**BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING**

Course No. : EEE 306

Course Title: Power System I Laboratory

Project Progress

Name of the Project: FORECASTING ELECTRICAL LOAD USING MACHINE LEARNING

**Group: 4**

**Members: 1906114**

**1906115**

**1906116**

**1906117**

**1906118**

**1906119**

## Title:

**FORECASTING ELECTRICAL LOAD USING MACHINE LEARNING**

## Group No: 4

## Group members:


**1906114 - Md. Asif Hasan**

**1906115 - Md. Mahi Hossain**

**1906116 - Tanvir Rahman Tasin**

**1906117 - Joydip Chakraborty**

**1906118 - Md. Meraz Rahman**

**1906119 - Jahin Arib**

## Abstract:

In this project, we have built a predictor model to forecast the short-term electrical load of a power system  using machine learning. The machine learning model used for this task is Artificial Neural network and Regression learner from machine learning. This model can forecast the hourly load demand of a power system given historical loads, the information about weekdays and the seasonality of the data. It has been trained using hourly load data information of Bangladesh Power Grid for the month of January in 2020 . The model can produce accurate forecasts with a mean square error of is 4.71e-04 in levenberg-marquardt and 1.3257e-6 in Bayesian regularization.

## Introduction:

Load forecasting means estimating the active load of a power system at various times to meet demand-supply equilibrium and to minimize utility risks. Nowadays our daily lives are heavily dependent on a continuous supply of electricity. Changes in demand may disrupt the supply chain, which may cause load-shedding and affect the price. Therefore, it is particularly important for energy suppliers to accurately predict the required amount of energy and build their networks accordingly. Usually, energy producers and suppliers use commercial software packages for load forecasting. But as these are third-party products, they are not transparent enough to ensure complete security. Building an at-home model can solve this issue while providing us with an opportunity to learn about the practical load demand and changes in load at various times in our country.

## Design and Analysis:

The data used in this project was collected from the PGCB website (Address: http://pgcb.gov.bd/site/page/420a1727-2eab-4874-a387-a018165b6334 ). They monitor the hourly load demand and generation data of the national grid. They also track the amount of load-shed, power generated and imported by various sources.

## ঘন্টাপ্রাতি উৎপাদন এবং লোডশেড

01-01-2020

| তারিখ | সময় | উৎপাদন(মেঃওঃ) | চাহিদা(মেঃওঃ) | লোডশেড | গ্যাস | তরল জ্বালানী | কয়লা | হাইড্রো | সৌর | ভেড়ামারা এইচভিডিসি | ত্রিপুরা | মন্তব্য |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ০১-০১-২০২০ | ২৪:০০:০০ | ৬৪০৯ | ৬৪০৯ | ০ | ৫৮৭৮ | ১৭ | ১৭৫ | ৪৬ | ০ | ২১৪ | ৭৮ | |
| ০১-০১-২০২০ | ২২:০০:০০ | ৭৪৩৯ | ৭৪৩৯ | ০ | ৬১৭৬ | ৪৪৩ | ১৭৫ | ৪৬ | ০ | ৫১১ | ৯৪ | |
| ০১-০১-২০২০ | ২১:০০:০০ | ৮০০১ | ৮০০১ | ০ | ৬২৬২ | ৯১৮ | ১৭৪ | ৪৪ | ০ | ৫১১ | ৯৮ | |
| ০১-০১-২০২০ | ২০:০০:০০ | ৮৩২৮ | ৮৩২৮ | ০ | ৬৩২৮ | ১১৭৯ | ১৭৪ | ৪৪ | ০ | ৫১১ | ৯৮ | |
| ০১-০১-২০২০ | ১৯:৩০:০০ | ৮৪০৫ | ৮৪০৫ | ০ | ৬৪১০ | ১১৭২ | ১৭৪ | ৪৫ | ০ | ৫১১ | ৯৮ | |
| ০১-০১-২০২০ | ১৯:০০:০০ | ৮৪৩৬ | ৮৪৩৬ | ০ | ৬৪৪০ | ১১৭২ | ১৭৪ | ৪৫ | ০ | ৫১১ | ১০০ | সন্ধ্যা পিক |
| ০১-০১-২০২০ | ১৮:০০:০০ | ৮০০৪ | ৮০০৪ | ০ | ৬১৪৭ | ১০৩২ | ১৭৪ | ৪৬ | ০ | ৫১১ | ১০০ | |
| ০১-০১-২০২০ | ১৭:০০:০০ | ৬৬৯১.৪ | ৬৬৯১ | ০ | ৫৬৭৩ | ২৪২ | ১৭৪ | ৪৭ | ০.৪ | ৪৬২ | ৮৪ | |
| ০১-০১-২০২০ | ১৬:০০:০০ | ৬৫২৫.৬ | ৬৫২৬ | ০ | ৫৭৩৮ | ১৭০ | ১৭৫ | ৪৩ | ৩.৬ | ৩১৩ | ৮৬ | |
| ০১-০১-২০২০ | ১৫:০০:০০ | ৬৫৪৫.৯ | ৬৫৪৬ | ০ | ৫৭৩৩ | ১৮৬ | ১৭৬ | ৪৭ | ৮.৯ | ৩১৩ | ৮৪ | |
| ০১-০১-২০২০ | ১৪:০০:০০ | ৬৫৫৬.৮ | ৬৫৫৭ | ০ | ৫৭২১ | ১৯৮ | ১৭৫ | ৪৭ | ১৪.৮ | ৩১৩ | ৯০ | |
| ০১-০১-২০২০ | ১৩:০০:০০ | ৬৮৪৬.৪ | ৬৮৪৬ | ০ | ৬০৫২ | ১৭৩ | ১৭৫ | ৪৭ | ৭.৪ | ৩১৩ | ৮২ | |
| ০১-০১-২০২০ | ১২:০০:০০ | ৬৭৪৯.৭ | ৬৭৫০ | ০ | ৬০০৩ | ১২৩ | ১৭৫ | ৪৭ | ৫.৭ | ৩০৭ | ৮৬ | ডে পিক |
| ০১-০১-২০২০ | ১১:০০:০০ | ৬৫১৫ | ৬৫১৫ | ০ | ৫৭৪৮ | ১৫১ | ১৭৫ | ৪৬ | ৫ | ৩০৭ | ৮০ | |
| ০১-০১-২০২০ | ১০:০০:০০ | ৬৫৬২.৪ | ৬৫৬২ | ০ | ৫৭৫৮ | ১৯২ | ১৭৫ | ৪৫ | ৪.৪ | ৩০৭ | ৭৮ | |
| ০১-০১-২০২০ | ০৯:০০:০০ | ৬৬৫৪ | ৬৬৫৪ | ০ | ৬০২৮ | ২০ | ১৭৫ | ৪৫ | ৬ | ৩০৭ | ৭৬ | |
| ০১-০১-২০২০ | ০৮:০০:০০ | ৬৩২৭.৫ | ৬৩২৮ | ০ | ৫৯১৬ | ০ | ১৬০ | ৪৬ | ৩.৫ | ১২৯ | ৭২ | |
| ০১-০১-২০২০ | ০৭:০০:০০ | ৫৮০৫.৭ | ৫৮০৬ | ০ | ৫৩৬১ | ০ | ১৬০ | ৪৬ | ০.৭ | ১২৯ | ৬৬ | |
| ০১-০১-২০২০ | ০৬:০০:০০ | ৫৩৪৬ | ৫৩৪৬ | ০ | ৪৯০২ | ০ | ১৬০ | ৪৬ | ০ | ১২৯ | ৬৬ | |
| ০১-০১-২০২০ | ০৫:০০:০০ | ৫১৪২ | ৫১৪২ | ০ | ৪৬৯৭ | ০ | ১৫৯ | ৪৬ | ০ | ১২৯ | ৬৮ | |
| ০১-০১-২০২০ | ০৪:০০:০০ | ৫২২৩ | ৫২২৩ | ০ | ৪৭৭৩ | ০ | ১৬০ | ৪৬ | ০ | ১২৯ | ৭০ | |
| ০১-০১-২০২০ | ০৩:০০:০০ | ৫৩৫১ | ৫৩৫১ | ০ | ৪৯০০ | ০ | ১৫৯ | ৪৭ | ০ | ১২৯ | ৭২ | |
| ০১-০১-২০২০ | ০২:০০:০০ | ৫৫৭০ | ৫৫৭০ | ০ | ৫১১৭ | ০ | ১৬০ | ৪৭ | ০ | ১২৯ | ৭৪ | |
| ০১-০১-২০২০ | ০১:০০:০০ | ৫৯১৩ | ৫৯১৩ | ০ | ৫৪৯৩ | ০ | ১৬০ | ৪৭ | ০ | ১২৯ | ৭৬ | |

