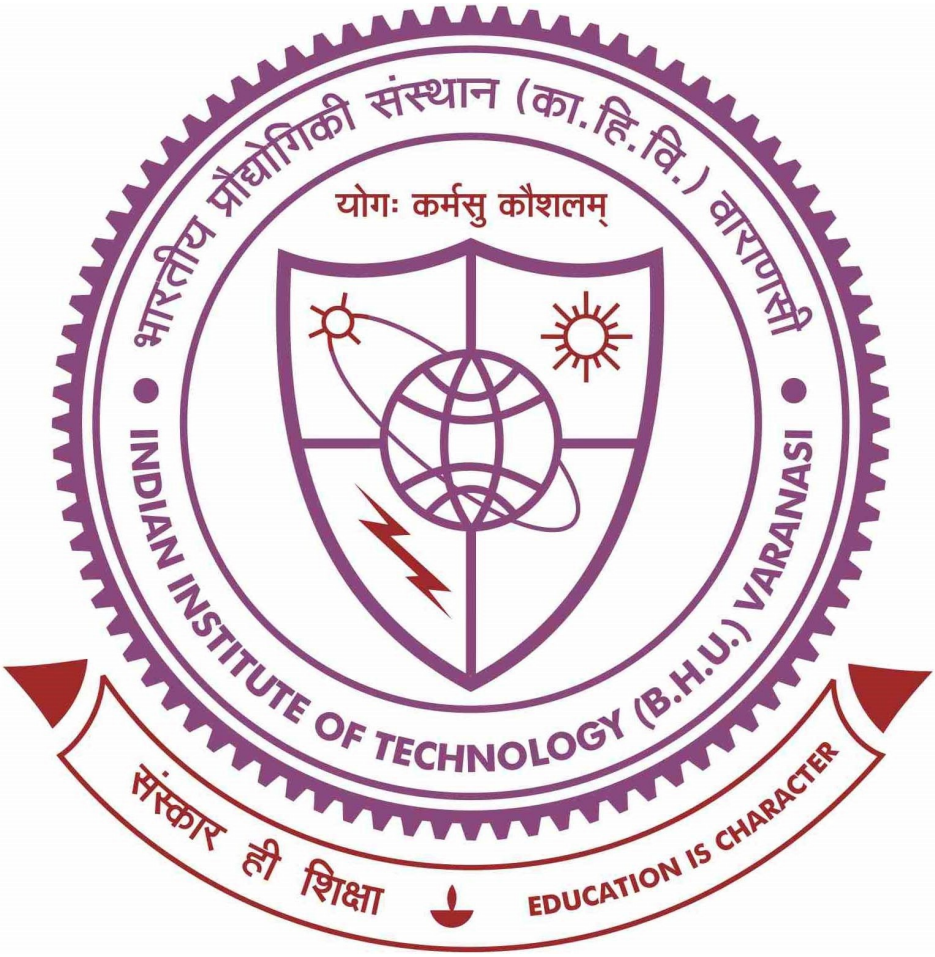


IIT (BHU) VARANASI

DEPT. OF ELECTRICAL ENGINEERING

B.TECH PROJECT REPORT

DR. S.P SINGH

TEAM MEMBERS

PIYUSH SINGH	18085045
PRADYUMN SINGH	18085047
USHMITA PAREEK	18085075
ANSHUMAN PATHAK	18085095

Medium-Term Load Forecasting using Meteorological and Historical Data

Abstract- Electrical load forecasting is very crucial for the optimal operation of an electrical system. The importance of these precise electric load forecasting gets even more important in the case of smart distribution grids for distributed energy management systems and demand response programs. This helps in pre-planning load demands and preparing for any possible surge or drop in supply. Short-term accurate forecasts help in maintaining the stability of the grid by balancing the demand and supply. Medium-term forecasts assist in planning maintenance of plants and purchase of fuel. Long-term forecasts lessen the financial risks. In this paper, we aim at medium-term forecasting. Hourly forecasts are made for one week ahead. In this project, we have used Seasonal ARIMA (SARIMA), Linear Regression, Gradient Boosted Regression, Multi-Layer Perceptron Regression and applied it to historical data from Ontario province.We analyse the predicted results, compare them with the actual values of system load that are recorded in the upcoming days, and calculate accuracy of these models. The source code for the STLF model proposed in this paper is available at https://github.com/spiyush19/UG-Project/blob/main/UG_Project.ipynb.

Introduction

As the technology moves towards smart energy grids and smart microgrids, medium term load forecasting has become an even more significant problem that engineers face. Solutions to this problems can be of different categories:

1. Numerical Weather Prediction methods
2. Statistical or machine learning models
3. Or combination of these two types of methods

This problem of load forecasting is considered very important, because it is very crucial to schedule power plant maintenance and purchase fuel to meet load demands. Inaccurate predictions can lead to system failure, faults, large penalties and higher market clearing prices.

This report presents a load forecasting method that considers both dynamic factors in load pattern as well as effect of meteorological factors, ie. weather, on the electric load. The paper makes use of the demand data of Ontario, Canada and the weather data set is taken from Toronto City Centre weather station. The period of analysis is 2004-2019.

Data Preprocessing

Data used for our report required several features that include:

1. Time and date and day of data collected
2. Weather attributes like temperature, relative humidity.
3. Electric load demand (target variable)

These attributes were suspected to play an important role in determining the electric load. The actual significance of these attributes in determining the exact value of the electric load can be found out only after our distribution generates probabilistic weight for them. Some attributes play a more significant role in electric load forecasting, while others less.
The demand data was collected from IESO Canada data directory. The weather data was collected from the Government of Canada's Environment and Climate Change website. The data scraped by sending following query using command line terminal of MobaXterm:

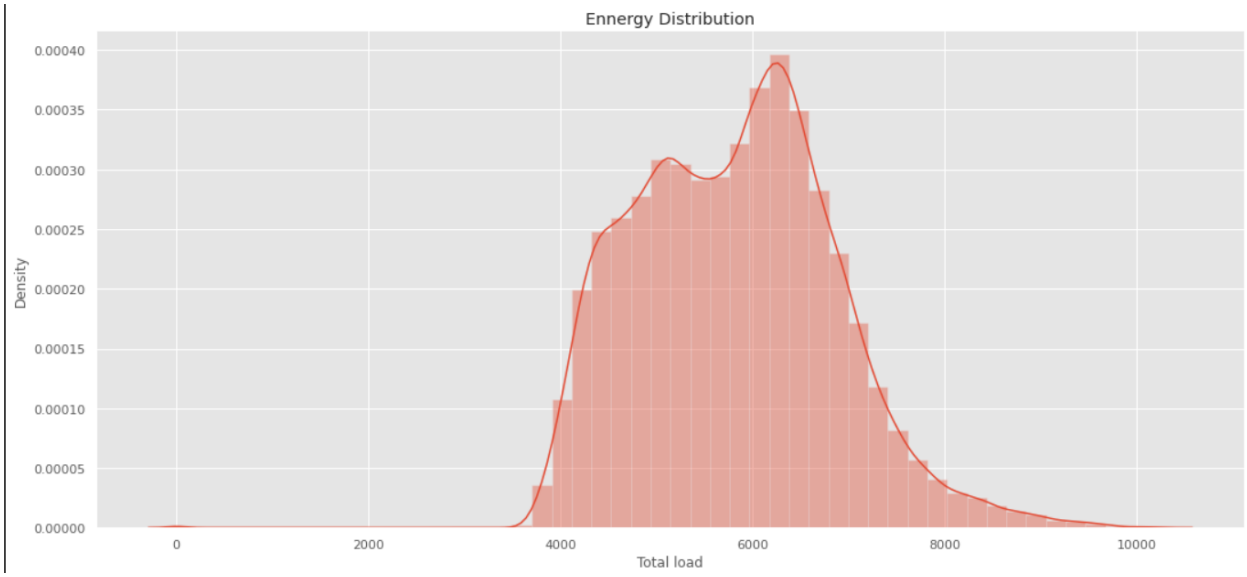
```
for year in `seq 2004 2019`;do for month in `seq 1 12`;do wget
--content-disposition
"https://climate.weather.gc.ca/climate_data/bulk_data_e.html?format=csv&sta
tionID=48549&Year=${year}&Month=${month}&Day=14&timeframe=1&submit=
Download+Data" ;done;done
```

This was the code used for merging the total demand data into one for merging it with the weather data.

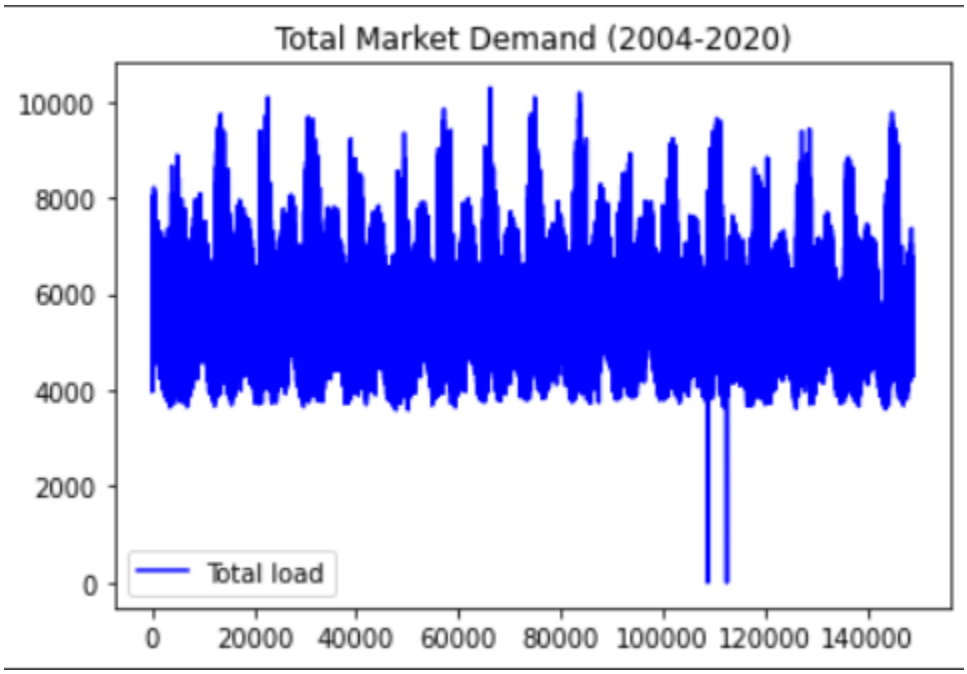
```
#combine all files in the list
df = []
dict = {}
directory = "/content/drive/My Drive/BTP/Demand"
for filename in os.listdir(directory):
  #print(filename)
  str1 = filename[16:20]
  #print(str1)
  dict[str1] = filename

filename = []
for i in range(2004,2021):
  s = str(i)
  path = directory + "/" + dict[s]
  filename.append(path)

#print(filename)
combined_csv = pd.concat([pd.read_csv(f) for f in filename])



#combined_csv = pd.concat([pd.read_csv(f) for f in filename])
#export to csv
combined_csv.to_csv("/content/drive/My Drive/BTP/Demand/combined_csv.csv", index=False)
```

Similarly,the data was merged into one for the weather data as well.
After that column of Toronto from the demand extracted according to the date-time and merged with the weather data and saved as one csv file.This file was further used for data processing.

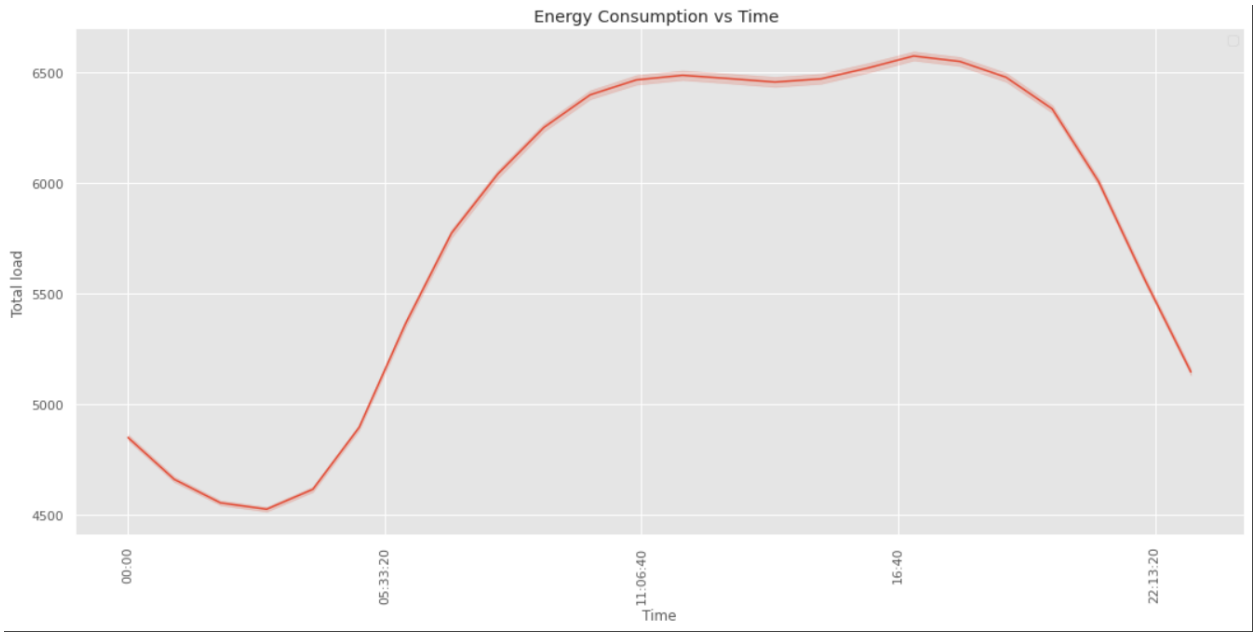
Here, we visualised our data with respect to various parameters to observe the trend and the dependency.



ENERGY DISTRIBUTION Vs TOTAL LOAD

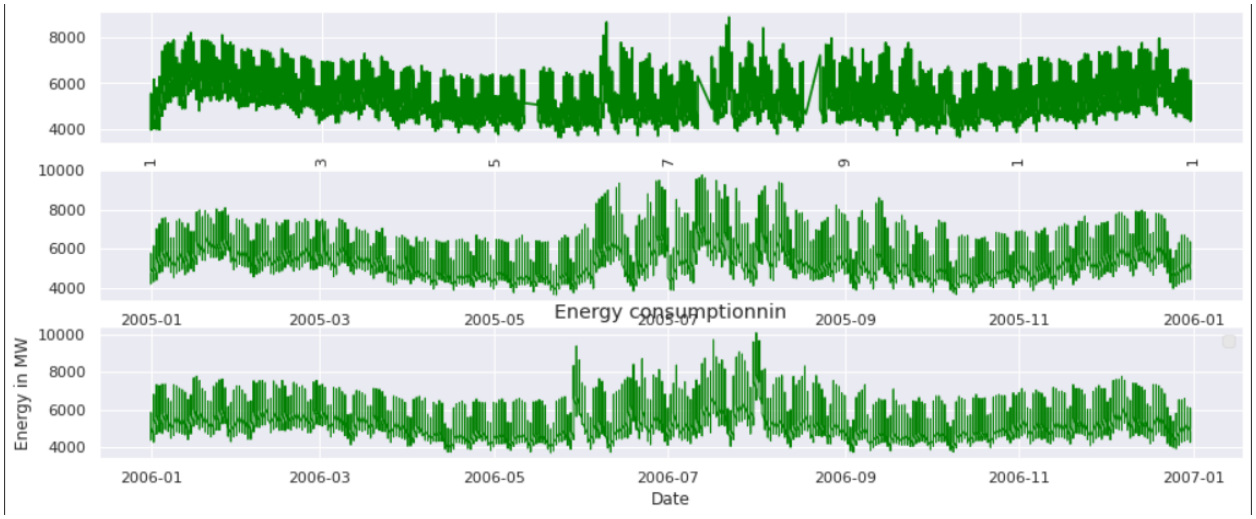


TOTAL MARKET DEMAND



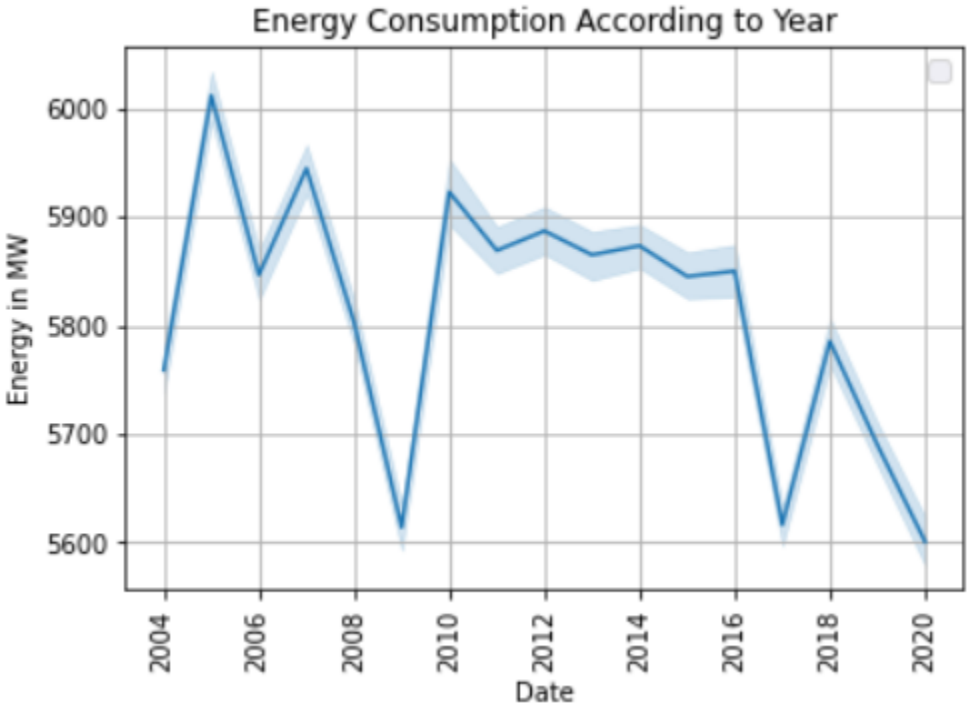
ENERGY CONSUMPTION Vs TIME (24-Hours Format)

Here,from the plot we can observe that the energy consumption is low as compared to other times of the day in the late night period.It is quite obvious because the loads are turned off at night.(i.e - Lights,Televisions,Laptops,etc).



ENERGY CONSUMPTION Vs MONTHS (In a particular year)

Here,from the plot we can see that energy consumption increases in the summer season.It is also quite obvious due the fact that many electric appliances operate at much load during these months.(i.e-Air Conditioners,Refrigerators,Fans,etc).



ENERGY CONSUMPTION Vs YEAR

Here,from the year-wise trends we can see that the load requirements tend to decrease via time (overall).The reason can be somewhat explained by the fact that smarter electrical appliances are evolving as the time passes.New technologies evolve and they require low energy consumption as compared to previous ones.For example - In the previous years (i.e 90's or the starting of 21st century) there were lots of 100 Watts electrical bulbs used in many houses.But now they are replaced by LED and smart lights which require energy in the range of 10-20 watts and lower also.

Data Cleaning

Collecting data can be tedious, and thus several conditions make it unfavourable to collect data accurately or collect data at all. This can be clearly seen in the dataset provided. The dataset contains lots of NaN values, which signify data is not available for that period of interest. Upon analysis, we found that NaN values were present only in the temperature and relative humidity columns.

```
[6]  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 140256 entries, 2004-01-01 00:00:00 to 2019-12-31 23:00:00
Data columns (total 4 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   weekday      140256 non-null  int64  
 1   load         140256 non-null  int64  
 2   temperature  139422 non-null  float64
 3   humd         139423 non-null  float64
dtypes: float64(2), int64(2)
memory usage: 5.4 MB
```

Assuming weather attributes change gradually, we used a backward fill method to fill the missing values. After this, all the data was non-null. But all these methods affect the accuracy of prediction of the model. For better prediction, collection of better and more complete data is beneficial.

Data Preparation

The *time* column, which contained the timestamp when the observation was taken, was set as the index of the dataframe. Next, we created some temporal features. These features are known to have an impact on electrical demand. The consumption pattern differs on a weekday from weekend. A *weekday* column was added which was 0 for Saturdays and Sundays, and 1 otherwise. Similarly, day of time also has an impact on the electrical demand pattern. Despite the fact that this is a time series problem, the model in this paper is decomposed into two sections. A direct feature must be implemented to capture the role of time in the second half. Because time is a recurrent variable, the hour variable is transformed using a sine/cosine transformation. The relative humidity, temperature and load are normalized.

Modelling Techniques

Time series modelling is a different task than a usual regression problem. This happens due to a variety of reasons:

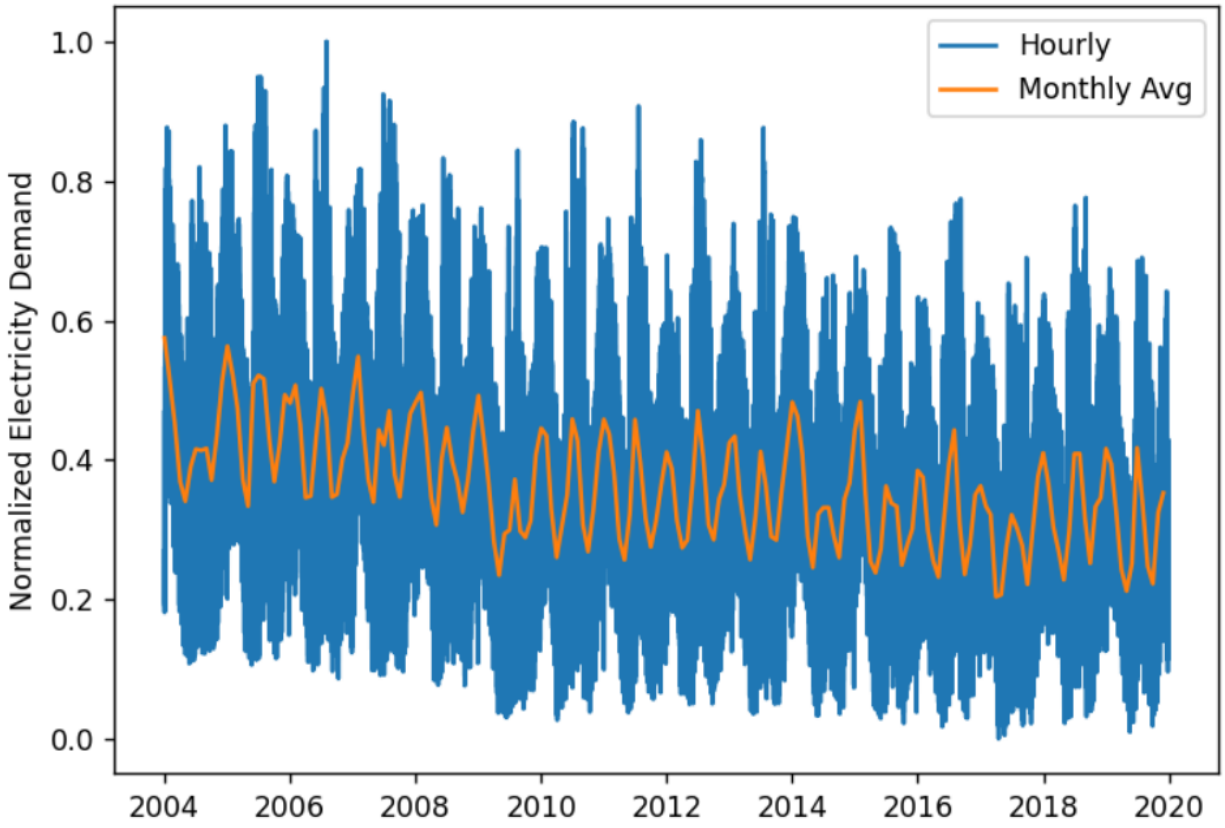
1. The lagged values which act as explanatory variables are not independent
2. The prediction problem is essentially extrapolation, the prediction intervals keep increasing or the confidence in prediction reduces especially for longer periods.
3. There is a need to isolate the inherent trend, cycles and seasonality to be able to uncover the underlying dynamics.

Among the multiple methods to forecast time series, ARIMA models, ARCH/GARCH models, VAR models are the most notable ones. With the emergence of better computation tools and more data, deep learning models for Sequential Data- RNN and its variations GRU and LSTM are very effective for forecasting time series.

One key approach in modelling the time series is by decomposing into simpler components. One of the methods is to disaggregate signal- trend, cycle or seasonal components, and noise.

We break our hourly data into a monthly average component and an hourly residual component. SARIMA is used to predict the monthly averages. The remaining component can be understood as hourly consumption above or below the average values. These hourly residuals are then predicted using three regression models and we compare and pick the best predictions.

In [11]:
```
monthly_norm = df_norm['load'].resample('MS').mean().bfill()
plt.plot(df_norm['load'], label='Hourly')
plt.plot(monthly_norm, label='Monthly Avg')
plt.ylabel('Normalized Electricity Demand')
plt.legend();
```



The above snippet shows calculation of monthly averages. The figure is a plot of hourly and monthly average for the entire data. As visible, there is a high yearly seasonal trend in the data. Data from 2004 to 2019 has been used. We dropped the year 2020 due to extreme changes in the consumption pattern.

Following train-validation-test split has been chosen:

1. Training Set- Year 2004 to 2016 (both inclusive)
2. Validation Set- Year 2017 and 2018
3. Test Set - Year 2019

A random sampling method has not been used because the time series data loses meaning in such cases.

SARIMA: Monthly Means Model

An ARIMA model has the following components:

1. Autoregressive (AR)- weighted sums of last ‘p’ terms

2. Moving Averages (MA)- weighted sums of last ‘q’ errors
3. Integrated of order ‘d’, if a series is differenced d times to make it stationary
4. Constant

Such a model is called ARIMA(p, d, q). In a SARIMA model, all these terms operate across a lag length ‘m’, denoting the number of periods in a season. More specifically, an SARIMA model is ARIMA(p, d, q), (P, D, Q) where the letters P, D, Q are seasonal counterparts of the original ARIMA model.
Since, we identify an yearly trend from the above plots, we conclude m = 12 for our monthly averages.

In statistics, Autocorrelation and Partial Autocorrelation plots are used to identify the parameters p, d, q, P, D, Q. However, we employ a grid search for these parameters. Grid search is used to improve a particular metric by scanning the data and building models with various hyperparameter combinations.
We have chosen Mean Absolute Error (MAE) instead of Mean Absolute Percentage Error (MAPE) due to normalization steps taken during preprocessing of data. As the load values (output, y) are scaled to lie between 0 and 1, MAPE can assume very large values.
MAE can be calculated as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y_i} - y_i|$$

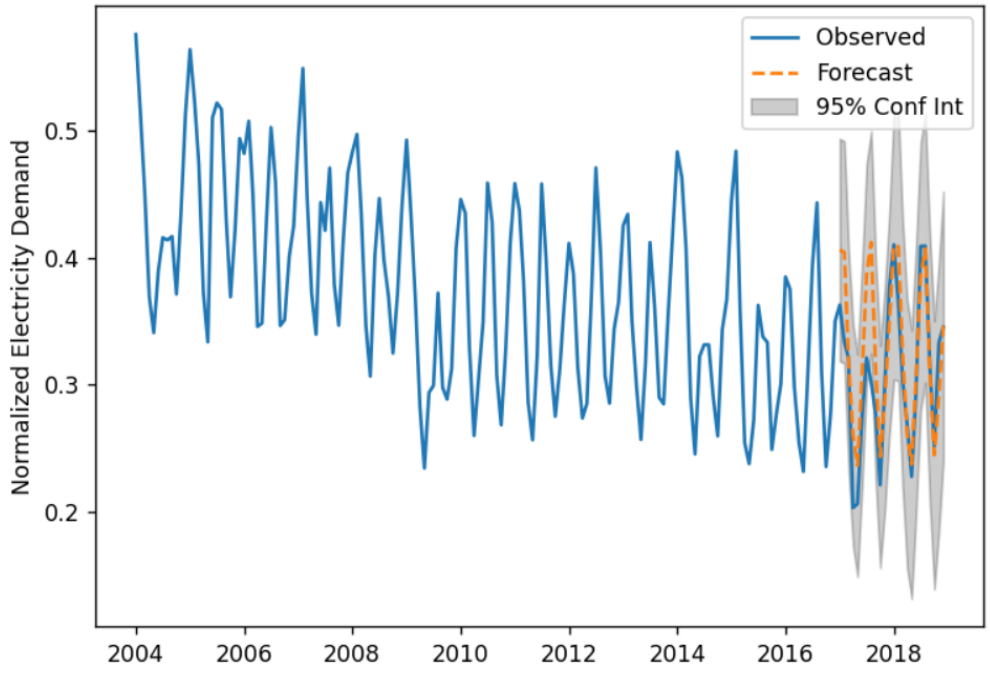
As we lose information in differencing the series, the following rules of thumb were kept in handy while evaluating the grid search results:

1. More than one order of seasonal differencing must not be used, i.e. D<=1.
2. In case, the seasonal trends are very strong, the total order of differencing (d + D) must have a maximum value of 2.

Out[13]:

	p	d	q	P	D	Q	mae
7	0.0	0.0	0.0	1.0	1.0	1.0	0.02828

The final SARIMA model for prediction is trained using the above parameters. The forecasted values on validation set approximate the actual values(below). Also, MAE is very low for the best model obtained by grid search.



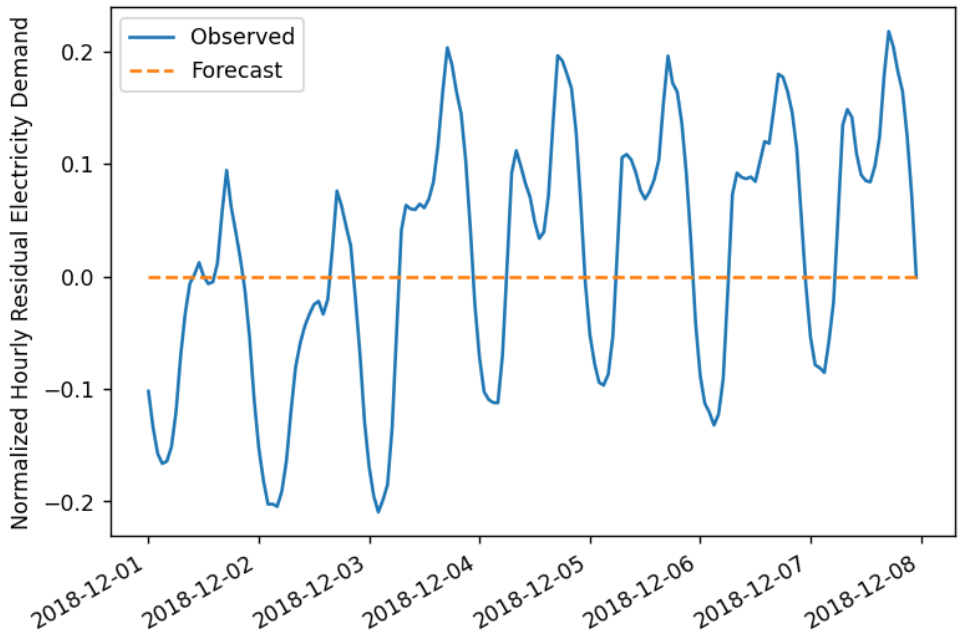
Regression: Hourly Residual model

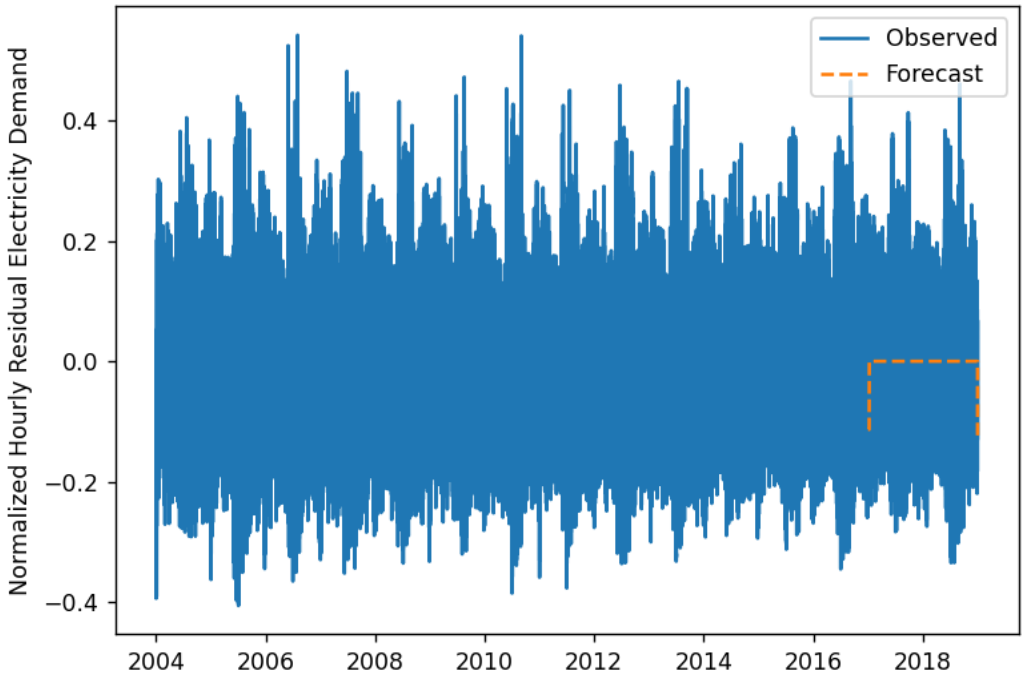
Hourly residuals are calculated from the original time series by subtracting monthly mean from it. Next, in order to make best predictions on hourly residuals, we deploy three different regression models. ARIMA model can also be used but that was found to give poorer results and it was also slow to train. The three regression models used are:

1. Linear regression
2. Gradient boosting regression (GBR)
3. Multi-layer perceptron (MLP) regression

Baseline

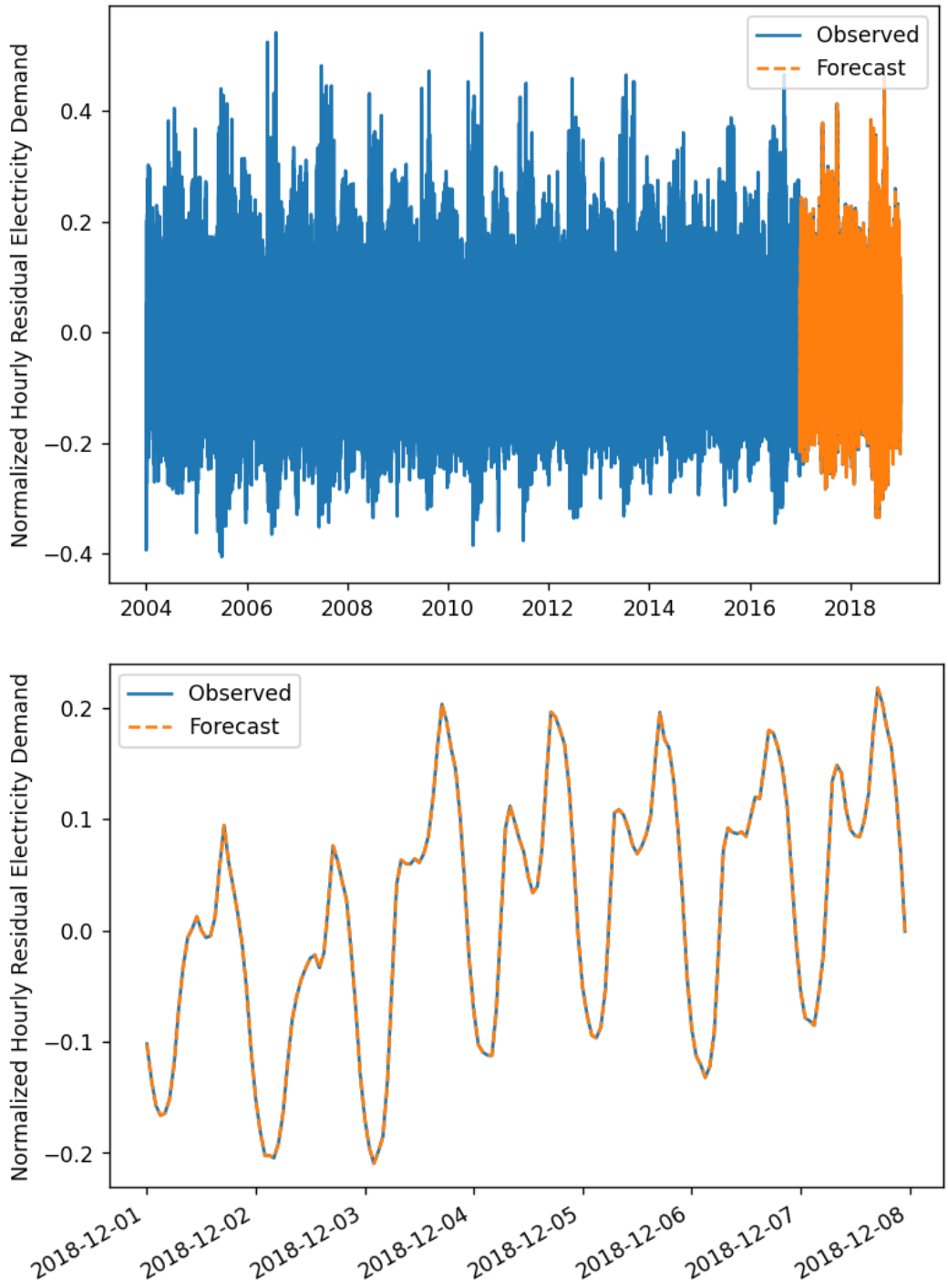
Our model needs a baseline to compare with. For sake of our model, we use values from previous week as the baseline for next week to predict hourly residuals.





Linear regression

In our first attempt we apply the simplest regression model, the Linear regression to our dataset. Our model without regularization has no hyperparameters. Following results have been recorded. We observe that it performs better than the baseline model, but the quality of prediction needs to be improved.

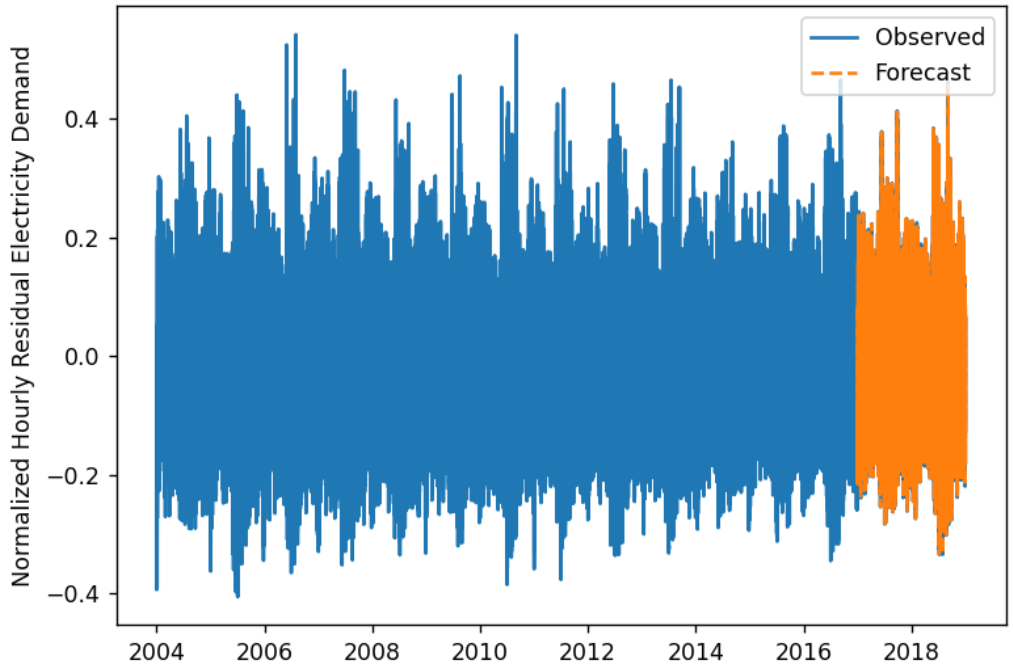


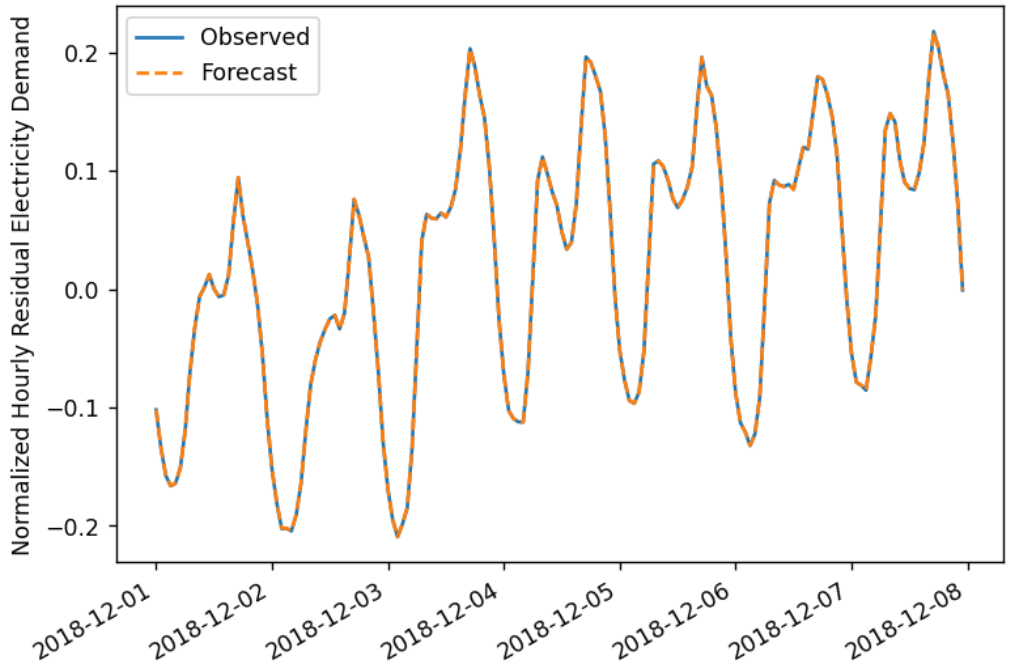
Gradient Boosted Regression

The GBR model has hyperparameters learning rate, max depth, that can be tuned to make accurate predictions. Following values were used for number of estimators(ne), learning rate (lr), max depth (md), mean absolute error (mae).

	lr	ne	md	mae
7	0.01	1000.0	3.0	0.000157

After which final model is trained whose results are as follows.





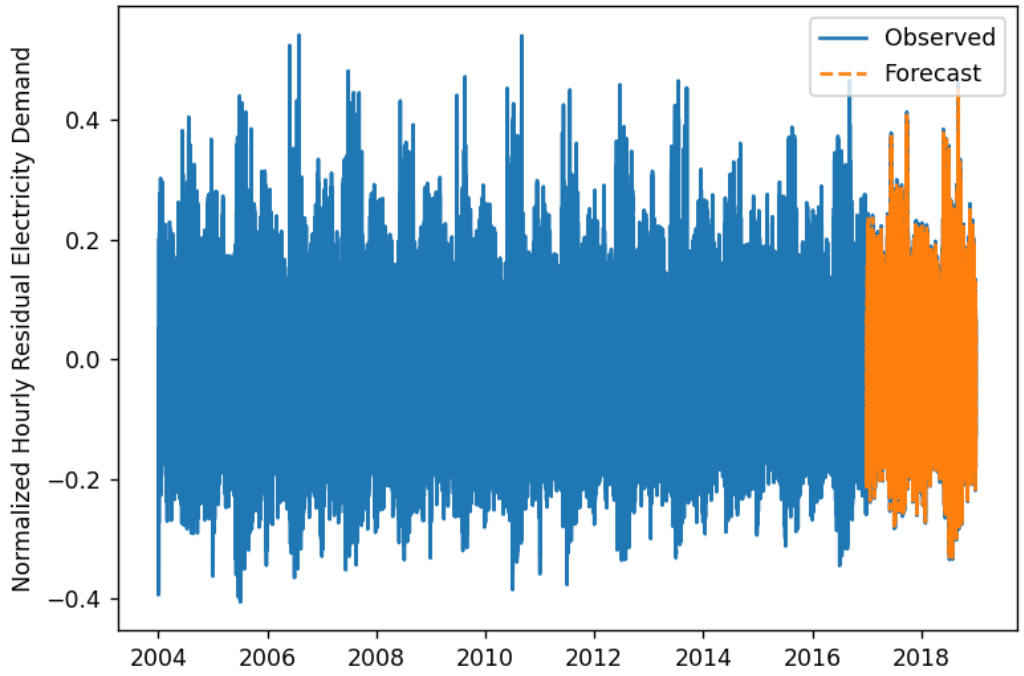
This model performs better than baseline as well as Linear regression and captures the peaks and valleys accurately.

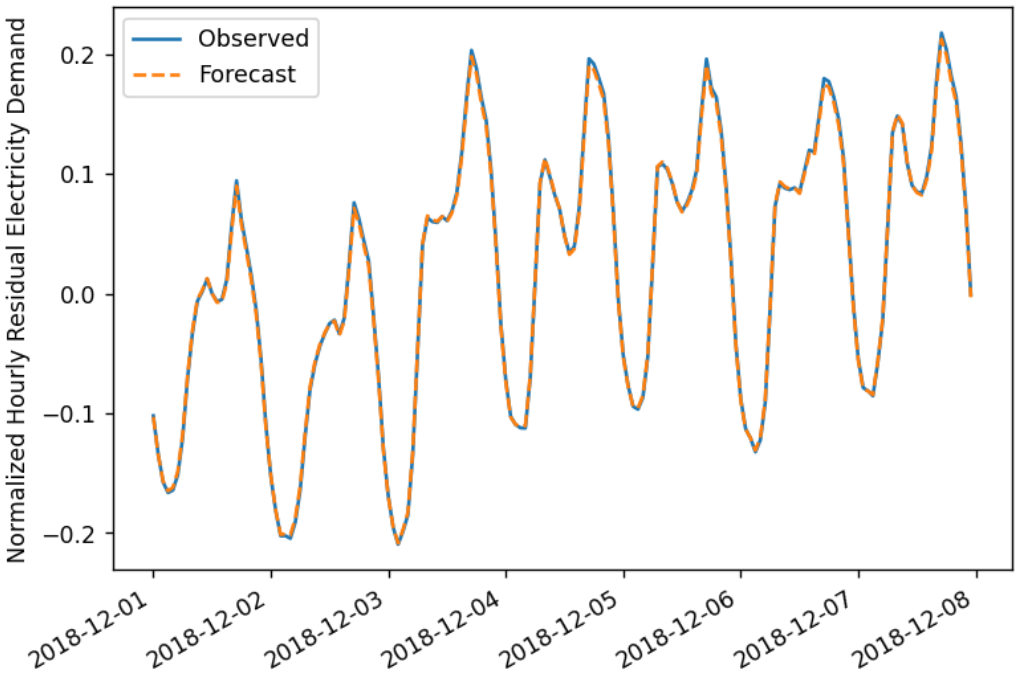
Multi-layer perceptron (MLP) regression

Lastly we have used a general type of neural network, the MPL model. This model has a higher number of hyperparameters compared to GBR such as the number of hidden layers (hl), number of neurons per layer, regularization (a), learning rate (lr), and maximum number of training iterations (mi).

	hl	a	lr	mi	mae
53	(500,)	0.001	0.01	10000	0.000296

Final model was trained based on these values of hyper parameters, Mean Absolute Error was achieved was `0.0023`

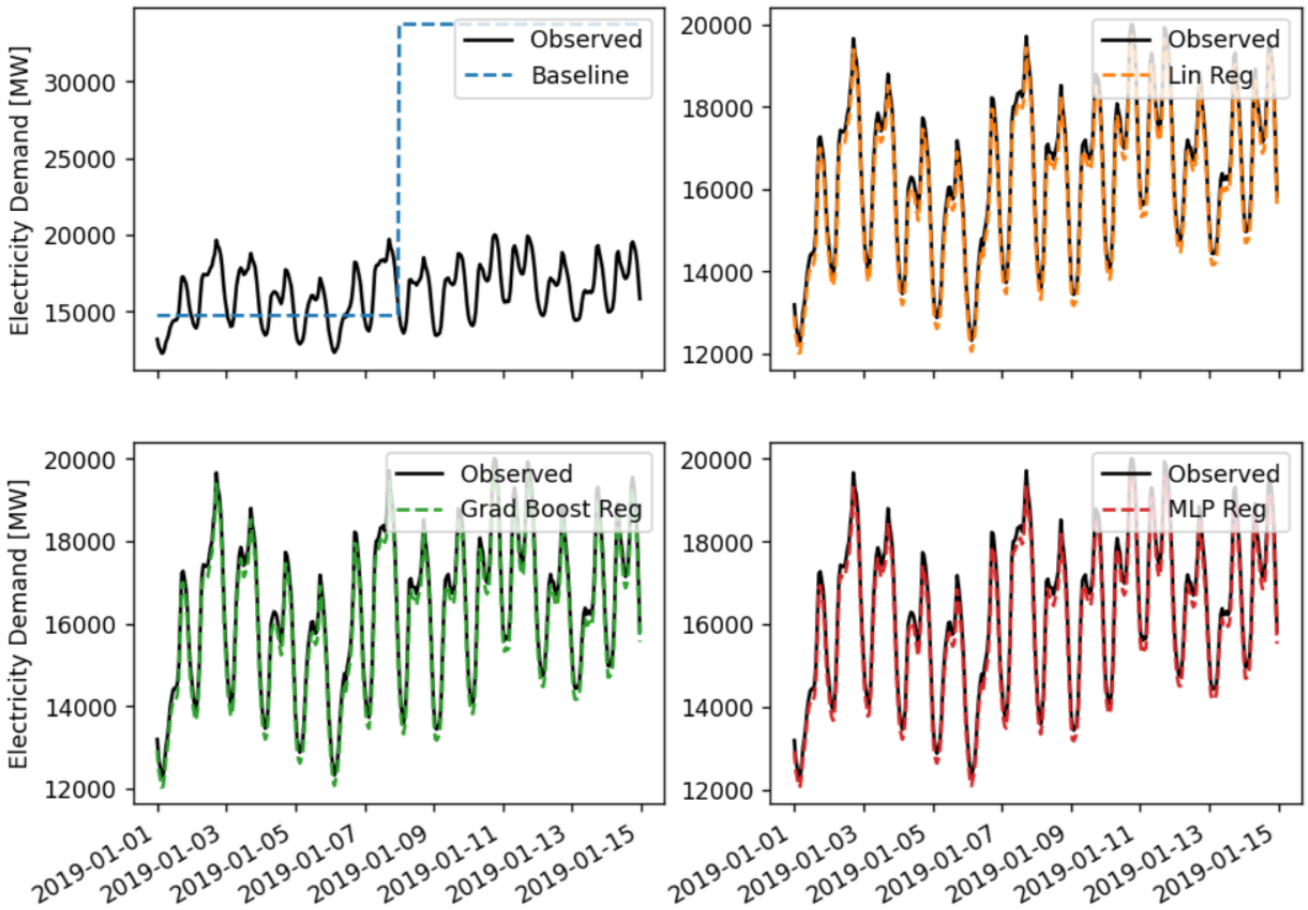




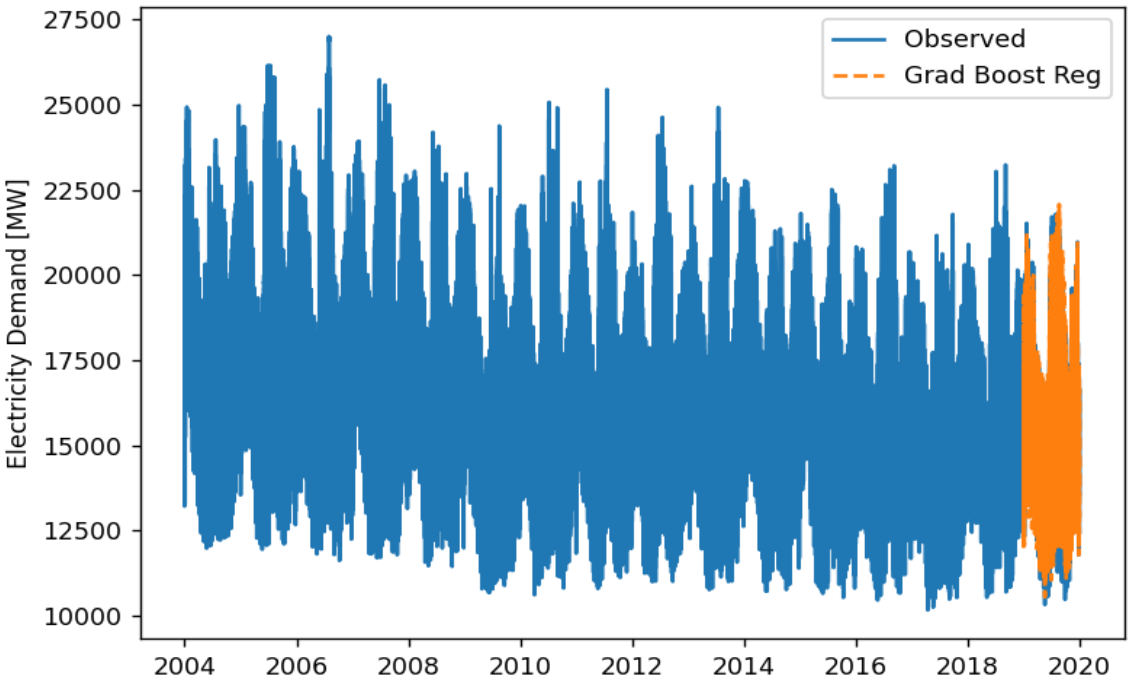
Evaluation of final model

To obtain final prediction for electric load we combined individual prediction of monthly average load and hourly residual. Firstly models are individually tested on the test dataset, i.e. Data of 2017. Next, both predictions are unnormalised to give actual values of load forecast. Finally both are added to give actual load forecasts.

Following are the results of trained different models against baseline model:



We find Gradient Boost Regression performs best on hourly data hence its results are combined with results of ARIMA for monthly average predictions to obtain overall prediction of future electric load. Load forecasting has been plotted along with actual values of electric load for the same period of time for comparison.



Results

The Data provided was processed, prepared and necessary parameters were extracted. Data was regularized and missing data was taken care of. Currently we trained our model based on two weather parameters for training and testing. However, we are working on preparing our model for handling various other parameters as well.

Here are the MAPE values obtained after validation using different models.

Baseline MAPE :- *1.12140159902467*
Linear Regression MAPE :- *0.026931833498880405*
Grad boost reg MAPE :- *0.026928717442635478*
MLP Reg MAPE :- *0.0270034683954952*

Therefore order of accuracy :-

GBR > LINEAR REG. > MLP

The best MAPE value was obtained in the case of **Gradient Boost Regression** .

But theoretically the order should be MLP > GBR > LINEAR REG, because MLP uses more hyperparameters than GBR, while Linear Regression uses none.

Reasons for not obtaining the theoretical result:

a) It can be seen that the gridsearch size of MLP was less ,which may result in low accuracy.
b) Due to the seasonal trend in data , linear regression overfits.
c) Maybe the reason can be the large dataset that we have,which allows the model to observe the pattern very much and predict it easily.

Conclusion

Time series forecasting gets more difficult when we try to predict load for a shorter period of time. We solved this problem by dividing the prediction into two components “Monthly average” and “Hourly residuals” and finally adding them for final prediction. As a result, the prediction is much more accurate than predicting load as a whole.
These forecasts prove very helpful to reduce average annual loss of power distributors because of oversupply or under-supply.

Improving the accuracy of load forecasting for the power grid is the biggest challenge in this process. The data provided from official sources contains several unavailable entries and also other inaccurately measured values as discussed above. If the data collection is better, the models work very well with satisfactory accuracy.

The model uses most obvious features for future load forecasts such as: month, time of day, temperature and relative humidity. Also study of those points can be very helpful, where the model is predicting with least accuracy. With help of that, we can try to look for additional features, inclusion of which can improve the model even further.

With the increase of smart meters in power distribution systems, highly efficient deep neural networks can be utilized for more accurate load forecasting.

References

1. Power Data
2. Historical Data - Climate - Environment and Climate Change Canada
3. Forecasting Electric Load by Aggregating Meteorological and History-based Deep Learning Modules
4. Time Series Forecasting with a SARIMA Model | by Andrew Graves
5. General seasonal ARIMA models -- (0,1,1)x(0,1,1) etc. (duke.edu)