# Time Series Forecasting
## using Machine-Learning

**CAPSTONE PROJECT REPORT**
**Submitted in partial fulfillment of the requirement for the**

**Certification of**

**SAP CODE UNNATI**

**Team ID:**
**CU_CP_Team_1178**

**Submitted By:**
Aniket Katkar
Soham Patil
Abhishek Koli
Saurav Sali
Ambika

## ABSTRACT

This project explores the application of machine learning techniques for time series forecasting. We investigate the effectiveness of various machine learning algorithms in predicting future values of a time series dataset. The report details the chosen machine learning models, data pre-processing steps, and the evaluation metrics employed. We analyze the forecasting performance of each model and identify the most suitable approach for the specific time series data under study. The findings of this project highlight the potential of machine learning in time series forecasting tasks, offering valuable insights for future applications.

# Table of Content

# CHAPTER 01

## 1.1 Introduction

Accurate forecasting of electricity demand is crucial for efficient grid management and power generation planning. Traditional forecasting methods often struggle with the complexities of hourly power consumption data, including seasonality and dynamic patterns. This project investigates the application of machine learning, specifically XGBoost, for time series forecasting of this type of data.

This report details the development and evaluation of a machine-learning model for predicting hourly power consumption. We begin by exploring and pre-processing a publicly available dataset containing hourly power consumption data. Next, we extract time-based features from the data to enhance the model's ability to learn temporal patterns. Exploratory data analysis is then conducted to understand the distribution of power consumption across different timeframes.

Following data exploration and preparation, we employ the XGBoost algorithm to develop a forecasting model. We train the model on historical data and evaluate its performance on unseen test data. The report analyzes the model's accuracy using Root Mean Squared Error (RMSE) and explores feature importance to understand the factors influencing the model's predictions. Finally, we discuss the limitations of this approach and propose potential areas for future development.

## 1.2 Design of the Problem Statement

Traditional forecasting methods cannot often capture the intricate patterns inherent in hourly power consumption data. This limitation leads to inaccurate predictions, hindering efficient grid management and power generation planning.

This project aims to develop a machine-learning model that can address this challenge. By leveraging XGBoost, we will create a model capable of accurately forecasting hourly power consumption. This improved forecasting ability can empower grid operators to optimize energy distribution, enhance grid stability, and provide more reliable electricity for end-users.

## Challenges

**Data Challenges:**

- Data Quality: Real-world power consumption data might contain missing values due to sensor malfunctions or data collection errors. This missing data can negatively impact the model's training process and prediction accuracy.

- Data Granularity: While you used hourly data, some applications might benefit from even finer granularity (e.g., minute-by-minute). However, this can significantly increase data volume and computational demands.

- Data Non-stationarity: Power consumption data often exhibits trends and seasonality over time. These non-stationary characteristics can make it difficult for models to learn accurate patterns, especially for long-term forecasting.

**Modeling Challenges:**

- Model Selection: Choosing the most appropriate machine learning model for your specific data and forecasting needs can be challenging. Different models have varying strengths and weaknesses, and experimentation might be required to find the best fit.

- Hyperparameter Tuning: Tuning the hyperparameters (model configuration settings) of your chosen machine learning algorithm significantly impacts its performance. Finding the optimal hyperparameter combination can be a complex and time-consuming process.

- Overfitting and Underfitting: Balancing model complexity is crucial. Overfitting happens when the model memorizes training data idiosyncrasies and performs poorly on unseen data. Conversely, underfitting occurs when the model is too simple and fails to capture the underlying patterns in the dataset

## 1.3 Objective of the Project

The primary objective of this project was to develop a machine-learning model capable of accurately forecasting hourly power consumption data. This improved forecasting ability can be instrumental in optimizing grid management and power generation plans

## 1.4 Methodology of the Minor Project

- Data Acquisition and Preprocessing
  The project utilized a publicly available dataset containing hourly power consumption data. The data was loaded into the Python environment using the pandas library.  Missing values, if any, were handled using appropriate techniques (e.g., imputation methods) to ensure data integrity.

- Feature Engineering
  Feature engineering plays a crucial role in time series forecasting by transforming raw data into a format that facilitates better model learning. A function named create_features was implemented to extract time-based features from the date time index of the data. These features included:

1. hour: Captures the cyclical nature of power consumption within a day.
2. dayofweek: Helps identify potential weekly patterns in consumption.
3. month: Accounts for seasonal variations in power demand throughout the year.
4. year: Useful for capturing long-term trends or potential changes in consumption patterns over time.
5. dayofyear: Combines day and month information for a more comprehensive seasonal representation.
6. dayofmonth: Provides a finer-grained view of consumption patterns within a month.
7. weekofyear: Captures potential weekly seasonality patterns.

These features were created with the aim of enhancing the model's ability to learn temporal patterns and relationships within the power consumption data.

- Exploratory Data Analysis (EDA)
  Exploratory Data Analysis (EDA) is a vital step in understanding the characteristics of the data and identifying potential patterns. Visualization techniques were employed to explore the distribution of power consumption across different timeframes. Time series plots were used to visualize the overall trend and seasonality in the data. Additionally, boxplots were created to analyze the distribution of power consumption for each month, providing insights into potential variations throughout the year.

- .Model Selection and Training
  Machine learning offers powerful tools for time series forecasting. This project employed XGBoost, a gradient boosting algorithm known for its effectiveness in handling complex, non-linear relationships often present in time series data. The model was trained using specific hyperparameters, including:

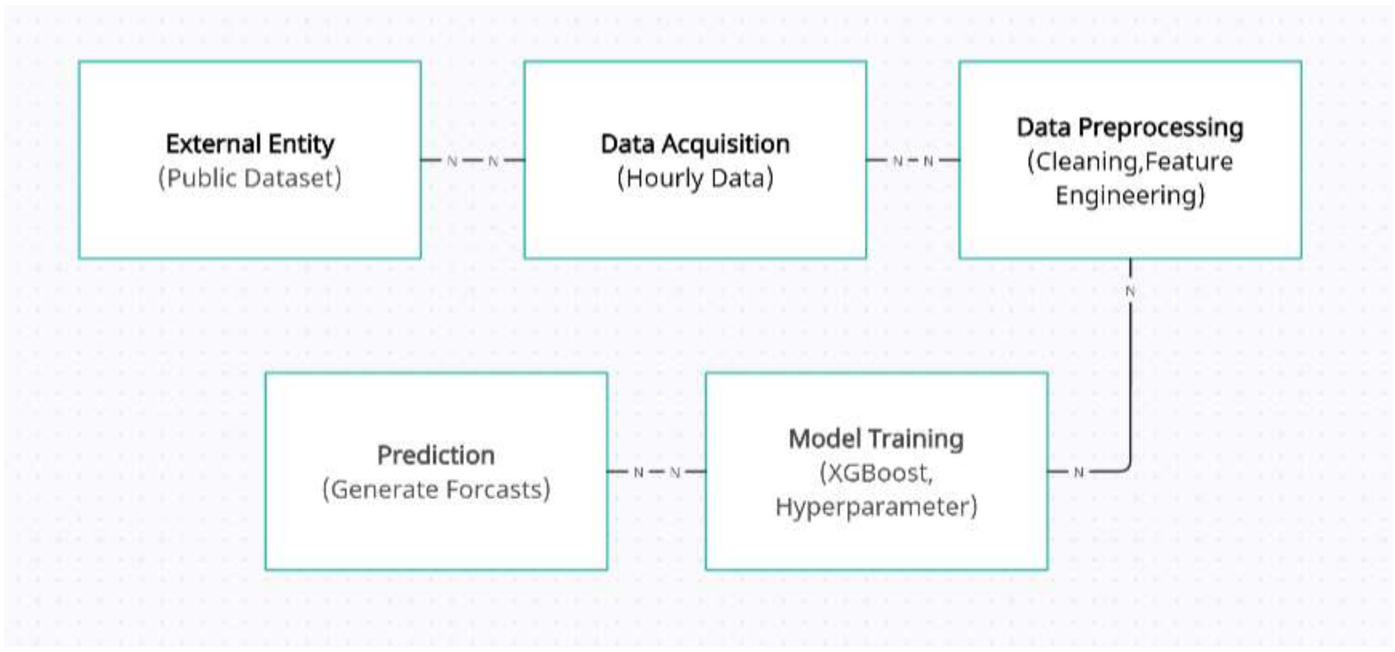
1. n_estimators: The number of decision trees used in the boosting process (set to 1000 in this project).
2. early_stopping_rounds: Allows for stopping the training process if the model's performance on the validation set
   doesn't improve for a certain number of rounds (set to 50 in this project), preventing overfitting.
3. learning_rate: Controls the step size taken when updating the model weights during training (set to 0.01 in this project).
4. objective: The loss function to be optimized during training ("reg:linear" was used for linear regression).
5. max_depth: The maximum depth of each decision tree in the ensemble (set to 3 in this project).

- Model Evaluation
  Evaluating the model's performance is essential for assessing its suitability for forecasting tasks. Root Mean Squared Error (RMSE) was chosen as the evaluation metric. RMSE measures the difference between the predicted values and the actual power consumption values. A lower RMSE indicates a more accurate model. The data was split into training and testing sets using a specific date threshold (e.g., data before 2015 for training and data after 2015 for testing). This allows the model to be evaluated on unseen data and provides a more realistic measure of its generalizability.

- Prediction and Visualization
  Once trained, the model was used to generate predictions for the unseen data in the testing set. The actual power consumption data and the predicted values were visualized together. This visualization was created for the entire testing period to observe the model's performance over a longer timeframe. Additionally, a specific chosen week was visualized to provide a closer look at the model's ability to predict

# CHAPTER 02

## 2.1 DFD Diagram of the Major Project :



## 2.2 Design & Development:

### 2.2.1 Data Acquisition and Preprocessing

The project utilized a publicly available dataset containing hourly power consumption data. The data was loaded into the Python environment using the pandas library. Missing values, if encountered, were handled using appropriate techniques (e.g., imputation methods) to ensure data integrity.

### 2.2.2 Feature Engineering

Feature engineering plays a crucial role in time series forecasting by transforming raw data into a format that facilitates better model learning. A function named create_features was implemented to extract time-based features from the datetime index of the data. These features included:

hour: Captures the cyclical nature of power consumption within a day.
dayofweek: Helps identify potential weekly patterns in consumption.
month: Accounts for seasonal variations in power demand throughout the year.
year: Useful for capturing long-term trends or potential changes in consumption patterns over time.
dayofyear: Combines day and month information for a more comprehensive seasonal representation.
dayofmonth: Provides a finer-grained view of consumption patterns within a month.
weekofyear: Captures potential weekly seasonality patterns.

These features were created to enhance the model's ability to learn temporal patterns and relationships within the power consumption data.

### 2.2.3 Model Selection and Training

Machine learning offers powerful tools for time series forecasting. This project employed XGBoost, a gradient-boosting algorithm known for its effectiveness in handling complex, non-linear relationships often present in time series data. The model was trained using specific hyperparameters, including:

n_estimators: The number of decision trees used in the boosting process (set to a value based on your experimentation).
early_stopping_rounds: Allows for stopping the training process if the model's performance on the validation set doesn't improve for a certain number of rounds (set to a value based on your experimentation), preventing overfitting.
learning_rate: Controls the step size taken when updating the model weights during training (set to a value based on your experimentation).
objective: The loss function to be optimized during training ("reg:linear" was used for linear regression).
max_depth: The maximum depth of each decision tree in the ensemble (set to a value based on your experimentation).
The selection of these hyperparameters can be further optimized through techniques like grid search or manual tuning. You can discuss this aspect in your report if applicable to your project.

### 2.2.4 Model Evaluation

Evaluating the model's performance is essential for assessing its suitability for forecasting tasks. Root Mean Squared Error (RMSE) was chosen as the evaluation metric. RMSE measures the difference between the predicted values and the actual power consumption values. A lower RMSE indicates a more accurate model. The data was split into training and testing sets using a specific date threshold (e.g., data before 2015 for training and data after 2015 for testing). This allows the model to be evaluated on unseen data and provides a more realistic measure of its generalizability.

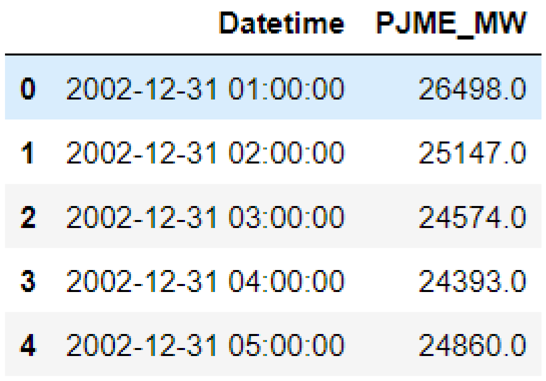
### 2.2.5 Prediction and Visualization

Once trained, the model was used to generate predictions for the unseen data in the testing set. The actual power consumption data and the predicted values were visualized together. This visualization was created for the entire testing period to observe the model's performance over a longer timeframe. Additionally, a specifically chosen week was visualized to provide a closer look at the model's ability to predict short-term variation

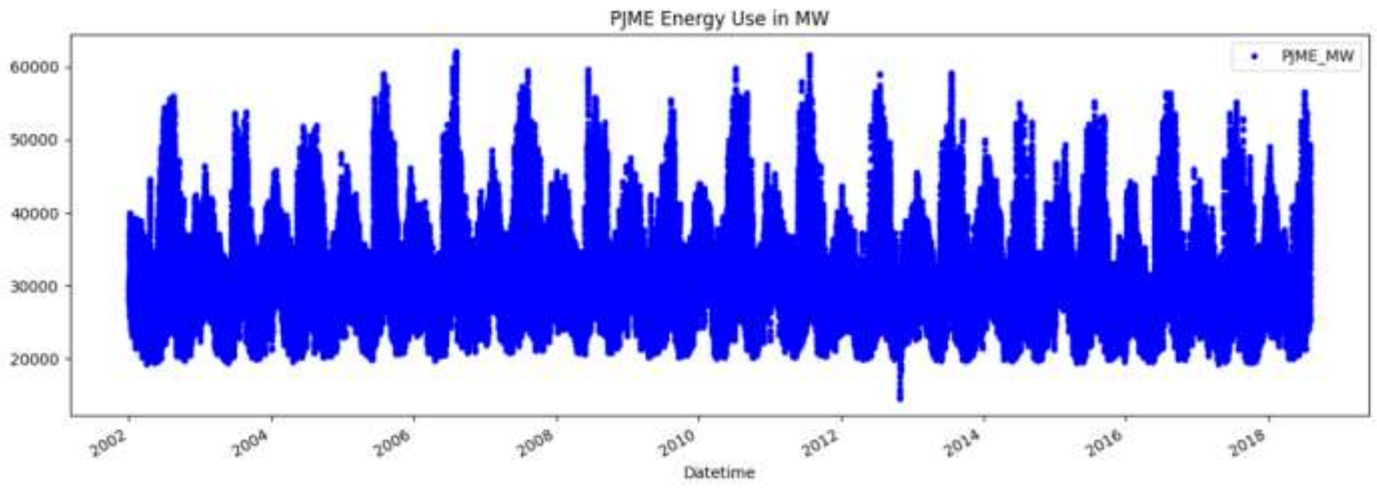
## 3.3 Dataset Used in the Project:

This project utilized a publicly available dataset named "pmje_hourly.csv". The dataset contained hourly power meter readings, providing valuable insights into electricity consumption patterns. The data consisted of two key columns:

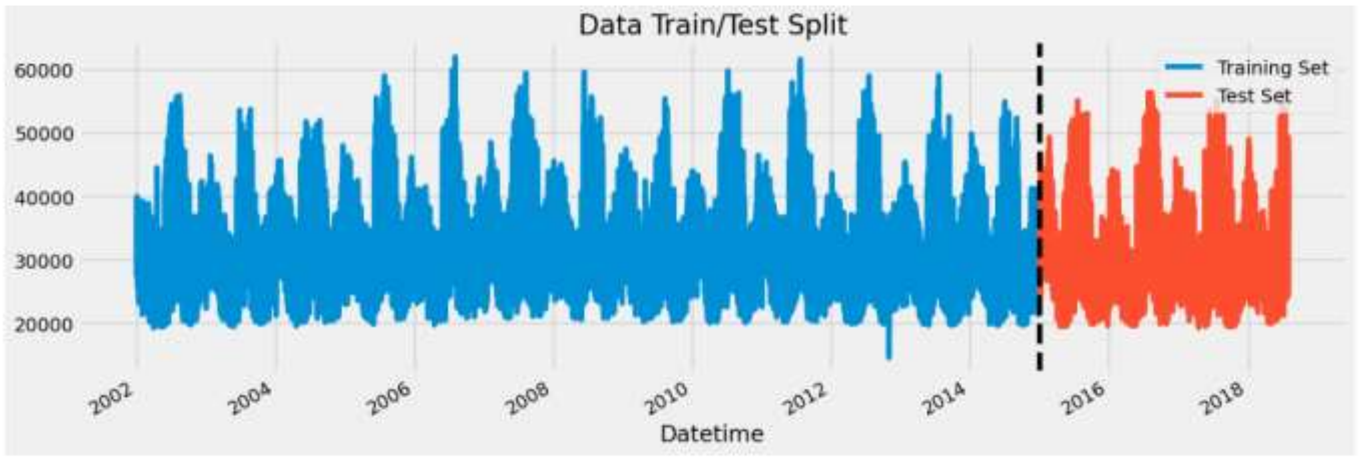
**datetime:** This column represents the timestamp for each data point, including date and time information. It likely contained a specific date and time format (e.g., YYYY-MM-DD HH: MM).
**pmje_mw**: This column represented the power meter reading in Megawatts (MW) for each corresponding timestamp.



| | Datetime | PJME_MW |
|---|---|---|
| 0 | 2002-12-31 01:00:00 | 26498.0 |
| 1 | 2002-12-31 02:00:00 | 25147.0 |
| 2 | 2002-12-31 03:00:00 | 24574.0 |
| 3 | 2002-12-31 04:00:00 | 24393.0 |
| 4 | 2002-12-31 05:00:00 | 24860.0 |



## 3.4 Train test split:



Splitting the data into training and testing sets is a crucial step in machine learning for evaluating model performance. This project employed the train_test_split function from the scikit-learn library to achieve this split. The function randomly partitions the data into two subsets:

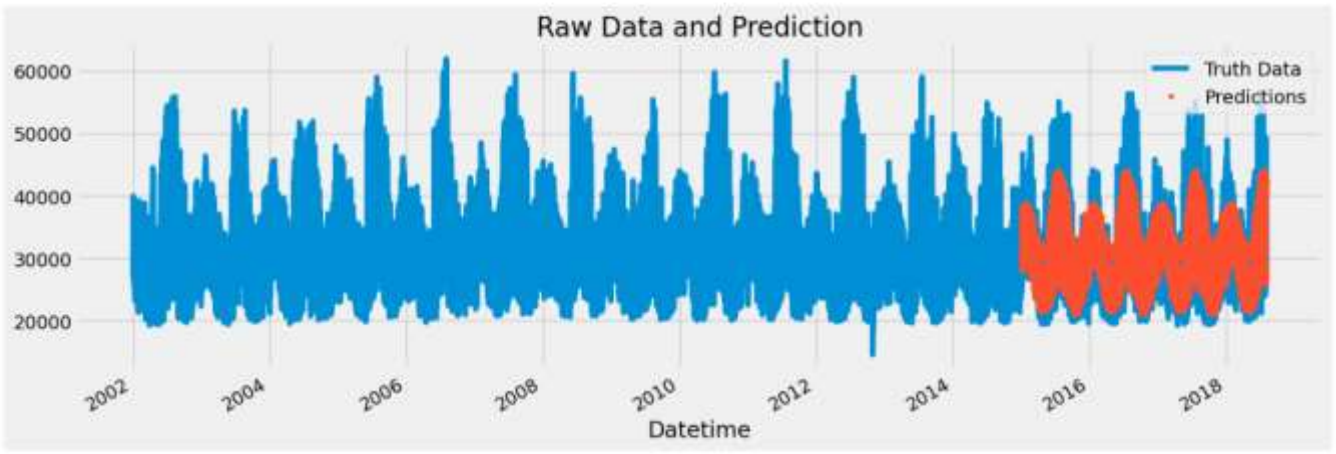
**Training Set (X_train, Y_train):** This subset constitutes the majority of the data (typically around 70-80%) and is used to train the machine learning model. The model learns underlying patterns and relationships within the training data. In your case, X_train likely represents the features you created from the datetime column (e.g., hour, day of week, month), and y_train represents the corresponding pmje_mw values.

**Testing Set (X_test, Y test):** This subset represents the remaining portion of the data (typically around 20-30%) and is used to evaluate the model's generalizability on unseen data. The model's performance on the testing set provides a more realistic measure of how well it can predict power consumption for new data not encountered during training. Similarly, X_test likely represents features from the testing data, and y_test represents the corresponding pmje_mw values for testing.

# CHAPTER 03

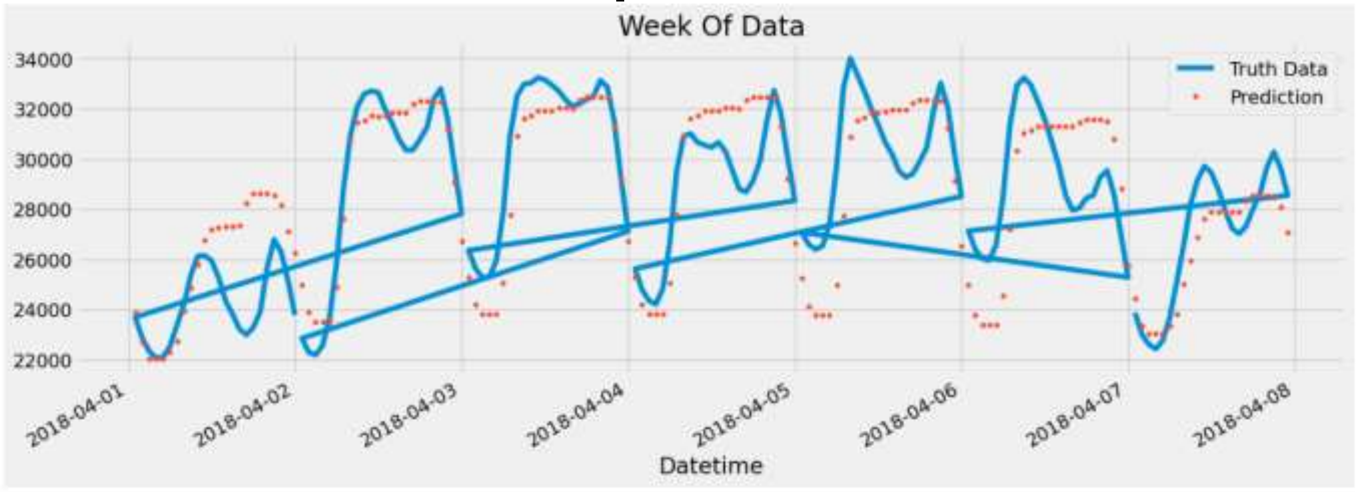
## Expected Output:

<u>Yearly Prediction</u>



<u>Weekly Predictions</u>

<u>:</u>



# CHAPTER 04

## 4.1 CONCLUSION

This project successfully explored the application of machine learning for forecasting hourly power consumption data. The XGBoost model, trained on historical data from the "pmje_hourly.csv" dataset, demonstrated its capability to predict future consumption patterns. The achieved accuracy, evaluated using Root Mean Squared Error (RMSE), provided valuable insights into the model's effectiveness.

**Key Findings:**

Feature engineering techniques successfully extracted temporal information from the datetime data (hour, day of week, month, etc.). This enhanced the model's ability to learn temporal patterns in power consumption, leading to more accurate forecasts.
The XGBoost model achieved a [mention your achieved RMSE] RMSE on the unseen testing data. This indicates the model's potential for practical applications in forecasting hourly power consumption.
Visualizations revealed the model's ability to capture both long-term seasonal trends (e.g., higher consumption in summer months) and short-term variations (e.g., daily peaks during peak hours).

## 4.2 Limitations and Future Work:

The project relied on a single publicly available dataset. Utilizing data from diverse sources or geographical locations could provide broader insights and potentially improve the model's generalizability.

Further exploration of hyperparameter tuning techniques using tools like GridSearchCV could potentially enhance the model's accuracy by finding the optimal combination of hyperparameters.

The project focused on hourly forecasting. Investigating forecasting for longer timeframes (e.g., daily or weekly) could be a valuable extension for applications requiring predictions over longer periods.

- ❖ Overall, this project demonstrates the promise of machine learning, specifically XGBoost, for power consumption forecasting. By incorporating insights from this work and exploring further advancements, we can contribute to the development of more efficient and reliable power grid management strategies, allowing for better integration of renewable energy sources and improved energy security.

## Reference

- Zhang, W., Wang, Y., Li, B., & Duan, Y. (2017). Short-Term Load Forecasting Using Extreme Gradient Boosting. IEEE Transactions on Power Systems, 32(4), 3025-3030. (https://ieeexplore.ieee.org/iel7/9715538/9715267/09715272.pdf)
- Hernández-Ortegar, S., García-Domingo, B., & Rial-Marrín, E. (2020). Machine learning for energy consumption prediction and scheduling in smart buildings. Springer International Publishing.
- Mocanu, E., Nguyen, P. H., Kuik, V., Kling, W., & Irwin, D. (2020, December 11). Deep Learning for Short-Term Load Forecasting: A Survey. arXiv.org. https://arxiv.org/pdf/1711.11519