Group 2 (John Buckheit, Nikhil Siddhartha, Shivasagar Boraiah)
AMS 559
Homework 1

Method 1: Nonlinear Input-Output Time Series Neural Network in MATLAB
(John Buckheit)

Approach
A nonlinear input-output neural network is used to predict the next 96 15-minute time steps (one full day ahead). Input parameters consists of daily time steps and weekly time steps, the output parameter is predicted energy demand load.

Neural Network
The network begins with an input layer of the first daily and weekly time steps. From this it makes a random guess at what the energy demand for that time step should be, giving us the output layer. This prediction is compared to the given actual energy demand, and the network adjusts the weights in the hidden layers to improve its guess on the next iteration. The network repeats this for every input/output, which trains the network to produce more accurate predictions.

Reasoning for input parameters
It is suspected that energy demand follows a daily and weekly cycle. By this we mean that every day sees a similar energy usage pattern, as well as every week. For example, every evening, users might begin to turn their lights on at a specific time, so we can expect an increase in energy load demand at that time. By classifying the 15-minute intervals into 96 daily steps, we can train the network to recognize similarities during each of these steps over the course of many days. Similarly, we could expect that users might use more electricity during the weekend since they might be spending more time at home during those days. This gives us a similar second input parameter, except it is classified into 672 15-minute steps, allowing the network to recognize weekly patterns.

Network Architecture
Separation of training/validation/testing: 70%/20%/10%
Algorithm: Levenberg -Marquardt
Delays: 1:2
Layers: 1,5,9,13

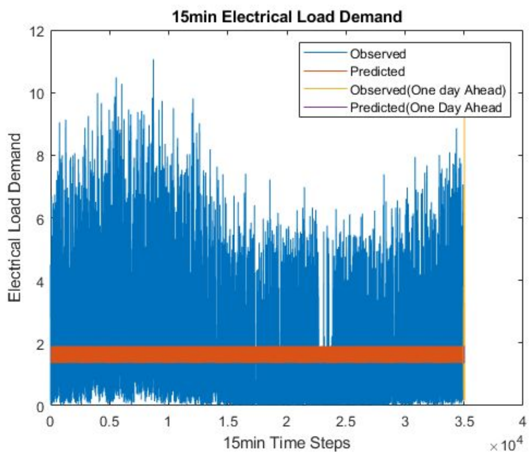
Results
Number of hidden layers were adjusted between 1,5,9,13 for each home in order to find the lowest Mean Absolute Error (MAE). This MAE is calculated on 364 days of data used for training/validation/testing, leaving the last day (365th) to be used as a comparison for the One-Day-Ahead (ODA) prediction. Based on this, we declare that Home 8 with 1 hidden layer produced the worst prediction, and Home 9 with 13 hidden layers produced the best prediction.

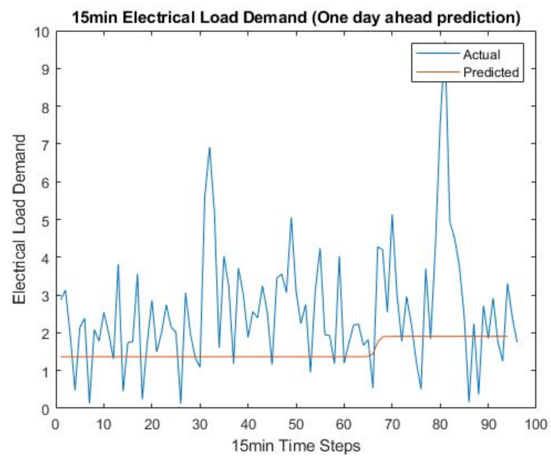
(Table 1)
MAE obtained for different number of hidden layers for each home

Home Number	Number of Hidden Layers			
	1	5	9	13
1	1.0274	1.0154	1.0165	1.0183
2	1.1273	1.1249	1.1252	1.1222
3	1.0597	1.0575	1.0582	1.0568
4	0.9711	0.9524	0.9494	0.9505
5	1.1241	1.1124	1.1130	1.1137
6	0.9839	0.9724	0.9684	0.9689
7	1.1488	1.1331	1.1285	1.1232
8	1.2723	1.2612	1.2598	1.2588
9	0.7981	0.7960	0.7955	0.7936
10	1.0342	1.0288	1.0288	1.0298
Average Error	1.0547	1.0454	1.0443	1.0436

