Report On Data Imputation and Predicting Future Power Demand

1. Introduction

At the Indian Institute of Technology Guwahati (IITG), electricity is supplied from two main sources: power imported from Assam Power Distribution Company Limited (APDCL) and on-campus solar power generated from Phase II solar PV systems. However, the APDCL import data, which provides hourly measurements of power in megawatts (MW), has missing values.

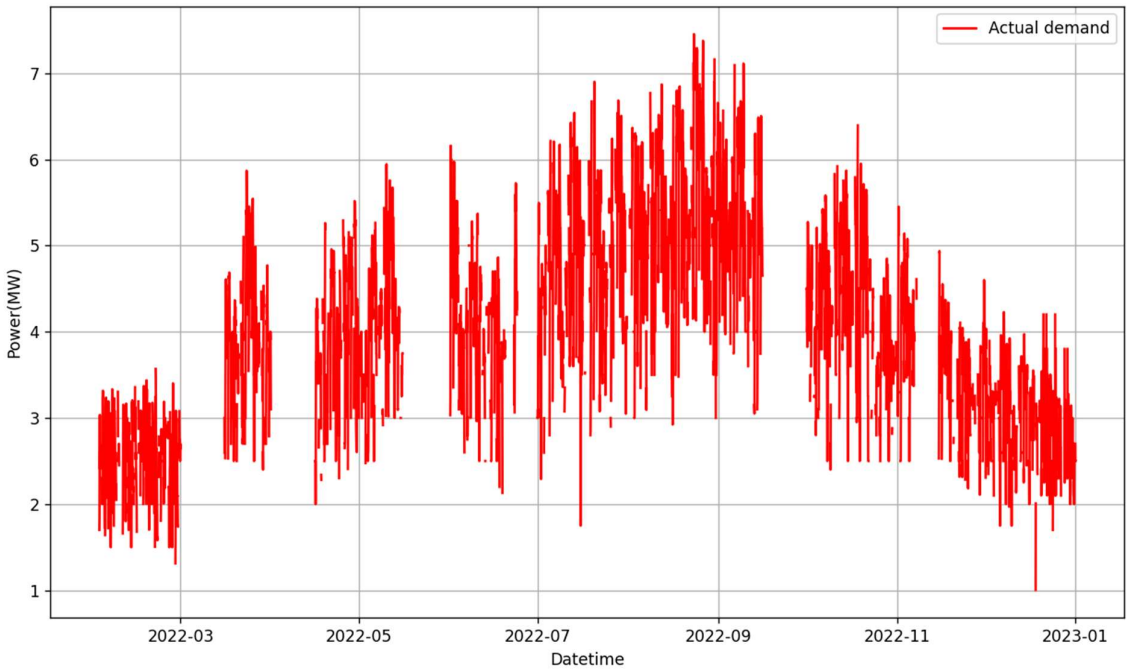


Figure 1: Actual power demand data with missing values

We can see in the above figure the plot of the original demand data which has missing points The goal of this project is to compare different ML models to predict demand data and therefore fill in the missing data in the power import records to create a continuous record of IITG's total power usage. The goal is also to compare the best model for the calculation of peak power demand and its datetime for a particular year. We made 3 different models for the prediction of demand data:

Model 1: It predicts hourly power demand using linear regression on hour-of-week load patterns and a threshold temperature feature.

Model 2: This model predicts hourly power demand using class-wise histogram-based gradient boosting on temperature and time features.

Model 3: This model predicts daily peak demand using a single histogram-based gradient boosting model with temperature, time, and encoded class features.

2. Procedure

2.1 Model 1

The estimation of values is performed using the following mathematical formula:

$$\hat{P}_k = \hat{Y}_k + S(T^k) \quad \text{…..……(i)}$$

where,

$$\hat{Y}_k = \sum_{j=0}^{N-1} \omega^{(j)} \, 1\{\text{j=k}\} \quad \text{…..……(ii)}$$

$$S(T^k) = \omega^{(N)} \begin{cases} (T - T^{Th}), T \geq T^{Th} \\ 0 \qquad\qquad , T < T^{Th} \end{cases} \quad \text{.....….(iii)}$$

$$= \omega^{(N)} \, S'(T^k)$$

$$\text{And, } 1\{j = k\} = \begin{cases} 1 & , j = k \\ 0 & , j \neq k \end{cases} \quad \text{……(iv)}$$

In equation (ii), Y^k is the predicted power for k^{th} hour, N is hours in a week, $\omega^{(j)}$ is the estimated coefficient of the regression model for j^{th} hour. $1\{j = k\}$ is the indicator function as shown in equation (iv).

