORMALIZATION

Min-max normalization is the method that used in this project and it performs a linear transformation on the original data values. This method preserves the relationship between original data and normalized data and the normalized data can be calculated using the below formula;

$$z = \frac{x - min_x}{max_x - min_x} \tag{13}$$

- x = input value
- z = normalized form of input

The input table and output table is normalized by using min-max normalization. The first day of the normalized input data are shown in The Table 4.3 is given below:

Table 4.3 The Normalized Table of 08.01.2016 Input Data

NorTemp	NorStdNum	NorPreWeekLoad	NorPreDayLoad	NorPre24HourAveLoad	Workday?	NorDay	NorMonth
0,257741014	0	0,047688323	0,010112479	0,08531412	1	0,233333333	0
0,235499656	0	0,052371997	0,010502785	0,08603671	1	0,233333333	0
0,23746207	0	0,052442962	0,019160487	0,086902782	1	0,233333333	0
0,242041036	0	0,053862257	0,017883121	0,087298651	1	0,233333333	0
0,240078622	0	0,05301068	0,015541284	0,087666861	1	0,233333333	0
0,251199633	0	0,053046163	0,016428343	0,088194111	1	0,233333333	0
0,228303472	0	0,065394032	0,065287585	0,088659127	1	0,233333333	0
0,205626254	0	0,070503495	0,118865983	0,089101671	1	0,233333333	0
0,209551062	0	0,074513004	0,130788064	0,090332495	1	0,233333333	0
0,269516163	0	0,059574921	0,127665614	0,092218492	1	0,233333333	0
0,318141975	0	0,055813788	0,137387787	0,092360244	1	0,233333333	0
0,328390581	0	0,055671859	0,159848135	0,091705072	1	0,233333333	0
0,353030888	0	0,070006742	0,165489834	0,090494992	1	0,233333333	0
0,356738114	0	0,06358443	0,15605152	0,090285821	1	0,233333333	0
0,365460176	0	0,058687861	0,16364475	0,088733461	1	0,233333333	0
0,353030888	0	0,057694355	0,154135472	0,086650394	1	0,233333333	0
0,327736443	0	0,057410496	0,153035518	0,085040987	1	0,233333333	0
0,299607843	0	0,060142639	0,140084448	0,083448867	1	0,233333333	0
0,281073045	0	0,069048717	0,143490757	0,082994223	1	0,233333333	0
0,242041036	0	0,059645886	0,107937409	0,082624284	1	0,233333333	0
0,22895761	0	0,065039208	0,107085832	0,082408198	1	0,233333333	0
0,216964858	0	0,049675336	0,077564489	0,082103949	1	0,233333333	0
0,209551062	0	0,050349501	0,05925558	0,081728824	1	0,233333333	0
0,203663165	0	0,050668843	0,025298939	0,081377901	1	0,233333333	0

4.4 ARTIFICIAL NEURAL NETWORKS DESIGNING

At this stage, the designer of the network decides to the number of hidden layers, neurons in each layer, transfer, and performance Function.

Levenberg-Marquardt backpropagation is used as a training Function instead of using scaled conjugate gradient Function or Bayesian regularization. Levenberg–Marquardt backpropagation (trainlm) algorithm reduced in each iteration of the algorithm and locates the minimum of a multivariate Function; these features make the function fastest. It works with loss Functions which take the form of squared errors.

4.5 TRAINING AND TESTING THE ANN

Matlab is programming is used for calculations and computer simulations developed by MathWorks. Matlab allows to matrix manipulations, plotting of Functions and implementation of algorithms. Also, Matlab has an optional toolbox such as Fuzzy Logic Designer, Curve Fitting, Signal Analysis and Neural Net Time Series etc. These applications allow the user to calculate or design some networks easily.

In this thesis, an application called as Neural Net Time Series is used. The input table divided into two parts in the ratio of 1 to 4 for obviating the neural networks memorize.

For creating a nonlinear autoregressive network with external input, the input and feedback delays and hidden layer number are needed to decide. Also, the data division ratio for training, validation, and testing are needed. A representative network variable "*net* " contains thorough information on the architecture of the created neural network. The values of basic network parameters can be obtained by inserting a named variable (e.g. net) directly in the command line.

ERROR MEASUREMENTS:

In this thesis, forecasting results of models are compared mean squared error (MSE) performance criterions which can be calculated as below:

$$MSE = \frac{1}{M} \sum_{i=1}^{m} (h(x_i) - y_i)^2 \tag{14}$$

$h(x_i)$ = output of the neural network,

x_i = the input value of the network,

The performance of the remaining part is more important than the performance of the selected part.

In the case of comparing errors with different methods ; sMAD (symmetric mean absolute deviation) ,sMAPE(symmetric mean absolute percentage error) and RMSE(root mean square error) are also used. sMAD,sMAPE and RMSE are calculated as follows:

$$sMAD = 100 * \frac{\frac{1}{n} * \sum_{i=1}^{n} |Ft - At\,|}{\frac{1}{n} * \sum_{i=1}^{n} |At|} \tag{15}$$

$$sMAPE = \frac{\left(\sum_{j=1}^{j=n} (|Ft - At|)\right)}{(|At| + |Ft|)/2} * \%100/n\,; \tag{16}$$

$$RMSE = \sqrt{\frac{1}{n} * \sum_{j=1}^{j=n} (|Ft - At|)} \tag{17}$$

Ft = the forecast value;

At = the actual value;

n = number of data;

If MAPE is small, the better is the model and the predictions are a good set of predictions.

MEASURE	MEANING	EQUATION	APPLICATION TO CHAPTER EXAMPLE
Mean absolute deviation (MAD)	How much the forecast missed the target	$\text{MAD} = \frac{\sum \lvert \text{Actual} - \text{Forecast} \rvert}{n}$ (4-5)	For $\alpha = .10$ in Example 4, the forecast for grain unloaded was off by an average of 10.31 tons.
Mean squared error (MSE)	The square of how much the forecast missed the target	$\text{MSE} = \frac{\sum (\text{Forecast errors})^2}{n}$ (4-6)	For $\alpha = .10$ in Example 5, the square of the forecast error was 190.8. This number does not have a physical meaning but is useful when compared to the MSE of another forecast.
Mean absolute percent error (MAPE)	The average percent error	$\text{MAPE} = \frac{\sum_{i=1}^{n} 100 \lvert \text{Actual}_i - \text{Forecast}_i \rvert / \text{Actual}_i}{n}$ (4-7)	For $\alpha = .10$ in Example 6, the forecast is off by 5.59% on average. As in Examples 4 and 5, some forecasts were too high, and some were low.

Figure 4.27 Comparision of Measures of Forecast Error

The neural network uses10 hidden layers and one output layer for the channel estimations with the Matlab programming. In Furthermore, the input and feedback delays are 1 to 10 and 1 to 2. The regression plot and the performance of the network, MSE, are influenced by these features. "Trainlm" function is used for the prediction and the result of MSE gets 0,0091. However, the performance of the remaining part is 0,030 which is highly more than acceptable error values.

The features are changed for the better performance, reduced error, and in parallel with these changes the regression plots as follows:

The final network model for training is shown in the Figure 4.28 :

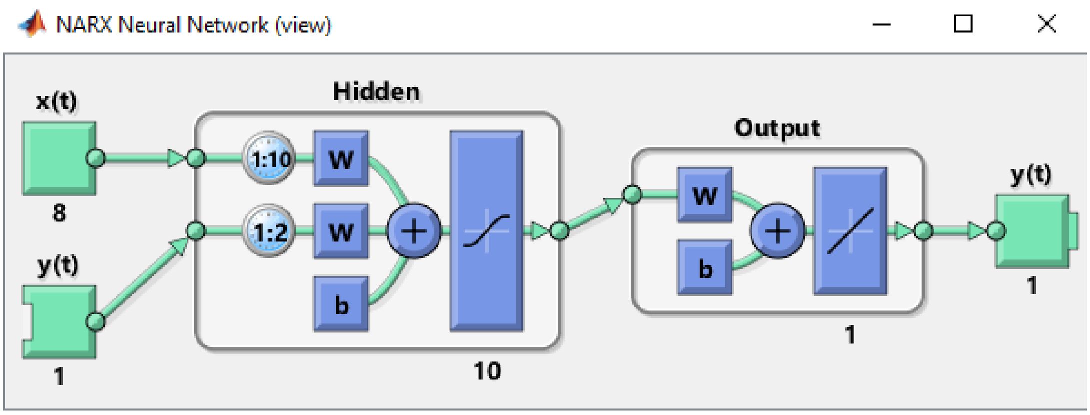

Figure 4.28 NARX Neural Network Model

In the figure 4.29 which named as Regression for remaining data, the data line is not close to 45 degrees and the network outputs are not equal to the targets. R values represent the correlation coefficient measure the correlation between outputs and targets. The r-value of 1 and 0 means a close, random relationship respectively.