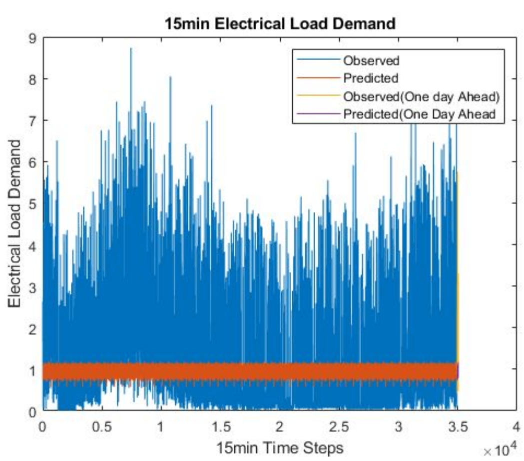
The following figures show the best and worst scenarios for homes/number of hidden layers, based on MAE. In addition, we show the results for the 365th day (ODA prediction), and the corresponding MAE for that day

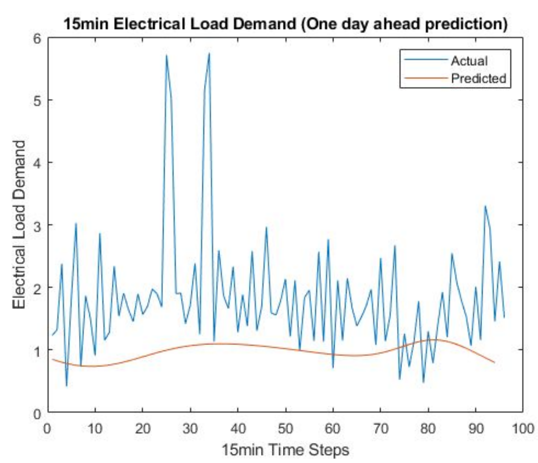
Figures a-d
Worst result:
Home 8, 1 hidden layer
MAE: 1.2723
MAE ODA: 1.3582
(Figures 1,2)





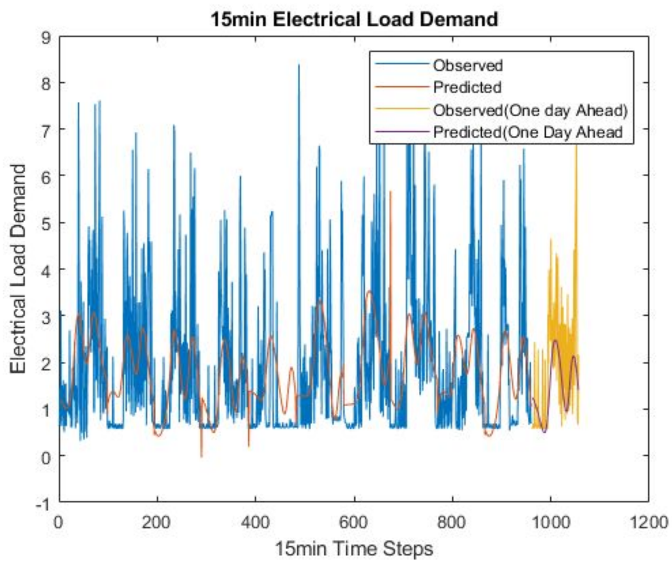
Best result:
Home 9, 13 hidden layers
MAE: 0.7936
MAE ODA: 0.9427
(Figures 3,4)