## 1.Building the Model

For building the model and load prediction we have used Artificial Neural network and Regression learner from machine learning.

## Neural network:

A neural network is a massively parallel distributed processor made up of simple Processing units that have a natural tendency for storing experiential knowledge and making it available for us. Artificial neural network (ANN) is a type of Artificial Intelligence technique that mimics the behavior of the human brain (Haykin, 2009).

ANNs have the ability to model linear and non-linear systems without the need to make assumptions implicitly as in most traditional statistical approaches. They have been applied in various aspects of science and engineering (Rivard & Zmeureanu, 2005; Chantasut et al., 2005).

ANNs can be grouped into two major categories: feed-forward and feedback (recurrent)

networks. In the former network, no loops are formed by the network connections, while
one or more loops may exist in the latter. The most commonly used family of feed-forward
networks is a layered network in which neurons are organized into layers with connections
strictly in one direction from one layer to another (Jain et al., 1996).

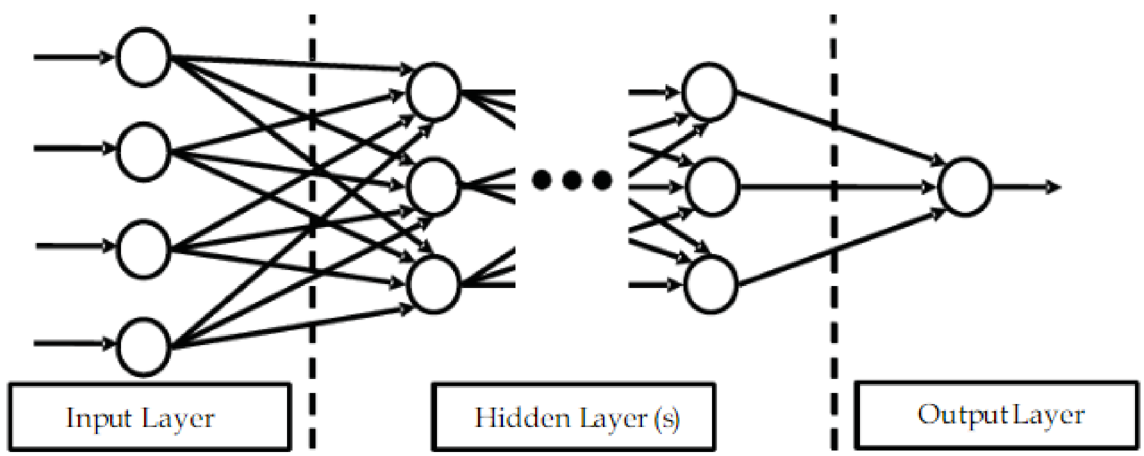



Fig. 1. A Multi-layered perceptron (MLP) network

After data collection, three data preprocessing procedures are conducted to train the ANNs
More efficiently. These procedures are: (1) solve the problem of missing data, (2) normalize

data and (3) randomize data. The missing data are replaced by the average of neighboring values during the same week. Normalization procedure before presenting the input data to the network is generally a good practice, since mixing variables with large magnitudes and small magnitudes will confuse the learning algorithm on the importance of each variable and may force it to finally reject the variable with the smaller magnitude (Tymvios et al., 2008).

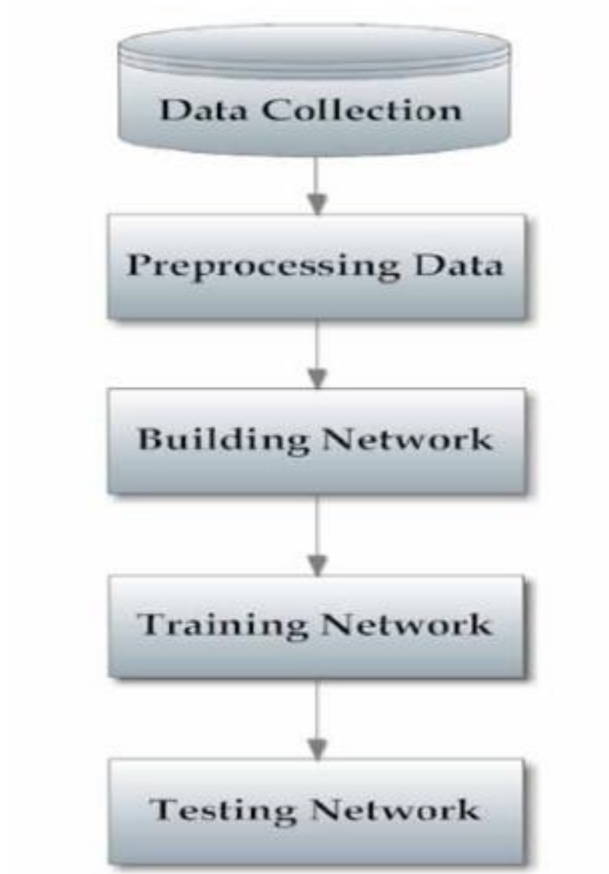
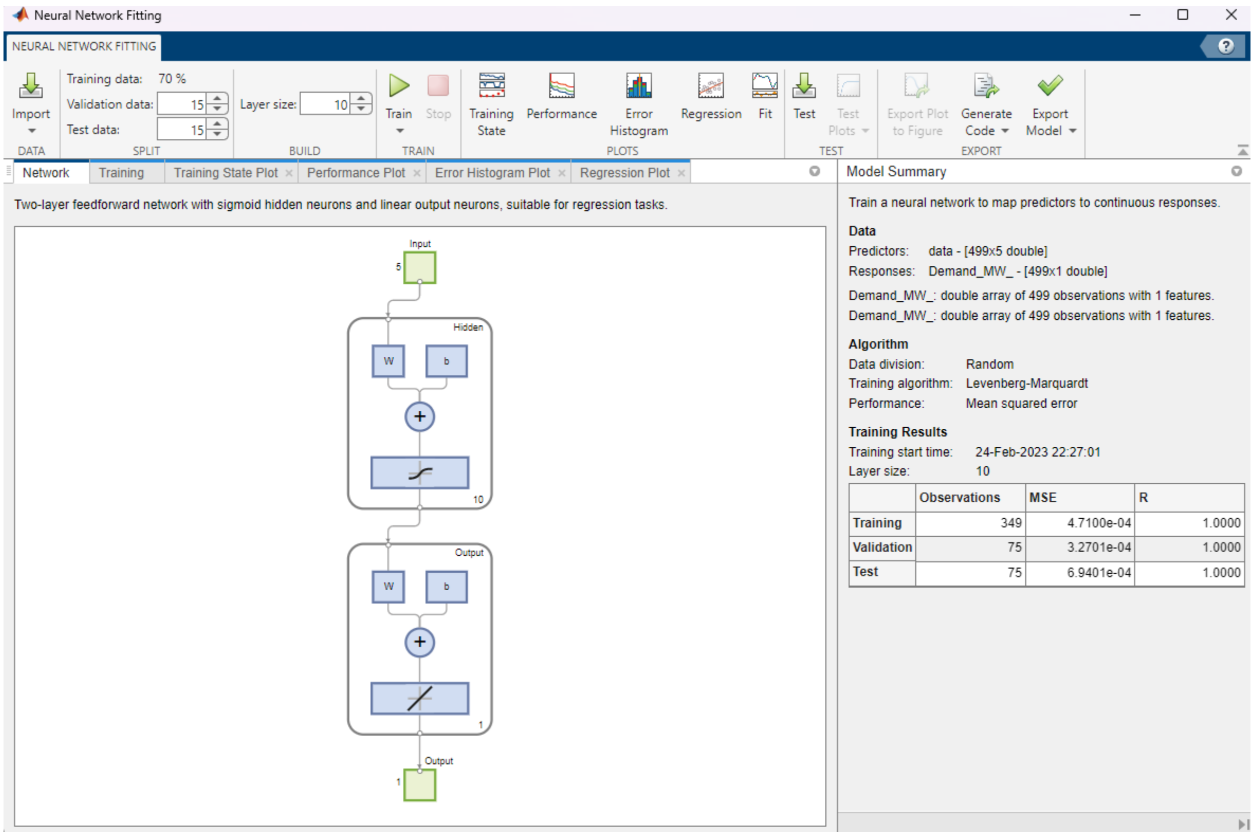



Fig. 6. Basic flow for designing artificial neural network model

# 2.Training the Model:




**This example shows how to train a feed forward neural network to predict Load**

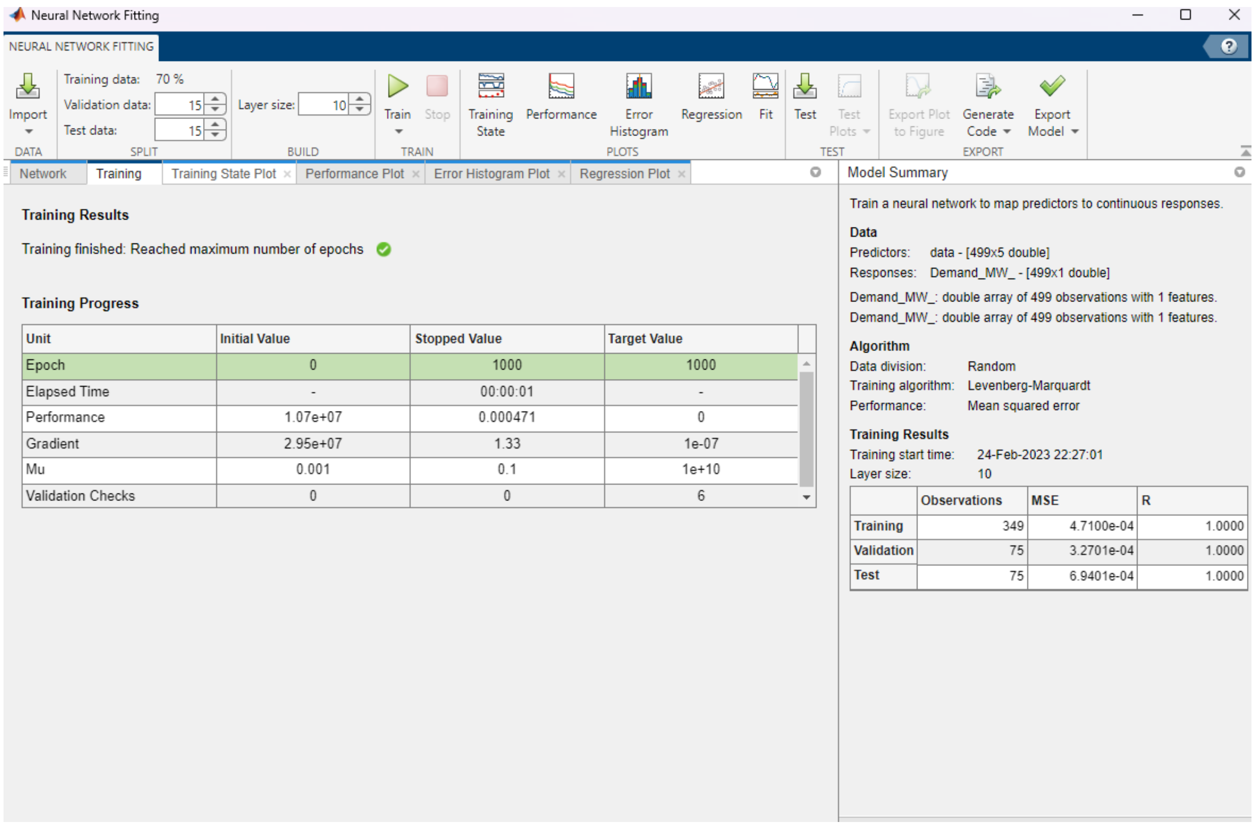
- Read Data from the Excel datasheet
- Assign Input Variables and Target Values

- Create and Train the Two-Layer Feed forward Network
- Use the Trained Model to Predict Data

Code generated alongside the network:

```matlab
% This script assumes these variables are defined:
%
%   data - input data.
%   Demand_MW_ - target data.

x = data';
t = Demand_MW_';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

After training the performance plot, Error histogram and the Regression plots of levenberg-marquardt




# Result and Output Graph:

# The mean square error vs epoch plot

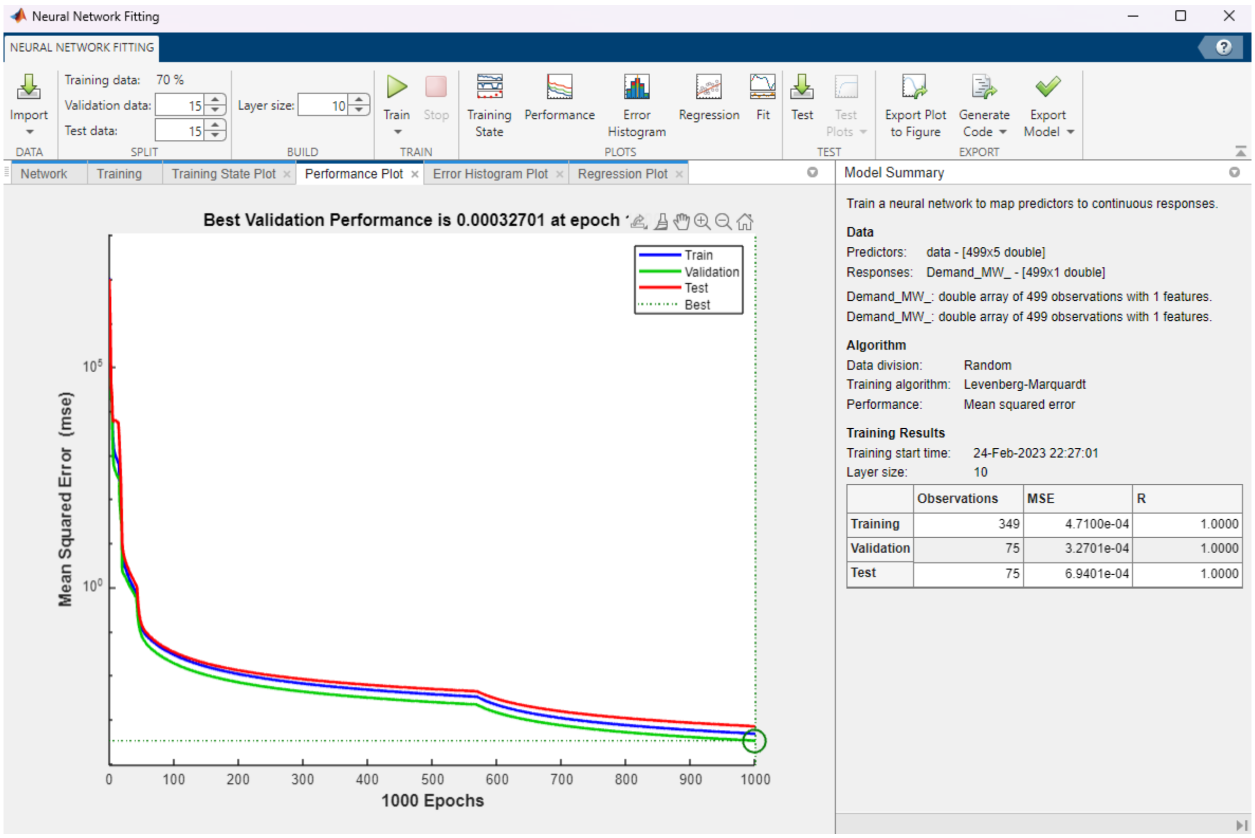



Figure: performance plot

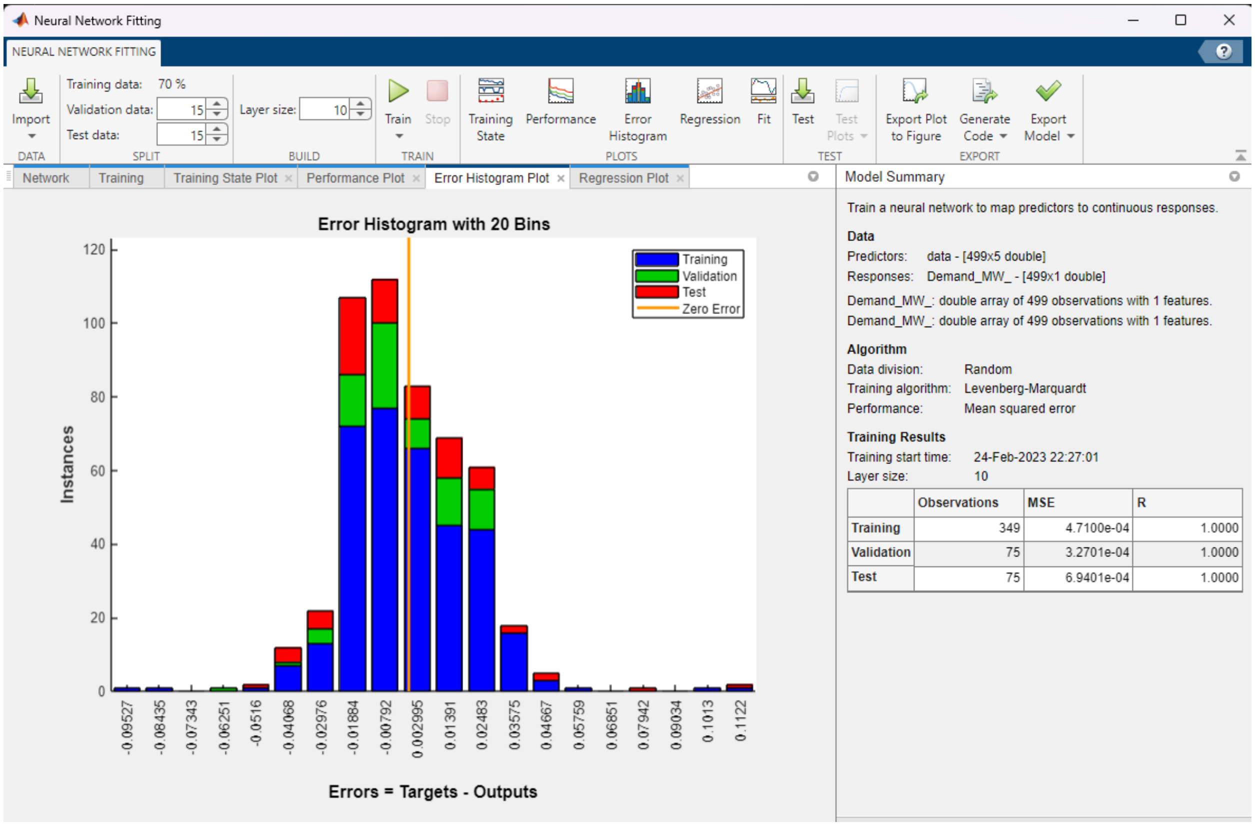



Figure: Error histogram

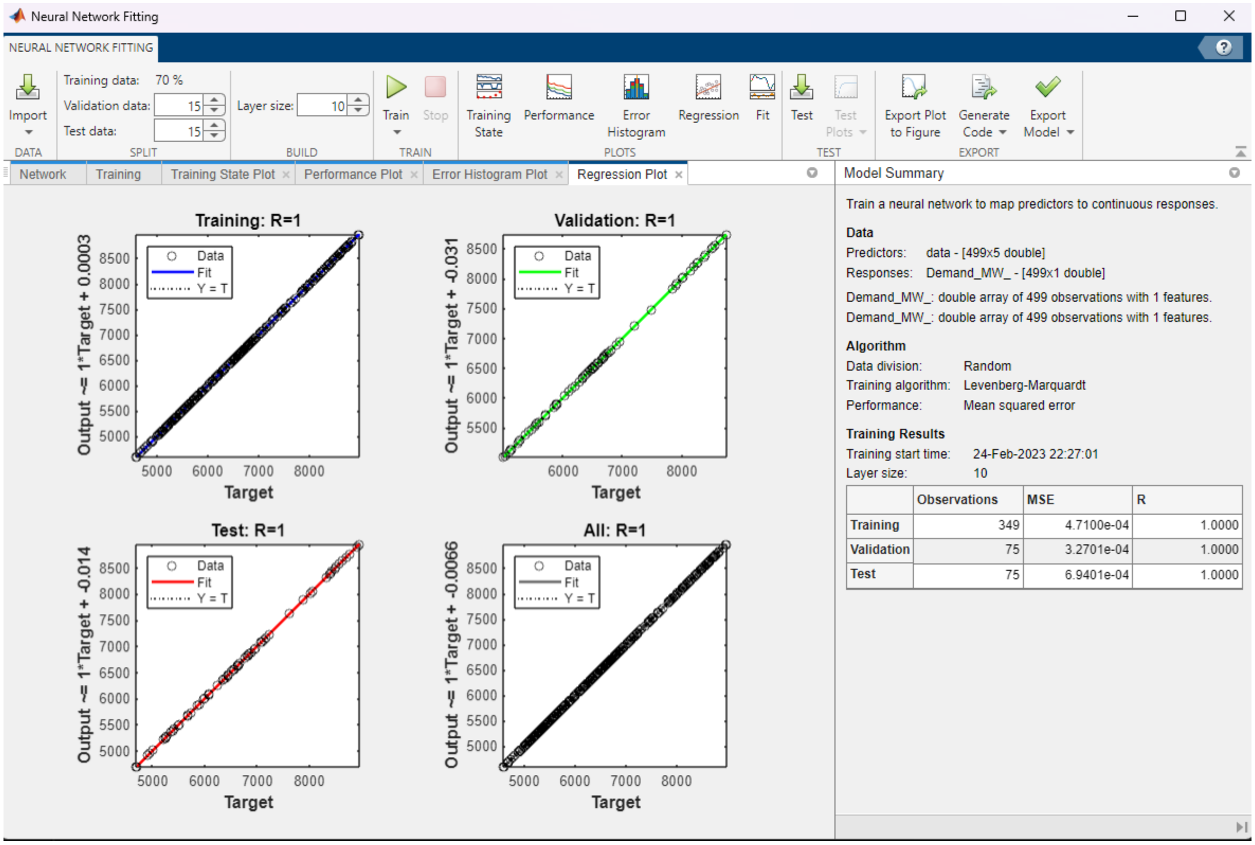
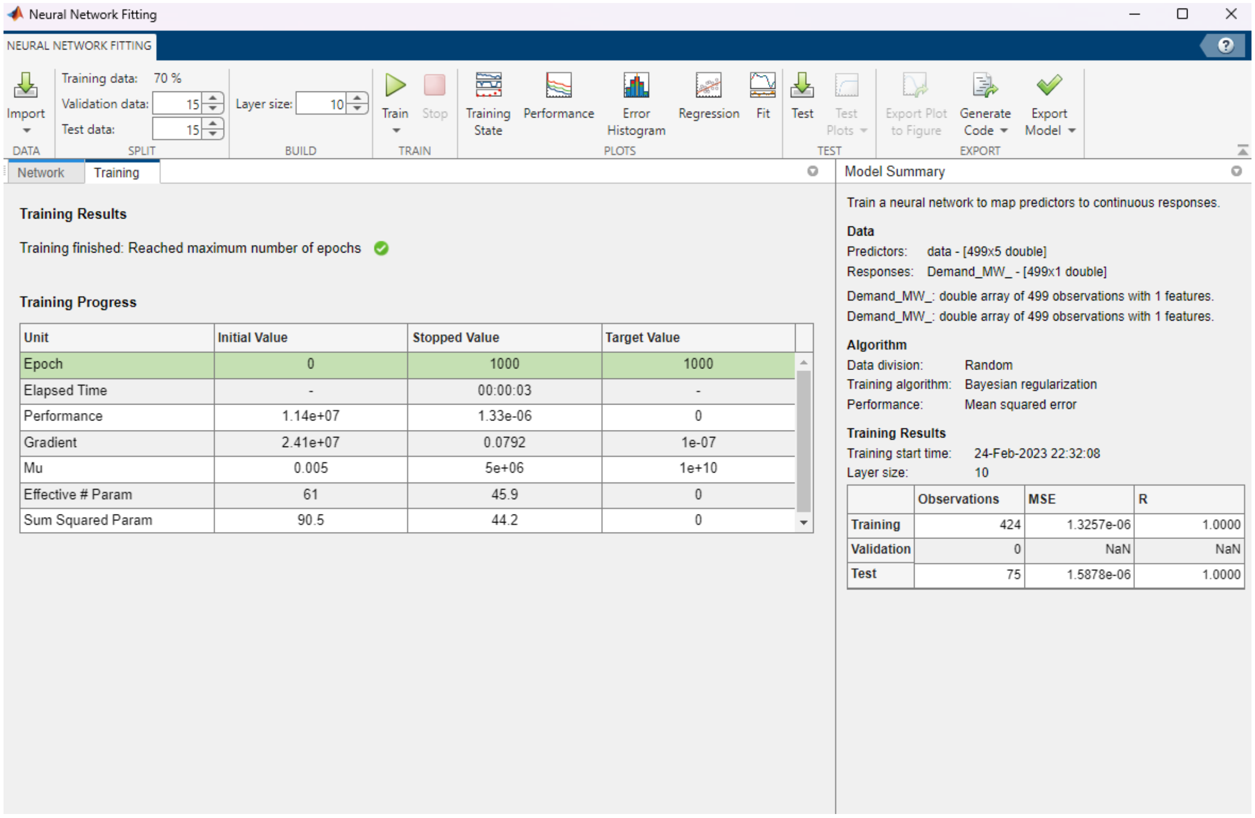



Figure: Regression plots

After training the performance plot, Error histogram and the Regression plots of Bayesian regularization.


Neural Network Fitting
NEURAL NETWORK FITTING
Training data: 70 %
Validation data: 15
Test data: 15
Layer size: 10
Import
Train
Stop
Training State
Performance
Error Histogram
Regression
Fit
Test
Test Plots
Export Plot to Figure
Generate Code
Export Model
DATA
SPLIT
BUILD
TRAIN
PLOTS
TEST
EXPORT
Network
Training
Training Results
Training finished: Reached maximum number of epochs
Training Progress
Unit | Initial Value | Stopped Value | Target Value
Epoch | 0 | 1000 | 1000
Elapsed Time | - | 00:00:03 | -
Performance | 1.14e+07 | 1.33e-06 | 0
Gradient | 2.41e+07 | 0.0792 | 1e-07
Mu | 0.005 | 5e+06 | 1e+10
Effective # Param | 61 | 45.9 | 0
Sum Squared Param | 90.5 | 44.2 | 0
Model Summary
Train a neural network to map predictors to continuous responses.
Data
Predictors: data - [499x5 double]
Responses: Demand_MW_ - [499x1 double]
Demand_MW_: double array of 499 observations with 1 features.
Demand_MW_: double array of 499 observations with 1 features.
Algorithm
Data division: Random
Training algorithm: Bayesian regularization
Performance: Mean squared error
Training Results
Training start time: 24-Feb-2023 22:32:08
Layer size: 10
| Observations | MSE | R
Training | 424 | 1.3257e-06 | 1.0000
Validation | 0 | NaN | NaN
Test | 75 | 1.5878e-06 | 1.0000

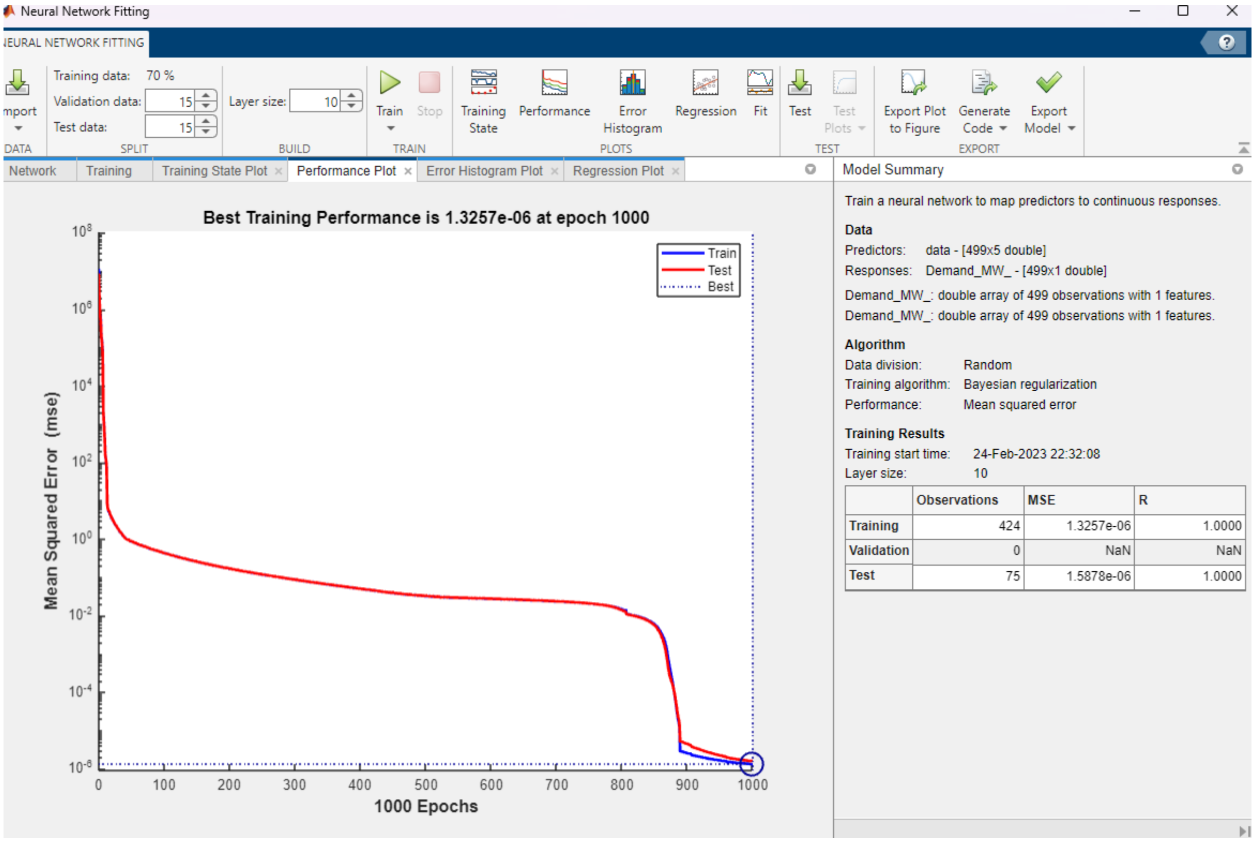



Figure: performance plot

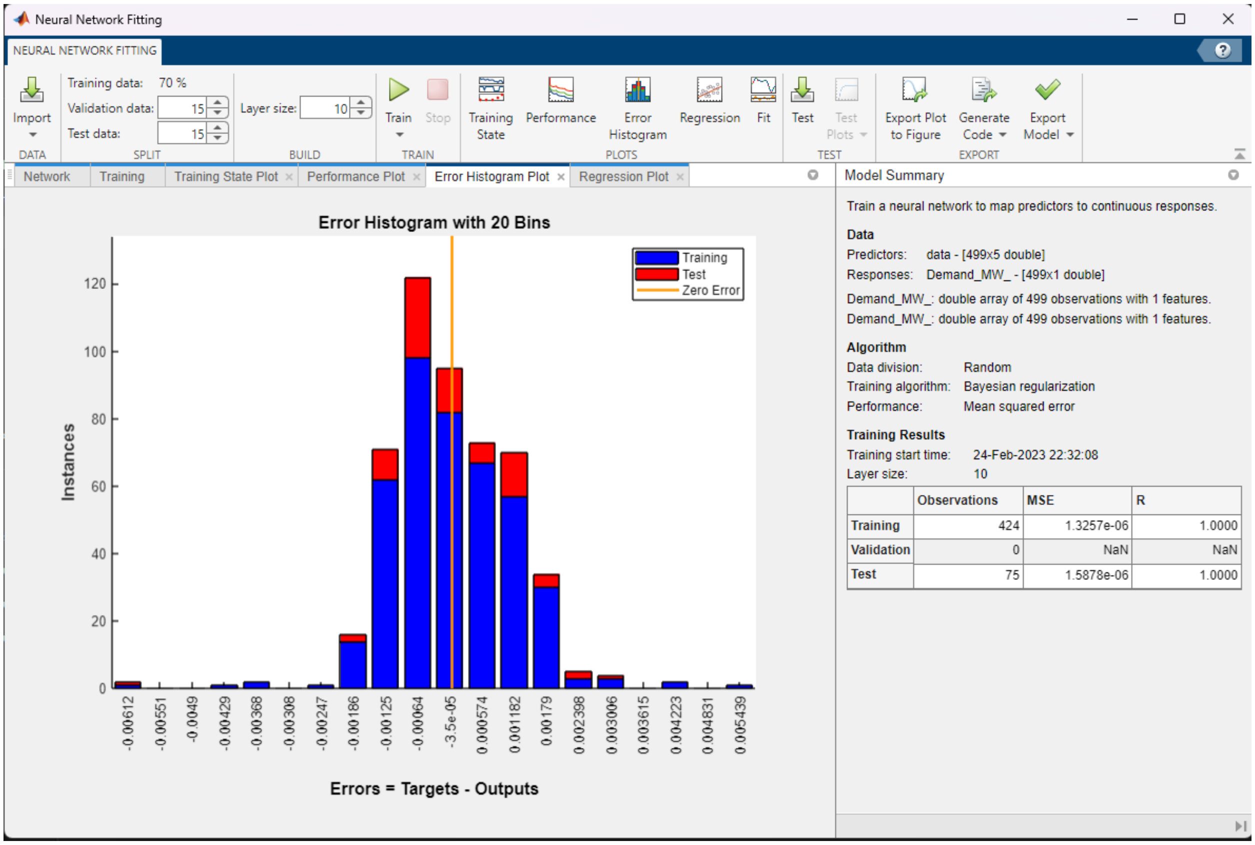



Figure: Error histogram

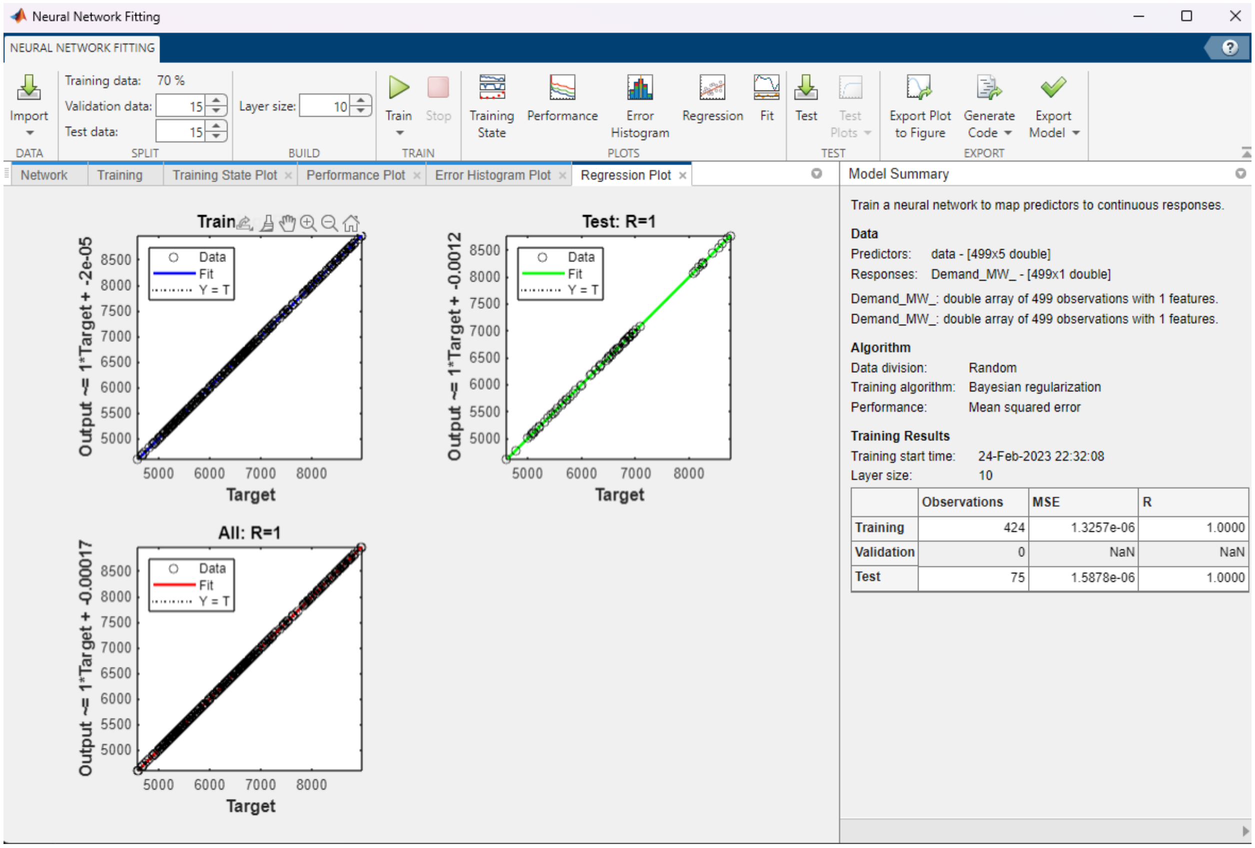



Figure: Regression plots

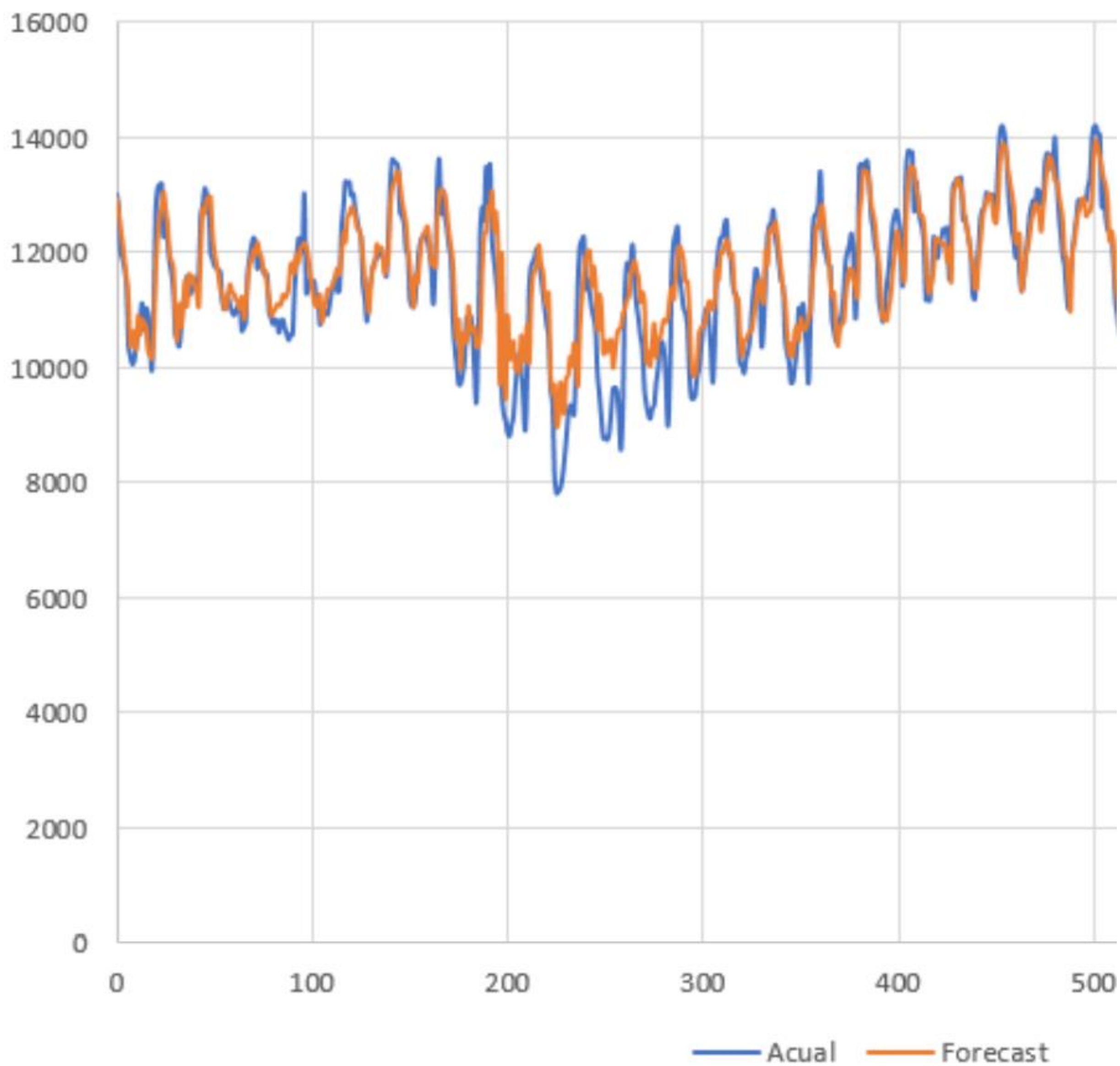
From the plot, we can see that our model is working quite well. There is a small amount of error in our forecasted data. In most conditions, the forecasted data matches the actual data perfectly. But we can also see

Using the code snippet below we were able to use the data from the network learning and compare the predicted and actual value.

untitled.m untitled2.m untitled3.m

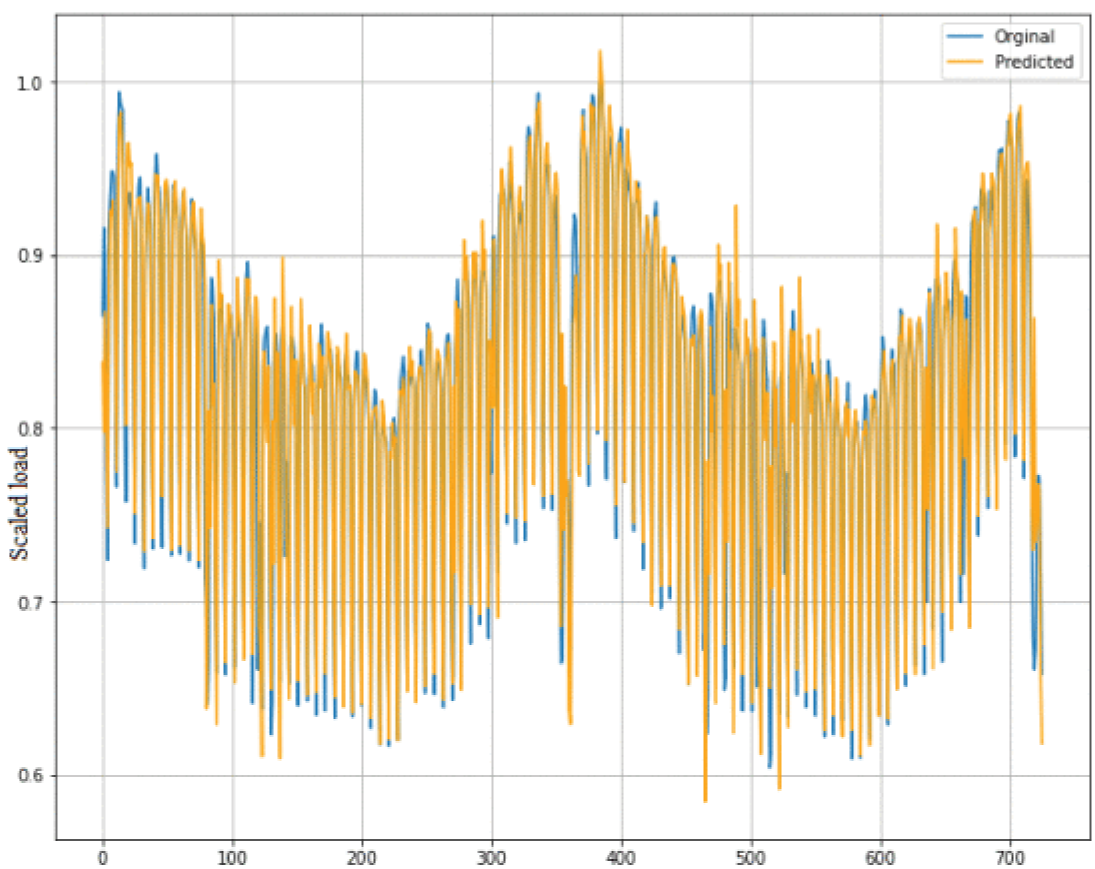
```
loadpredict=sim(net,x);
MAPE=((y-loadpredict).*100)./(y.*500)

n=1:499;
plot(n,loadpredict,'DisplayName','loadpredict','Color','m');
hold on
plot(n,y,'DisplayName','actualload','Color','b');
hold off
```

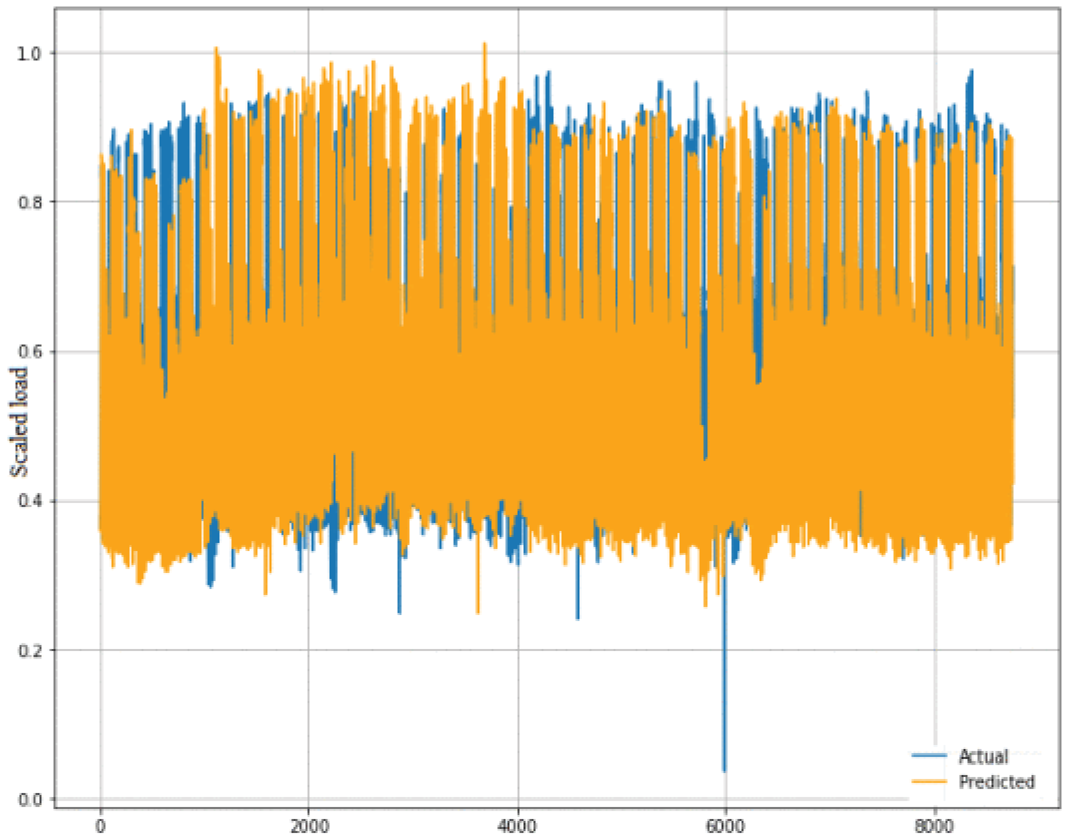
| Name | Value |
|---|---|
| y | 1x499 double |
| x | 5x499 double |
| trainFcn | 'trainlm' |
| tr | 1x1 struct |
| textdata | 500x6 cell |
| t | 1x499 double |
| r | 1x499 double |
| performance | 19.9691 |
| net_1 | 1x1 struct |
| net | 1x1 network |
| n | 1x499 double |
| MAPE | 1x499 double |
| mape | 0 |
| loadpredict | 1x499 double |
| hiddenLayerSize | 10 |
| e | 1x499 double |
| Demand_MW_ | 499x1 double |
| data | 499x5 double |
| colheaders | 1x1 cell |




**Comparative table with  other papers:**

Here are some of the forecasting vs predicted data of Germany, Malaysian :






## Conclusion:

Analyzing the result, we can say that our ANN model and regression learning model is working very accurately under normal conditions. The use of seasonality in training data increased the accuracy more. This model can be improved by adding other parameters such as temperature conditions, fuel prices, holidays etc. to the training data. As Load forecasting is extremely useful in power system design, our project can play a key role in making a power system efficient.

Load forecasting helps utility companies to plan well regarding future consumption or load demand. It will provide enough information to these companies to take economically viable decisions for future investments in generation and transmission. It also helps to determine the number of required resources such as fuels for generating power in power plants so that uninterrupted and economic power generation will be possible. According to the forecasted demand planners can plan the future location, type, and size of future power plants so that the maximum benefit can be achieved. Deciding and planning for the maintenance of a power system can also be made much easier and more effective. By understanding the demand, utility companies will know when to conduct the maintenance and ensure the minimum impact on the consumers. By using load forecasting, power plants can avoid under-generation or over-generation and ensure efficient use of resources. The target of our project was to show how machine learning techniques can be used in power systems. We have also learned how important load forecasting is in power system analysis. Although not accurate at all times, our model is forecasting load efficiently with a low error percentage. So, overall, this project is serving its purpose

**Problem and future scope:**

Load forecasting task is difficult due to the complex nature of loads which may vary depending on the seasons and the total consumption for two similar seasons may vary. It is sometimes difficult to accurately fit the numerous complex factors that affect demand for electricity into the forecasting models.

Load forecasting **minimizes utility risk by predicting future consumption of commodities transmitted or delivered by the utility**. Techniques include price elasticity, weather and demand response/load analysis, and renewable generation predictive modeling.

## Reference:

1. On Short-Term Load Forecasting Using Machine Learning Techniques and a Novel Parallel Deep LSTM-CNN Approach

BEHNAM FARSI1, MANAR AMAYRI2, NIZAR BOUGUILA1, (Senior Member, IEEE),AND URSULA EICKER

**2. Neural networks for short-term load forecasting: a review and evaluation**
**Publisher: IEEE**

H.S. Hippert; C.E. Pedreira; R.C. Souza