Electrical Load Consumption Forecasting using Machine Learning

Md. Alamgir Hossain (1806186), Md. Meherab Hossain (1806182), Fahim Shahriar Talukder (1806193), Md. Rokonuzzaman Rokon (1806180), Asif Quadir (1806187)

***Abstract* - Load forecasting is a technique used by power companies to predict the power or energy needed to balance the supply and load demand at all the times. It is mandatory for proper functioning of electrical industry. It can be classified in terms of time like short-term (a few hours), medium-term (a few weeks up to a year) or long-term (over a year). In this paper, for medium- and long-term forecasting end use and econometric approach is used. Whereas for short-term forecasting various approaches are used like regression models, time series, neural networks, statistical learning algorithms and fuzzy logic.**

I. INTRODUCTION

Electrical energy cannot be stored. It must be generated whenever there is a demand for it. It is, therefore, imperative for the electric power utilities that the load on their systems should be estimated in advance. This estimation of load in advance is commonly known as load forecasting. It is necessary for power system planning. Power system expansion planning starts with a forecast of anticipated future load requirements. There is a growing tendency towards unbundling the electricity system. This is continually confronting the different sectors of the industry (generation, transmission, and distribution) with increasing demand on planning management and operations of the network [1]. The operation and planning of a power utility company requires an adequate model for electric power load forecasting. Load forecasting succor an electric utility to make conclusions regarding decision on generating and purchase electric power, load switching, voltage control, network reconfiguration and infrastructure development. Electric load forecasting is the process used for forecasting future electric load, given historical load and weather information and current and forecasted weather information. In the past few decades, several models have been developed to forecast electric load more accurately. With an introduction of deregulation of power industry, many new challenges have been encountered by the participants of the electricity market. Forecasting of wind power, electric loads and energy price have become a major issue in power systems [2]. Following needs of the market, various techniques are used to forecast the wind power, energy price and power demand. The market risk related to trading is considerable due to extreme volatility of electricity prices [3]. Considering the uncertain nature of future prices in competitive electricity markets, price forecasts are used by market participants in their operation planning activities. The objective of this paper is to provide a brief overview of various forecasting problems and techniques in power systems. Depending on the time zone of planning strategies the load forecasting can be divided into following three categories namely:

1. Short term load forecasting: this forecasting method is usually having period ranging from one hour to one week. It can guide us to approximate load flow and to make decisions that can intercept overloading. Short term forecasting is used to provide obligatory information for the system management of daily operations and unit commitment [1].
2. Medium term load forecasting: this forecasting method has its period ranging from one week to one year. The forecasts for different time horizons are important for different operations within a utility company [1].

Medium term forecasting is used for the purpose of scheduling fuel supplies and unit management.

3. Long term load forecasting: this forecasting method has its period which is longer than a year. It is used to supply electric utility company management with précised prediction of future needs for expansion, equipment purchases or staff hiring [1].

II. METHODOLOGY

A. Dataset Collection

We collected Dataset from a Kaggle Competition. Title of the dataset was "Hourly Energy Demand Generation and Weather". It originally contained two individual datasets.

1. Energy dataset: This dataset contains hourly electricity generation, demand, and pricing information for years 2015 to 2018 of Spain (total 28 columns).
2. Weather dataset: This dataset contains hourly weather information like Temperature, Humidity, Windspeed, atmospheric pressure of five major cities of Spain.

Five major cities were taken from five different regions to represent weather information of the whole country [4]. These cities are shown below:

Figure 01: Map of Spain with cities

B. Dataset Preparation

Few steps were followed to prepare the dataset for modeling

Step 1: We removed all the duplicate data from the dataset

Step 2: We checked for outliers and removed them.

Step 3: We removed qualitative data from the dataset.

Step 4: We merged the two-dataset basis on their hourly timestamp

Step 5: After merging, we removed the incomplete rows from the merged dataset.

Finally, we had a dataset of size 35064×89.

C. Feature Selection

As we had too many columns in the dataset we got in the previous step, we were required to choose useful columns to make a useful model. At first, we selected total load actual as our target column as this is the parameter we intend to forecast in our project. Then we select two kinds of columns as input to our model. These are:

Time Information: Day, Month, Year and Hour. (4 columns)

Weather Information: Temperature (K), atmospheric pressure (hPa), humidity (%) and wind speed (m/s) for each of the five cities. Total 4×5 = 20 columns.