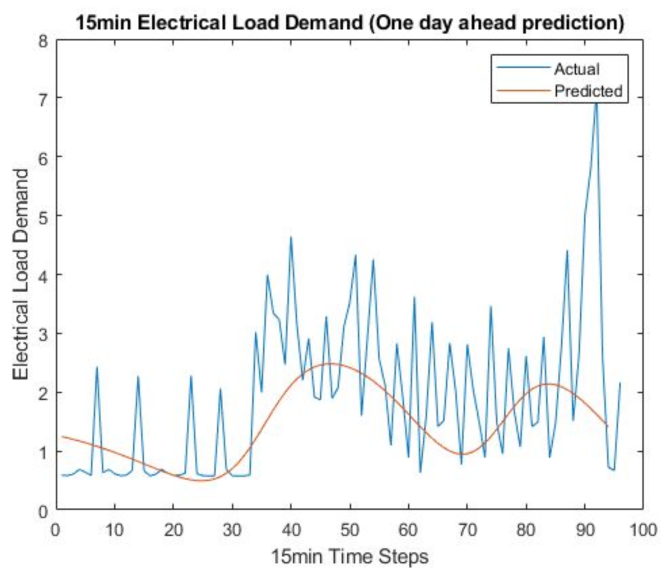




The results from the network's prediction of Home 1 energy load are shown below with a time scale restricted to the first 10 days, with the 11th day being ODA prediction. This gives a better visual interpretation of the networks predictive abilities as opposed to viewing all 365 days at once.

(Figures 5,6)

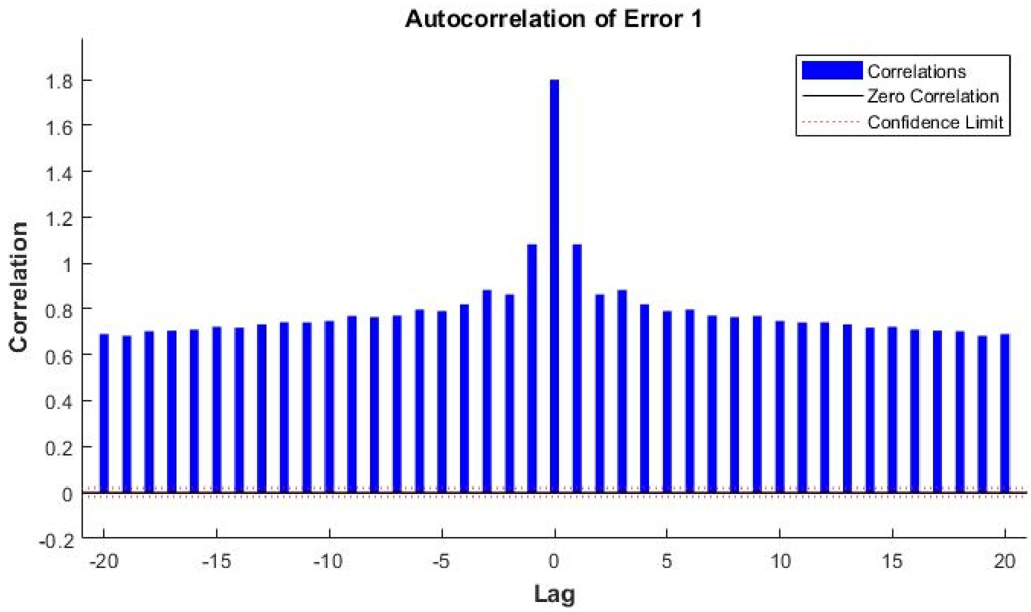


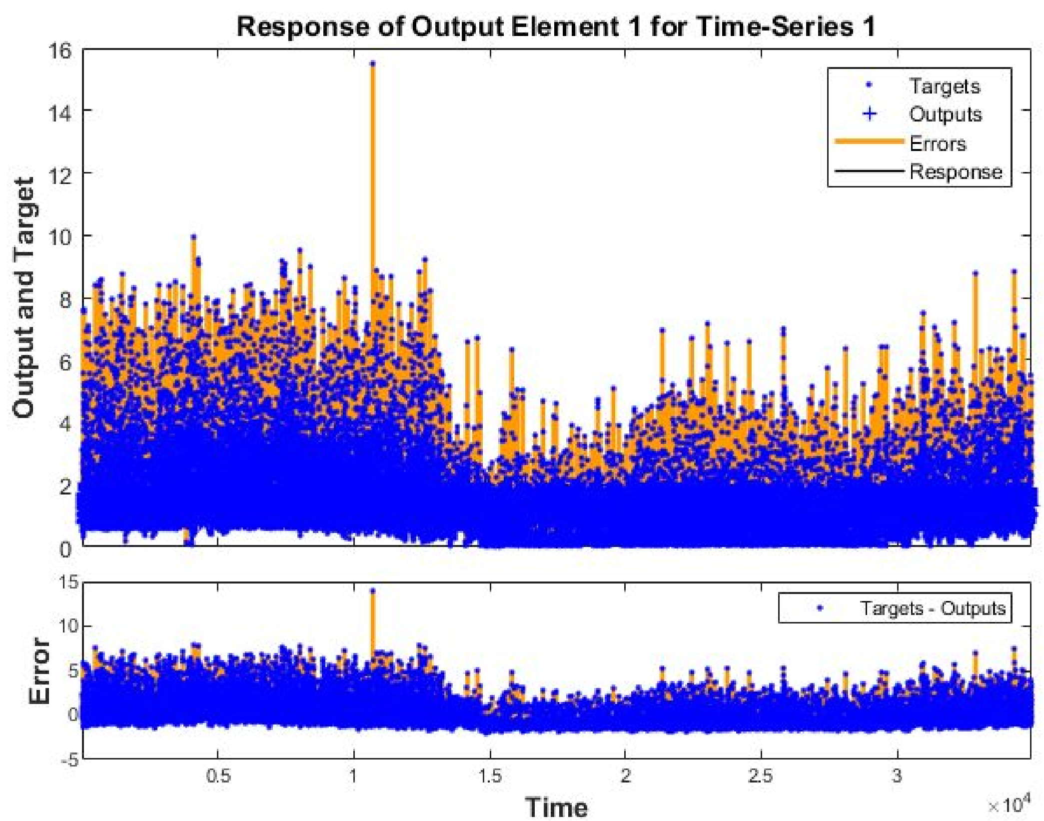


Discussion

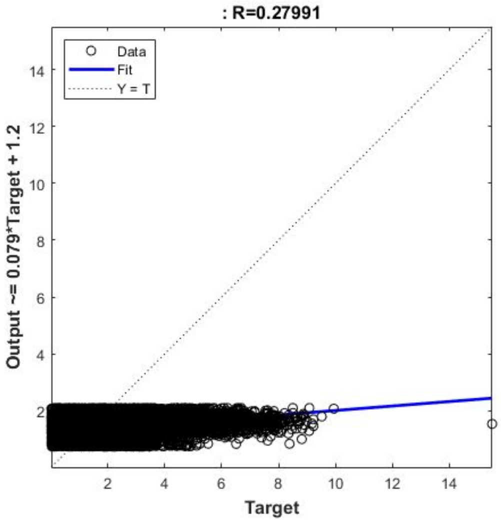
This neural network provides a good baseline for improvement on energy prediction. A better understanding of network architecture could lead to better results; as well as repeated runs with different numbers of hidden layers. It should be noted that the success of the results was based on MAE for the training/validation/testing data. If success was based on the MAE of ODA prediction, we might declare a different worst/best case scenarios.

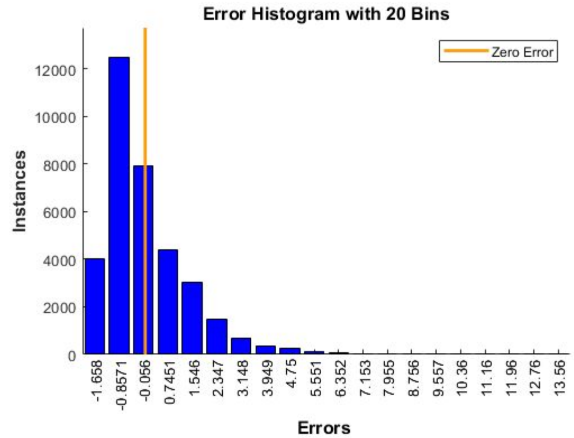
Addition Figures for Nonlinear Input-Output Neural Network

