Net Load Prediction Model

Mingyue Guo, Mingjun Wei, Mingzhe Liu, Zheng O'Neill
J. Mike Walker 66' Department of Mechanical Engineering, Texas A&M University
Contact: Mingyue Guo; Email: mingyue.guo@tamu.edu; Lab page: https://hvac.engr.tamu.edu/

1. Workflow of Model

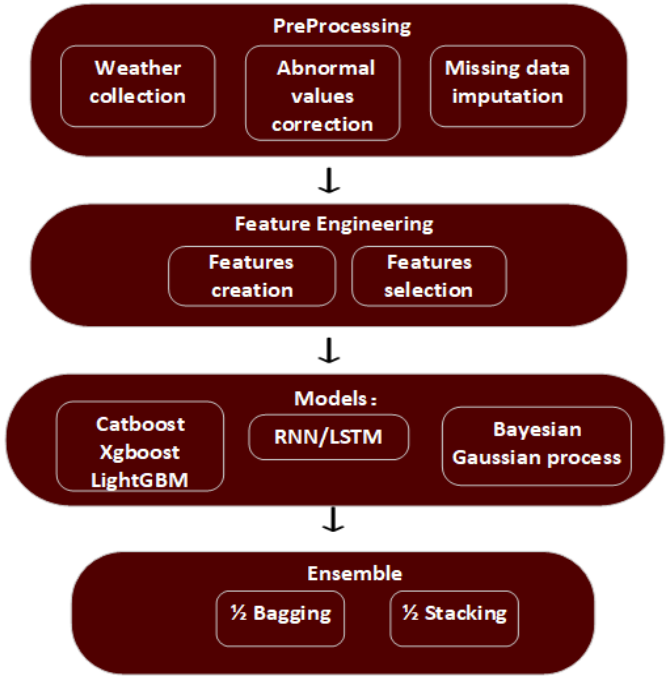


Figure 1: workflow of model

The team will develop a data-driven model to predict the net load. Figure 1 shows the workflow which includes 4 steps. First, we need to prepare the data for the data-driven model through preprocessing. Second, various types of features are created and selected as inputs to the data-driven model. Then, multiple algorithms are used to train the data-driven model. Finally, multiple data-driven model will be ensembled to get the finally prediction.

Step 1: Data Preprocessing

1.1 Exploratory Data Analysis

Univariate descriptive statistics

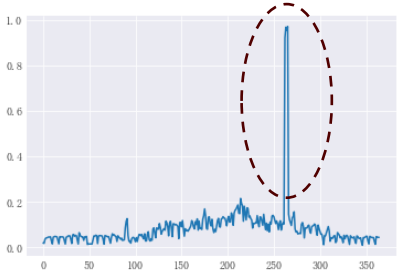
- Statistical feature of data (maximum, median, minimum)
- What is the variation/spread/range?
- What is the distribution of data, bell curve, bathtub curve, etc.?

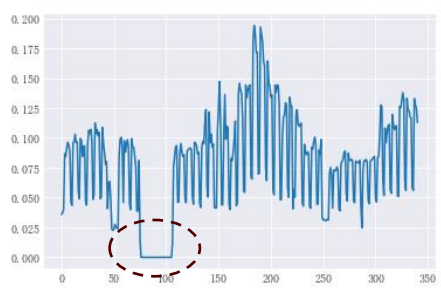
Bi-/multi- variable descriptive statistics

- Identifying relationships between variables

1.2: Data Cleaning and Wrangling

- Abnormal data recognition: outlier, dead value, etc.
- Outliers detection and cleaness
- Check dimensions (number of rows/cases, number of columns/variables)
- Check data types (categorical, ordinal, or numerical/discrete/continuous) of each variable.
- Check for missing values, encoding errors, etc.
- Fill missing value according to the missing gap





Step 2: Feature Engineering

Feature creation

- To get better model performance
- Encode (e.g one-hot encode) categorical data if needed
- Target encoder to capture statistic feature
- Create physical-based feature by domain knowledge

Feature selection

- Drop features that have low variability
- Drop features that have no relation to target
- Drop features that are highly related to other features
- Select features by **LOFO** (leave one feature out method), keep the feature if both training and cross-validation evaluation score are enhanced

Step 3: Train Models

Data split

- Split data into different train, valid and test dataset to cross validation

Training model and hyper-parameter tuning

- Using different popular machine learning algorithm to train individual base model
- Using GA or Bayesian to get better hyper-parameter

Step 4: Ensemble

- Different base model may capture different relationship between features and target
- Using bagging and stacking methods to combine base models to reduce overfitting