TEAM 11

CONTENT

Overview

Electrical Energy is one of the most important sources for the social and economic development of all nations. The growth in energy consumption is essentially linked with the growth in the economy. Building energy consumption has become one of the three major energy-consuming industries, in addition to the industrial and transportation industries. Office buildings are a symbol of the post-industrial era and of today's global knowledge economy. Large corporate building energy consumption is several times that of ordinary buildings. Therefore, it is necessary to study the characteristics of corporate buildings to determine their energy consumption situations and the important factors affecting their energy consumption.

The ability to forecast electricity demand accurately is one of the most important tasks for power system planning. Load prediction helps in reliable and economical operation of the power system. In general, if the forecasted load is greater than the demand there will be an unnecessary commitment of units, whereas if the forecasted load is lower than the demand, it will lead to the purchase of power at higher prices in the deregulated market.

Hence, we aim to develop efficient time series models for Dream Vidyut and help them to better forecast electricity consumption of provided corporate buildings. We have used data visualization tools, to generate insights and developed intelligent features to support our forecasting models. Identification of anomalies was done, assignable causes were identified, and they are replaced with mean values to develop robust models. We have developed various statistical, machine learning and deep learning-based models, and evaluated them on the validation set for model selection. Finally, we stacked results from the two best performing models to produce roust predictions.

Exploratory Data Analysis

Data Summary

Summary Statistics

The data provided consists of the reading in the main meter and two sub-meters of five buildings from 01-04-2017 to 31-12-2017. The given timestamp is divided into 15-minute intervals. The dataset consists of electricity consumption of five buildings across three different meters for each timestamp. The table presents the aggregated summary statistics for the given dataset for all the buildings. To see individual building-wise summary statistics refer to Annexure A.1.

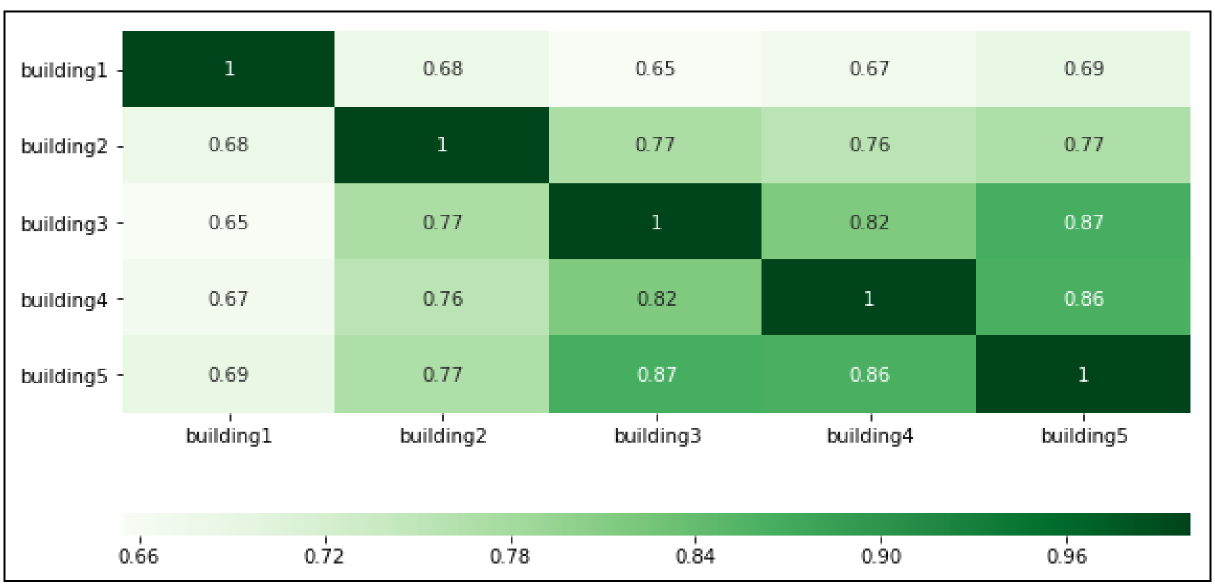
	Main meter	Submeter 1	Submeter 2
Mean	6095.76	2092.88	776.83
Std	3485.12	1135.88	883.49
Min	61.63	0.00	0.00
25%	3455.91	1365.00	4.85
50%	5362.18	2075.67	621.91
75%	7685.40	2772.50	1365.89
Max	23174.60	17206.16	4392.07

From the mean, 75% and the maximum readings of the meters, it can be seen that the **dataset consists of outliers** that have to be treated for better prediction. It can also be seen from the above table that 25% of the Sub Meter 2 readings are below 4.85, which suggests that **electricity consumption through sub meter 2 is only during some portion of time.**

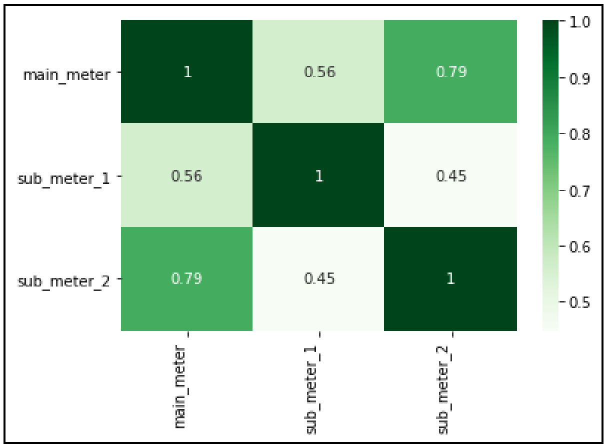
Correlation Matrices

Correlation Between Buildings for different meters

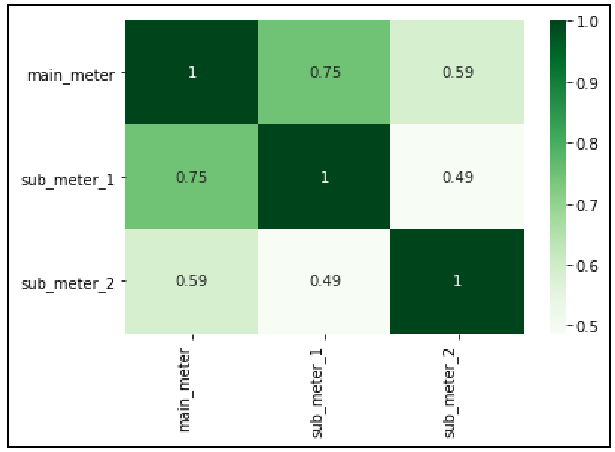
We have calculated the correlation matrix across buildings for all the metres and found that the electricity consumptions between the corporate buildings are **fairly correlated**, which shows that the model can be **generalized to other new corporate buildings** too. Below is the correlation between buildings for main-meter readings. For submeter-1 and sub meter-2 readings see Annexure A.2.