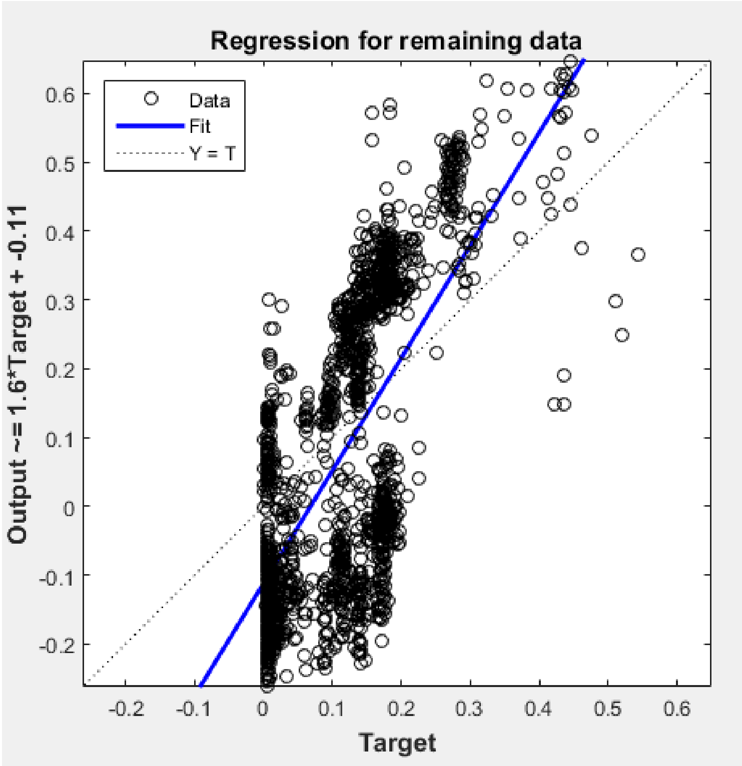

Figure 4.29 Regression Line of The Remaining Data of Trainlm Function with 10 Hidden Layer Size

The following regression plots in the figure 4.30 display the network outputs with respect to targets for training, validation, and test sets. For greater accuracy, the data line should nearly be a 45-degree line, where the network outputs are equal or close to the targets.

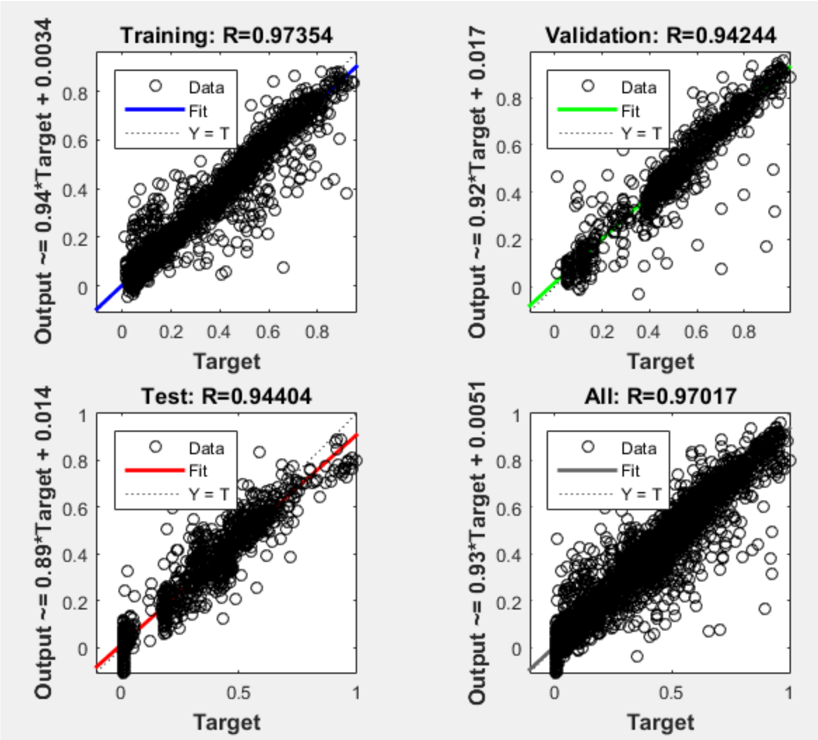

Figure 4.30 Regression Plots Of ANN During Training, Testing And Validation Of Trainlm Function With 10 Hidden Layer Size

For this plot, R values are measured as 0.97, 0.94, 0.94 and 0.97 for the training, validation, testing and whole dataset. These values are very close to the actual values, but the R-value of the remaining data set is not successful.

Secondly, the hidden layer size is changed from 10 to 15 for showing its influence. The training Function, input and feedback delays are remains. The performance results are getting as 0.46 and 1,50 respectively. The error values are decreased, however the remainin part performance is still too high for tolerance.

The structure of the model with 15 hidden layer size is given figure 4.31:

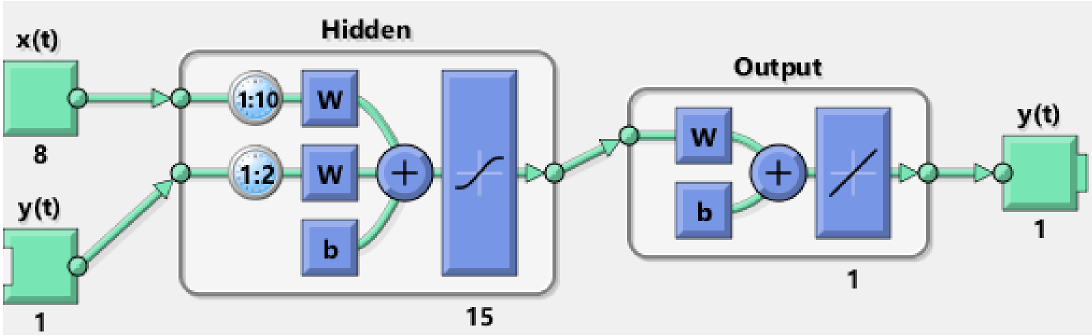

Figure 4.31 Structure Of Back Propagation Neural Network In The Case Under Study

While the error value is reduced, the regression plots of remaining data in Figure 3.31 shows that the predicted outputs are not close to actual output data. The Figure 4.32 and is given below:

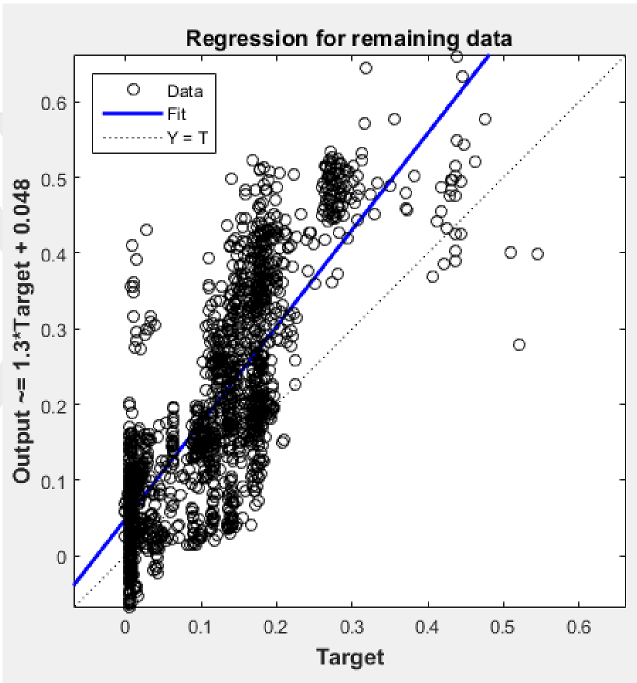

Figure 4.32 Regression Line of The Remaining Data of Trainlm Function with 15 Hidden Layer Size

In the Figure 4.33, The Regression plots of Training, Validation, Testing and All data set are shown. R values of 0.96, 0.93, 0.92 and 0.96 are obtained for the training, validation, testing and whole dataset. The values are pretty close to 1 as it should be, on the other hand the performance can be improved by changing network parameters.

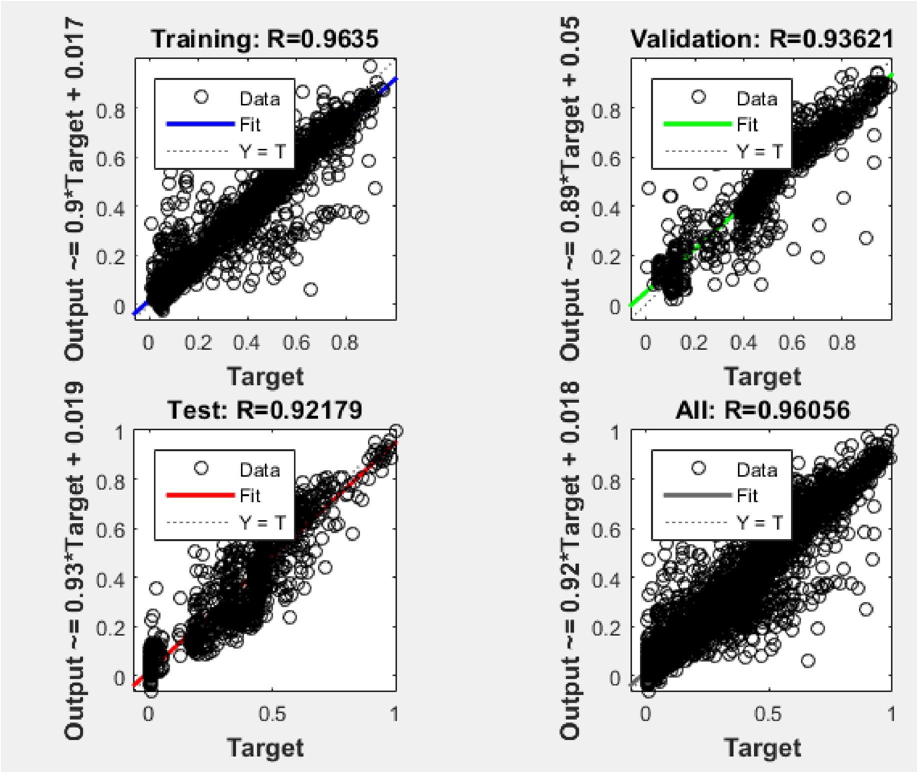

Figure 4.33 Regression Plot For Training, Testing And Validation Of All Products ANN Model Of Trainlm Function With 15 Hidden Layer Size

As mentioned above, using of the bayesian regularization Function can be slower but gives more reduced error in training. The network was trained once more while using bayesian regularization with 10 hidden layer size. The performance of the whole data set is measured as 0,0029 and the performance of the remaining part is measured as 0,00295.

The Figure 4.34 is named as “the plot of the regression for the remaining part” shows that the error value can not be acceptable for this prediction.

Performance of the remaining input is nearly equal to the performance of trainlm with 10 hidden layers. R values of 0.98, 0.90, 0.97 are obtained for the training, testing and whole dataset.