Equation (iii) shows temperature threshold feature, where T^{Th} is threshold temperature. We estimated the value of T^{Th} using a graph of Power vs Temperature as shown in the figure below:

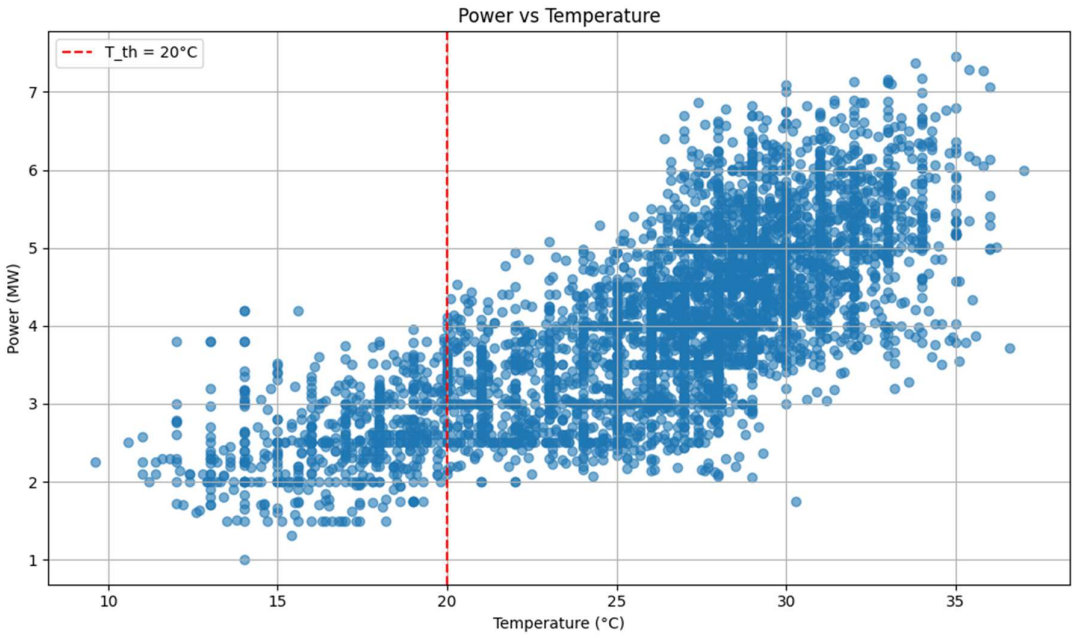


Figure 2: Power vs Temperature plot to find T^{Th}

We observe that the power demand goes significantly high after 20°C so, we selected T^{Th} =20°C.

2.1.1 One-Hot Encoding of Hour-of-Week Feature

Since linear regression models cannot directly handle categorical variables, the hour-of-week feature was encoded using one-hot encoding. To capture systematic time-of-use patterns in electricity demand, categorical information related to the position of each timestamp within a week was explicitly encoded using one-hot encoding. Each hourly timestamp t was first mapped to a unique hour-of-week index defined as:

$$hour\_of\_week = 24 \times day\_of\_week + hours \qquad \text{.....(v)}$$

where,

day_of_week $\in\{0,1,\ldots,6\}$represents the day of the week (Monday = 0, Sunday = 6) and,

hour $\in\{0,1,\ldots,23\}$ represents the hour of the day.

This transformation yields a discrete categorical variable taking integer values from 0 to 167, uniquely identifying each hour within a weekly cycle (e.g., Monday 00:00 as 0 and Sunday 23:00 as 167).

2.1.2 Model Training Using One-Hot Encoded Features

For each hourly timestamp t, the input feature vector was defined as:

$$X(k) = [h_0(k), h_1(k), h_2(k), \ldots, h_{167}(k), S'(T^k)]$$

where,

$h(k) \in \{0,1\}$ are the one-hot encoded hour-of-week indicators, and

$S'(T^k) = \max((T(k) - T^{Th}), 0)$ is the thresholded temperature feature.

The corresponding target variable y(k) is the observed electrical power demand.

An intercept term (β) is included to capture the average baseline power demand of the system, enabling the hour-of-week coefficients to model deviations around this mean. So the final equation becomes like:

$$\hat{P}_k = \sum_{j=0}^{N-1} \omega^{(j)} 1\{j=k\} + \omega^{(N)} S'(T^k) + \beta \quad \text{.....(vi)}$$

Only time steps with valid (non-missing) power measurements were used for model training. These observations were randomly divided into a training set (90%) and a test set (10%). The training set was used to estimate the model parameters, while the test set was reserved for performance evaluation using R-squared score.

2.1.3 Prediction of Future Power Demand

The gaps of power demand of 2022 are imputed using the predicted power and the model is trained again using the fully imputed data of 2022. Hourly temperature and timestamp information from the target period (2023) are converted into the same feature representation using the encoder fitted during training. The trained regression model is then applied to estimate demand for each hour.

2.2 Model 2

The model that we used here is called Histogram based Gradient Boosting Regressor. The hourly demand data is labelled with classes of multiple events in accordance to the academic calendar of IITG such as, ‘Working_days’, ‘Short_Holiday’, ‘Long_Holiday’, ‘Exam’, ‘Weekend’, ‘Fest’, ‘Vacation’. Separate models are trained for each class, and during prediction, each sample is evaluated using the model trained on its corresponding class.

2.2.1 Model Training Using Class-wise Gradient Boosted Decision Trees

The model starts with a very simple prediction, typically the average demand of the training data. Each model in the ensemble is a regression decision tree. A tree predicts demand by repeatedly splitting the feature space into regions and assigning a constant value to the initial prediction.

The full model does not use just one tree. It adds 600 small trees, but each tree makes a small correction to the final value. The final working mathematical formula is:

$$\hat{P}^{(c)}(x) = \bar{P}_c + \eta \sum_{k=1}^{600} g_k^{(c)}(x) \qquad \text{..... (vii)}$$

In the above equation,

$\hat{P}^{(c)}(x)$ = predicted power demand for input feature x, using the model trained for class `c`.

$\bar{P}_c$ = initial baseline prediction.

η = Learning rate controls how much each tree contributes, here it is 0.03.

$g_k^{(c)}(x)$ = Output of the k^{th} Regression Tree

For each timestamp *t*, we construct a feature vector:

$$X_t = [T_t, h_t, d_t, m_t] \qquad \text{..... (viii)}$$

where Tt is the temperature, h_t is the hour of the day, d_t is the day of the week, and m_t is the month. The model was trained with 90% of valid (non-missing) data and 10% random data was reserved for testing.

2.2.2 Prediction of Future Demand Data

The gaps of 2022 power demand are imputed using predicted power and the model is trained again using the imputed power data. The trained models are used to forecast electricity demand for the year 2023. The same set of input features used during training is constructed for the 2023 timestamps.

2.3 Model 3

This model works almost like Model 2, instead we use daily peak power for training and prediction. Also in feature vector, the academic calendar based classes are encoded using one-hot vectors. For each day d, we construct a feature vector:

$$X_D = [T_d^{pk}, T_d^{avg}, w_d, m_d, c_1, c_2, \ldots, c_k] \quad \ldots\ldots\ldots\ldots\ldots \text{(ix)}$$

T_d^{pk} is the maximum temperature of the day, T_d^{avg} average temperature of the day, w_d day of the week, m_d is month, c_k are binary variables representing the type of academic day (only one of them is 1).

The working equation for this model is:

$$\hat{P}(x) = \bar{P} + \sum_{k=1}^{600} \eta . g_k\,(x) \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(x)}$$

In the above equation,

$\hat{P}(x)$ = predicted power demand for input feature x.

$\bar{P}$ = This is a constant value, the average of all training targets.

η = Learning rate controls how much each tree contributes, here it is 0.03.

$g_k(x)$ = Output of the k^{th} Regression Tree

Here we do not make separate models for separate classes, instead we train one single function for all days, and classes are part of the input vector. Remaining process is same as model 2.

3. Results

The performances of the models were evaluated and it is represented in the following table:

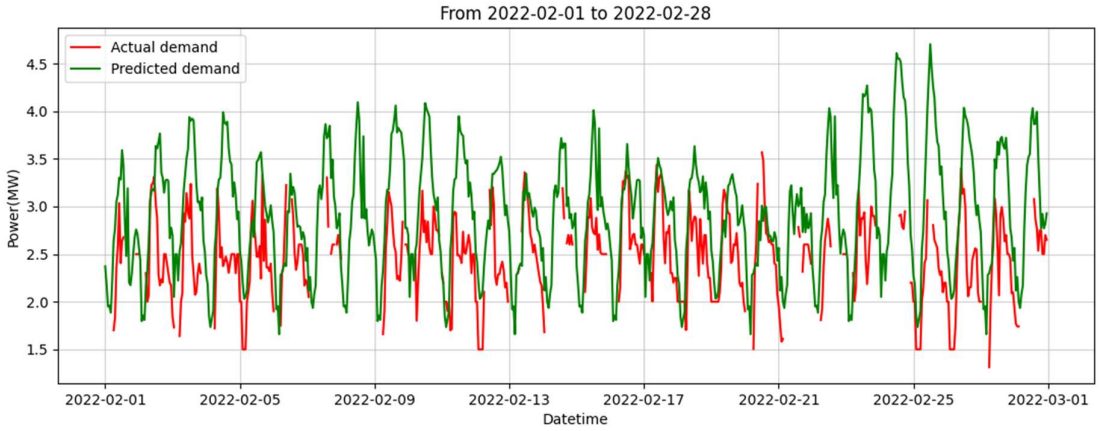
	Model 1	Model 2	Model 3
R2 score for 2022	0.693	0.922	0.973
MAE for 2022	0.526 MW	0.249 MW	0.182 MW
Percentage error for 2022	14.07 %	6.71 %	4.04 %
Peak Power Prediction for 2022 (actual = 7.452 MW)	6.218 MW	6.723 MW	7.155 MW
Peak Power Datetime Prediction for 2022 (actual = 2022-08-23 12:00:00)	2022-08-24 12:00:00	2022-09-09 11:00:00	2022-08-24
R2 score for 2023	0.623	0.687	0.659
MAE for 2023	0.633 MW	0.593 MW	0.661 MW
Percentage error for 2023	15.52 %	14.72 %	13.26 %
Peak Power Prediction for 2023 (actual = 7.861MW)	7.477 MW	6.775 MW	6.941 MW
Peak Power Datetime Prediction for 2023 (actual = 2023-08-08 11:00:00)	2023-05-10 13:00:00	2023-08-25 12:00:00	2023-08-29

Table 1: Performance evaluation of all models

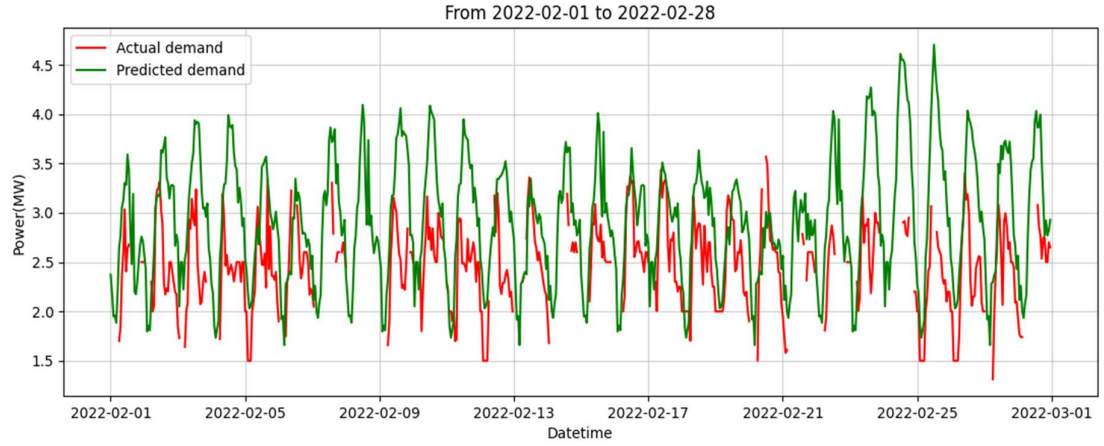
Below are the plots of all the models, representing the actual and predicted demand for the year 2022, using which we did the training. For simplicity we showed 3 months of data – February, August and November of 2022:

For February 2022:

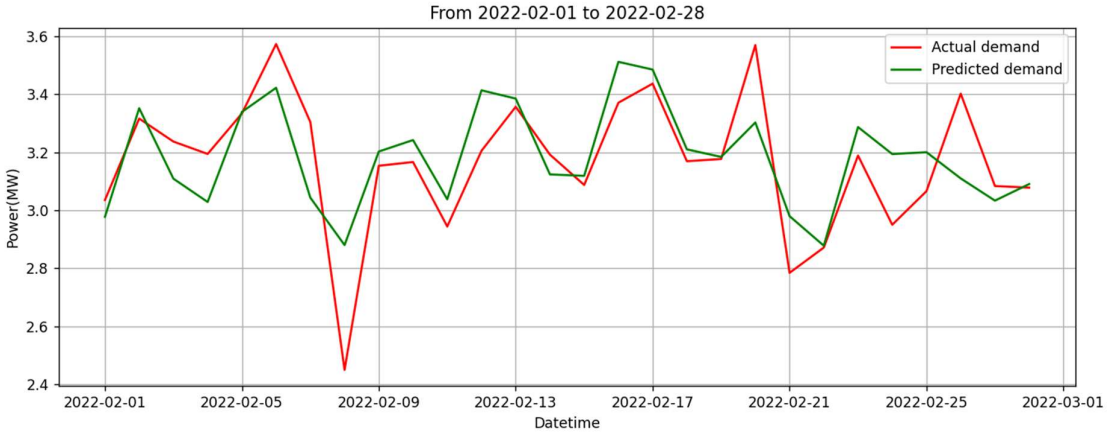
Model 1:



Model 2:

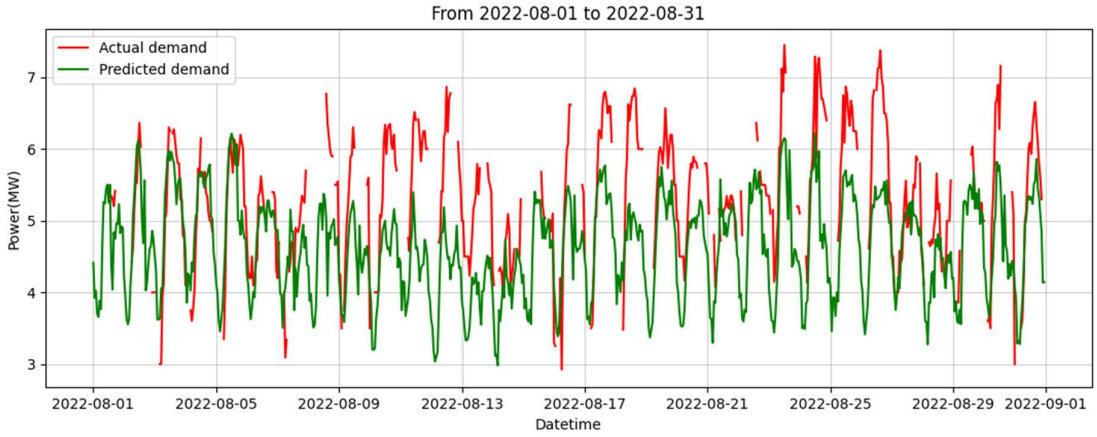


Model 3:

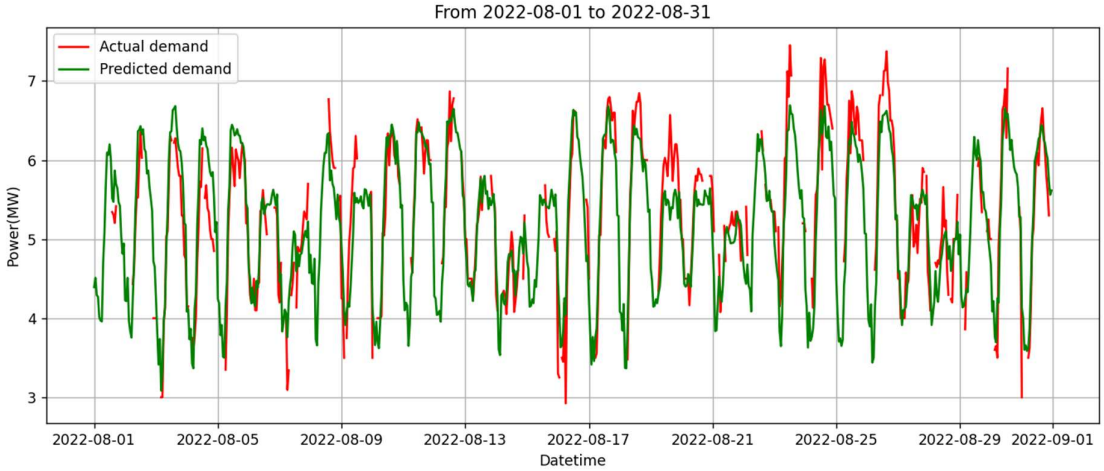


For August 2022:

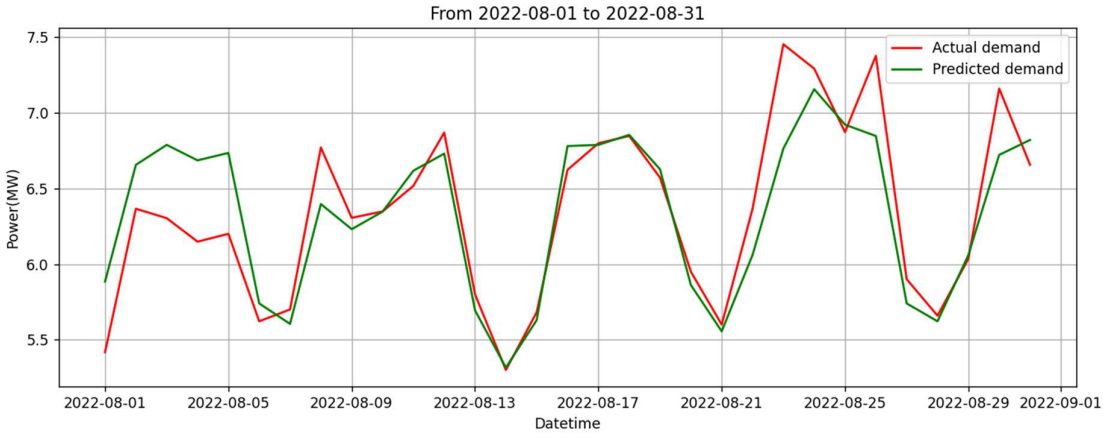
Model 1:



Model 2:

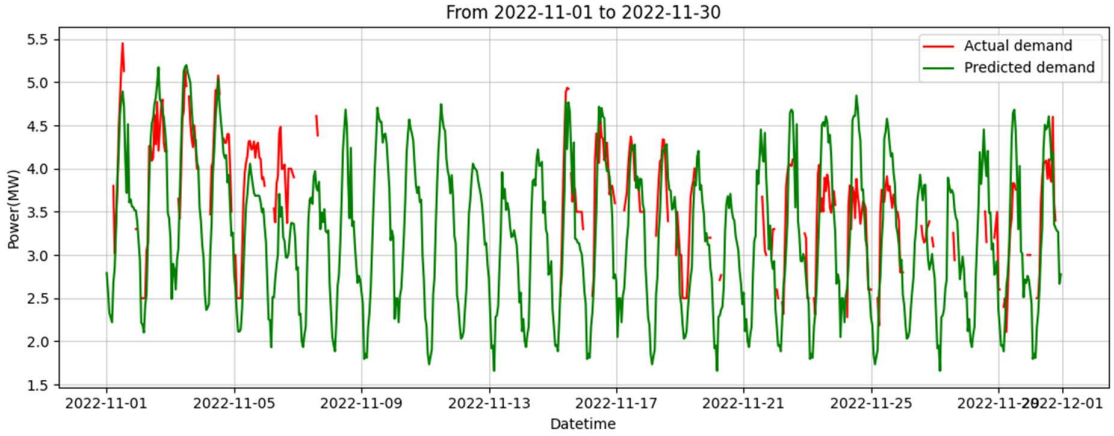


Model 3:

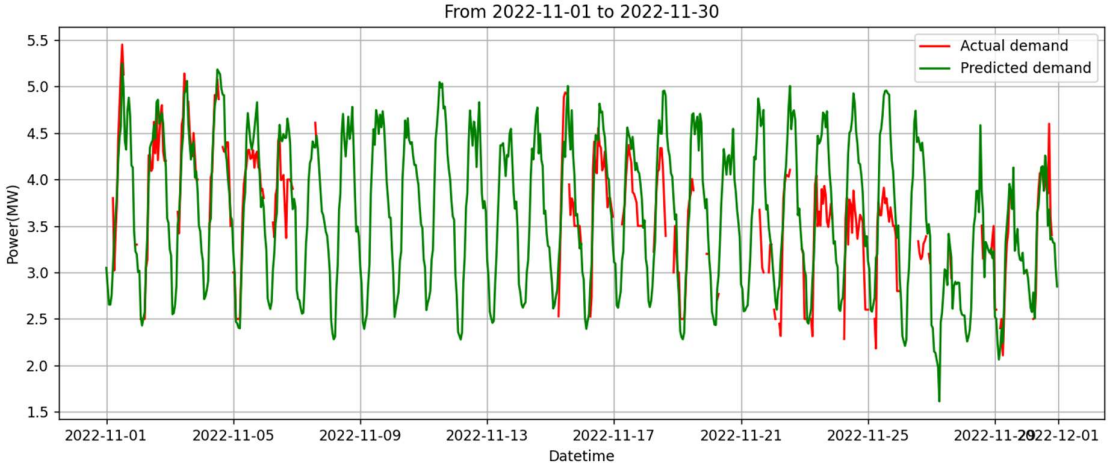


For November 2022:

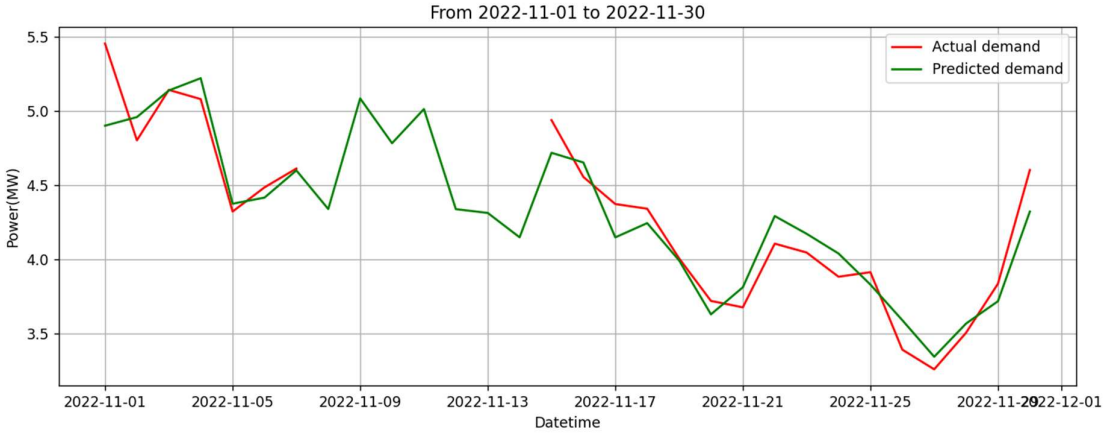
Model 1:



Model 2:



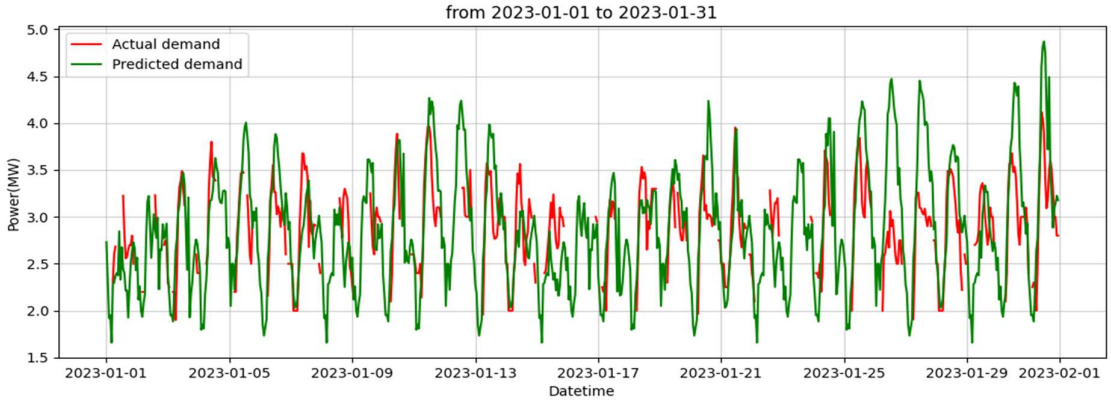
Model 3:



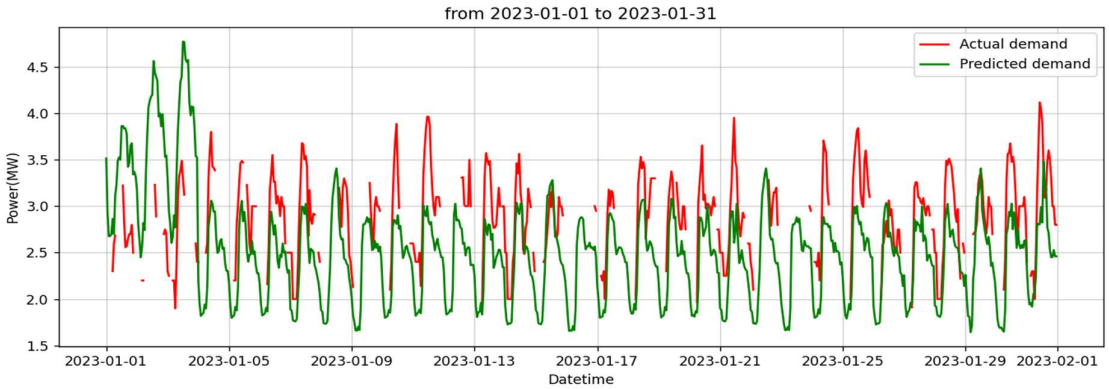
Now the models were retrained with the imputed power data of 2022 and using that we predicted power demand for 2023. Below graphs shows the prediction for the month of January, June and December of 2023:

For January 2023:

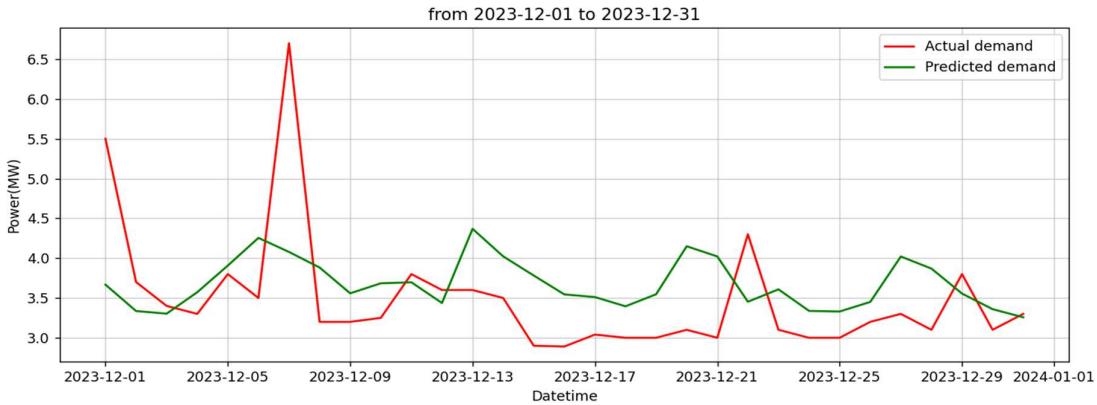
Model 1:



Model 2:

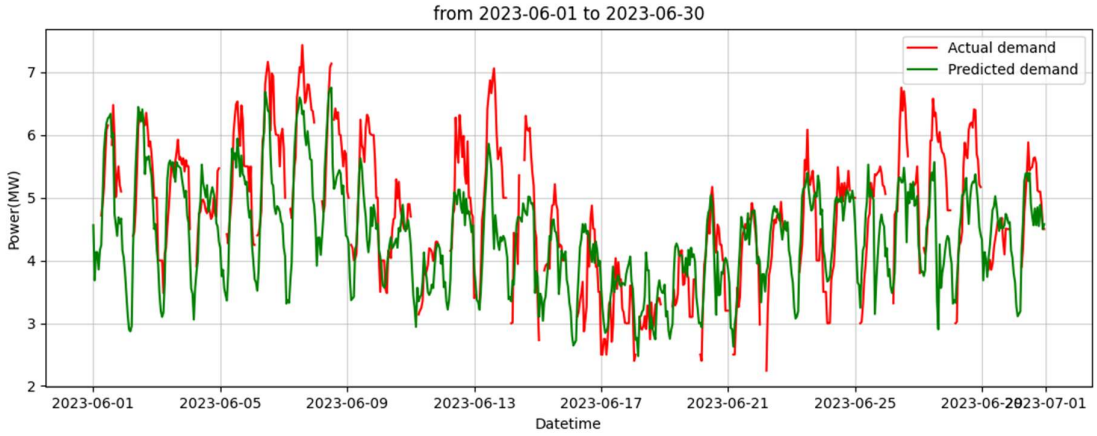


Model 3:

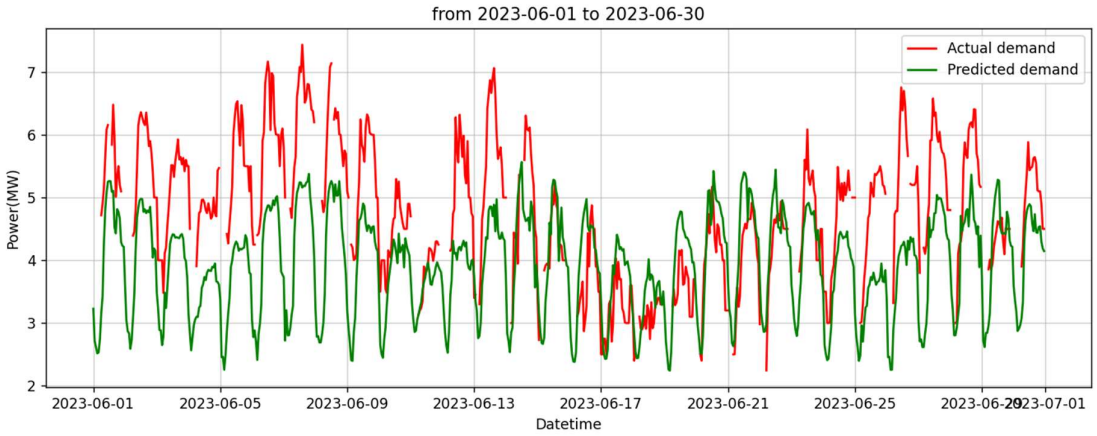


For June 2023:

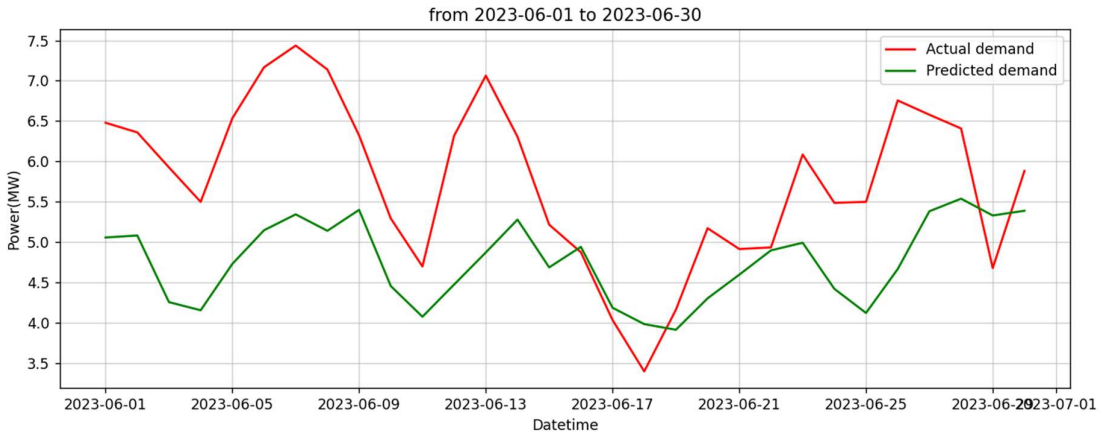
Model 1:



Model 2:

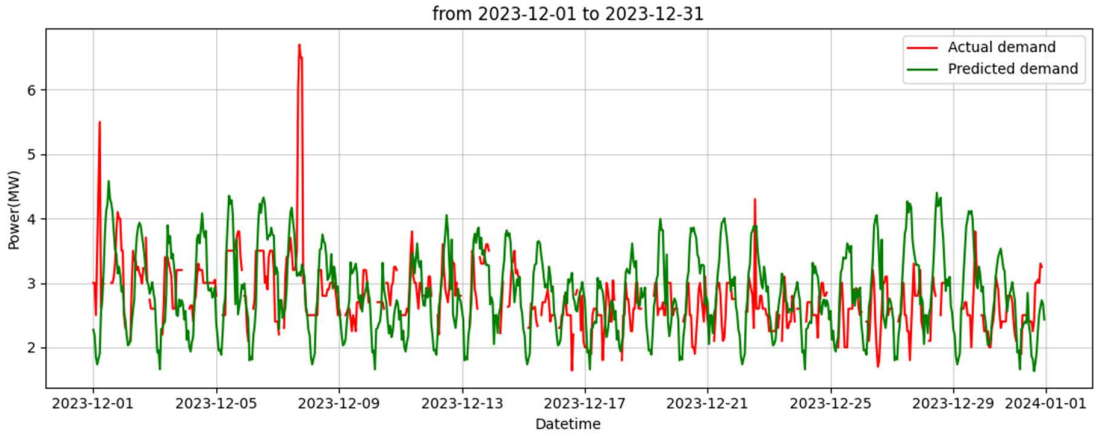


Model 3:

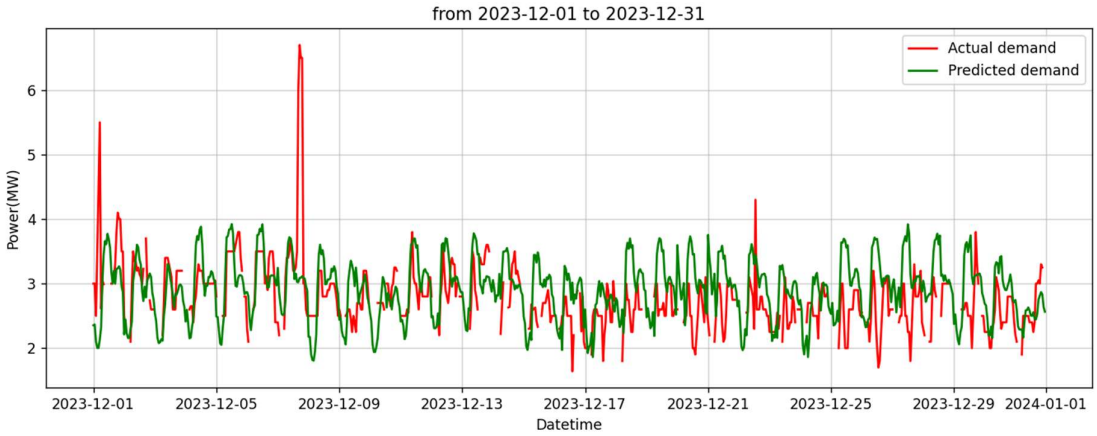


For December 2023:

Model 1:



Model 2:



Model 3: