ANALYSIS OF PEAK ENERGY DEMAND WITH ENVIRONMENTAL VARIABLES USING THE BANE-16 DATASET

BY JAIYEON CHOI

1. Introduction.

During hot summer season, people sometimes experience the power outage since the maximum generation of power cannot catch up the total demands increasing. The project aims for capturing how each of climate factors impacts total energy demands and is expected to be good reference for the insight of facilitating sustainable energy supply plan. Since the project are focusing on energy demand of warmer season, the project looks the example of Bangladesh, where hot or warmer season continues for all year.

2. Data.

2.1. *Dataset.*

The BanE-16 dataset (Salehin et al., 2023) used for this project is sourced from the Bangladesh Power Development Board (DPDB) and Daffodil International University spanning January 2018 to April 2023.

The list of the variables in the dataset is given below:

Name of Data Column	**Description**
Day	The Day in which data were recorded.
Month	The month in which data were recorded.
Year	The year in which data were recorded.
Temp2(c)	Represents temperature at a certain location measured by a sensor in Celsius.
Temp2_max(c)	Highest temperature recorded during a specific time period in Celsius.
Temp2_min(c)	Lowest temperature recorded during a specific time period in Celsius.
Temp2_ave(c)	Average temperature recorded during a specific time period in Celsius.
Surface_pressure(pa)	Atmospheric pressure measured at the surface level.
Wind_speed50_max(m/s)	Maximum wind speed measured at 50 meters above the ground.
Wind_speed50_min(m/s)	Minimum wind speed measured at 50 meters above the ground.
Wind_speed50_ave(m/s)	Average wind speed measured at 50 meters above the ground.
Prectocorr	A measurement related to precipitation correlation.
Total_demand(mw)	Total electrical demand measured in megawatts.
Max_generation(mw)	Maximum power generation capacity recorded in megawatts.

TABLE 1
variable list

2.2. *Exploratory Data Analysis.*

For initial checkup, we plotted the histograms for all variables. Since most of models implemented in this project assume normality of the input values, it is desirable that each of histograms is normally distributed. Because the date variables, day, month, and year are not able to be normally distributed for the data collected everyday, we looked into other variables before moving on to data pre-processing. Wind speed variables and prectotcorr showed right-skewed pattern while temperature variables except Temp2_max(c) showed left-skewed pattern. In order to obtain elaborated models, we also have to check the presence of multicollinearity between different explanatory variables. As we can see, **temp2_c (Temp2(c) in Table 1)**, **temp2_min_c (Temp2_min(c) in Table 1)**, **wind_speed50_ave.m.s (Wind_speed50_ave(m/s) in Table 1)**, **wind_speed50_min.m.s (Wind_speed50_min(m/s) in Table 1)** showed overly high correlation(≥ 0.9) with other variables.

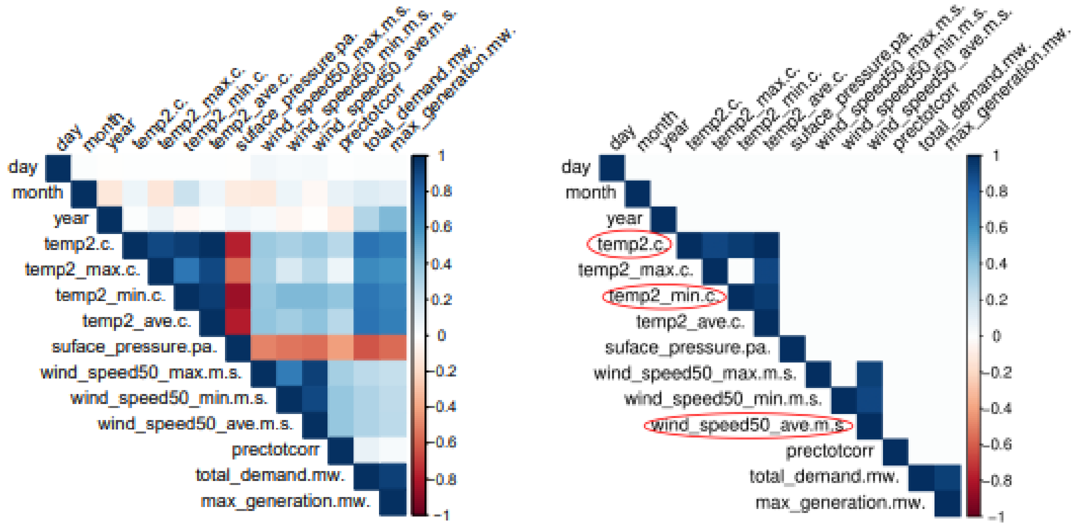

Fig 1: Overall Correlation Heatmap(left) and Large Correlation(≥ 0.9) Heatmap(right)

3. Method.

3.1. *Data preprocessing.*

For data imputation, we applied spline interpolation to each of columns. We were not ready to use time series model at this point, but since all data was collected in time-order, to keep the consistency for data collected everyday, we implemented interpolation with timely ordered values using the spline interpolation method in R, which is *na.interpolation* with spline option to fill missing values in each of columns.

In order to make the data normally distributed, we applied power transformation to each of data columns except time variables. Optimizing algorithm to find the transformed data having the minimum absolute skewness was also implemented during the power transformation process. Two histograms below show how a data column of non-zero skewness changed to normal-like distribution.

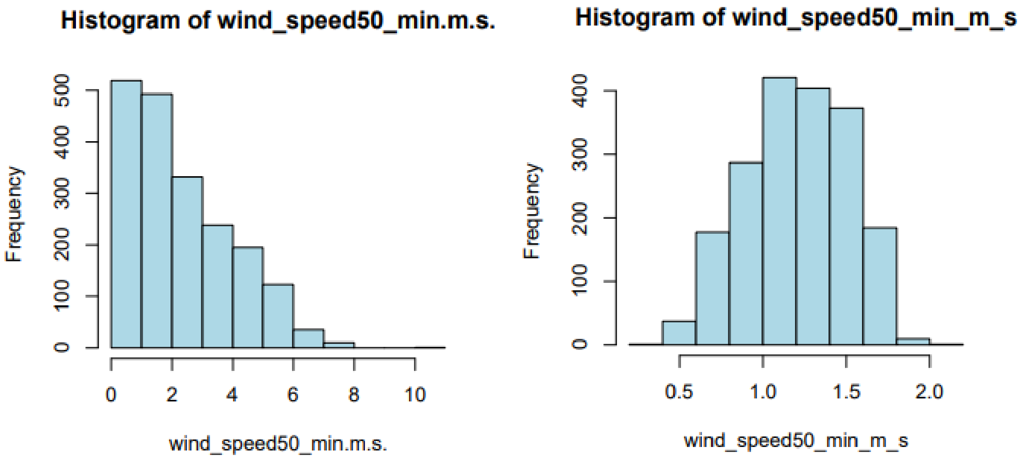

Fig 2: A histogram from original data(left) vs transformed histogram(right)

Before removing multicollinearity between different explanatory variables, We had to decide to leave minimum wind speed variable or average wind speed variable. Since **average wind speed variable(Wind_speed50_ave(m/s) in Table 1)** showed higher correlation with **total_demand(mw)**, we decided to drop **minimum wind speed variable (Wind_speed50_min(m/s) in Table 1)**. The variables removed are circled in Figure 1.

Since time variables are not able to be transformed, we trained models with two different pre-processed datasets - the dataset with date variables and the dataset without date variables.

Before training process, all the data columns were normalized with column mean(μ) and column standard deviation(σ) respectively.

Let normalized input for original input x be $\tilde{x}$,

$$\tilde{x} = \frac{x - \mu}{\sigma}$$

3.2. *Model training.*

Each of datasets is split into 10% for testing and 90% for training.

In this project, 10 models are trained with two different datasets.

- Simple Linear Regression
- Weighted Least Squares
- K Nearest Neighbours(with different Ks(5, 21, 51))
- Generalized Additive Model B-spline applied(GAM)
- Ridge Regression
- LASSO Regression
- Projection Pursuit
- Basic multi-layer Neural Network(Werbos, 2021)
- Elastic Net model
- Elastic Net model with XGBoost(Chen and Guestrin, 2016)

Only one model listed below is trained with the dataset ordered in the time sequence. Since we use the time series model here, we removed all date variables and input the dataset in the time sequence.

- Multivariate time series model by Long Short-Term Memory(LSTM)(Hochreiter and Schmidhuber, 1997)

The reason why we selected LSTM for multivariate time series model is that LSTM model performs well for long term period(356 days).

We compared 10 trained models and select the best model among the non-times series model and then compared the best non-time series model to the LSTM multivariate time series model. Each of criteria is described in the next subsection.

3.3. *Cross-validation and model evaluation.*

For all cross-validation process, K-fold cross-validation where k = 5, is implemented. We compared Mean-Squared-Errors(MSE) obtained from cross validations for all models and selected the model with the lowest MSE as the best model among non-time series models. We repeated this process for two different dataset.

After the model selection, we compared the best non-times series model and LSTM Multivariate time series model.

Since the K-fold cross validation is not valid for time series models, we simply compared two MSEs from predictions of two models.

4. Result.

4.1. *Comparison of MSE from K-fold CV and Time dependency of the model improvement.*

From two MSE comparison charts, except XGBoost elastic net and simple neural network, we can easily see that models are all improved. In fact, even for neural network model, cross-validation errors(CV errors) decreased negligibly(from 0.9915 to 0.9824). However, for XGBoost elastic net, CV error increased from 0.1517 to 0.1526, which is negligible change. At this point, we concluded that date variables can contribute to the improvement of models and assumed that time series models can be well-fitted for this dataset.

In the meantime, elastic net fitted with XGBoost was the best model among the 10 non-time series models for any of two datasets.