The figures of the regression plots are Figure 4.34 and Figure 3.35 are given below:

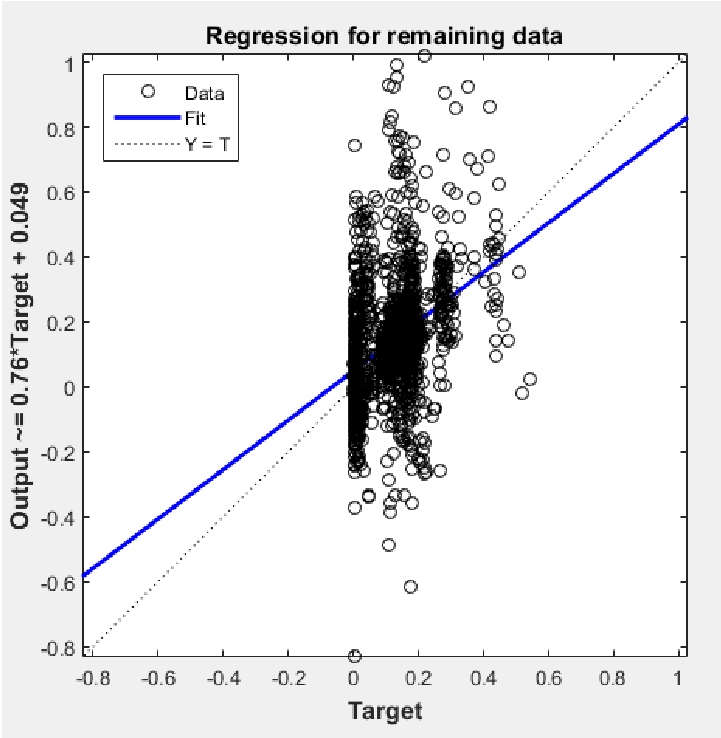

Figure 4.34 Regression Line of The Remaining Data of Trainlm Function with 10 Hidden Layer Size

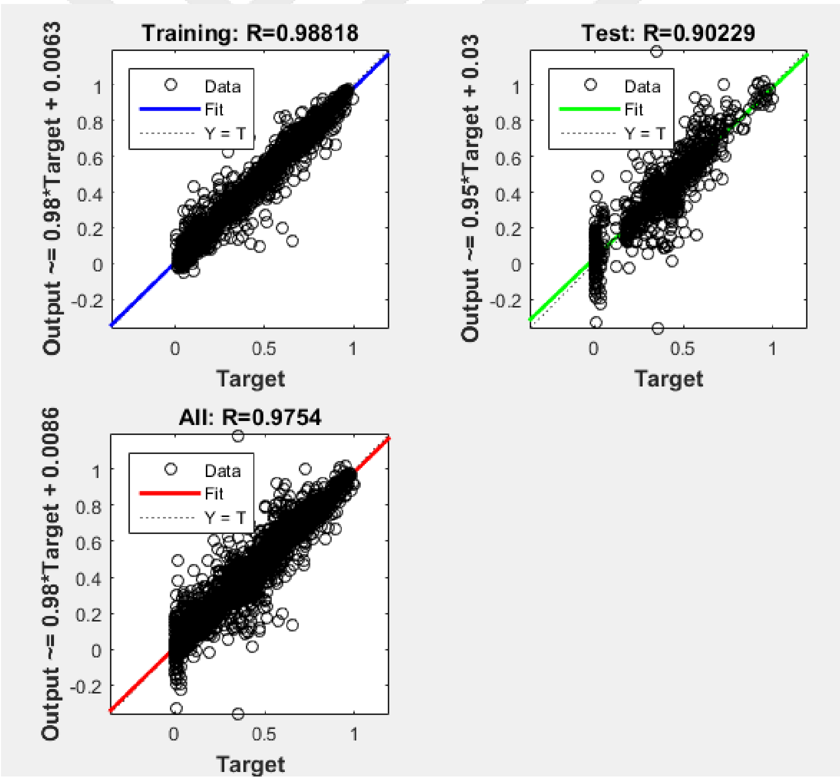

Figure 4.35 Regression Plot For Training, Testing And Validation Of All Products ANN Model Of Trainlm Function With 10 Hidden Layer Size

However the performances are unacceptable for training.Therefore, getting less error rate, the network was trained with 15 hidden layer size, the results are given below:

- trainFcn = 'trainbr';
- inputDelays = 1:10;
- feedbackDelays = 1:2;
- hiddenLayerSize =15;
- performance = 0.0024;
- performance of remaining = 0,0348;

The predicted outputs are shown in the figure 4.36 which is given below:

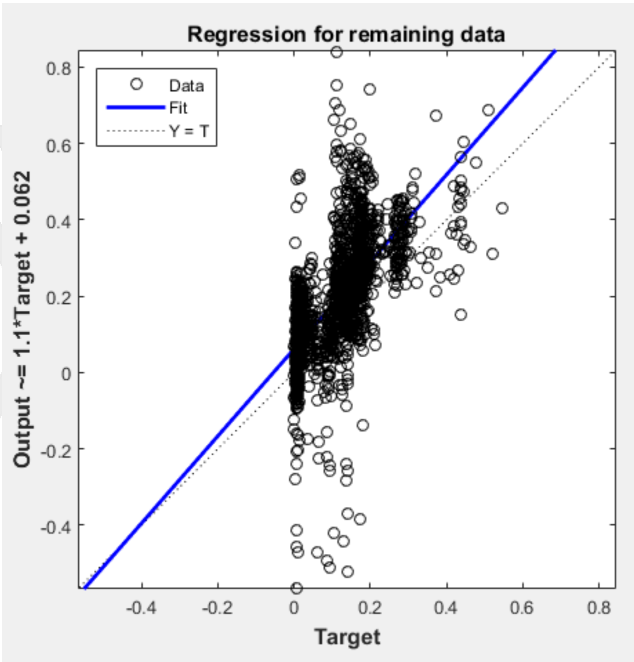

Figure 4.36 Regression Line Of The Remaining Data Of Trainbr Function With 15 Hidden Layer Size

In the figure 4.37, R values of 0.98, 0.91 and 0.97 are obtained for the training, testing and whole dataset. These values are very close to 1 which shows the outputs are similar to actual output. However, the regression plot's failure shows that the network memories the whole data set and not successful with the randomly selected data.

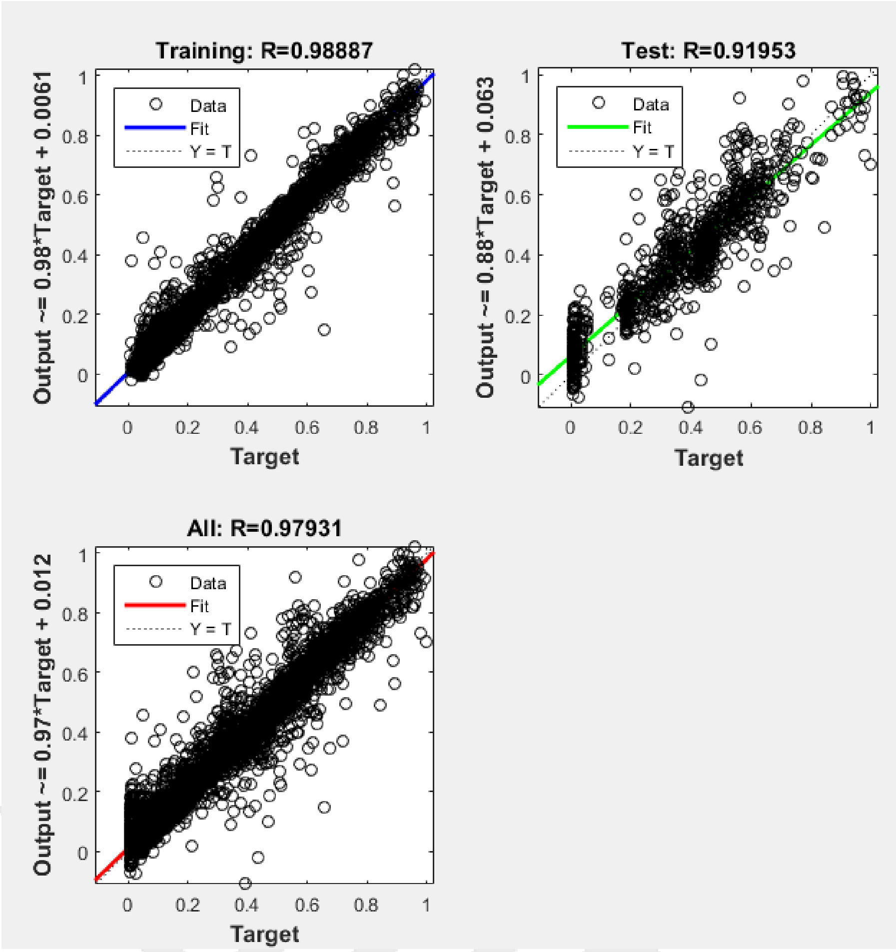

Figure 4.37 Regression Plot For Training, Testing And Validation Of All Products ANN Model Of Trainlm Function With 15 Hidden Layer Size

After training our network with changing Functions and hidden layer sizes, results of the analysis is not convincing, because of the regression plot of remaining data is not matched with the predicted values. For reducing the errors of the network, the input data numbers are increased and trained. January 2017 to July 2017 load demand, student number etc recorded, collected and normalized before added to the table of 2016.

The first plot is the Figure 4.38 regression plot of remaining data and it belongs to the training network with hidden layer size 10 and Levenberg-Marquardt Function. The regression plots are more accurate than 2016 input data. The error or remaining data is 0,0025 which it was before 0,0030.