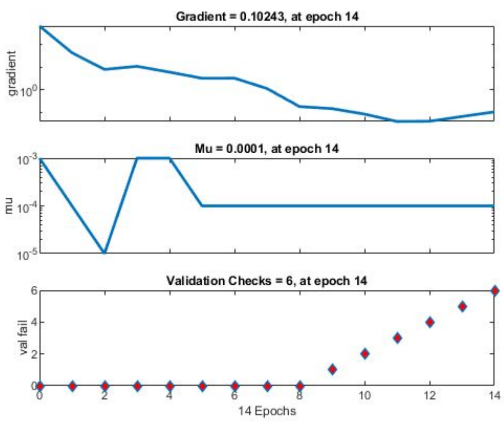


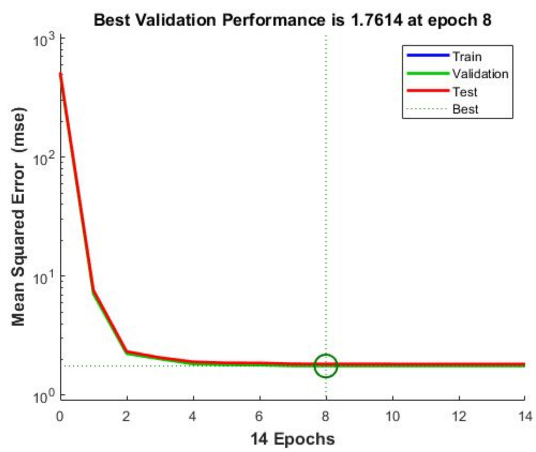


Addition Figures for Nonlinear Input-Output Neural Network









Methods 2,3,4,5: Linear Regression, Random Forest Model, SARIMAX, Naive method

(Nikhil Siddhartha, Shivasagar Boraiah)

Separation of Data:
Taken as input in the form of the index of the time-slot. I have taken care of the maximum and minimum value of the input that is valid.
Min value = 96 (index of time-slot after 1 day)
Max value = 34944 (index time-slot after 364 days)

Then, I train on all the data before the time-slot
Prediction and testing is done for the next day from the time-slot given. (i.e. next 96 items)

Features Considered:
To create these features, I used the fact that time stamp data is created after every 15 minutes. Hence, I populated the following fields for every 15 minutes, starting from 12/1/2014 to 11/30/2015.

The features that were built are:

1. Month - to model the yearly trends in energy consumption (e.g summer vs winter)
2. Day - to model the monthly trends in energy consumption (r.g month beginning vs. month end)
3. Weekday - to model the weekly trends in energy consumption (e.g weekday vs weekend)
4. Minute - to model the daily trends in energy consumption (e.g. morning vs evening)

Models used:
2. Linear Regression
3. Random Forest Model

Mean Absolute Errors (MAE):
2. Linear Regression:

```
MODEL : LINEAR REGRESSION
MAE: House[1], 1.3842771912426108
MAE: House[2], 1.6575772741759216
MAE: House[3], 1.757195698276184
MAE: House[4], 0.8508289271089308
MAE: House[5], 1.6159036257161234
MAE: House[6], 1.0995928273446747
MAE: House[7], 1.7165909204053094
MAE: House[8], 1.6850882502435782
MAE: House[9], 1.264611838218349
MAE: House[10], 1.3748836072516915
```

3. Random Forest Model

```
MODEL : RANDOM FOREST
MAE: House[1], 1.1018142677083334
MAE: House[2], 1.4142444208333333
MAE: House[3], 1.0788536979166667
MAE: House[4], 0.8294385572916667
MAE: House[5], 1.0180408385416666
MAE: House[6], 1.0075472694444445
MAE: House[7], 1.0564943020833333
MAE: House[8], 1.2987328552083335
MAE: House[9], 0.7396080083333333
MAE: House[10], 1.1589714322916664
```

4. SARIMAX (Home1)

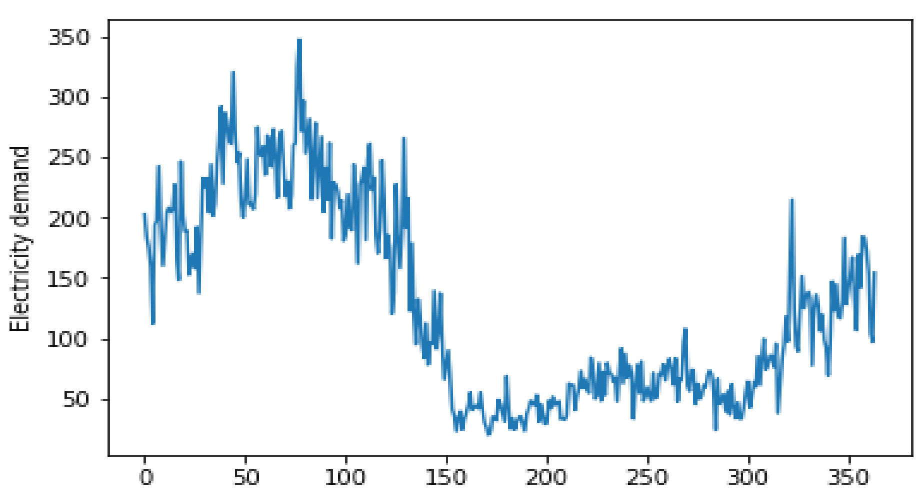
```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3702      0.022     17.106      0.000       0.328       0.413
ma.L1         -0.9859      0.005   -187.622      0.000      -0.996      -0.976
ma.S.L96      -1.0000    231.002     -0.004      0.997    -453.756     451.756
sigma2         1.6777    387.579      0.004      0.997    -757.964     761.319
==============================================================================
```