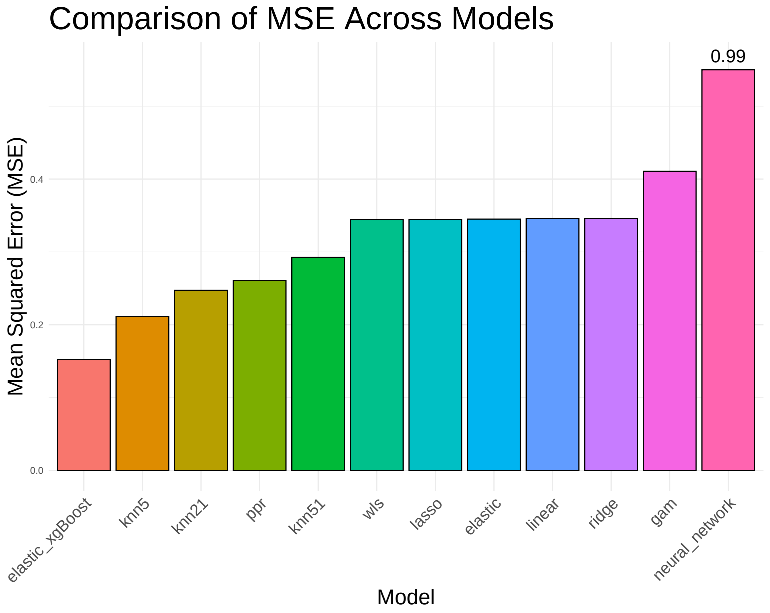

Fig 3: MSE comparisons with date variables

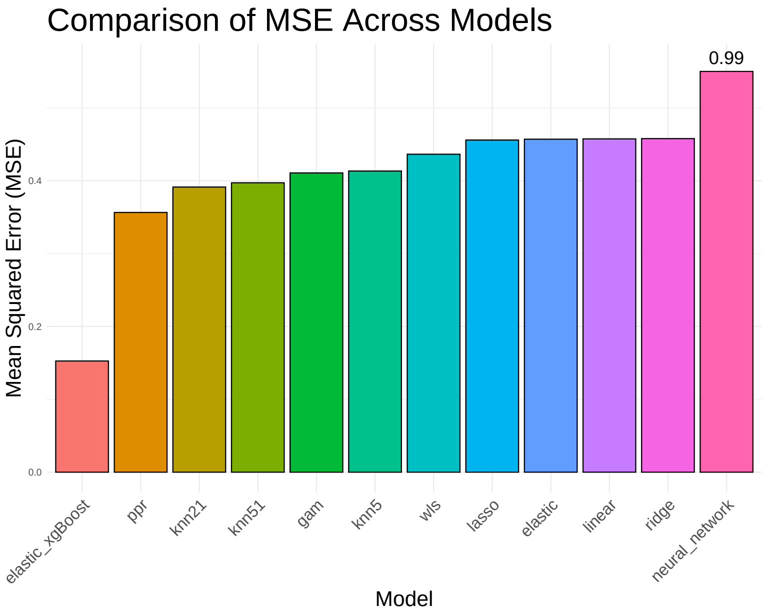

Fig 4: MSE comparisons without date variables

4.2. *Comparison of non-time series model and time series model.*

We compared MSEs from LSTM time series model and our best non-time series model(elastic net with XGBoost) on on the test dataset(10% of overall datasets). The corresponding dates are 2022-10-22 to 2023-04-30 and the blue line is true value from the test dataset. The MSE of LSTM time series model(0.335) was even smaller than the MSE of our best non-time series model(0.645). However, forecasting plot shows that elastic net with XGBoost(green line) performed better for total demand of average days and LSTM(red line) performed better for annual drastic changes.

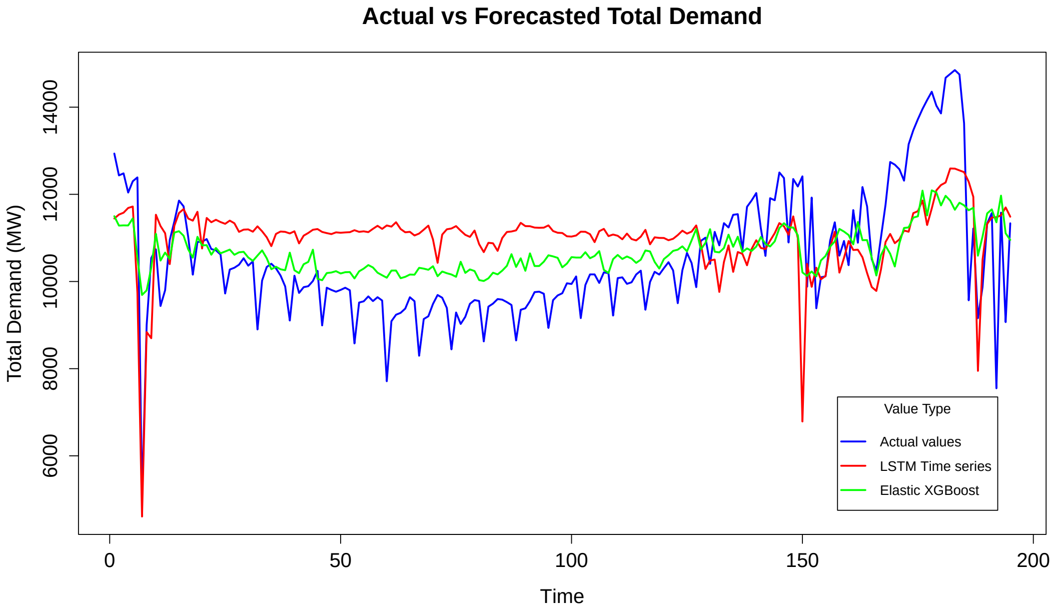

Fig 5: Time-order predictions

4.3. *What factors contribute to non-time series model?.*

From Feature Importance plot of the elastic net with XGBoost, it is not surprising to see that date(year, month, day) variables also contribute to model performance more than other multiple variables. It also implies that daily average temperature(Temp2_ave(c)) and precipitation(prectotcorr) significantly contributed to the model performance.

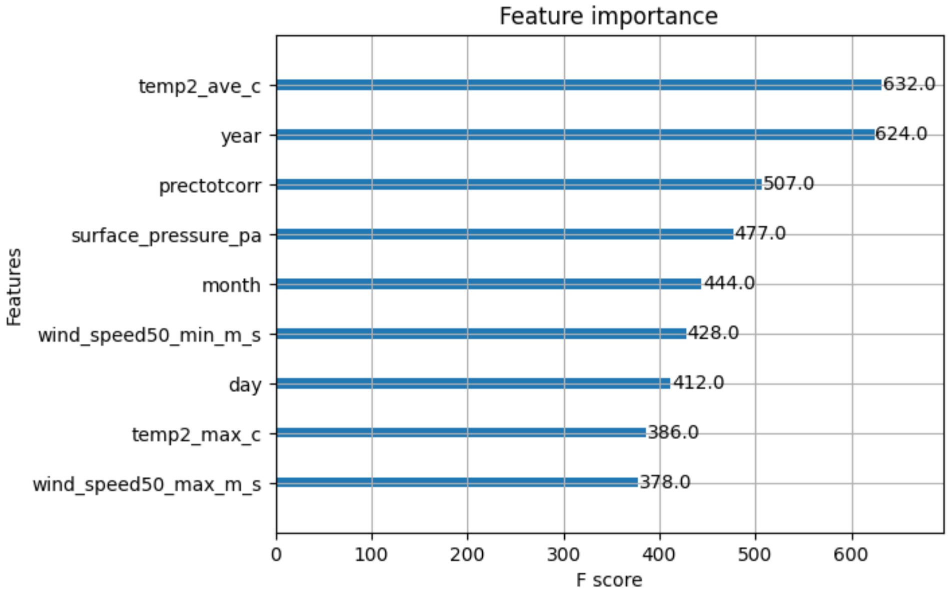

Fig 6: Feature Importance

5. Conclusion.

From two model predictions and the significance of date variables for models, the overall trend of total demand has increased and long term climate factors, the temperature and precipitation impact on the upward trend. This implies that the total demand will keep increasing in the future due to climate change. Moreover, because the reason why the date variables contribute to models seems to be the total demand showing seasonality, the Bangladeshi government should be prepared with energy supply plan considering drastic changes in November, May and June to prevent the waste of energy resource and power outage.

REFERENCES

SALEHIN, I., NOMAN, S. M., and HASAN, M. (2023). Electricity Energy Dataset "BanE-16": Analysis of Peak Energy Demand with Environmental Variables for Machine Learning Forecasting. *Data in Brief*, **52**. DOI: 10.1016/j.dib.2023.109967.

HASTIE, T., TIBSHIRANI, R., and FRIEDMAN, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, New York.

CHEN, T. and GUESTRIN, C. (2016). *XGBoost: A Scalable Tree Boosting System*. Available at: https://doi.org/10.48550/arXiv.1603.02754

HOCHREITER, S. and SCHMIDHUBER, J. (1997). *Long short-term memory*. Available at: https://doi.org/10.1162/neco.1997.9.8.1735

GEEKSFORGEEKS. (2024). Multivariate Time Series Forecasting with LSTMs in Keras. Available at: https://www.geeksforgeeks.org/multivariate-time-series-forecasting-with-lstms-in-keras/

WERBOS, P. (1982). *Applications of advances in nonlinear sensitivity analysis*. Available at: https://werbos.com/Neural/SensitivityIFIPSeptember1981.pdf

GEEKSFORGEEKS. (2021). Training Neural Networks with Validation using PyTorch. Available at: https://www.geeksforgeeks.org/training-neural-networks-with-validation-using-pytorch/