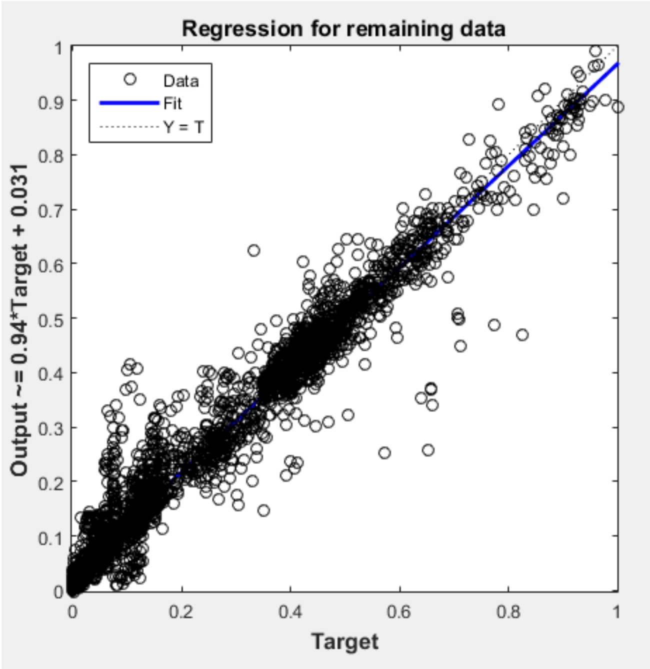

Figure 4.38 Regression Line of The Remaining Data of Trainlm Function with 10 Hidden Layer Size

The regression plot of the whole data set is shown below in the Figure 4.39 .It has a 0,0024 error rate and the data line degree seems to 45 degrees and the distribution of predicted output datas are on the line. It shows that the network is successful with randomly selected data.

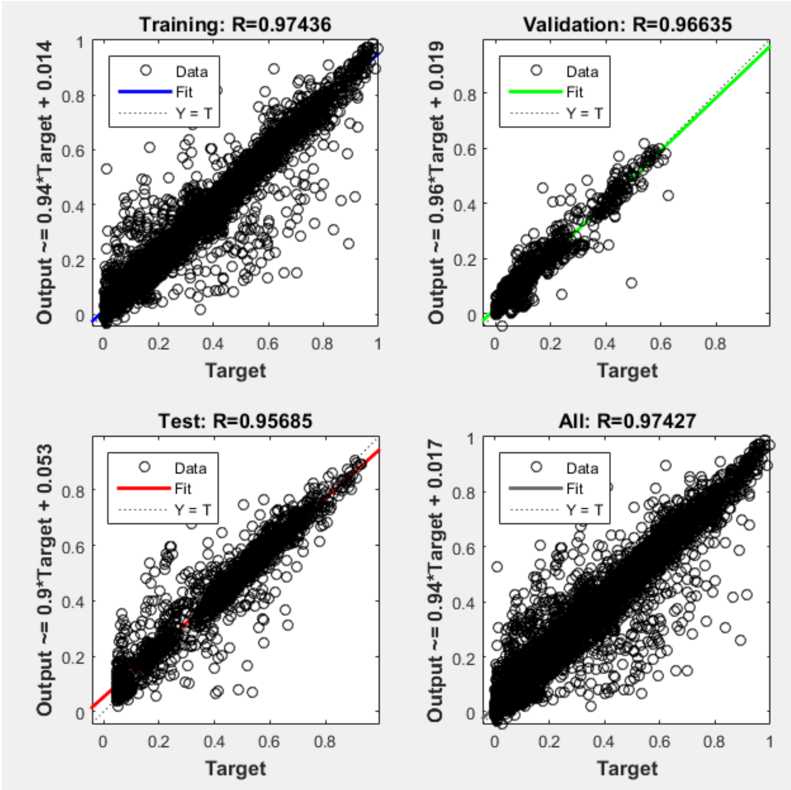

Figure 4.39 Regression plot for training, testing, and validation of ANN model with Trainlm Function and10 Hidden Layer Size

After changing the network parameters for observation of the performance of the network, the network structure is changed as open loop and closed loop respectively.

Firstly, open-loop structure is applied to the network by using “trainlm” functiın with 15 hidden layer size. The network model is given below in the Figure 4.40:

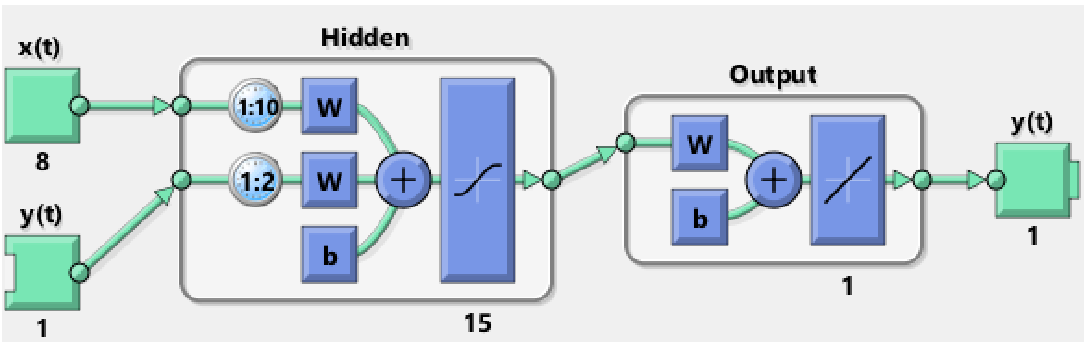

Figure 4.40 Open-Loop Architecture Network Model

Figures 4.41 and Figure 4.42 show that the regression of NARX plot with training, validation, testing and output data plots. R-value represents the Regression value which indicates the relationship between output and target values. If regression value is close to 0 means that the outputs are not similar to targets, however, if R is close to 1 there is a linear relationship between outputs and inputs. In this regression plot, the R-value of output is 0,97214 which is quite close to 1 indicates a good fit.

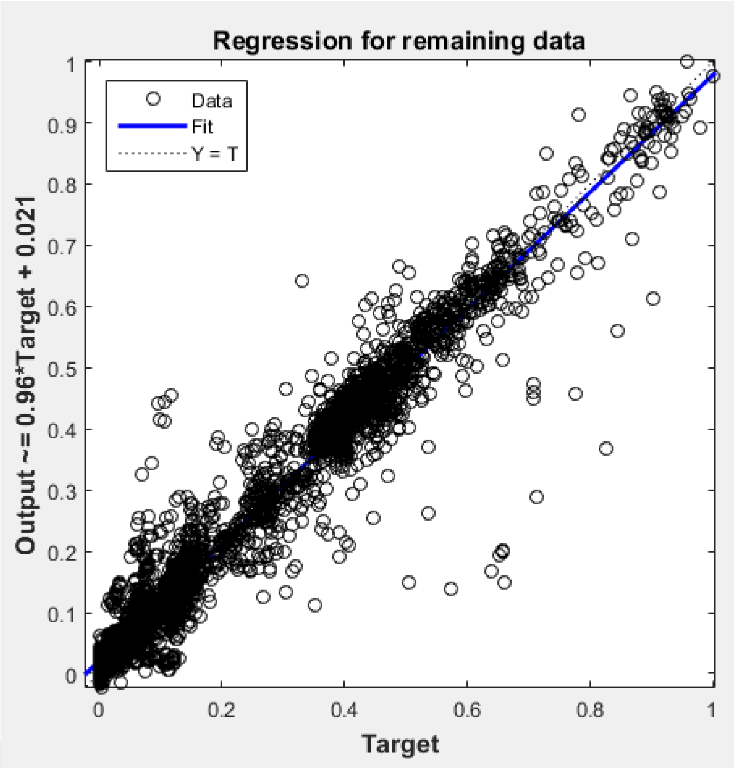

Figure 4.41 Regression of NARX plot of Remaining Data (plotregression) of Trainlm Function with 15 Hidden Layer Size

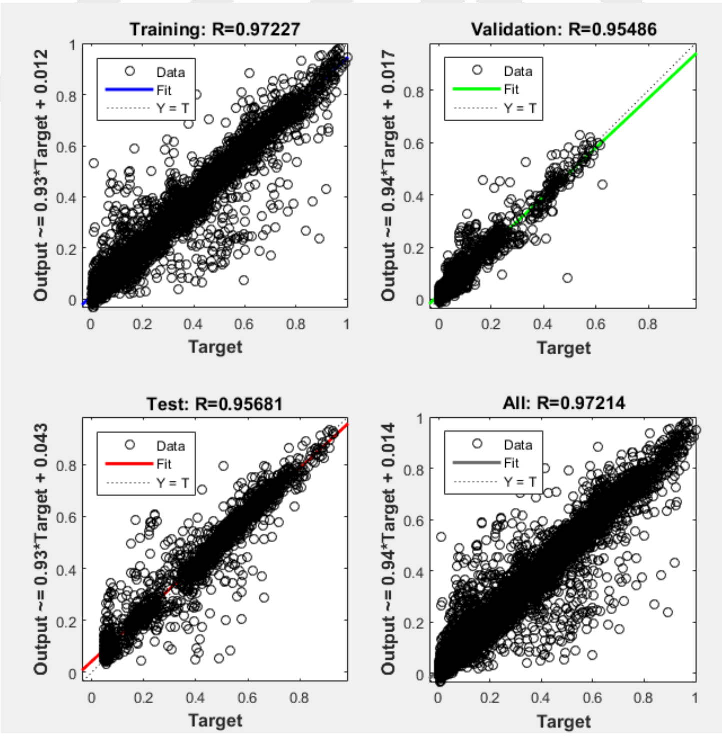

Figure 4.42 Regression of NARX plot (plotregression) of Trainlm Function with 15 Hidden Layer Size

- trainFcn = 'trainbr';
- inputDelays = 1:10;
- feedbackDelays = 1:2;
- hiddenLayerSize =10;
- performance = 0.0019;
- performance of remaining = 0,0026;
- Timing: 10:27

By using Bayesian regulation training Function and 10 hidden layers, the result of MSE of remaining data is 0,0026 which close to the resulting of network training by 10 hidden layer size and Levenberg-Marquardt optimization. However, the MSE result of this network is measured as 0,0019 highly better than Levenberg-Marquardt optimization.