Correlation Between Meters for different buildings



Building 1



Building 2

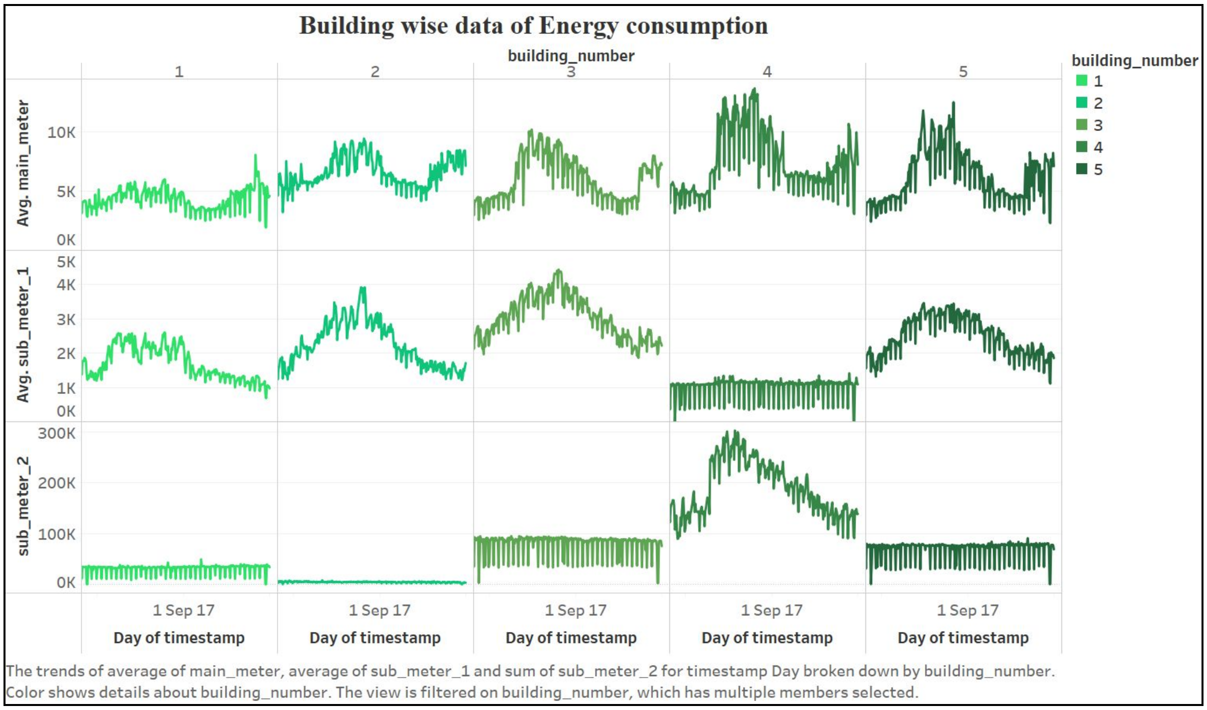
It is visible that **meter readings are also fairly correlated with each other**, which can be justified as the electricity consumption generally depends on the day-wise factors such as temperature, humidity etc. Above is the Correlation matrices for Buildings 1 and 2, rest are shown in Annexure A.2.

Data Visualisation and Insights

Data visualization is one of the first steps of building data-driven forecasting models. It is a well-known fact that the representation of data in terms of the graph, charts etc grabs our interest and helps us to internalize the patterns quickly. Study of Graphical representation of data not only allows us to identify the general trends but it also helps us in identifying the specific patterns and outliers present.

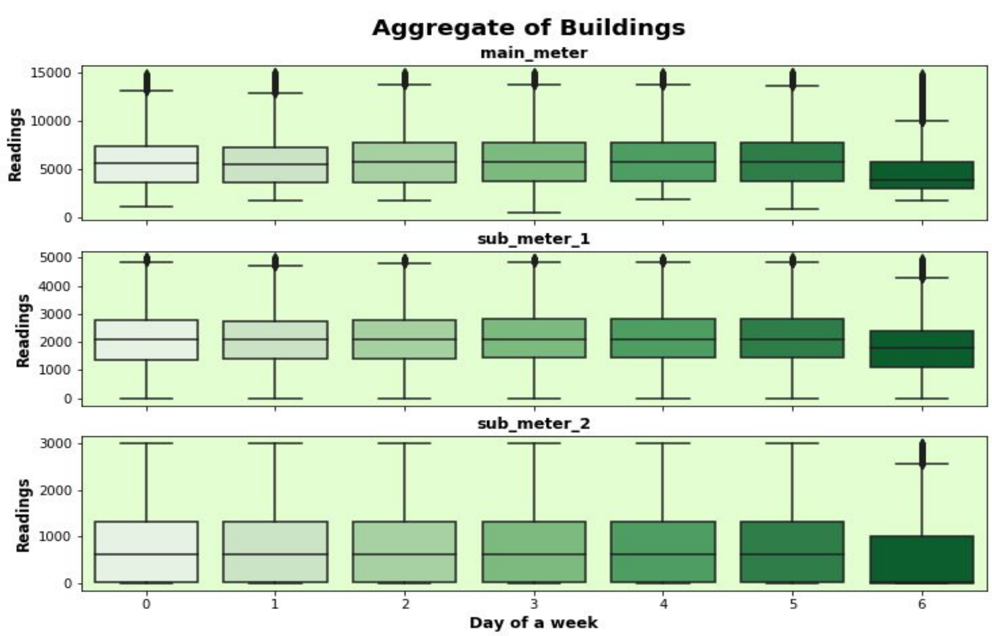
General Trends in Time Series

We first plotted a canvas of line charts of each time series, ranging over all the buildings, to get a brief overview of what kind of time series we have, and what further analysis we should perform. It is noticeable that the **variance of readings of the main meter are considerably higher** than that of the sub-meters. Also, it is visible that as compared to submeter 1, **sub-meter 2 has fairly less variance**. It is also clear that **submeter 2 has higher consumption** when compared to that of submeter 1 and mainmeter.



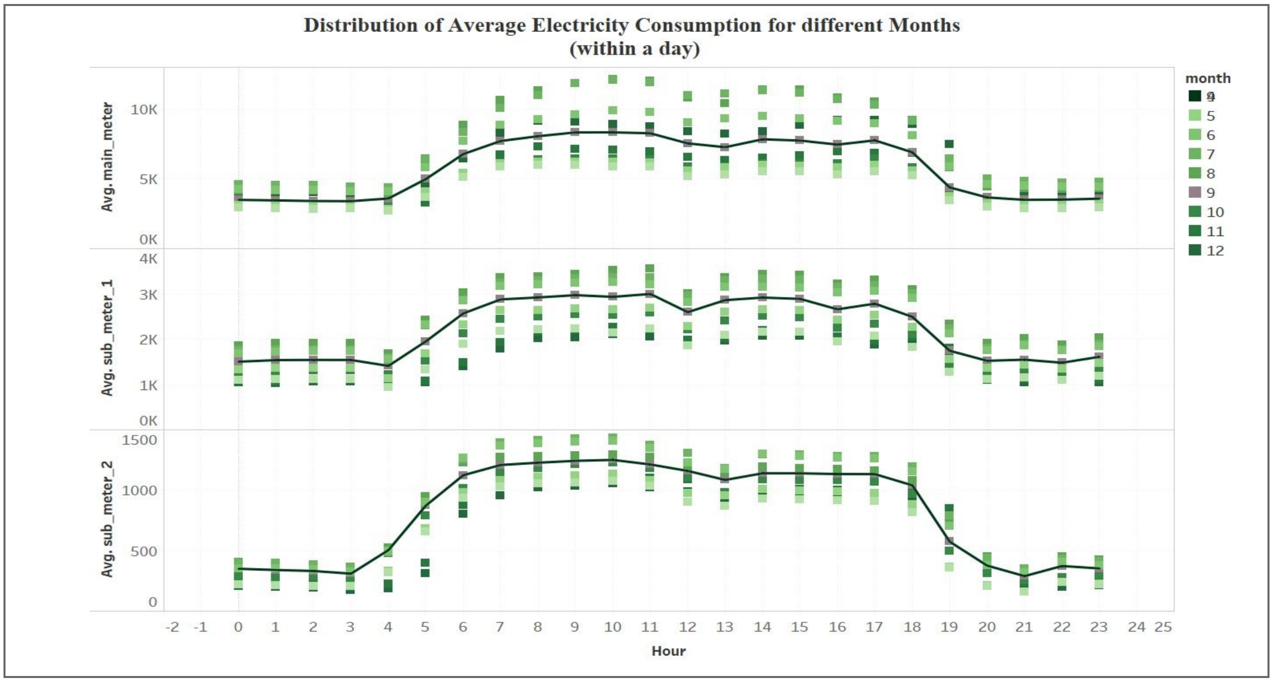
Seasonality within a Week

We had an intuition that there should be a **significant difference** between the distribution of energy consumption **during the weekdays as compared to weekends**. Our intuitions were further strengthened by plotting the boxplots which clearly shows a significant difference between the energy consumption on Sunday and other days. (See Annexure A.3)



Seasonality within a Day

As the given building is a corporate building, an obvious question arises if the readings while the data from **corporate & non-corporate hours** (08:00-20:00) might follow a certain trend. We confirmed this intuitive idea by plotting the aggregated day-wise meter readings. It is clearly visible that there is a **significant difference between the electricity consumption during corporate and non-corporate hours.**



Feature Engineering

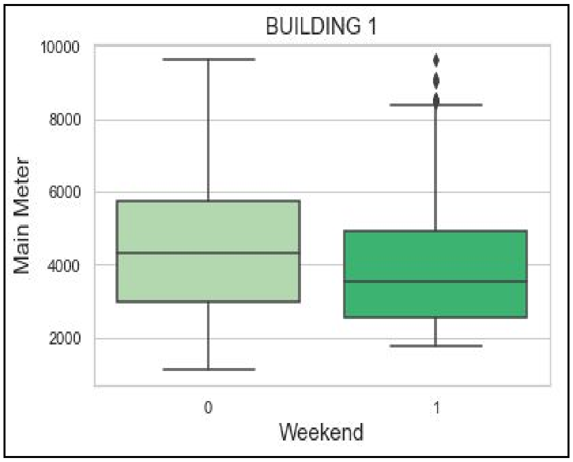
Weekend Feature

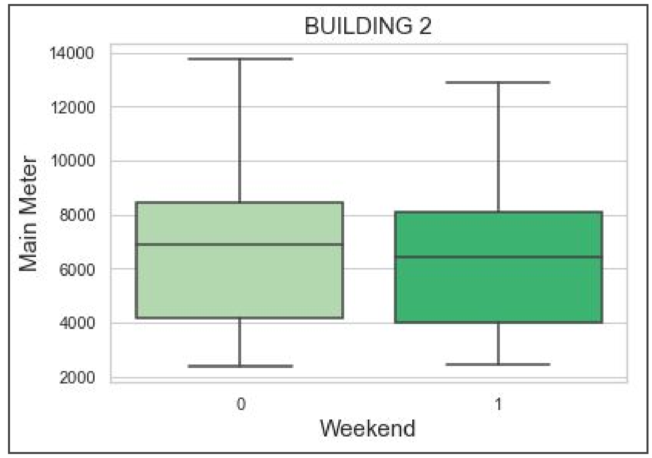
We have identified through the above graphs that the distribution of energy consumption during the weekdays is very different from that of Weekends. So, we have **made a new feature called Weekend** which has a **value of 1 on weekends and a value of 0 on weekdays.** We have conducted a **T-test** to check the significance of this feature. The table below gives us the T-Statistic for the weekend feature:

Meter	Mean Difference	Standard Error	|T-Statistic| (|t|)	Critical Value (α)	P-Value
Main Meter	1603.547	43.673	36.718	1.645	0.000
Sub Meter 1	408.776	16.672	24.519	1.645	0.000
Sub Meter 2	304.267	12.369	24.600	1.645	0.000

Confidence Level = 0.95, Degree of Freedom = 32998
The **null hypothesis is rejected** and the **time series are significantly different** for the added weekend feature.

The box plots also show the significance of the added features. (See Below and Annexure A.4)





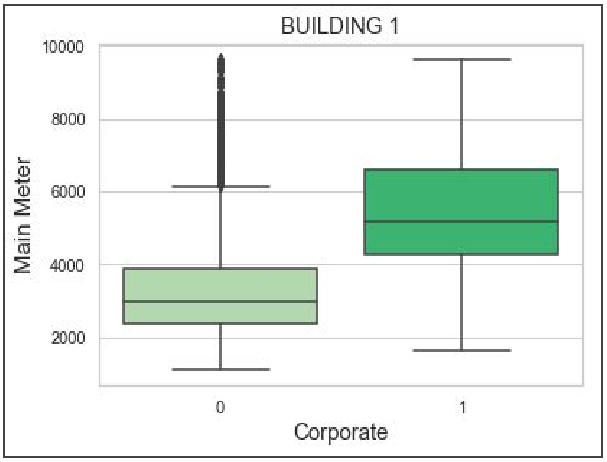
Corporate Feature

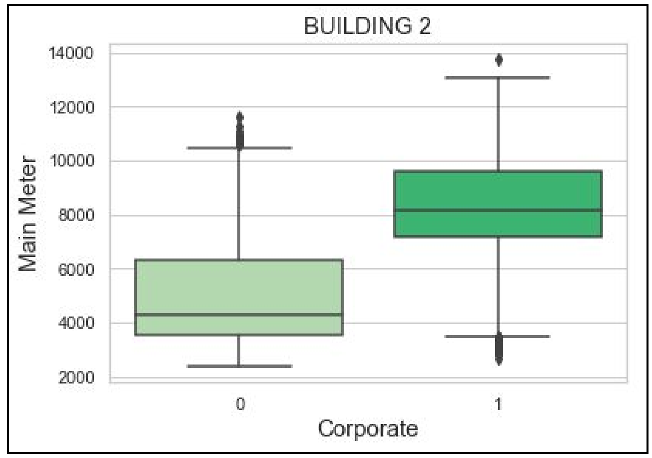
As we know, the meters have significantly different readings during **corporate hours (8 AM to 8 PM)**. So, a new feature called corporate was made which had a value of 1 during corporate hours and a value of 0 during non-corporate hours. The table shows the result of the t-test conducted:

Meter	Mean Difference	Standard Error	|T-Statistic| (|t|)	Critical Value (α)	P-Value
Main Meter	3468.002	27.564	106.073	1.645	0.00
Sub Meter 1	935.615	10.888	84.227	1.645	0.00
Sub Meter 2	605.682	6727	67.174	1.645	0.00

Confidence Level = 0.95, Degree of Freedom = 32998
The null hypothesis is rejected and the time series is significantly different for the added corporate feature.

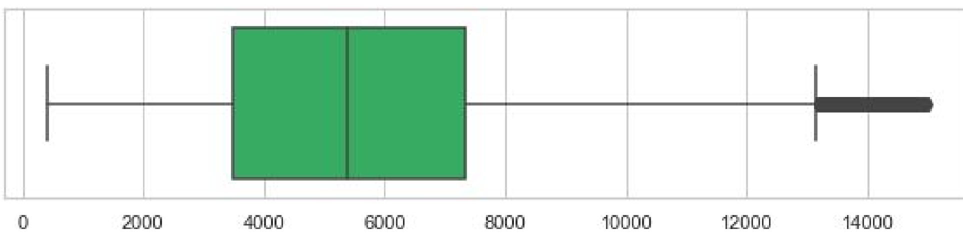
The box plots also show the significance of the added features. (See below and Annexure A.4)



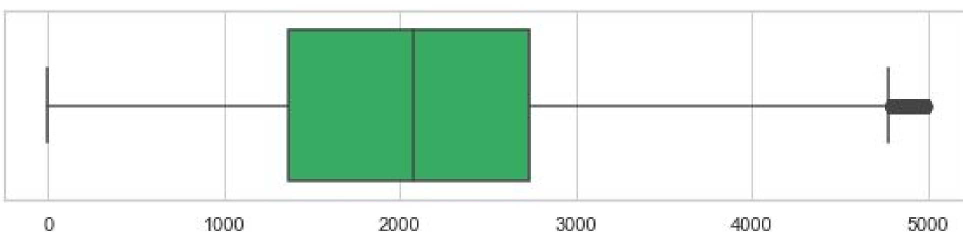


Anomaly Detection

We wanted to see if there were any significant outliers with respect to the meter values which we were predicting, so we decided to box-plot these meter values.



Main Meter



Sub Meter 1

Sub Meter 2

As we can see from these plots, there are **clear outliers**, so we decided to **replace these outliers with the median values** of each of the series. The percentage of data which was replaced are as follows:

Main Meter	Sub Meter 1	Sub Meter 2
2.65%	0.45%	2.13%

Since these constitute a **very insignificant percentage** of the series, removal of these should not affect the series adversely.

Time Series Models

There are a variety of Time Series forecasting models available and are used in a variety of fields such as Weather forecasting, Demand forecasting in Industries, Stock price prediction in financial markets etc. We will be using these models to build an efficient and robust electricity demand forecasting model.

We applied various Time Series models from different classes such as Statistical (Holtz-Winter's Method, ARMA, **ARIMA**, Garch), Machine Learning based (**XGBoost Regressor**, Random Forest Regressor) as well as Deep Learning-based (RNNs, LSTMs, **GRU**) models. We have presented below the results for the best models from each class and our intuition of why these models performed the way they have performed.

Next, we also wanted to explore if **making predictions at different aggregate timestamp levels** had any advantage to it. So, we had to decide the optimal level of timestamps for making the predictions, so, we decided to make predictions at various aggregate timestamp levels such as 30 minutes, 1 hour, 2 hours and 4 hours other than the original predictions at 15 minutes. We made these predictions across different models as well. We **compared the evaluation metric values for the validation set at these different aggregate levels**. We observed that prediction made at **an aggregate level of 1 hour provided the best results** across different models. You can see the graph of the results of the same in Annexure A.5

In the interest of time and space, we have mentioned the details of architecture and parameters of the best models in Annexure A.6.

ARIMA

Theory

An ARIMA model is a class of statistical models for analyzing and forecasting equally spaced univariate time series data and also predicts a value in a response time series as a linear combination of its own past values, past errors and current and past values of other time series. ARIMA models allow both autoregressive (AR) components as well as moving average (MA) components. AR component model the "change since last time". MA components capture smoothed trends in the data. The ARIMA model is formulated as:

$$\left(1-\sum_{i=1}^{p}\varphi_i L^i\right)(1-L)^d X_t = \boldsymbol{\delta} + \left(1+\sum_{i=1}^{q}\theta_i L^i\right)\epsilon_t$$

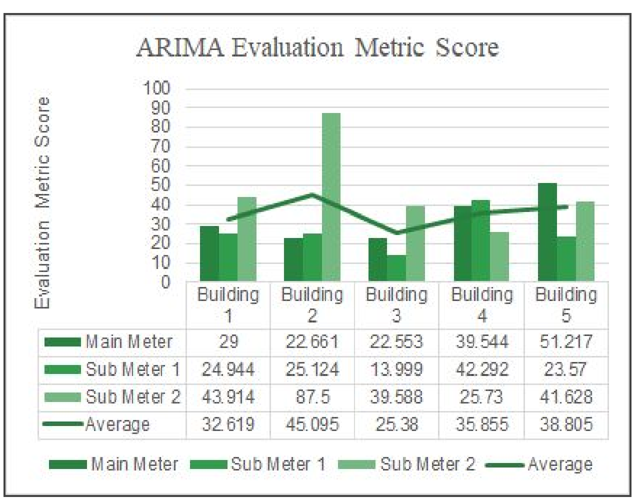
$X_t \rightarrow$ Time Series | p $\rightarrow$ order of AR part | d $\rightarrow$ order of differencing
q$\rightarrow$ order of MA part | θ $\rightarrow$ parameters of MA part | L $\rightarrow$ Lag Operator
$\varepsilon \rightarrow$ error terms | ϕ $\rightarrow$ parameters of AR part

Why ARIMA?

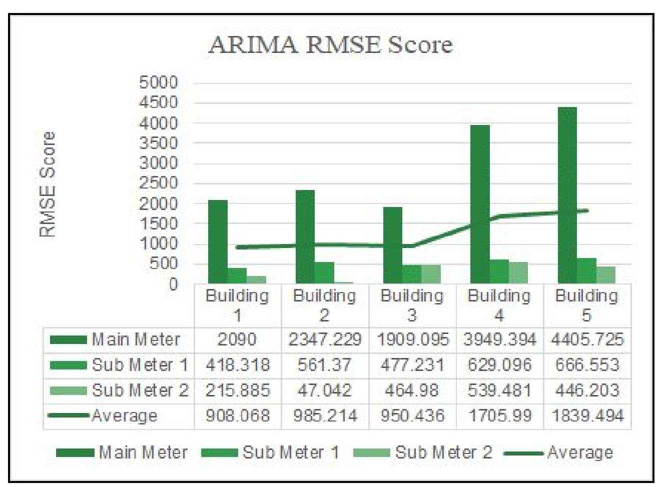
ARIMA is one of the best models when it comes to time series forecasting. It determines the level of differencing to use, which helps make the data stationary. It occupies the middle-range area of being simple enough to not overfit while being flexible enough to capture some types of relationships you'd see in the data. We have seen that ARIMA has fairly captured the simple pattern as expected.

Results

Evaluation Metric on Validation Set



RMSE on Validation Set



XGBoost (eXtreme Gradient Boosting)

Theory

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. It is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

Why XGBoost?

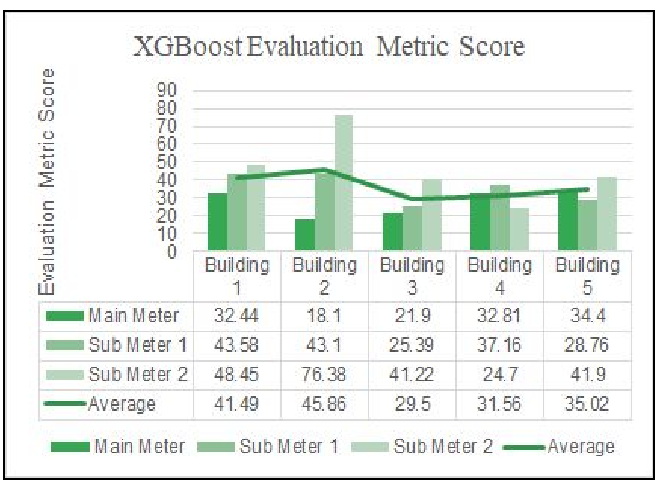
The two reasons to use XGBoost are also the two goals of the project:

- Execution Speed: It is really fast when compared to other implementations of gradient boosting.
- Model Performance: It dominates structured or tabular datasets on classification and regression predictive modelling problems.

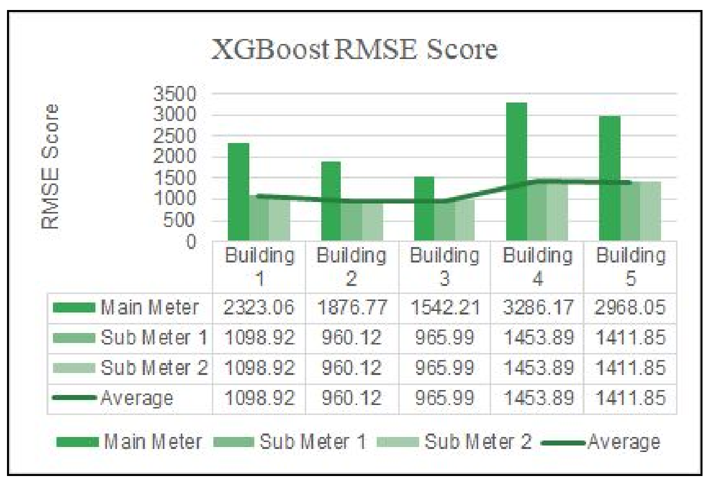
Finally, the evidence is that it is the go-to algorithm for competition winners on the Kaggle competitive data science platform.

Results

Evaluation Metric on Validation Set

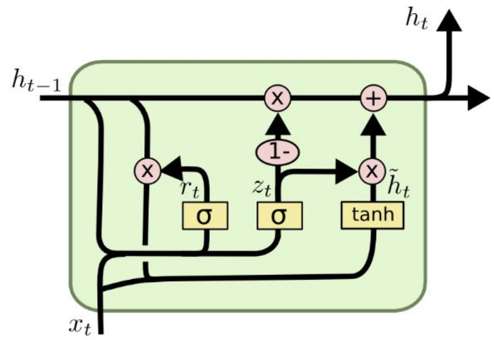


RMSE on Validation Set



GRU (Gated Recurrent Unit)

Theory



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

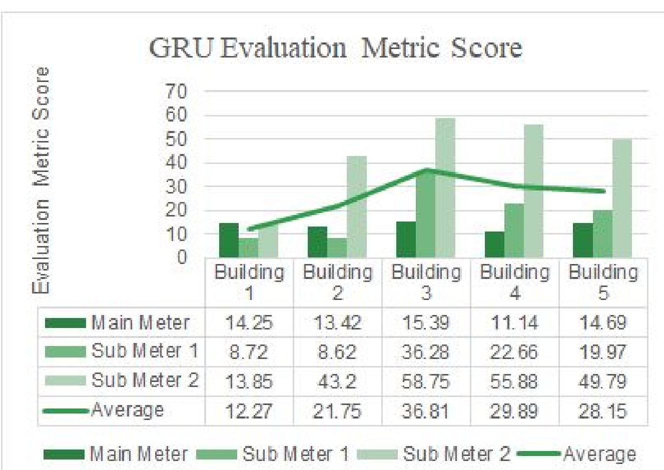
GRU is an updated version of RNN which solves the vanishing gradient problem of standard Recurrent Neural Network. GRUs remember the past sequence using forget gates and reset gates. GRUs is preferred over LSTMs as it is easy to modify, doesn't need memory units and is computationally effective.

Why GRU?

GRU is a model which can use both the time series data as well as the engineered features together, and hence in theory it should learn both the high level as well as low level patterns in our dataset, and give good results.

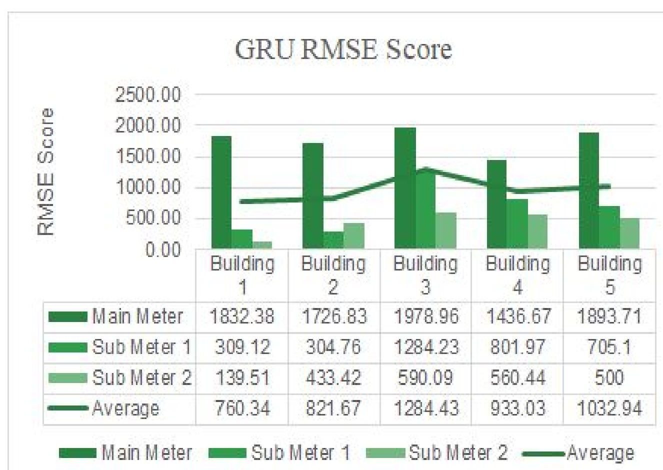
Results

Evaluation Metric on Validation Set



	Building 1	Building 2	Building 3	Building 4	Building 5
Main Meter	14.25	13.42	15.39	11.14	14.69
Sub Meter 1	8.72	8.62	36.28	22.66	19.97
Sub Meter 2	13.85	43.2	58.75	55.88	49.79
Average	12.27	21.75	36.81	29.89	28.15

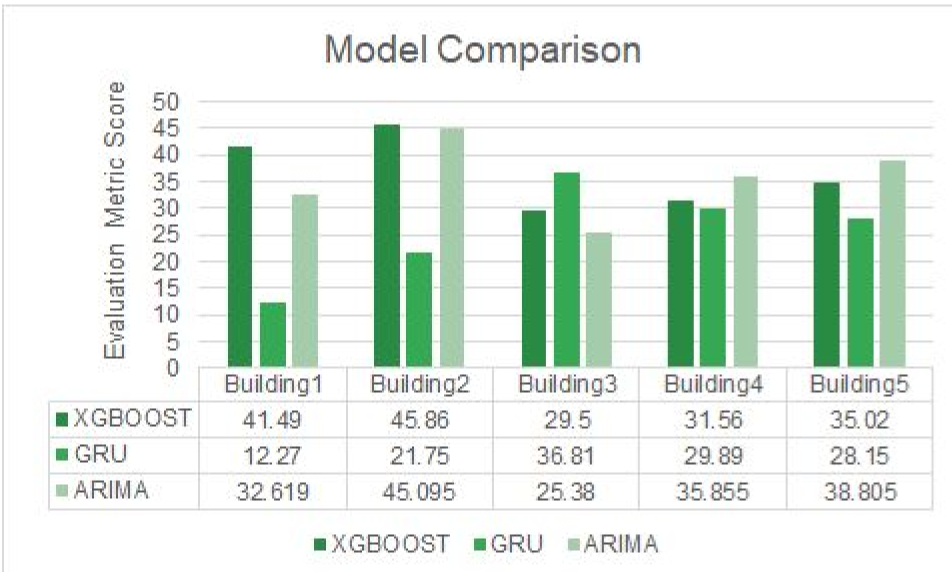
RMSE on Validation Set



	Building 1	Building 2	Building 3	Building 4	Building 5
Main Meter	1832.38	1726.83	1978.96	1436.67	1893.71
Sub Meter 1	309.12	304.76	1284.23	801.97	705.1
Sub Meter 2	139.51	433.42	590.09	560.44	500
Average	760.34	821.67	1284.43	933.03	1032.94

Comparative Analysis

The graph below shows the comparison between models on the basis of the value of the evaluation metric score on the validation set. It is clear that **GRU has a lower evaluation metric score for most of the buildings**, then that of ARIMA and XGBoost. **ARIMA has performed well in the meters where the variance is less**, while GRU is able to learn more complex patterns in the dataset. XGBoost's results are good among the other machine learning algorithms but its performance is not at par with other models which are tailor-made for sequential models.



	Building1	Building2	Building3	Building4	Building5
XGBOOST	41.49	45.86	29.5	31.56	35.02
GRU	12.27	21.75	36.81	29.89	28.15
ARIMA	32.619	45.095	25.38	35.855	38.805

Stacked Predictions

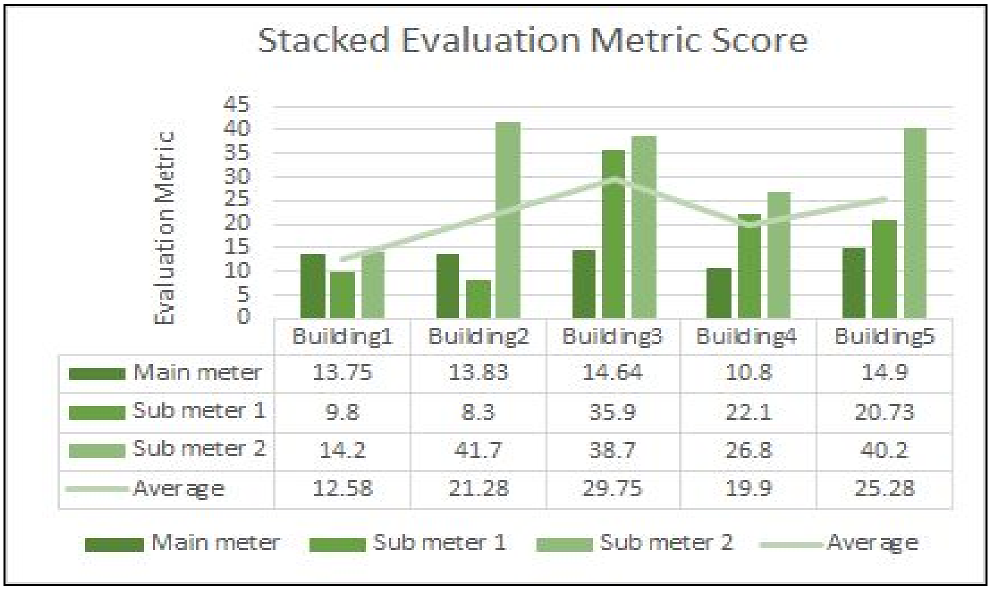
Stacked predictions allow you to make "out-of-sample" predictions and prevent misleadingly high accuracy scores. Given that the data used to train the model is different than the data on which ultimately predictions need to be delivered, stacking provides robustness to our model. We are **stacking the predictions from GRU & ARIMA Models**, and we **use the value of the evaluation metric** for the model **on the validation set** to decide the weights. Weights are determined using the formulas below, and **final predictions** are given as a **weighted average** of predictions from each model.

$$W_X = \frac{\frac{1}{x^2}}{\frac{1}{x^2}+\frac{1}{y^2}}$$

$$W_Y = \frac{\frac{1}{y^2}}{\frac{1}{x^2}+\frac{1}{y^2}}$$

Here, W_X → Weight of model X. | x → Evaluation metric score of model X.
W_Y → Weight of model Y. | y → Evaluation metric score of model Y.

Final Results on the Validation set.



Conclusion

From the results obtained, we see that there is no “One Model that fits all”. For example, GRU model consistently performs well in the case of main meter predictions. But in the case of submeters, there is no such general trend, ARIMA performs well for certain buildings while XGB performs for others. But we observe that both GRU and ARIMA consistently outperform XGBoost and this is expected because while both ARIMA & GRU are models specifically targeted towards time-series data, XGBoost isn’t.

The main meter and the submeter time series have two aspects associated with it, first the mean and second the variance of the series. We observed that the ARIMA model was adept at capturing the mean associated with the time series while the GRU model could capture variance associated with the time series. So, both these models constitute very strong predictors. Hence, to get the best of both the worlds of ARIMA & XGB, we decided to stack the predictions from both these models. Stacking of strong predictors has been consistently used to win data science competitions as well. And we have seen that it increases the accuracy in our case as well.

