Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 11 Issue 3, March - 2024

# Extreme Gradient Boosting (XGBoost) Model For Characterization And Forecasting Of Injection Substation Feeder Load

**Imoh, Isaiah UDofot**[1]
Department of Electrical/Electronic Engineering,
University of Uyo, Akwa Ibom, Nigeria

**Trust Christopher Oguichen**[2]
Department of Computer Engineering
Capt. Elechi Amadi Polytechnic, Rumuola
Port Harcourt, Rivers State, Nigeria
trust.oguichen@portharcourtpoly.edu.ng

***Abstract*— In this work, XGBoost model-based injection substation load prediction is presented. The case study dataset consist of AKA feeder which are part of an injection substation in Uyo Akwa Ibom State. The parameters considered are the historical load data and the feeder state. The data cleaning algorithm, data trimming analytical model, the details of the XGBoost model are presented along with the description of the case study injection substation load dataset. Specifically, 4 months hourly historical data were used in the study. The performance metric used for the model assessment is the Mean Squared Error (MSE). The entire dataset was split into 70% for training while 30% were used test set. XGBoost regression models were created for the AKA feeder ; the number of estimators were defined as 1000, the tree threshold was set to 50 while the learning rate was set to 0.001. The results of the MSE for the XGBoost model predictions is 113.19. The MSE values is slightly high. As such, it is recommended that further studies should be conducted with other machine learning models to see which model can give better prediction performance than the XGBoost.**

***Keywords*— XGBoost Model, Injection Substation, Dimension Reduction, Load Prediction, Machine Learning**

## 1.0 Introduction

Load prediction and forecasting is a quintessential requirement in the electrical industry due to its ability to provide quality information on load resource demands, hence, aid effective management of the limited available resource [1,2,3]. A proper and accurate forecast can mitigate over-estimation or under-estimation of required power system equipment [4,5]. Furthermore, the need for a precise estimation of required power system equipment calls for an accurate prediction and forecasting model which yields minimum prediction errors.

Notably, some researches carried out on electricity cost analysis and market design unanimously confirms that inaccurate predictions result in overestimated load which in turn results in exorbitant cost depending on the prediction error magnitude [6,7,8]. Other consequences for inaccurate predictions include contract retraction for market members [9,10], provision shortage [11,12] and and expensive makeup services [13]. Other than giving enhancing efficiency on the overall power market, accurate predictions give a clear insight into the power system dynamics. Accordingly, in this work, XGBoost model is used for the prediction of load demand on an injection substation [14]. The model utilized time stamped historical data of the load demand on the feeders in the injection substation to both train and validate the model. The details of the model description, the data cleaning procedure and the model training and validation procedure along with the prediction performance evaluation are presented.

## 2.0 Methodology

### 2.1 Dataset Description

A typical dataset from distribution station is used in the proposed model for prediction and of load demand. The case study dataset is obtained for AKA feeder which is part of distribution substation in Uyo, Akwa Ibom State. The parameters considered are the historical load data and the feeder dataset as shown in Figure 1. These dataset are considered as time series since they are recorded on hourly bases and each row of the data has a unique timestamp.

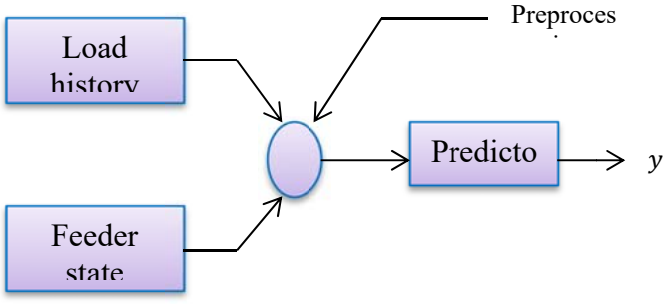


Figure 1: Required input dataset for prediction

### 2.2 Data Cleaning

Prior to data normalization, all null values for each column are compiled using Equation 1;

$$d_{null_k} = \sum_{i=0}^{N} \mathcal{F}(r_i \nexists \mathbb{R}); \quad -\infty < \mathbb{R} \leq \infty \qquad (1)$$

Where, $d_{null_k}$ denotes the output vector which contains all the null or non-numeric values in the $k^{th}$ column, $N$ denotes the total number of data , $\mathcal{F}$ is the filter function, $r_i$ is the $i^{th}$ data point to be tested, while $\mathbb{R}$ is the real number space which spans between $-\infty$ and $\infty$. If $r_i$ is not within the range of $\mathbb{R}$, then $r_i$ is appended to the $d_{null_k}$ vector. The null percentage for each data column can be computed as:

$$\aleph_k = \frac{d_{null_k}}{N} \qquad (2)$$

Where $\aleph_k$ is the null percentage which is computed for the $k^{th}$ column, and $N$ is the total data points. The valid columns are retrieved using the criteria that $\aleph_k$ must be less than or equal to 10% ($\aleph_k \leq 10$). It is assumed that if null percentage exceeds 10% in any given column, then the predicted output may not be reliable. Another important task to perform is to ensure that all data fits into its correct data type. Various data types that may exist include string, float, datetime, integer, and Boolean. However, due to the nature of the data used in this work , only two data types are considered which are: datetime and float. Datetime is used for the date column which gives information on when the data was captured. This column is also referred to as the unique id (UID) columns since no two columns have the same time stamp. On the other hand, the float data type is used for the historical load data and the feeder state since these two parameters record real numeric values. The data cleaning algorithm is presented in Algorithm 1

**Algorithm 1: Data cleaning algorithm**

1: **Begin**

2: **Initialize** $\aleph_k$, $d_{null_k}$, $\mathbb{Z} \rightarrow Normalized\ data\ output$

3: **Require** $\mathbb{C} \rightarrow Raw\ data\ input$

4: **foreach** data column $d_k$ in $\mathbb{C}$

5: Compute $d_{null_k}$ based on Equation 1

6: Compute $\aleph_k$ based on Equation 2

7: **if** $\aleph_k \leq 10$ **then**

8: Set all null entries in $d_k$ to zero

9: append $d_k$ to $\mathbb{Z}$

10: **endif**

11: **return** $\mathbb{Z}$

12: **end for**

13: **end**

### 2.3 Load prediction Based on XGBoost Model

In this work, XGBoost model is used to predict the load on the substation. In the XGBoost model, multiple weak learners are amalgamated into a common strong learner. This property reduces the loss function, $L_{xgb}$. Supposed the dataset to be trained is defined as $\mathbb{D}_T = \{x_i, y_i\}^N$, XGBoost model must obtain the approximation $\hat{F}(x)$ for the mapping function $F(x)$ which maps the input vector $x$ to the corresponding output vector $y$. In this case the loss function $L_{xgb}$ must be optimized $L_{xgb} \rightarrow L\{y, F(x)\}$. An extensive approximation $F^*(x)$ can be computed as a weighted aggregate of functions.

$$F_j(x) = F_{j-1}(x) + \omega_j \mathbb{Q}_j(x) \qquad (3)$$

Where, $\omega_j$ denotes weight of the $j^{th}$ function $\mathbb{Q}_j(x)$. The function list represents the decision tree. An iterative method is used to formulate the approximations. For the initial function, the approximation can be obtained as:

$$F_0(x) = arg\ \min_{\alpha} \sum_{i=1}^{N} L(y_i, \alpha) \qquad (4)$$

Where, $\alpha$ denotes an hyper parameter. The successive functions must minimize

$$\{\omega_j, \mathbb{Q}_j(x)\} = arg\ \min_{\omega,\mathbb{Q}} \sum_{i=1}^{N} L\left(y_i, F_{j-1}(x_i) + \omega \mathbb{Q}(x_i)\right) \qquad (5)$$

For the gradient drop optimization of the mapping function, every $\mathbb{Q}_j$ is considered as a gradient step. This requires that $\mathbb{Q}_j$ must be trained with the most recent dataset $\mathbb{D}_T$ where

$$\mathbb{D}_T = \{x_i, r_{ij}\}_{i=1}^{N} \qquad (6)$$

Where, $r_{ij}$ denotes the spurious residual and can be computed as:

$$r_{ij} = \left[\frac{\partial L(y_i, F(x))}{\partial F(x)}\right]_{F(x)=F_{j-1}(x)} \qquad (7)$$

The gradient drop step can be damped in order to normalize the gradient boost. The damper can expressed as:

$$F_j(x) = F_{j-1}(x) + v\omega_j \mathbb{Q}_j(x) \qquad (8)$$

In this work , $v = 0.1$, this lower value is selected to improve the learning rate. The decision tree complexity can be scaled down using the dissimilarity of the loss function:

$$L_{xgb} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{j=1}^{J} (\mathbb{Q}_j) \qquad (9)$$

$$\mathbb{Q}_j = \sigma L_T + \frac{\|\varphi\|^2}{2} \qquad (10)$$

Where, $\sigma$ denotes the loss reduction gain controller, $L_T$ denotes the number of leaves for the specified tree, and $\varphi$ denotes the outcome of the leaves. It should be note that $\sigma$ is inversely proportional to tree complexity. In other words, tree complexity is reduced as $\sigma$ is increased. In order to improve the learning rate, the XGBoost model mitigates the difficulty of computing the best segment. The segmentation function lists all segment members and chooses the one that has the best gain. This implies that for every node, there must be a search through over the listed properties to compute the best segment.

## 3. Results and Discussion

The historical load dataset for AKA feeder which is part of substation in Uyo metropolis in Akwa Ibom State is used. The dataset used in the study where presented in CSV file format and those historical data are captured on hourly basis for a period of 4 months (specifically May to August

of 2022. The Mean Squared Error (MSE) is used for performance evaluation.

From the raw data collated, the following columns were extracted: "TIME", "AKA", "FDR.1", as shown in Figure 2 where AKA represents data for AKA feeder. The "TIME" column Figure 2 is used as the unique index for referencing each row of data. The un-scaled data slice for AKA dataset is shown in Figure 3 while visualization of the historical load and feeder state for AKA is shown in Figure 4. The entire dataset was split into 70% for training while 30% were used as test set, as shown in Figure 5 for AKA feeder.

XGBoost regression models were created for the AKA feeder dataset and the number of estimators which represents the number of boosting trees were defined as 1000. Before fitting the models on each of the training set, the features of the training set were extracted. These feature include the seasonal pattern of the dataset. The tree threshold was set to 50. This means that if the test set fails to improve after 50 trees, the model evaluation will stop. The learning rate was set to 0.001 to improve the learning process.

| TIME | TIME | AKA | FDR.1 | KVA | PF | RP |
|---|---|---|---|---|---|---|
| 2022-05-01 01:00:00 | 01/05/2022 01:00 | 150 | 2.5 | 180 | 3 | 11.1 |
| 2022-05-01 02:00:00 | 01/05/2022 02:00 | 150 | 2.5 | 180 | 3 | 11.1 |
| 2022-05-01 03:00:00 | 01/05/2022 03:00 | 150 | 2.5 | 180 | 3 | 11.1 |
| 2022-05-01 04:00:00 | 01/05/2022 04:00 | 150 | 2.5 | 180 | 3 | 11.1 |
| 2022-05-01 05:00:00 | 01/05/2022 05:00 | 150 | 2.5 | LS | LS | 11.1 |
| ... | ... | ... | ... | ... | ... | ... |
| 2022-08-25 20:00:00 | 25/08/2022 20:00 | L/S | L/S | 10.8 | 0.9 | 0.3 |
| 2022-08-25 21:00:00 | 25/08/2022 21:00 | L/S | L/S | 10.8 | 0.9 | 0.3 |
| 2022-08-25 22:00:00 | 25/08/2022 22:00 | L/S | L/S | 10.8 | 0.9 | 3.9 |

Figure 2 : A portion of cleaned dataset

| TIME | AKA | FDR_AKA |
|---|---|---|
| 2022-05-01 01:00:00 | 150.0 | 2.5 |
| 2022-05-01 02:00:00 | 150.0 | 2.5 |
| 2022-05-01 03:00:00 | 150.0 | 2.5 |
| 2022-05-01 04:00:00 | 150.0 | 2.5 |
| 2022-05-01 05:00:00 | 150.0 | 2.5 |
| ... | ... | ... |
| 2022-08-25 20:00:00 | 0.0 | 4.3 |
| 2022-08-25 21:00:00 | 0.0 | 4.3 |
| 2022-08-25 22:00:00 | 0.0 | 4.3 |
| 2022-08-25 23:00:00 | 0.0 | 4.3 |
| 2022-08-26 00:00:00 | 0.0 | 4.3 |

2808 rows × 2 columns

Figure 3: Un-scaled data slice for AKA dataset

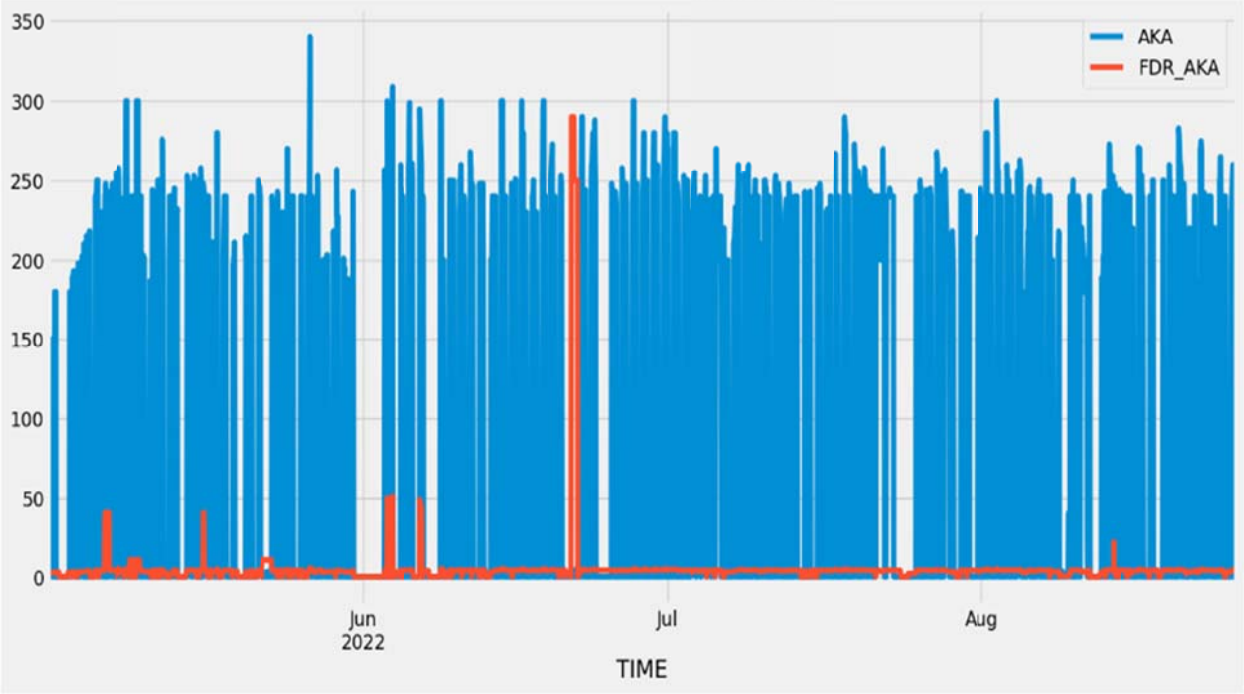


Figure 4 Visualization of the historical load and feeder state for AKA

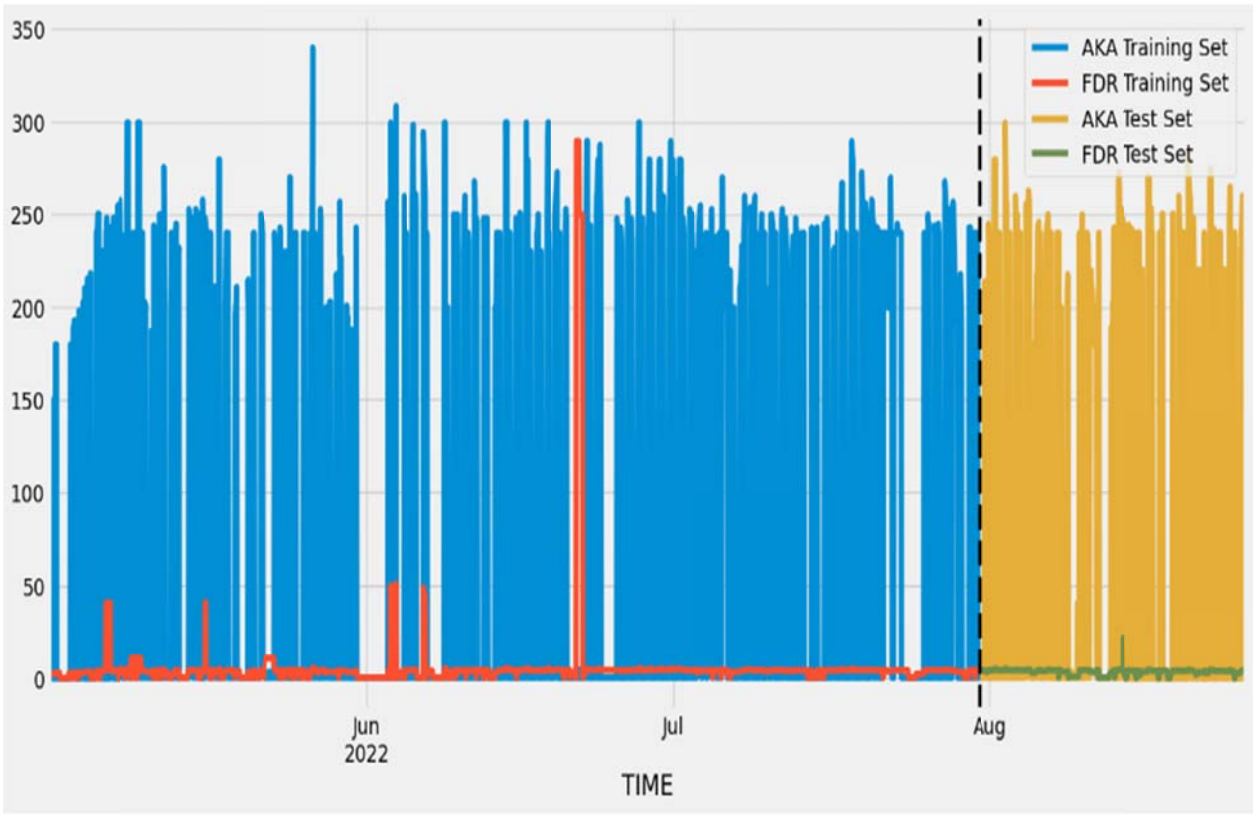


Figure 5: Training/test set splitting for AKA dataset

The feature importance for the model on the AKA is presented in Table 1. The important features are plotted in Figure 6 for AKA feeder. The results extracted from the features importance on the feeder dataset show that the XGBoost model mainly use the hourly property of the dataset to make predictions followed by the FDR data property and then the days of the week data property. Notably, the week property is meant to feed the model with the seasonal pattern of the datasets.

Now the forecast on the test set was performed with the training model and the results are shown in Figure 7 for the AKA feeder dataset. From the results presented, the MSE for the XGBoost model predictions is 113. 19 for the AKA feeder.

Table 1: Feature importance for the model on AKA and IBB feeder dataset

| Column Property | AKA |
|---|---|
| FDR | 0.400109 |
| Hour | 0.532639 |
| Days of the week | 0.067252 |

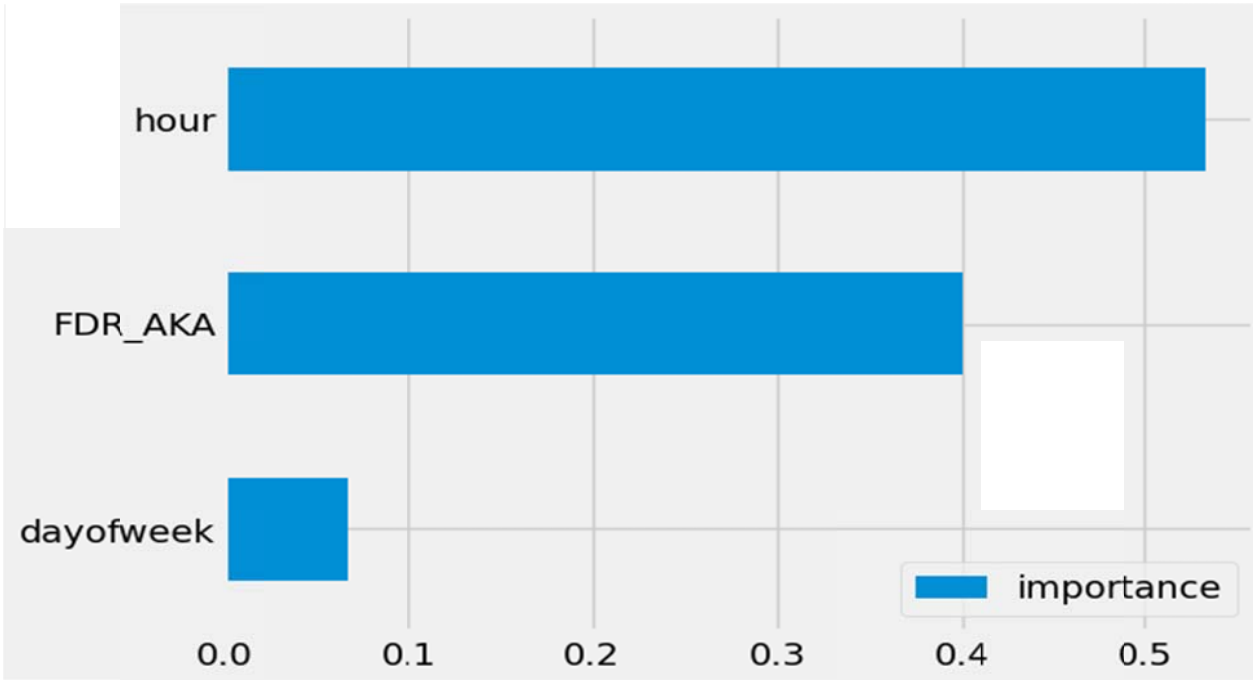


Figure 6: Feature importance for the model on AKA feeder dataset

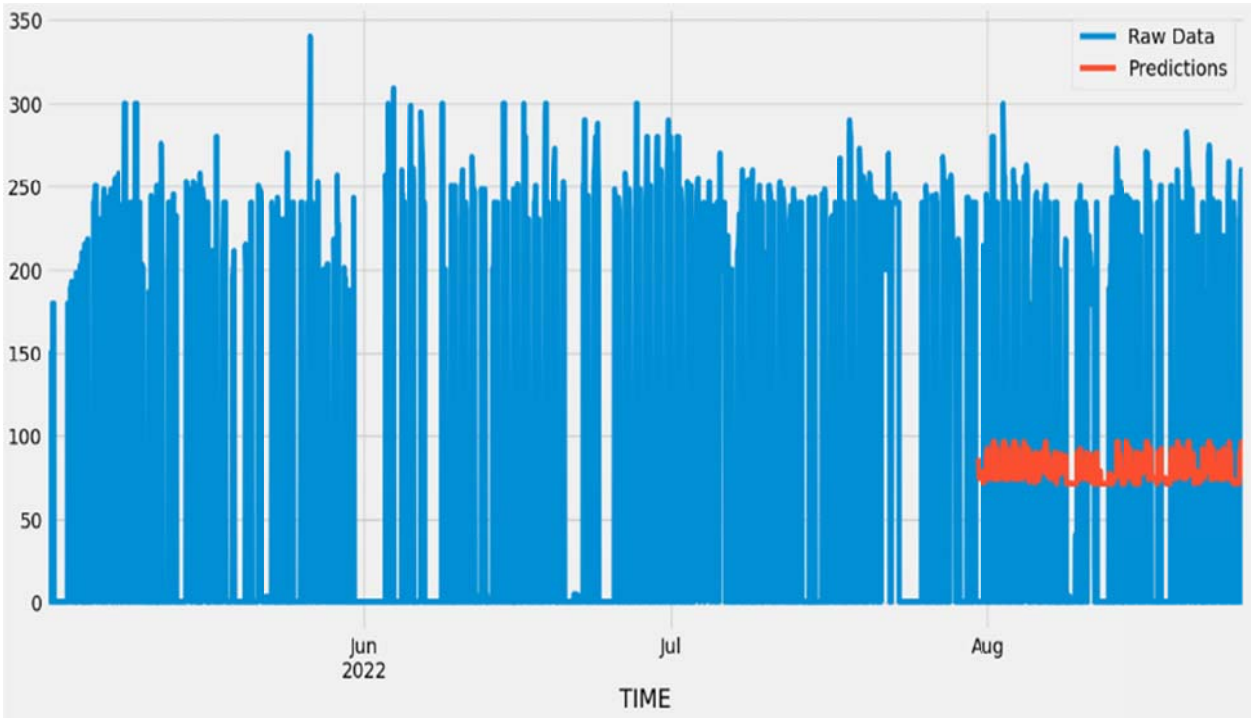


Figure 7: Model prediction for AKA data using XGBoost

## 4. Conclusion

Prediction of load at a feeder of an injection substation using XGBoost model is presented. The model description, dataset preprocessing, model training and the model prediction process are presented along with the performance metric which is mean square error. The case study data consist of time stamped hourly data of a feeder in an injection substation. The results showed a relatively high root mean error. As such, it is recommended that for the given injection substation load dataset, other machine learning models should be used to model and predict the load data to see if better prediction performance can be achieved.

## References

1. Bhuiyan, E. A., Hossain, M. Z., Muyeen, S. M., Fahim, S. R., Sarker, S. K., & Das, S. K. (2021). Towards next generation virtual power plant: Technology review and frameworks. *Renewable and Sustainable Energy Reviews*, *150*, 111358.

2. Ravi, S. S., & Aziz, M. (2022). Utilization of electric vehicles for vehicle-to-grid services: Progress and perspectives. *Energies*, *15*(2), 589.
3. Singh, A., & Saroha, S. (2022, December). A Study on Solar Forecasting using Artificial Neural Networks Technique. In *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (pp. 1635-1641). IEEE.
4. Saxena, D., & Singh, A. K. (2021). Workload forecasting and resource management models based on machine learning for cloud computing environments. *arXiv preprint arXiv:2106.15112*.
5. Chen, Y., Ye, T., Du, H., Shan, J., & Gao, H. (2023, April). Power Grid Dispatch with High Proportion of New Energy Integration. In *2023 Panda Forum on Power and Energy (PandaFPE)* (pp. 1962-1966). IEEE.
6. Emde, A., Märkle, L., Kratzer, B., Schnell, F., Baur, L., & Sauer, A. (2023). Effects of Load Forecast Deviation on the Specification of Energy Storage Systems. *Designs*, *7*(5), 107.
7. Zhao, X., Gao, W., Qian, F., & Ge, J. (2021). Electricity cost comparison of dynamic pricing model based on load forecasting in home energy management system. *Energy*, *229*, 120538.
8. Dudek, G., Piotrowski, P., & Baczyński, D. (2023). Intelligent Forecasting and Optimization in Electrical Power Systems: Advances in Models and Applications. *Energies*, *16*(7), 3024.
9. Liao, Z., Pan, H., Huang, X., Mo, R., Fan, X., Chen, H., ... & Li, Y. (2021). Short-term load forecasting with dense average network. *Expert Systems with Applications*, *186*, 115748.
10. Foldvik Eikeland, O., Bianchi, F. M., Apostoleris, H., Hansen, M., Chiou, Y. C., & Chiesa, M. (2021). Predicting energy demand in semi-remote Arctic locations. *Energies*, *14*(4), 798.
11. Mir, A. A., Alghassab, M., Ullah, K., Khan, Z. A., Lu, Y., & Imran, M. (2020). A review of electricity demand forecasting in low and middle income countries: The demand determinants and horizons. *Sustainability*, *12*(15), 5931.
12. Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., ... & Ziel, F. (2022). Forecasting: theory and practice. *International Journal of Forecasting*, *38*(3), 705-871.
13. Fildes, R., Ma, S., & Kolassa, S. (2022). Retail forecasting: Research and practice. *International Journal of Forecasting*, *38*(4), 1283-1318.
14. Sukumar, S., Pindoriya, N., Ahuja, A., & Verma, R. (2020, December). Forecast and energy management system (F-EMS) framework for optimal operation of sewage treatment plants. In *2020 21st National Power Systems Conference (NPSC)* (pp. 1-6). IEEE.