The regression lines of remaining data and the all data set are figure 4.43 and the figure 4.44 are given below:

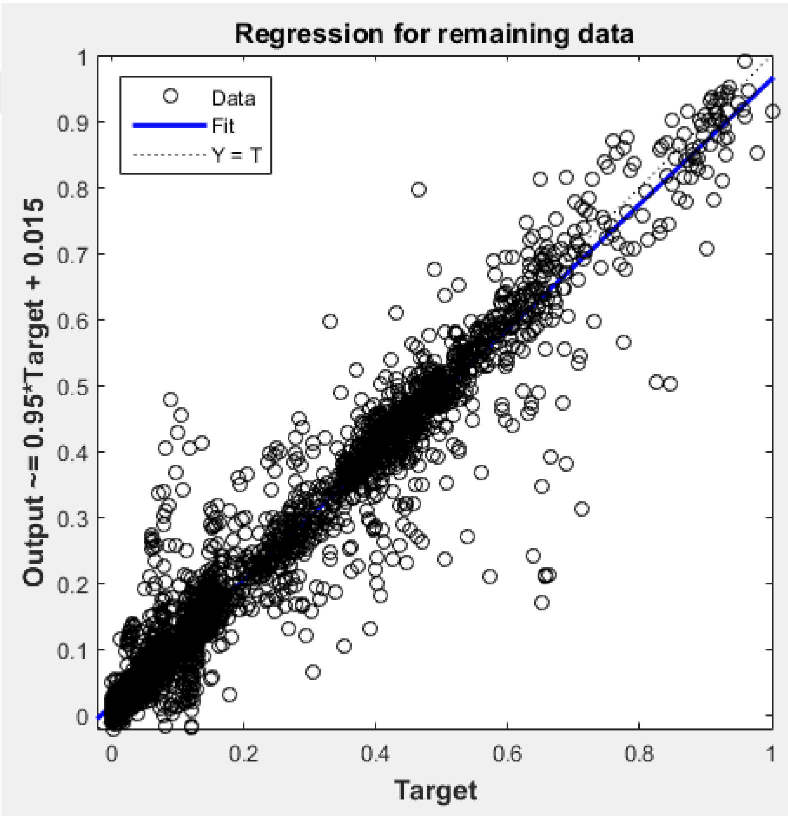

Figure 4.43 The Regression Line Of The Remaining Data of Trainbr Function with 10 Hidden Layer Size

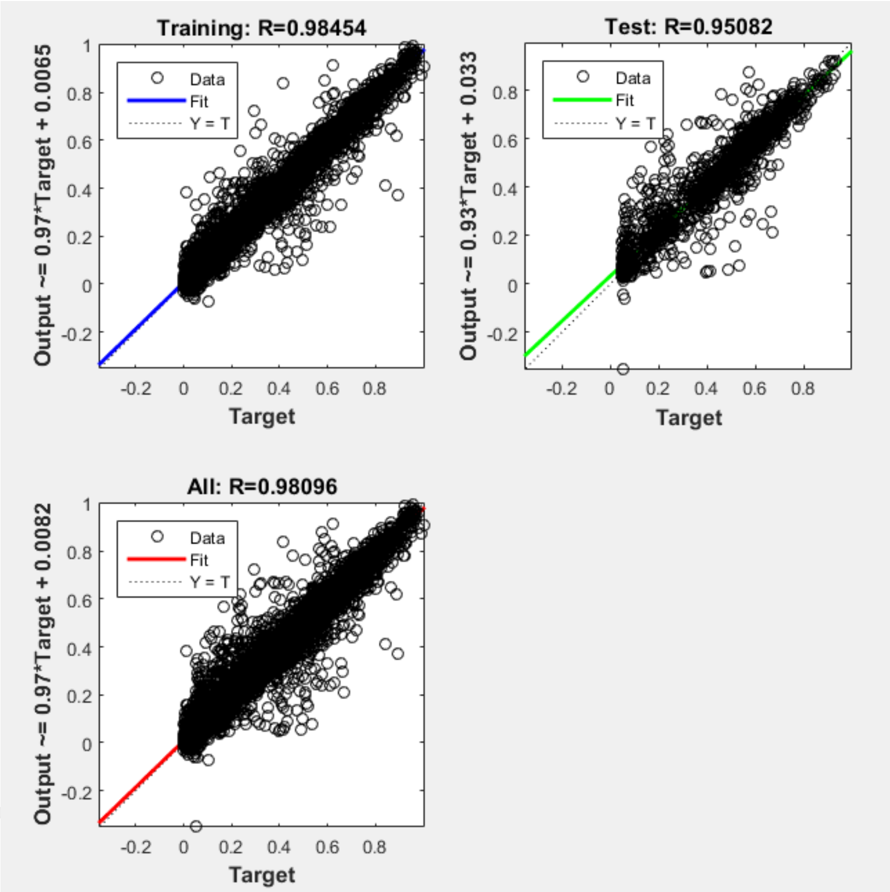

Figure 4.44 The Regression Lines Of Training, Testing And Output Data Of Trainbr Function With 10 Hidden Layer Size

- trainFcn = 'trainbr';
- inputDelays = 1:10;
- feedbackDelays = 1:2;
- hiddenLayerSize =15;
- performance = 0.0021;
- performance of remaining = 0,0026;
- Timing: 26:06

By using Bayesian regulation training Function and 15 hidden layers, the result of MSE of remaining data is 0,0026 which equal to the resulting of network training by 10 hidden layer size and Levenberg-Marquardt optimization. However, the MSE result of this network is measured as 0,0021 highly better than Levenberg-Marquardt optimization which the performance is 0,0027 but the training timing is measured as 26 minutes which is higher than the trainlm training Function.

The regression lines of remaining data and the all data set are figure 4.45 and the figure 4.46 are given below:

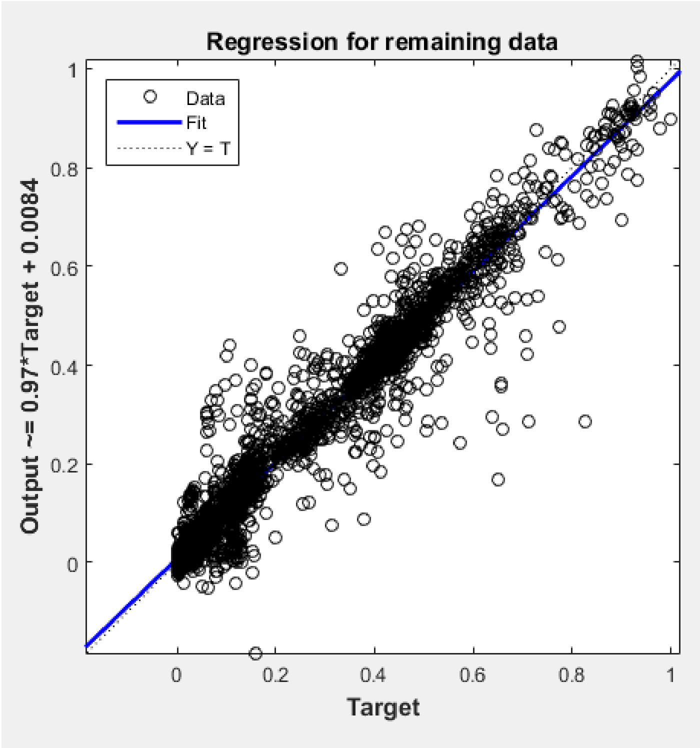

Figure 4.45 The Regression Line Of The Remaining Data of Trainbr Function with 15 Hidden Layer Size

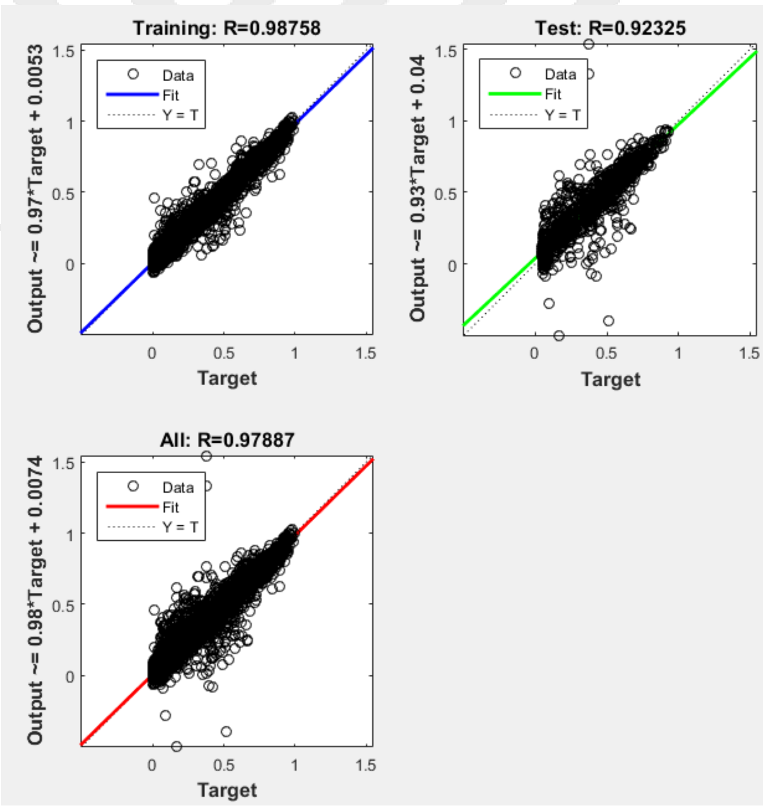

Figure 4.46 The Regression Lines Of Training, Testing And Output Data Of Trainbr Function With 15 Hidden Layer Size

- trainFcn = 'trainlm';
- inputDelays = 1:10;
- feedbackDelays = 1:2;
- hiddenLayerSize =10;
- performance = 0.0041;
- performance of remaining = 0,0058;
- Timing: 26:06

Closed-loop(series architecture) network model Figure 4.47 is given below:

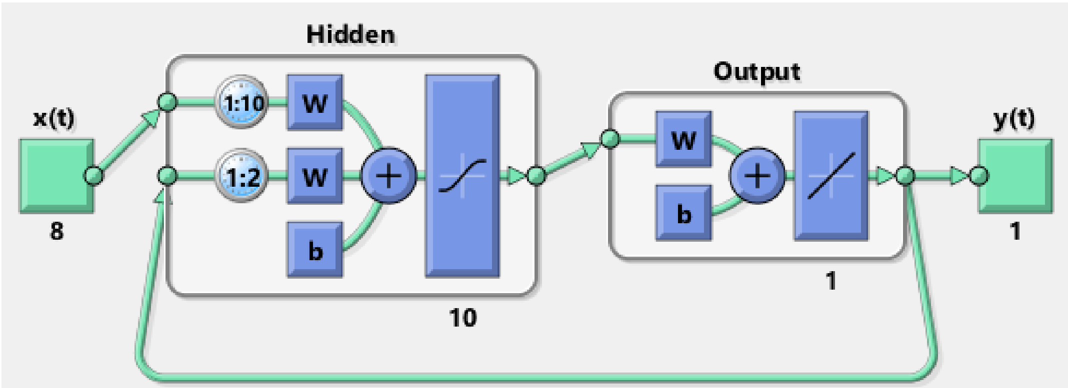

Figure 4.47 Closed-Loop(Series Architecture) Network Model

The Figure 4.47 shows the difference between closed-loop and open-loop architecture types. In the closed-loop network model, the feedback comes from the output is used with the input value.

The plot of the closed loop regression is obtained using 10 neurons in the hidden layer and the Levenberg-Marquardt algorithm. The performance of the closed loop is measured as 0,0044 which is quite a bit more than the open loop performance and the Regression is 0,95642. It is seen that the open loop regression value is closer to 1 than the closed loop regression value. :

The regression lines of remaining data and the all data set in the figure 4.48 and the figure 4.49 are given below:

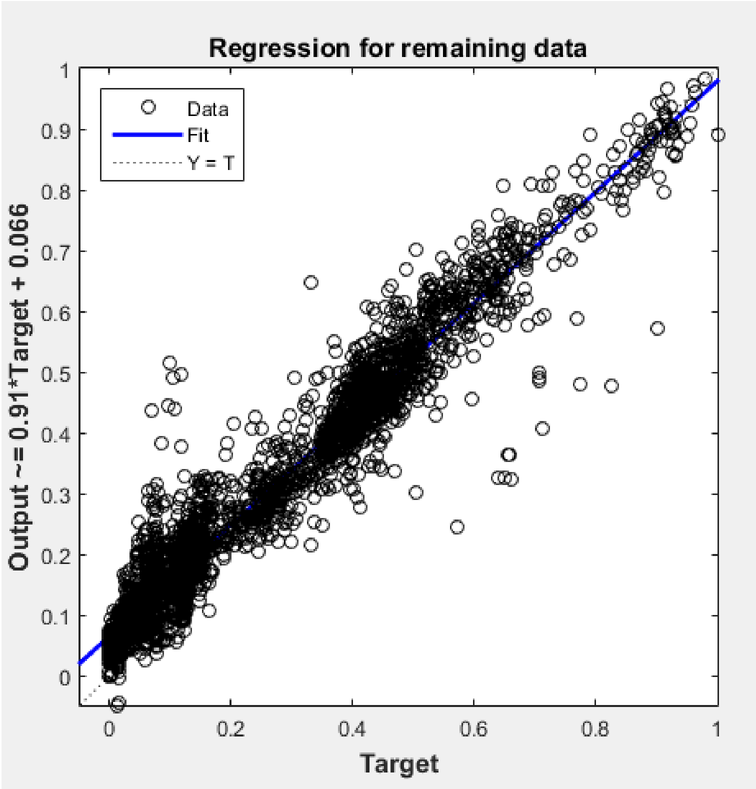

Figure 4.48 The Regression Line Of The Remaining Data Closed Loop Architecture

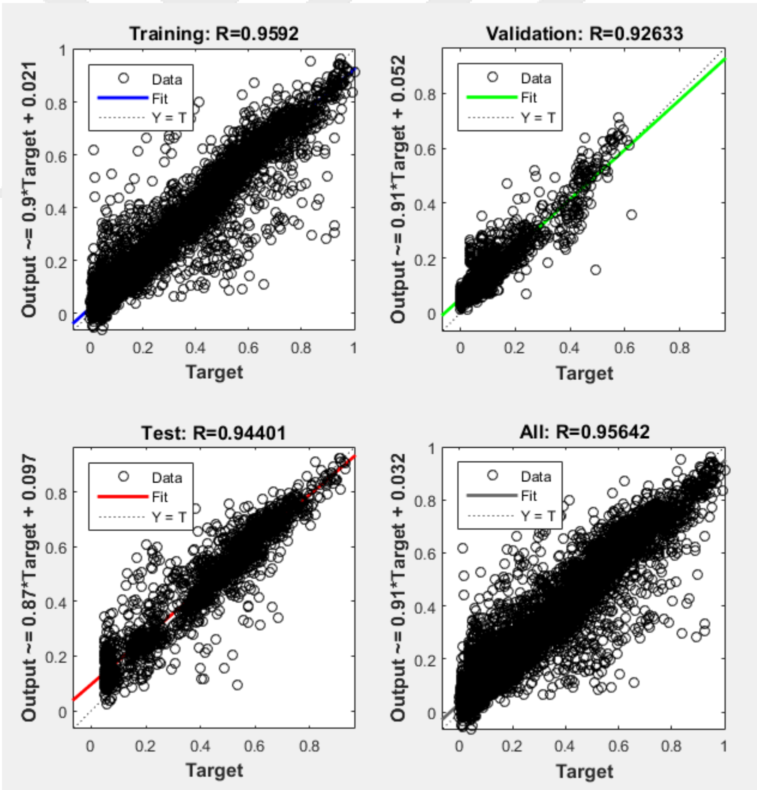

Figure 4.49 The Regression Line Of The Closed Loop Architecture

The final result of the experements by changing parameters are tabled as below:

Table 4.4 The Performance Results Of 2016 Data Inputs

features	1	2	3	4
Function	trainlm	trainlm	trainbr	trainbr
iputdelays	01:10	01:10	01:10	01:10
feedbackdelays	01:02	01:02	01:02	01:02
hiddenlayersize	10	15	10	15
performance	0,0091	0,0046	0,0029	0,0024
performance_remaining	0,030	0,0150	0,0025	0,00348

The results are measured for closed loop model multistep prediction surveyed after adding data recorded in 2017 summarized in the Table 4.5 given below:

Table 4.5 The Performance Results Of 2016 and 2017 Data Inputs

Denklem 1

features	1	2	3	4
Function	trainlm	trainlm	trainbr	trainbr
inputdelays	01:10	01:10	01:10	01:10
feedbackdelays	01:02	01:02	01:02	01:02
hiddenlayersize	10	15	10	15
performance	0,0024	0,0026	0,0019	0,0021
performance_remaining	0,0024	0,0026	0,0026	0,0026
RSME	0,0458	0,0648	0,1453	0,00463
SMAPE	34,78%	44,62%	47,04%	38,29%
SMAD	25,75%	37,36%	51,50%	25,98%

The increasing number of input data also increasing the performance by reducing the error parameter MSE as can be seen from above performance tables 4.4 and the table 4.5 . Using Bayesian Function instead of Levenberg–Marquardt backpropagation takes much more time but gives more accurate results according to parameter MSE. The best result gets while using Levenberg–Marquardt backpropagation, 10 hidden layer size and input delays while using closed-loop model for multi-step prediction.

After training the network, the relationship between predicted output and actual output is given at Figure 4.50:

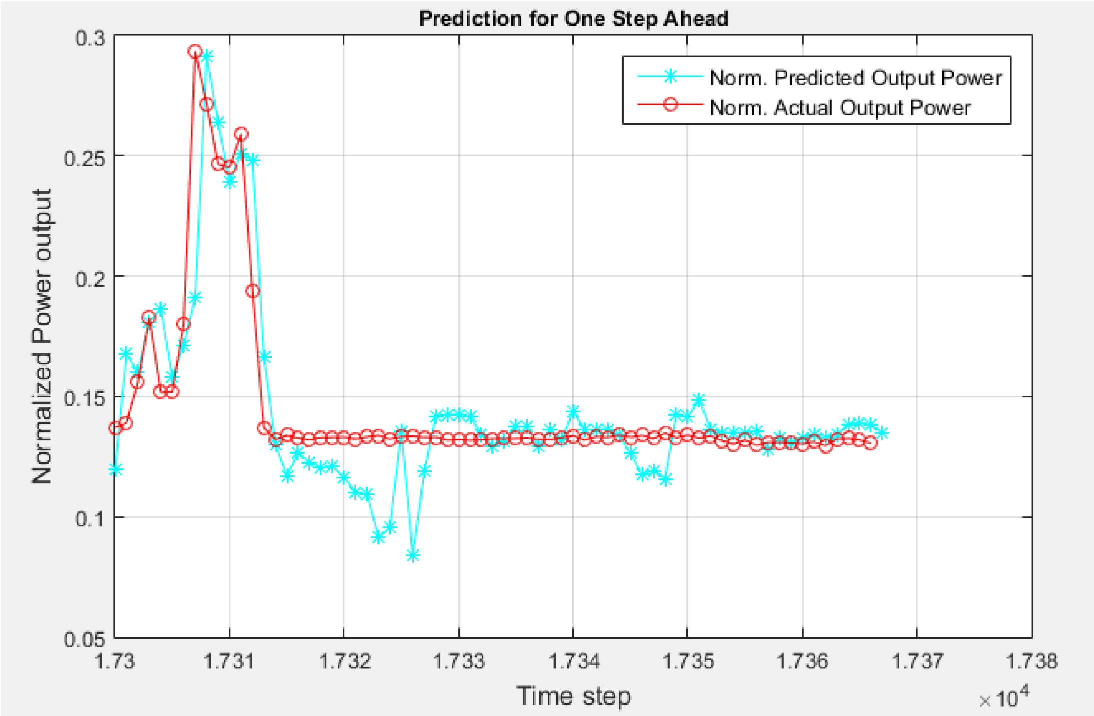

Figure 4.50 Comparison Between Normalized Actual And Normalized Forecasted Load

The Figure 4.50 belongs to network with the training parameters such as 10 hidden layer, open-loop structure for the one-ahead-prediction. MSE of 0,0024 while using “trainlm” Function and the regression value R is 97.38. Figure 4.50 shows that the predicted output power is a quite close to actual output power.

The Figure 4.51 shows that the de-normalized values comprasion of predicted and actual output datas:

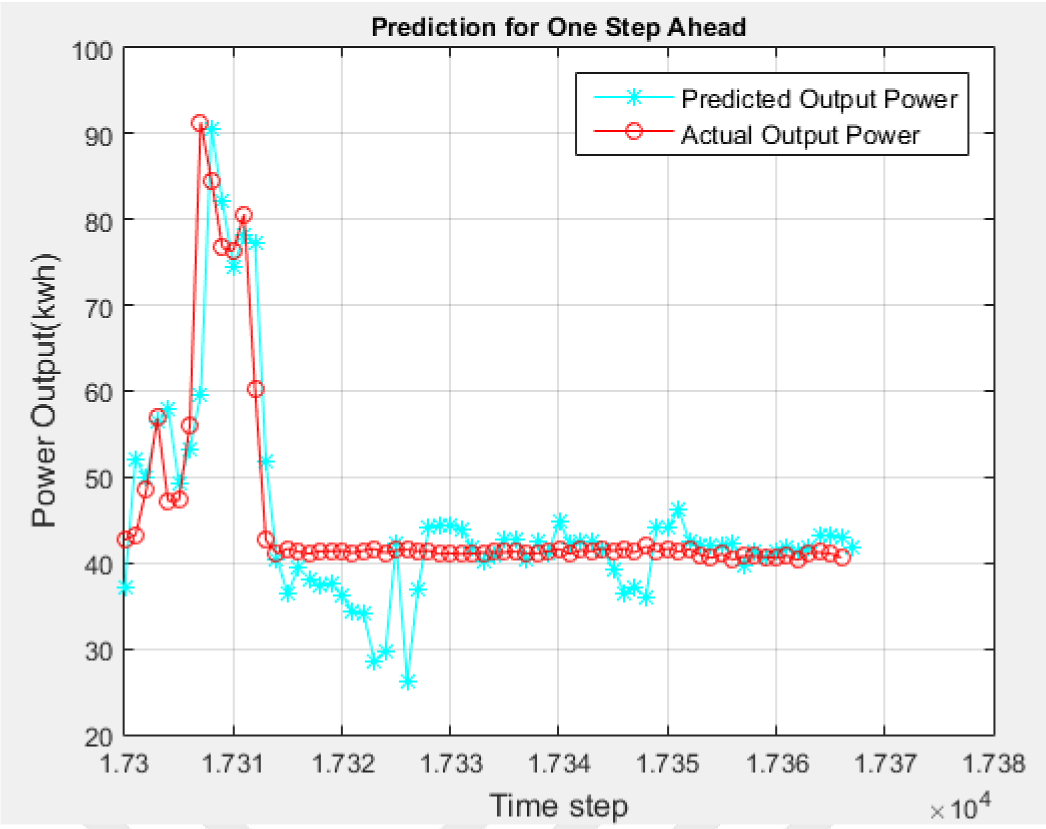

Figure 4.51 De-Normalized Power Output vs Actual Output

Closed-loop prediction method is also used for multi-step predictions in NARX. Figure 4.52 shows the relationship between the multi-step predicted values and the actual output values. The parameters are not changed for this prediction, 'trainlm' function is used and the delays are choosen as 2.

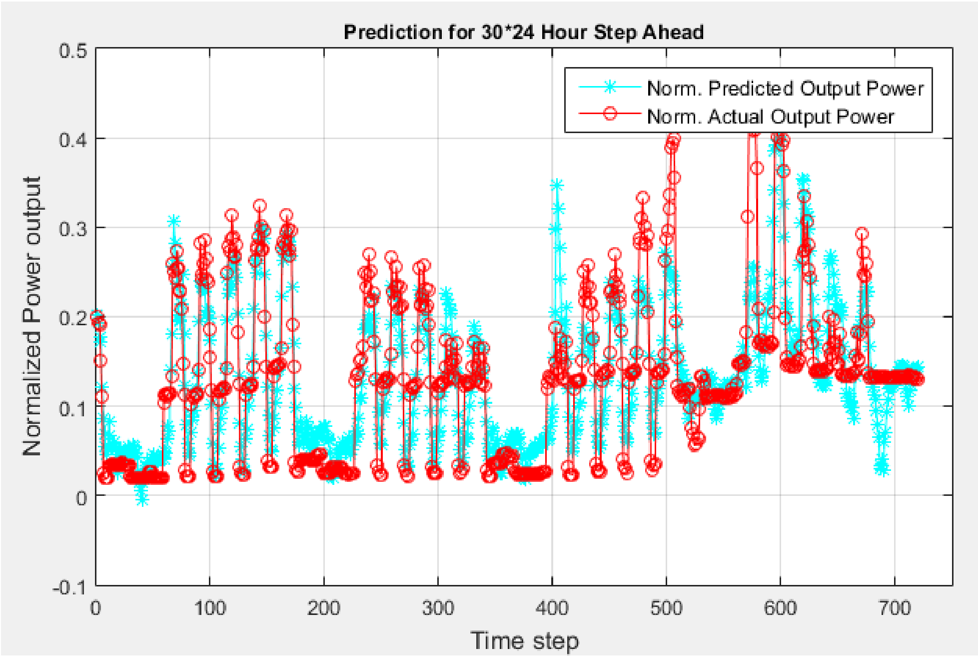

Figure 4.52 Multi-Step Prediction for 720 Hours

Re-normalization process is made for predicted and actual power outputs to understanding the real relationships. Figure 4.53 shows the re-normalized predicted output power versus actualoutput power graphs below:

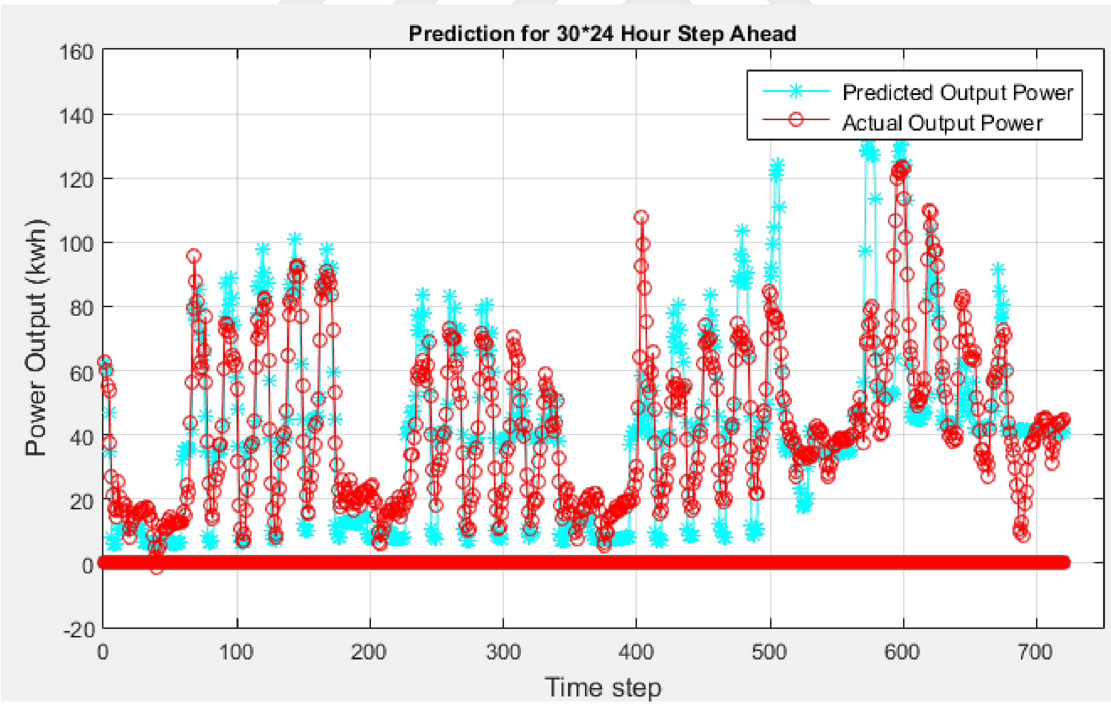

Figure 4.53 De-Normalized Power Output vs Actual Output for Multi-Step Prediction

5. CONCLUSION

There are many ways to be a successful organization; however, without having a well-done electricity consumption management program, the company cannot achieve its objectives. The company should learn the behavior of the power system. By learning behavior, the budget allocation for consumption of electricity can be arranged. The artificial neural network system can provide to learn behavior and features of the system.

In this thesis, the artificial neural network system is developed which aim to forecast Yasar University's short-term electricity demand. The hourly mechanical energy consumption of Yasar University Y-block which recorded at 2016 and 2017 and used for forecasting. After collecting energy consumption data, student number, weather temperatures, and holiday dates are also searched and aggregated.

Load forecasting can be classified into three types by having regard to forecasting time periods. The first one is short-term forecasting which is started with few hours to one week. The second one is the medium term forecasting is the type of forecasting which predicts load from few weeks up to few months. The last one is long-term load forecasting which predicts the load longer than a year. In this thesis, short-term load forecasting type is used for the prediction.

In the literature, there are many methods are invented for load forecasting. The most used ones are simple linear and multiple regression method, time series, expert systems, and fuzzy logic however artificial neural network method has more accurate results than the other methods discussed in the research.

The artificial neural network is a mathematical model takes as an example of human brain processes information over neurons. The architecture types of the Artificial Neural Network are explained. The first one is Feed forward Neural Network and it can be classified by their layer types. Single layer neural networks have one input layer and one output layer and they are connected only in one direction(no feedback) and the Multi-Layer Feed forward Neural Network which have one input layer, one output layer and more than one hidden layers of processing units without feedback connections.

In this thesis, an ANN based model is used to predict hourly load consumption forecasting by using the nonlinear autogregressive network with exogenous input (NARX) which is a recurrent dynamic network with the feedback connections.The exogeneous inputs are the temperature, student number and the day type.

The importance of the temperature can be seen clearly in the Table 2.1 TheAverage Load Consumption and Air Temperature of 2016 in Izmir. The table 2.1 shows that the highest average temperatures in Izmir are recorded at July and August with 28 ℃ and 30 ℃ respectively, which also have the highest recorded average load consumption with 88,51 kWh and 110,85 kWh respectively. The usage of air conditioners considerable affect the load consumption. The Table 2.2 TheAverage Load Consumption and Air Temperature of 2017 in Izmir premediates this allegation.

The day type is also an important factor for the network. The figure 2.1 The Daily Load Consumption Of September 2016 includes to The Sacrifice Holiday dates. The load consumption is almost equal to zero in a similar vein of the Figure 2.2 which shows to holiday of Ramadan Bairam.

The first steps of the construct of the network are recording and collecting the input datas and normalization process by using the Min-max normalization. Matlab programming Narx network is used for the construction. The artificial neural network has many features but one of the most powerful features is their ability to *learn* and generalize from a set of training data. Therefore, avoiding to memorize the results, all data set is divided into two parts at the rate of 80% and 20% respectively; selected part and remaining part. As can seen in the Table 3.4, the best result of selected data set is getting by using “trainbr” Functions and 15 hidden layer size with MSE of 0,0024.
Also, the best result of remaining part is getting by using “trainlm” Functions and 15 hidden layer size with MSE of 0,015 However, the MSE is more over than 1% and it cannot be acceptable for the network type.

Consequently, the number of the inputs are increased by adding 2017 data samples to the input data set. In the table 3.5 indicates that the lowest error result of the selected part is improved to 0,0029 to 0,0019 and the error rate of the remaining part is 0,0025.

The results are show that the number of sampling number affects the performance of the network. Also, using Bayesian regularization Function is improved the accuarcy of the network nevertheless network training timing is more over than Levenberg-Marquardt Function training. It can be a disadvantages for the networks with the multiecheon inputs.

6. FUTURE WORK

After training our network model with 2016 load consumption data as input, the network cannot be successful and adding 2017 data to input the error is reduced. For more accurate results, the input data numbers should be increased.

The air temperature was used in this research, for a future search, some other important weather parameters should be added like humidity, airspeed, the sunshine duration can be used.

Also, general enrolled in school student number taken as student number in this forecast. By measuring the real student number in the university Y block campus at that specific hour, the dependability of the network will be increased.

7. REFERENCES

Abdi. H,(2010). Normalizing Data. In Neil Salkind (Ed.), Encyclopedia of Research Design. Thousand Oaks, CA: Sage. 2010

Adhikari, R., & Agrawal, R. K. (2013). An introductory study on time series modeling and forecasting. arXiv preprint arXiv:1302.6613.

Amani, P., Kihl, M., & Robertsson, A. (2011, November). NARX-based multi-step ahead response time prediction for database servers. In ISDA (pp. 813-818).

Arı, A., & BERBERLER, M. E. Yapay Sinir Ağları ile Yaklaşım ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı. Acta INFOLOGICA, 1(2), 55-73.

Artificial Intelligence - Expert Systems(2018), Retrieved from //www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_expert_systems.htm.

Boussaada, Z., Curea, O., Remaci, A., Camblong, H., & Mrabet Bellaaj, N. (2018). A Nonlinear Autoregressive Exogenous (NARX) Neural Network Model for the Prediction of the Daily Direct Solar Radiation. Energies, 11(3), 620.

Cavallaro, F. (2005). Electric load analysis using an artificial neural network. International Journal of Energy Research, 29(5), 377-392.

Da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., & dos Reis Alves, S. F. (2017). Artificial neural networks(p. 22-28). Cham: Springer International Publishing.

Dilhani, M. S., & Jeenanunta, C. (2016, March). Daily electric load forecasting: Case of Thailand. In Information and Communication Technology for Embedded Systems (IC-ICTES), 2016 7th International Conference of (pp. 25-29). IEEE.

Elgarhy, S. M., Othman, M. M., Taha, A., & Hasanien, H. M. (2017, December). Short term load forecasting using ANN technique. In Power Systems Conference (MEPCON), 2017 Nineteenth International Middle East (pp. 1385-1394). IEEE.

Feinberg, E. A., & Genethliou, D. (2005). Load forecasting. In Applied mathematics for restructured electric power systems (pp. 269-285). Springer, Boston, MA.

Forecasting Fundamentals.(2018) Retrieved from http://mech.at.ua/Forecasting.pdf

Halici,U.(2008). From Biological to Artificial Neuron Model. Retrieved from http://users.metu.edu.tr/halici/courses/543LectureNotes/lecturenotes-pdf/ch1.pdf

Highley, D. D., & Hilmes, T. J. (1993). Load forecasting by ANN. IEEE Computer Applications in Power, 6(3), 10-15.

Ishik, M. Y., Göze, T., Özcan, İ., Güngör, V. Ç., & Aydın, Z. (2015, April). Short term electricity load forecasting: A case study of electric utility market in Turkey. In Smart Grid Congress and Fair (ICSG), 2015 3rd International Istanbul (pp. 1-5). IEEE.

Jacobson, L.(2014, March). Introduction to Artificial Neural Networks Part 2 – Learning. Retrieved from http://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-2-Learning/8

Khotanzad, A., Afkhami-Rohani, R., & Maratukulam, D. (1998). ANNSTLF-artificial neural network short-term load forecaster-generation three. IEEE Transactions on Power Systems, 13(4), 1413-1422.

Kiartzis, S. J., Bakirtzis, A. G., & Petridis, V. (1995). Short-term load forecasting using neural networks. Electric Power Systems Research, 33(1), 1-6.

Kumar, R., Aggarwal, R. K., & Sharma, J. D. (2013). Energy analysis of a building using artificial neural network: A review. Energy and Buildings, 65, 352-358.

Lagandula A.C.,(2018). McCulloch-Pitts Neuron—Mankind's First Mathematical Model Of A Biological Neuron. Retrieved from https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1.

Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation Function can approximate any Function. Neural networks, 6(6), 861-867.

McCormick, C. (2013). Radial Basis Function Network (RBFN) Tutorial. Retrieved from http://mccormickml.com/2013/08/15/radial-basis-Function-network-rbfn-tutorial/

Memarian, H., & Balasundram, S. K. (2012). Comparison between multi-layer perceptron and radial basis Function networks for sediment load estimation in a tropical watershed. Journal of Water Resource and Protection, 4(10), 870.

Mijwel, M. M. (2018, January), Artificial Neural Networks Advantages and Disadvantages

Moghram, I., & Rahman, S. (1989). Analysis and evaluation of five short-term load forecasting techniques. IEEE Transactions on power systems, 4(4), 1484-1491.

ÖZDEN, S., & ÖZTÜRK, A. (2018). Yapay Sinir Ağları ve Zaman Serileri Yöntemi ile Bir Endüstri Alanının (İvedik OSB) Elektrik Enerjisi İhtiyaç Tahmini. International Journal of InformaticsTechnologies, 11(3).

Papadopoulos, T. A., Giannakopoulos, G. T., Nikolaidis, V. C., Safigianni, A. S., & Panapakidis, I. P. (2016). Study of electricity load profiles in university campuses: The case study of Democritus University of Thrace.

Penya, Y. K., Borges, C. E., & Fernández, I. (2011, September). Short-term load forecasting in non-residential buildings. In AFRICON, 2011 (pp. 1-6). IEEE.

Radivojac, P.(2005). Data Transformation. In Indiana University Bloomington,

Rahman, A., Srikumar, V., & Smith, A. D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Applied Energy, 212, 372-385.

Roberts, History: The 1980's To The Present (2009), Retrieved from https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history2.html

Saatwong, P., & Suwankawin, S. (2016, June). Short-term electricity load forecasting for Building Energy Management System. In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2016 13th International Conference on (pp. 1-6). IEEE.

Sazli, M. H. (2006). A brief review of feed-forward neural networks. Communications, Faculty of Science, University of Ankara, 50(1), 11-17.

Singh, A. K., Ibraheem, S. K., Muazzam, M., & Chaturvedi, D. K. (2013). An overview of electricity demand forecasting techniques. Network and complex systems, 3(3), 38-48.

Swingler, K. Single Layer Perceptrons. Retreieved from http://www.cs.stir.ac.uk/courses/ITNP4B/lectures/kms/2-Perceptrons.pdf

Tanaka, Y. (1993, September). An overview of fuzzy logic. In WESCON/'93. Conference Record, (pp. 446-450). IEEE.

Yadav, N., Yadav, A., & Kumar, M. (2015). An introduction to neural network methods for differential equations.(p.13-15), Berlin, Germany: Springer.

Zadeh, L. A. (1996). Fuzzy logic= computing with words. IEEE transactions on fuzzy systems, 4(2), 103-111.