Thus, finally we had 4+20 = 24 columns altogether as input columns and one as target.

D. Checking Outliers and Correlations

Before starting modeling, we again checked for outliers and missing data. However, no outliers or missing data was found. Below graph shows the result for outliers checking for the temperature column of Barcelona city. The horizontal axis shows the data index, and the vertical axis shows temperature in K (Kelvin).

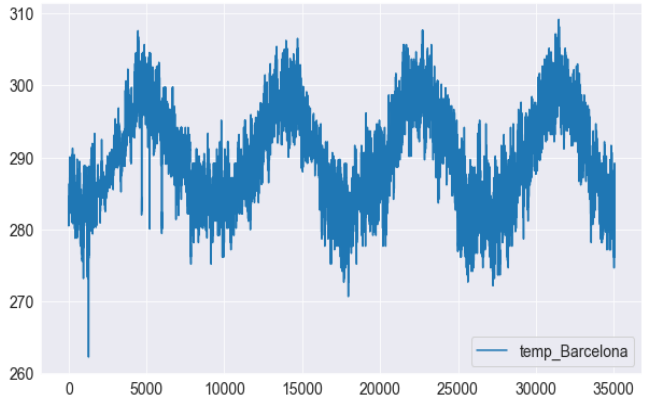

Figure 02: Demo of Checking outliers from a single column

As we have selected only four parameters for each city to create our model, we investigate the correlations between our targeted parameter and input parameters before modeling.

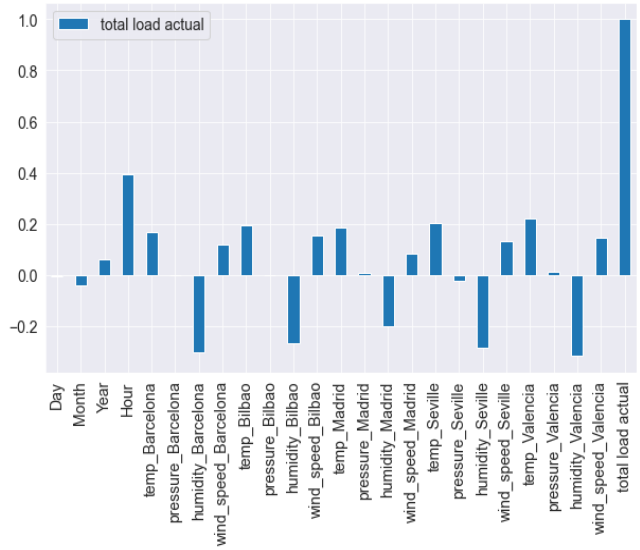

Figure 03: Calculating correlation of different columns with the target column

Here, we can see load demand has quite strong positive correlation with temperature, wind speed and hour of the day. Also, it has quite strong negative correlation with the humidity of air. However, correlation with the atmospheric pressure is very low.

E. Dataset Splitting

For creating a proper machine learning model and testing it we divided the data into 3 sets. The data from year 2015 to 2017 (26305 data) were used as Train dataset which was used to train the model. The data of first 6 months of 2018 (4415 data) were used as a Validation set. We used the dataset to optimize the model. Lastly the last six months of data of 2018 (4344 data) were used to as Test set which was used to find accuracy of the model

F. Machine Learning Algorithm

The machine learning algorithm we've used is called Extreme Gradient Boosting Regressor (XGBoost Regressor). The algorithm uses sequential decision trees to create the model [5].

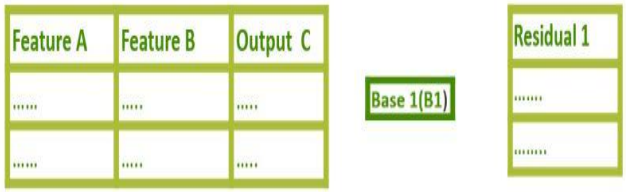

Figure 04: Working of model fig(1)

To understand the model, we can think of a data with 2 features and one output. The base value B1 is the average of output C which is the basic predicted value. From the predicted value and the actual output, a new table can be created which contains the individual error or residual values. These residual values go through a proper decision tree and creates an output table.

Figure 05: Working of model fig (2)

From that table using proper learning rate new predicted value can be determined using the equation Output 2 (B2) = B1 + α1*T1.

Figure 06: Working of model fig (3)

Again, comparing the prediction value with the actual output value new error table can be created which will be sent to a different decision tree and a new tree output will be determined. Using that output and selecting a new learning rate new predicted value can be determined using Output 3(B3) = B2 + α2*T2.

The process will go on. The residual value will keep decreasing every time and the model will get better. When the error values are minimized, the model will be trained [6].

When predicting test data or validation data directly from the features all the features of the data will go through same decision trees and will create similar outputs. Using the learning rates as before altogether the new output values can be predicted.

Output = B1 + α1*T1 + α2*T2 + …….. αn*Tn

G. Training Model and Optimization

G.1. Training

The training dataset is used to create the model. The data goes through XGBoost Regressor algorithm and will create a model. Then the validation dataset was used to get new predicted values. Then the predicted values were compared with the actual values. According to the error the model was optimized later.

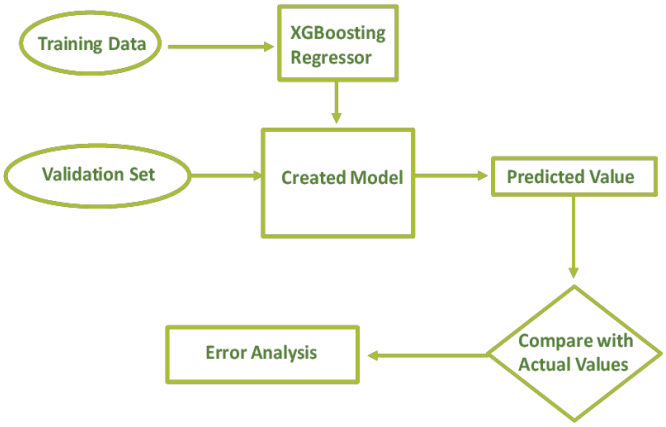

Figure 07: Demonstration of How the Modeling works

G.2. Optimization

Every machine learning model has some hyperparameters that are used in many ways changing the parameters will change the way output is produced. For optimizing a model for a proper data, the parameters need to be changed in a way that model doesn't overfit or underfit the data.

In this model the same thing was followed. At first the model had some overfitting problems. After changing the hyperparameters both manually and in a program new value of hyperparameters were selected for which the model was optimized.

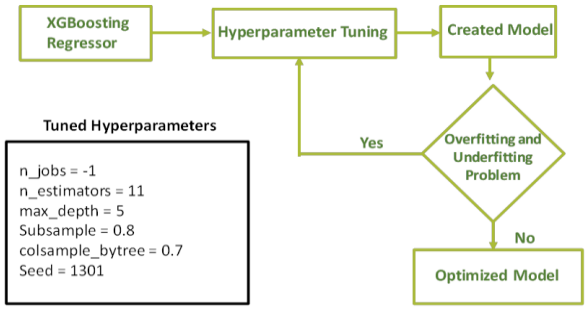

Figure 08: Demonstration of Optimization and hyperperameters

The hyperparameters in the figure above were used in our model.

n_jobs = -1
n_estimators = 11
max_depth = 5
Subsample = 0.8
colsample_bytree = 0.7
Seed = 1301

H. Checking Performance with Validation Set and Test Set:

After the model was optimized, we predicted new output values for both Validation and Test dataset. Then we compared with the actual values.

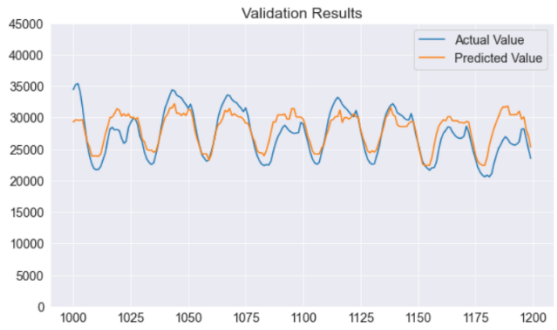

Figure 09: Performance of the validation set

Mean absolute deviation of the validation data was 8.29%.

We also checked for the test dataset.

Figure 10: Performance of the test set

Mean absolute deviation of the test data was8.37%.

I. Application Architecture

We have created a web application to provide live demonstration of our project. The web application interacts with the user by taking user inputs. Hence provides the desired load forecasted output to the user.

Like any other web application our application consists of two parts namely a backend and a frontend.

1. Backend:

We have used Heroku service as our web server. Afterwards, already trained model is deployed to the web server. Then we created a backend using python's flask framework and connected the model with the server using this backend. To communicate with the frontend, we created a rest api that takes request from the frontend and responses back with prediction using the server [7].

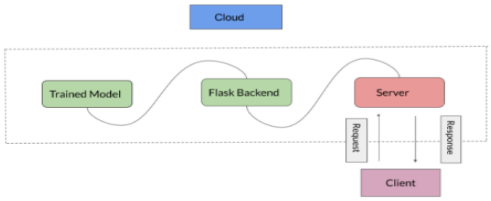

Figure 11: Architecture of the Backend Service

2. Frontend

The frontend is used to interact with the user. Basically, it takes input from the user and shows the output. The frontend is created using the technologies like React JS, html, JavaScript, and CSS. It is deployed to an online hosting platform named Vercel. The workflow is simple. Firstly, the frontend receives necessary inputs from the user for prediction. Then the frontend makes a post request to server via REST API service. After receiving the post request, the server responses with the prediction. Hence, the prediction is shown in the frontend as the desired output [8].

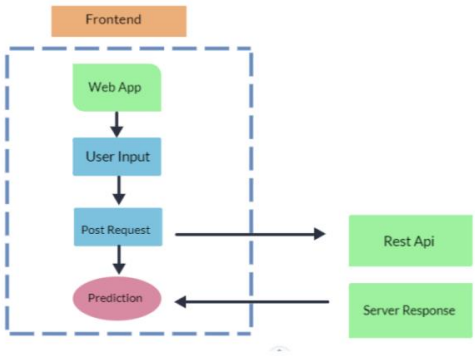

Figure 12: Workflow of the Frontend

J. Discussion

The model was created with the data of 2015 to 2017. We predicted the results for the year 2018. As we can

see in the graph almost all the cases where there is error the model predicted the load to be less than the actual value. Due to the increase of electricity usage the load has been increased significantly. Therefore, the model struggles a little bit to predict the actual result.

III. CONCLUSION

In this project we tried to forecast electrical load with machine learning techniques. Machine learning techniques work well when there is a lot of data available. Our project also suffered to predict recent load demand due to unavailability of recent data. Recent data shows the trend of load demand and so helps machine learning model to learn the trend. So, as an improvement we can add recent data to our collection. Hence our model can perform better in the future.

REFERENCES

[1] B. Sharma, T. Anwar, K. Chakraborty, and H. Sirohia, "Introduction to Load Forecasting," *Artic. Int. J. Pure Appl. Math.*, vol. 119, no. 15, 2018.

[2] C. Kuster, Y. Rezgui, and M. Mourshed, "Electrical load forecasting models: A critical systematic review," *Sustainable Cities and Society*, vol. 35. 2017. doi: 10.1016/j.scs.2017.08.009.

[3] M. A. Hammad, B. Jereb, B. Rosi, and D. Dragan, "Methods and Models for Electric Load Forecasting: A Comprehensive Review," *Logist. Sustain. Transp.*, vol. 11, no. 1, 2020, doi: 10.2478/jlst-2020-0004.

[4] D. Rolnick *et al.*, "Tackling Climate Change with Machine Learning," *ACM Computing Surveys*, vol. 55, no. 2. 2023. doi: 10.1145/3485128.

[5] S. Liang, "Comparative Analysis of SVM, XGBoost and Neural Network on Hate Speech Classification," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 5, 2021, doi: 10.29207/resti.v5i5.3506.

[6] M. Chen, Q. Liu, S. Chen, Y. Liu, C. H. Zhang, and R. Liu, "XGBoost-Based Algorithm Interpretation and Application on Post-Fault Transient Stability Status Prediction of Power System," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2893448.

[7] S. P. Ong *et al.*, "The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles," *Comput. Mater. Sci.*, vol. 97, 2015, doi: 10.1016/j.commatsci.2014.10.037.

[8] H. M. Abdullah and A. M. Zeki, "Frontend and backend web technologies in social networking sites: Facebook as an example," 2014. doi: 10.1109/ACSAT.2014.22.