MAE: 1.7452254974474286

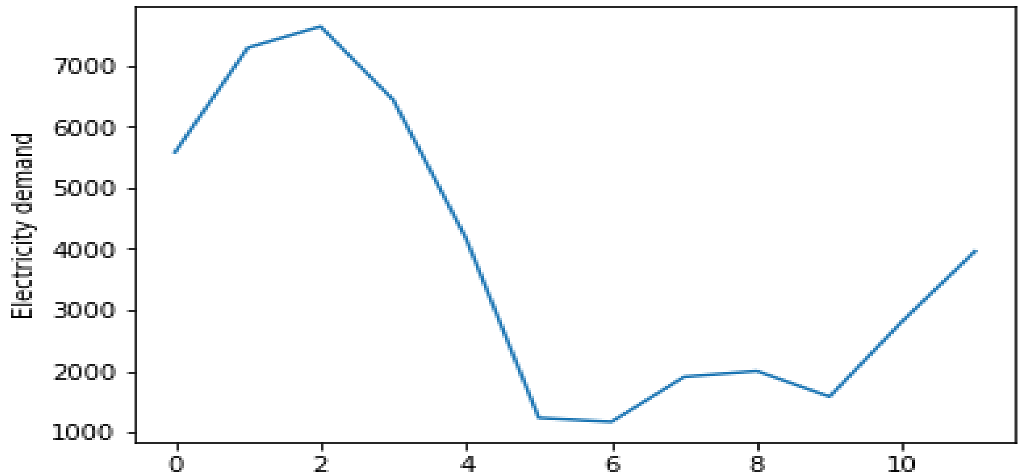
5. Naive Method (Home1)
MAE: 2.186657056890723

Plots Visualization:

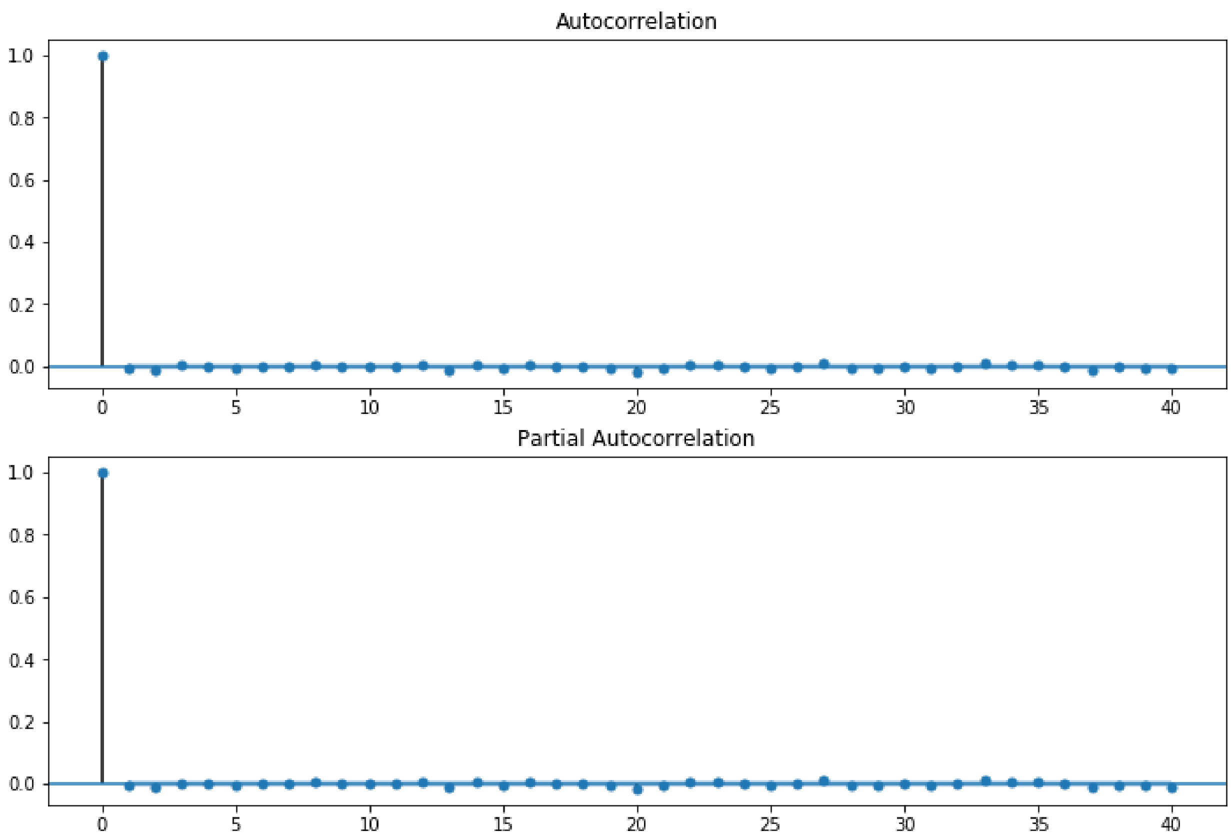
1. Energy Demand plot on daily decomposition (Home1)



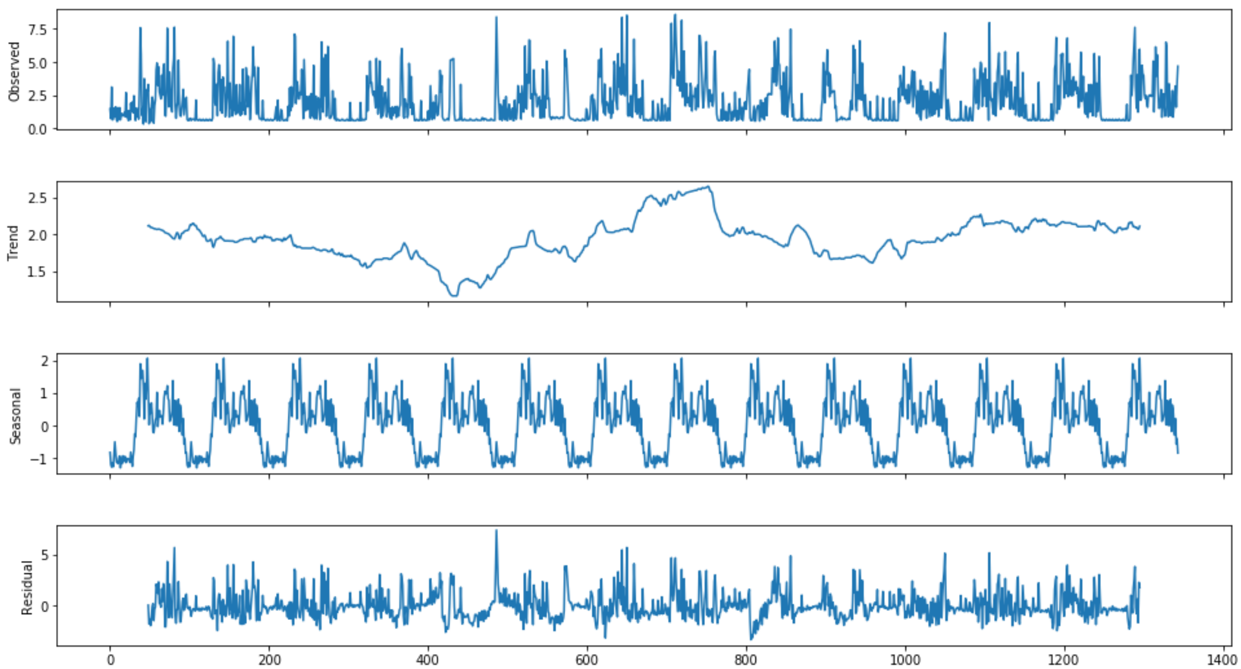
2. Energy Demand plot on monthly decomposition (Home1)



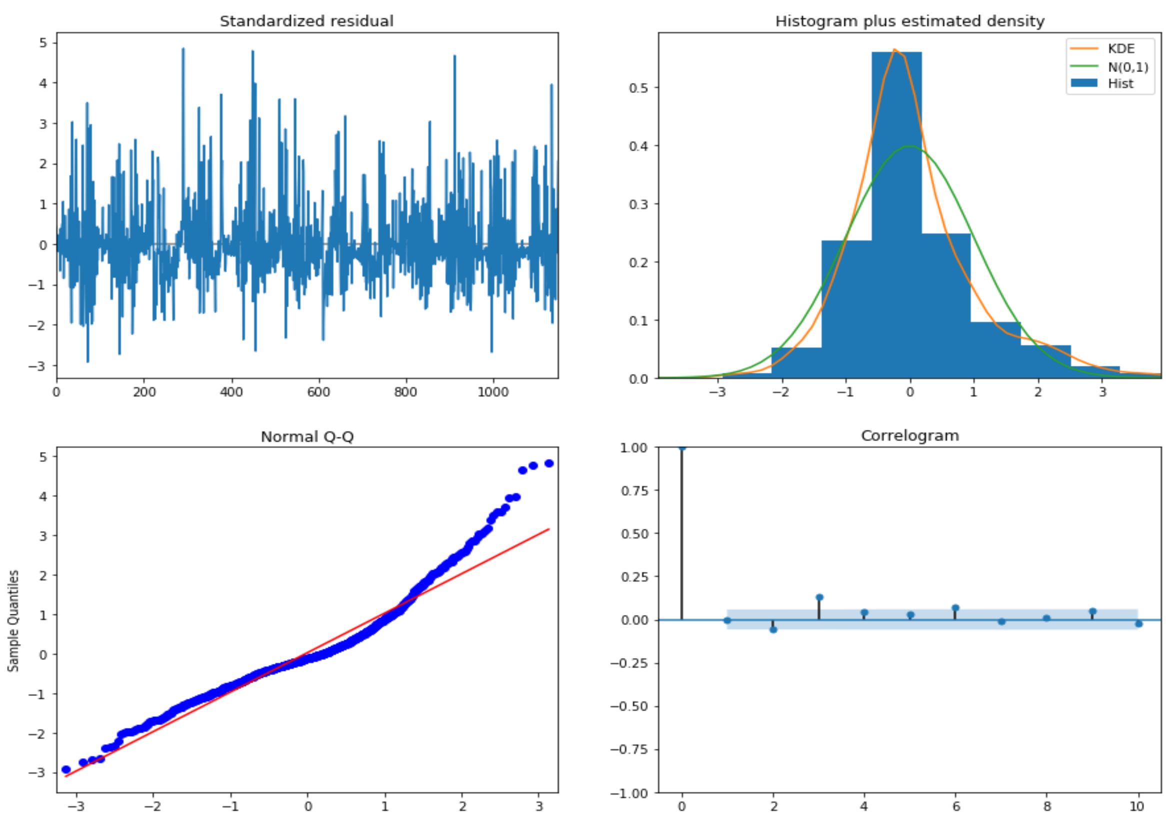
3. AutoCorrelation (Home1)



4. Seasonal Decomposition of data (Home 1)



5. Results from SARIMAX model:



Comparison of Methods

All three methods are compared in the following table:

(Table 2)

MAE obtained for each method for each home

Home Number	Method				
	1: Neural network (13 hidden layers)	2: Linear Regression	3: Random Forest Model	4: SARIMAX	5: Naive Approach
1	1.0183	1.3842	1.1018	1.7452	2.1866
2	1.1222	1.6575	1.4142		
3	1.0568	1.7571	1.0788		
4	0.9505	0.8508	0.8294		
5	1.1137	1.6159	1.0180		
6	0.9689	1.0995	1.0075		
7	1.1232	1.7165	1.0564		
8	1.2588	1.6850	1.2987		
9	0.7936	1.2646	0.7396		
10	1.0298	1.3748	1.1589		
Average Error	1.0436	1.4407	1.0704		

The first three methods performed relatively well, with the Neural network and Random forest performing slightly better than the Linear Regression. The Naive Approach had the largest error, which was expected. SARIMAX took too long to train so only home 1 was considered.