Annexure

Annexure A.1

(Building wise Summary Statistics)

	Building 1			Building 2			Building 3		
	Main meter	Submeter 1	Submeter 2	Main meter	Submeter 1	Submeter 2	Main meter	Submeter 1	Submeter 2
Mean	4349.49	1697.61	321.87	6592.04	2232.39	44.07	5984.49	3042.0	830.91
Std	1815.29	645.93	316.16	2498.84	951.32	48.03	2952.99	907.04	730.69
Min	61.63	0.00	0.00	2363.11	553.49	0.00	1632.34	1118.01	0.00
25%	2791.32	1200.77	0.22	4129.87	1462.72	1.08	3325.27	2329.6	27.30
50%	4090.15	1601.91	311.82	6794.70	2084.04	46.55	5653.98	2876.4	1481.79
75%	5609.02	2123.16	626.71	8337.40	2843.53	79.53	7629.98	3621.8	1509.89
Max	10684.3	4295.65	756.75	13770.9	7030.58	490.72	16414.1	5434.9	1968.79

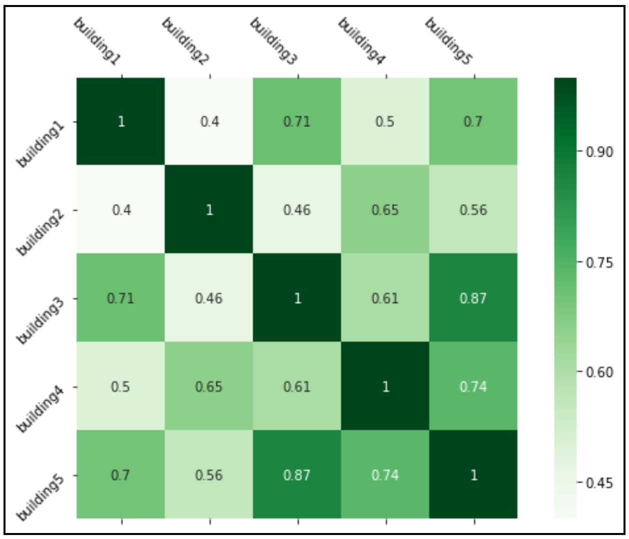
	Building 4			Building 5		
	Main meter	Submeter 1	Submeter 2	Main meter	Submeter 1	Submeter 2
Mean	7352.70	1013.17	1962.09	6200.09	2479.20	725.19
Std	4684.14	971.18	803.96	3851.81	919.21	634.29
Min	422.50	0.00	179.97	1662.09	567.38	0.00
25%	4031.30	7.25	1296.78	2964.96	1675.68	2.95
50%	5666.11	596.84	1896.11	5364.19	2408.67	1309.74
75%	8446.49	2073.69	2468.68	8773.46	3231.33	1320.16
Max	22921.3	2111.29	4290.66	19671.6	4854.52	1337.11

Annexure A.2

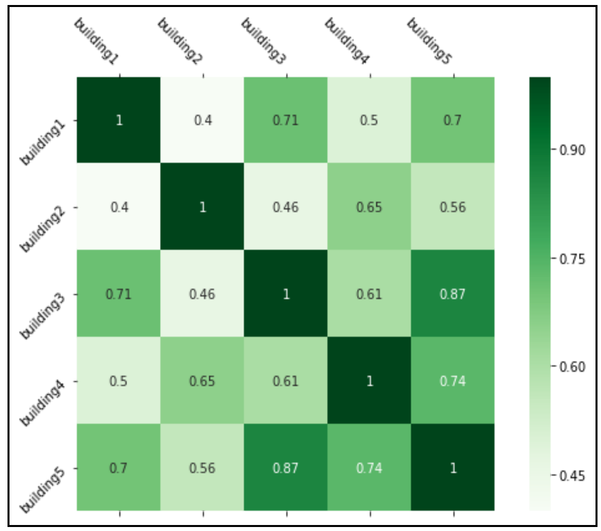
(Correlation Matrices)

Correlation between buildings for sub-meters.

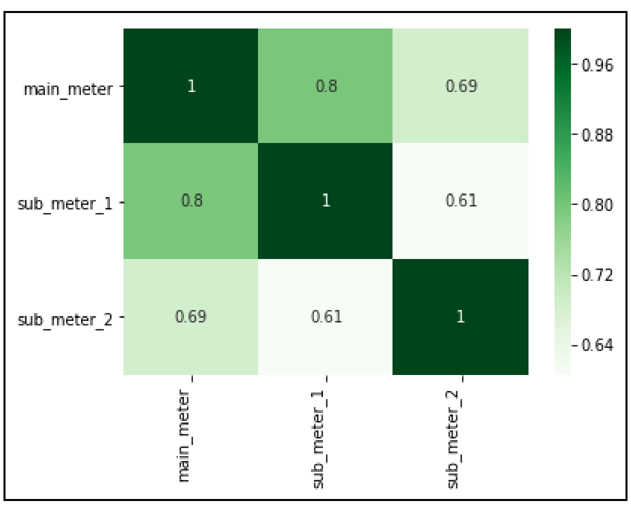
Sub-meter 1

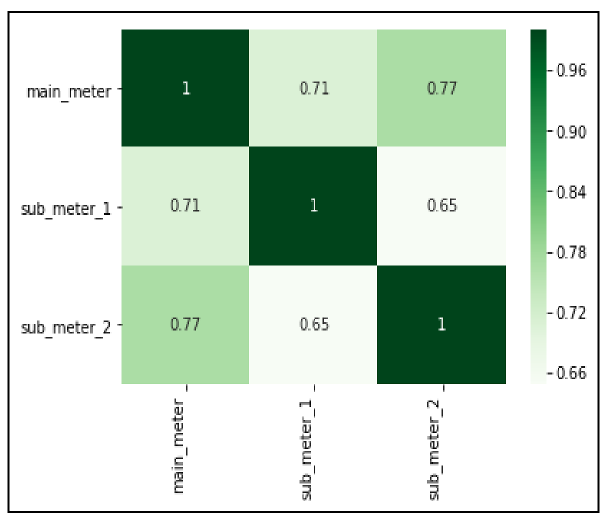


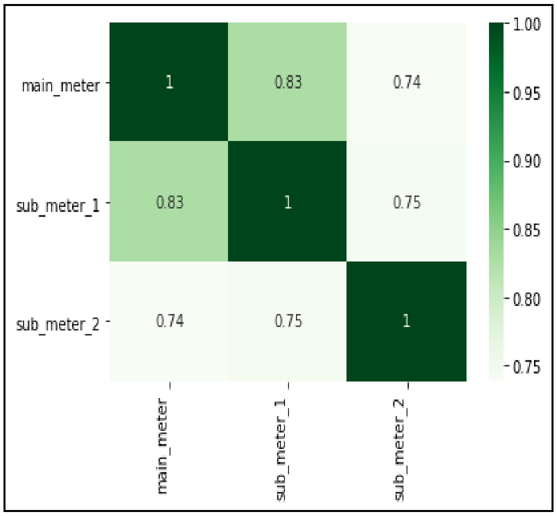
Sub-Meter 2



Correlation between meters for buildings 3, 4 and 5.

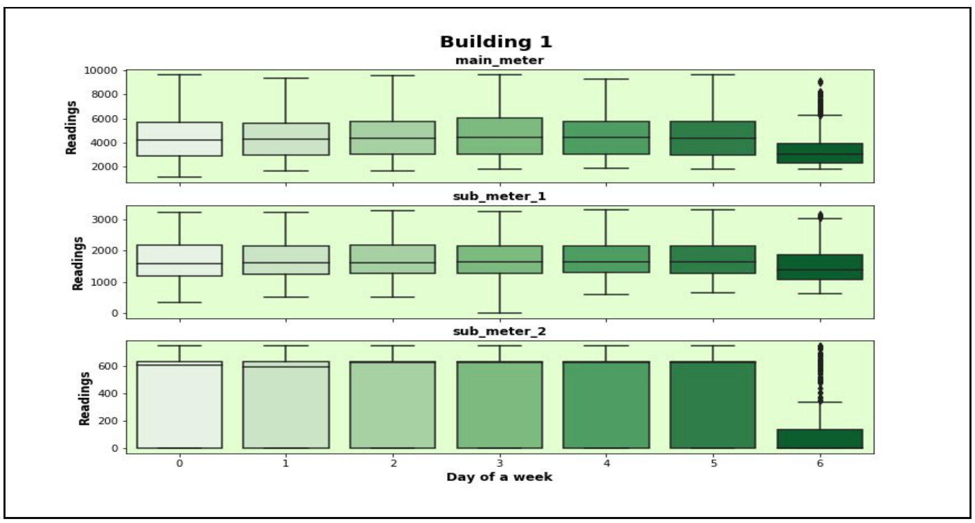


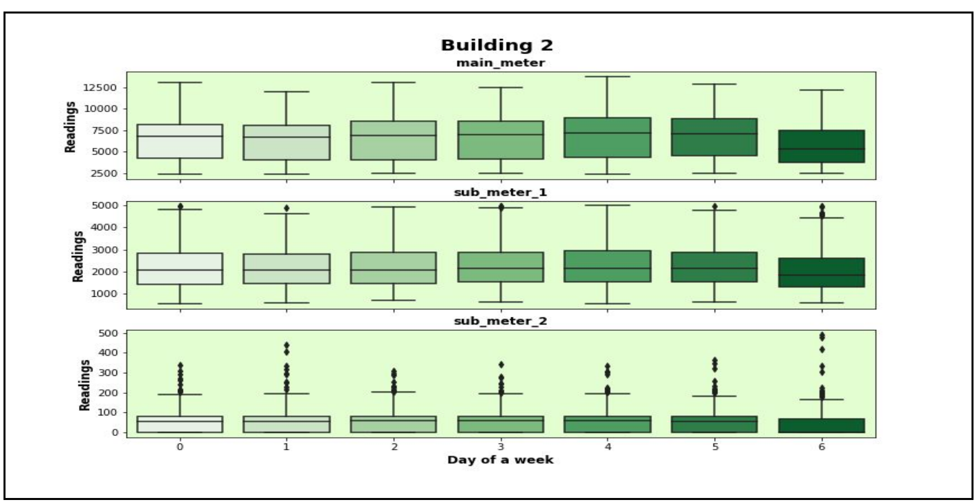


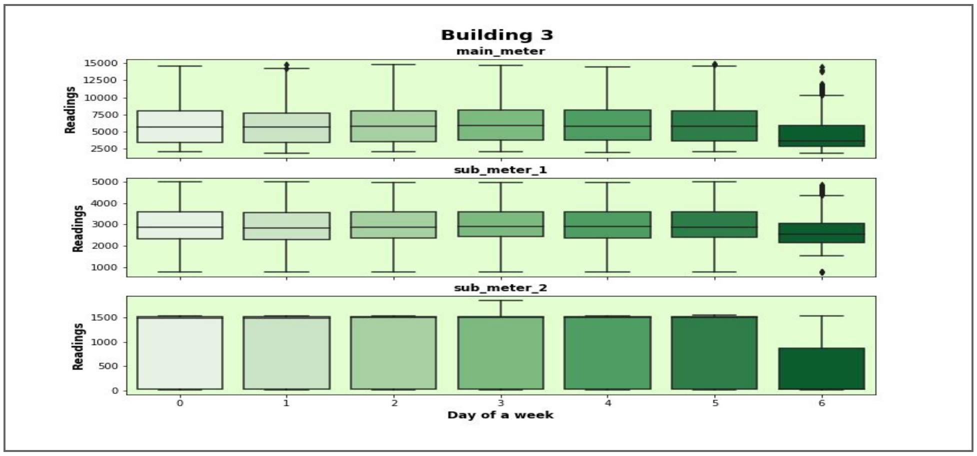


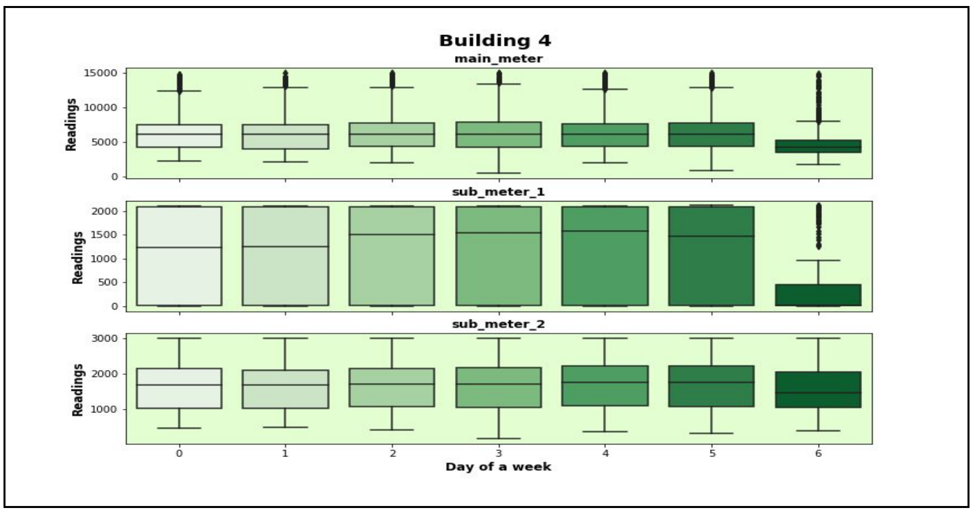
Annexure A.3

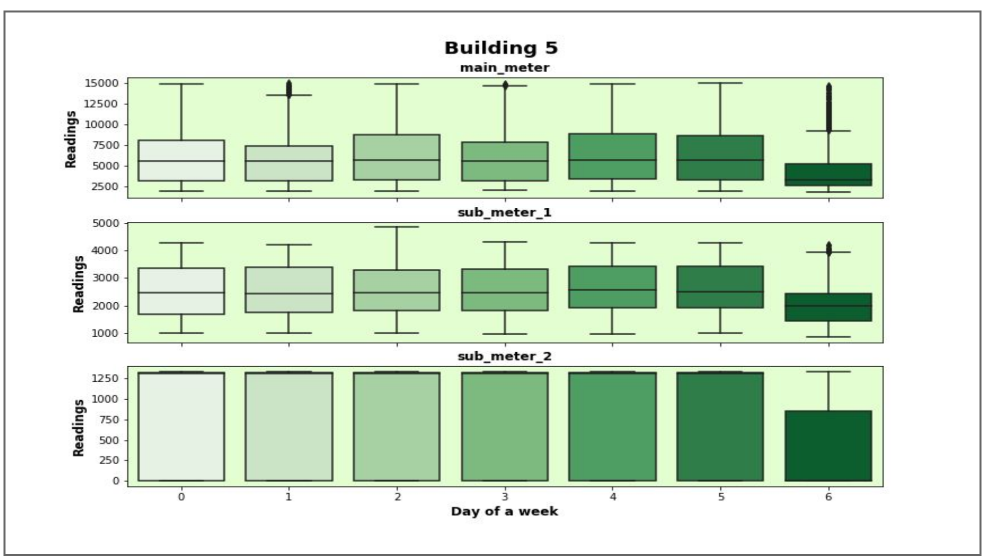
(Building-wise weekly seasonality)






Building 3
main_meter
sub_meter_1
sub_meter_2
Readings
Day of a week

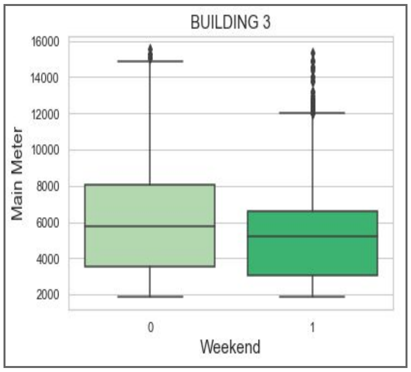

Building 4
main_meter
sub_meter_1
sub_meter_2
Readings
Day of a week

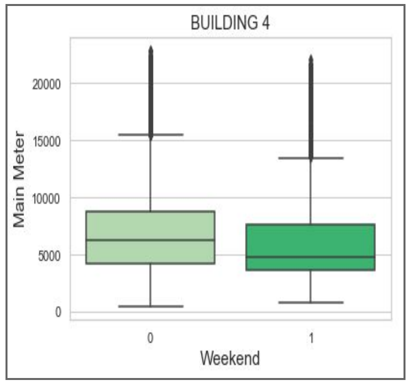

Building 5
main_meter
sub_meter_1
sub_meter_2
Readings
Day of a week

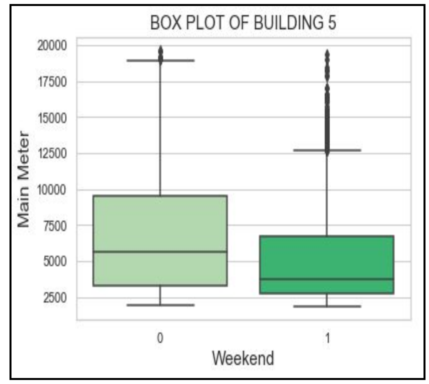
Annexure A.4

(Box plots supporting Feature Engineering)

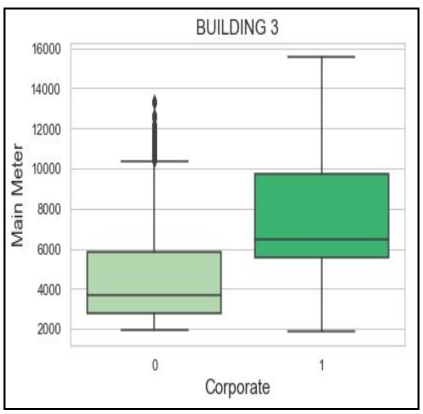
Weekend Feature

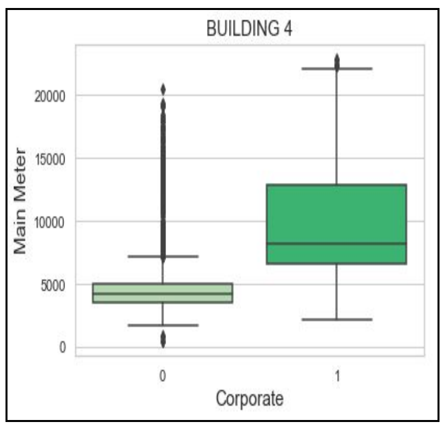


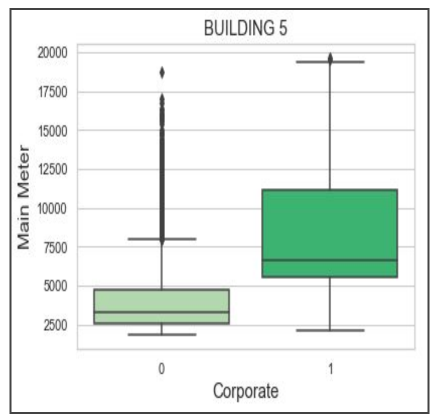




Corporate Feature

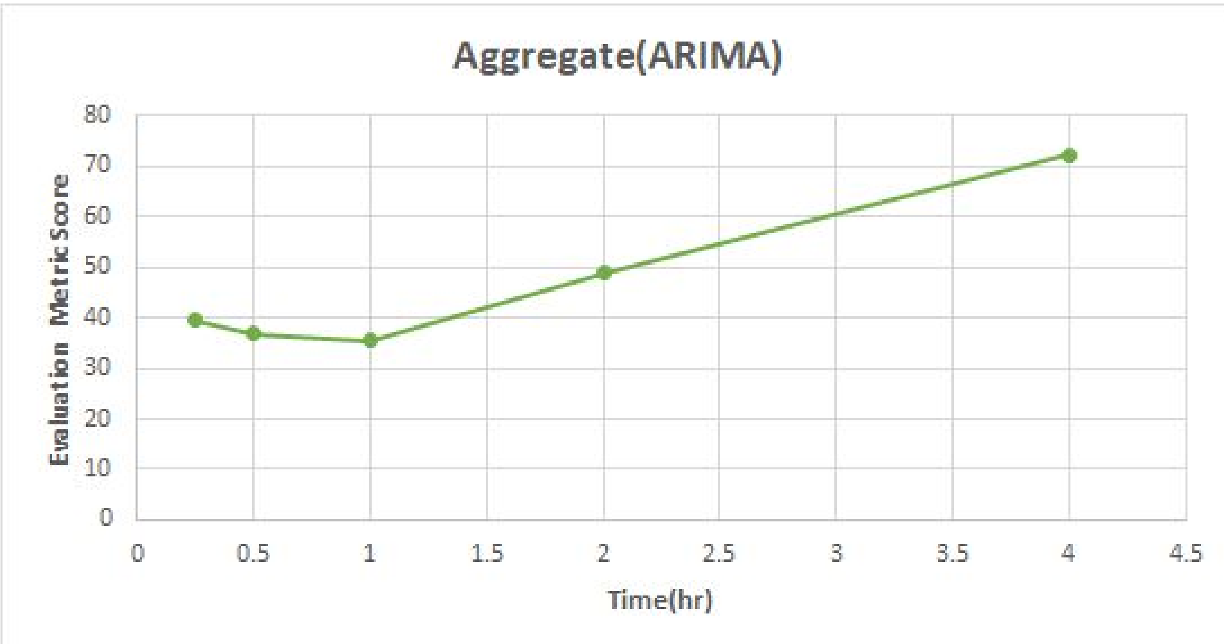


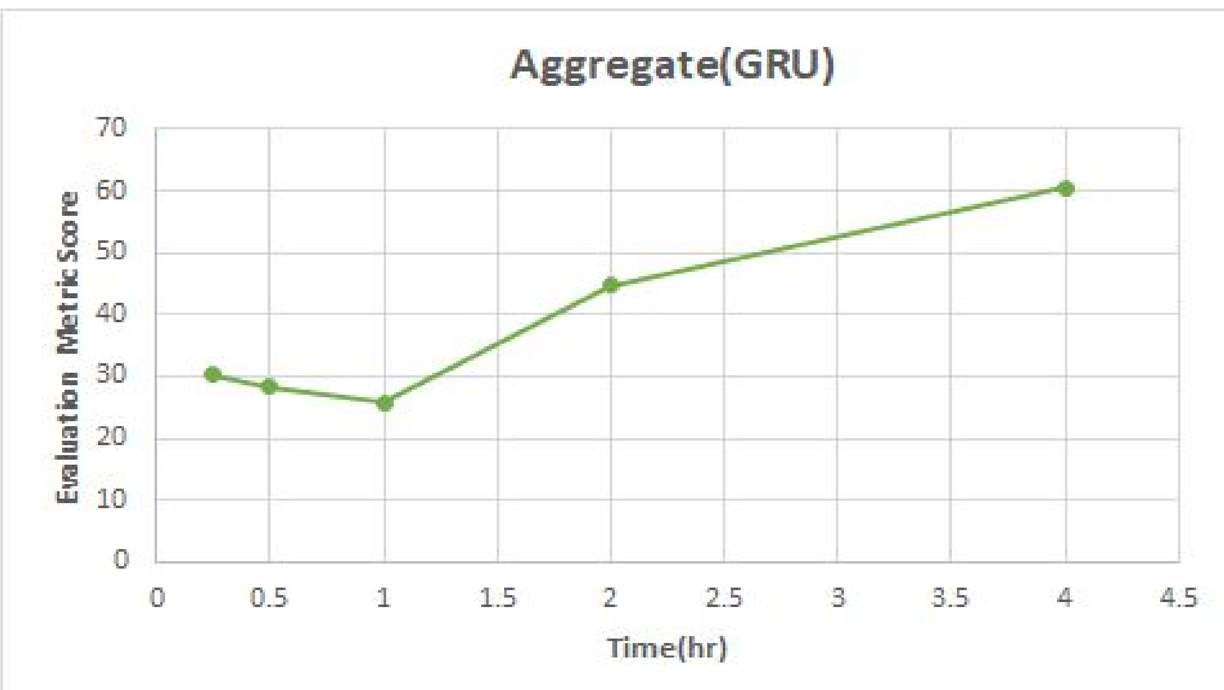




Annexure A.5

(Line plots for deciding optimal Aggregate timestamp level)





Annexure A.6

(Details of the Final model used)

ARIMA (p, d, q) MODELS

We have trained 15 different models for different meters of different buildings, the tuned parameters (p, d, q) for each of them are mentioned in the table below:

(p,d,q)	Building 1	Building 2	Building 3	Building 4	Building 5
Main Meter	(5,1,3)	(3,1,4)	(3,1,4)	(3,1,5)	(4,1,4)
Sub Meter 1	(5,1,4)	(5,1,5)	(2,1,1)	(4,0,3)	(5,1,5)
Sub Meter 2	(3,0,5)	(1,0,0)	(3,0,5)	(4,1,4)	5,0,3)

GRU Model Architecture

We have tried different configurations of GRU models, we have tried both training 15 different models for each meter of each building, and generalised building specific models too. We found that generalised building-wise model, which predicted all the three-meter readings simultaneously worked better than that of 15 different models, which is due to the high correlation between the meter readings as shown in section 2 of the report. The final architecture used is as follows: