Improved Electricity Forecasting

Group:
Snowboarders - Wanxin Luo, Tina Cao

Agenda

Note: We are rewriting most parts of the deliverable due to mistakes spotted in the previous group's work

Introduction & Business Problem

Improvement Highlights

Data Extraction, Diagnostics, & Processing

Target Variables

Predictive Variables

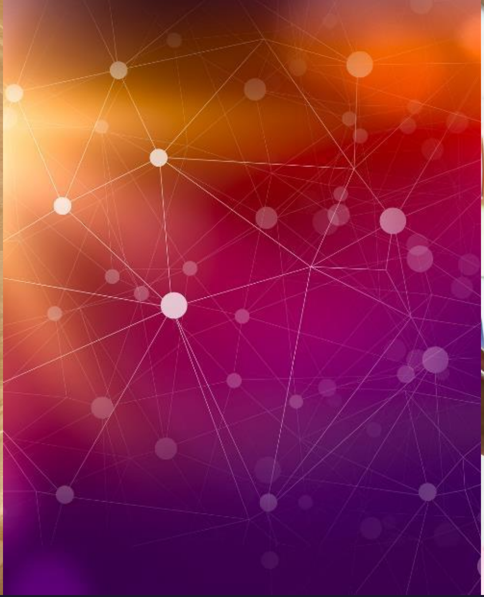
Pre-modeling

Modeling

Model Performance Comparison

Best Performing Model Analysis - Random Forest

Further Improvements



Introduction

The Electricity Consumption Forecasting Project is designed to utilize multiple time-series analytics methods to improve the operational efficiency and strategic planning of energy suppliers. By analyzing consumption patterns, trends, and seasonal variations in electricity usage data, this project aims to predict future electricity demands accurately. This involves a thorough analysis of historical consumption data to identify patterns and trends that significantly influence energy usage levels.

Objective

The project will employ a range of advanced predictive modeling techniques, encompassing both statistical and machine learning models. The objective is to develop robust forecasting models that can reliably predict future energy demands. These models are expected to serve as essential tools for energy managers, enabling them to optimize electricity generation and distribution, thus reducing costs associated with overproduction and minimizing the risk of energy shortages.

Business Problem

The outcomes of this project are anticipated to have a great impact on the operational strategies of energy suppliers. By providing accurate demand forecasts, this project will empower decision-makers with the insights needed to make informed operational and strategic choices. These choices will span various aspects of the business, from resource allocation and pricing strategies to customer service and grid management, ultimately enhancing the overall efficiency and responsiveness of the energy sector.

Improvement Highlights

We have made several improvements based on the work from the previous group. It is listed below:

- New Features
- More Models
- Better Approach
- Better Performance

Improvement Highlights - New Features

New features:

We have significantly enhanced our feature engineering to improve the granularity and accuracy of our predictive models. Our expanded feature set now includes:

- **Temporal Features**: We have introduced detailed time indicators such as year, month, day, and hour, along with the day of the week, which are essential for capturing seasonal and daily patterns in electricity usage. Additionally, a weekend binary indicator has been added to distinguish between weekdays and weekends, recognizing the different consumption patterns on these days.
- **Holiday Feature:** The “is_holiday” binary feature identifies public holidays, which typically have unique electricity usage patterns compared to regular days.
- **Moving Average:** We also added a moving average of the previous hour's electricity consumption. This feature helps in stabilizing the predictions by reducing short-term fluctuations and adding contextual relevance to the observed data.

These new features are designed to provide our models with a comprehensive view of the temporal dynamics affecting electricity consumption, enabling more accurate and reliable forecasting.

Improvement Highlights - More Models

More models:

To broaden our analytical scope and leverage the strengths of various advanced algorithms, we have integrated additional predictive models into our evaluation framework. This includes:

- **Facebook Prophet:** A tool designed for forecasting time series data, especially useful in scenarios with seasonal variations.
- **Elastic-Net Regularization:** A linear regression model that combines L1 and L2 regularization penalties, effective in reducing overfitting and feature selection with highly correlated predictors.
- **Random Forest:** An ensemble method that leverages multiple decision trees to improve prediction accuracy and robustness.
- **XGBoost:** A machine learning model using gradient boosting technique.

These models were selected for their strengths in handling complex and large-scale data with seasonal variations, thus providing a comprehensive approach.

Improvement Highlights - Better Approach

Better Approach:

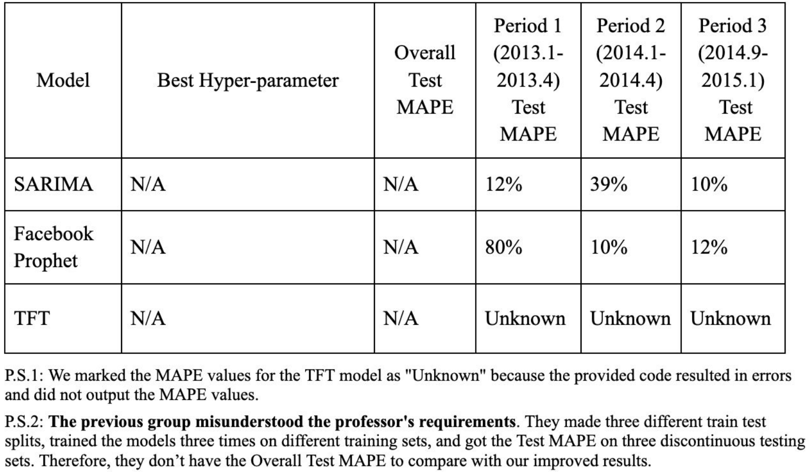
The dataset was divided into an 80% training set and a 20% testing set. For each modeling technique, we further split the training set into 60% for training and 20% for validation, as specific methods like Random Forest utilize time series cross-validation (TSCV) via GridSearchCV.

One mistake we would like to point out from the previous group is their misunderstanding of the requirement to divide the testing results into three or four time periods or regions of the same size, and for each region, to report MAPE and create a box plot of errors. The previous group interpreted this as using three different iterations, each iteration splitting the data differently to ensure the final test set has the same amount of data. However, we understand it as needing to split the test data into three equal parts and calculate their MAPE. Hence, we used our approach when fitting the models.

Improvement Highlights - Better Performance

Better Performance:

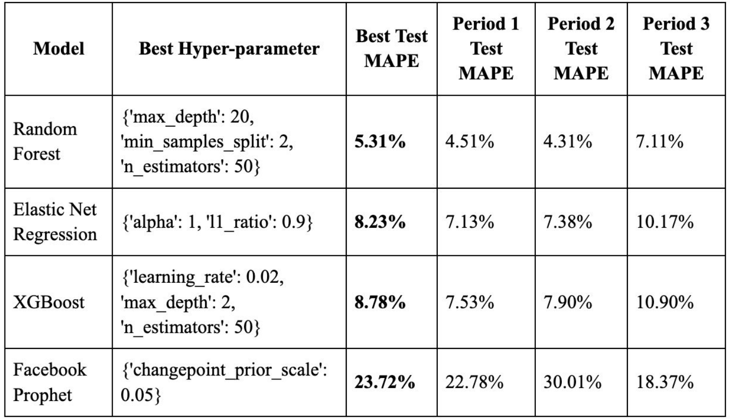
Compared to previous group, our best model has demonstrated significantly enhanced performance. The Random Forest model consistently delivered strong results across three periods, achieving Mean Absolute Percentage Errors (MAPE) well below 10%. This consistent and stable performance underscores the effectiveness of our model improvements and positions it as a highly reliable tool for forecasting.



Model	Best Hyper-parameter	Overall Test MAPE	Period 1 (2013.1-2013.4) Test MAPE	Period 2 (2014.1-2014.4) Test MAPE	Period 3 (2014.9-2015.1) Test MAPE
SARIMA	N/A	N/A	12%	39%	10%
Facebook Prophet	N/A	N/A	80%	10%	12%
TFT	N/A	N/A	Unknown	Unknown	Unknown

P.S.1: We marked the MAPE values for the TFT model as "Unknown" because the provided code resulted in errors and did not output the MAPE values.

P.S.2: **The previous group misunderstood the professor's requirements**. They made three different train test splits, trained the models three times on different training sets, and got the Test MAPE on three discontinuous testing sets. Therefore, they don't have the Overall Test MAPE to compare with our improved results.



Model	**Best Hyper-parameter**	**Best Test MAPE**	**Period 1 Test MAPE**	**Period 2 Test MAPE**	**Period 3 Test MAPE**
Random Forest	{'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 50}	**5.31%**	4.51%	4.31%	7.11%
Elastic Net Regression	{'alpha': 1, 'l1_ratio': 0.9}	**8.23%**	7.13%	7.38%	10.17%
XGBoost	{'learning_rate': 0.02, 'max_depth': 2, 'n_estimators': 50}	**8.78%**	7.53%	7.90%	10.90%
Facebook Prophet	{'changepoint_prior_scale': 0.05}	**23.72%**	22.78%	30.01%	18.37%

Data

The dataset for this project consists of historical electricity consumption data for 370 clients, recorded at 15-minute intervals over a period spanning from 2011 to 2014. Each observation in the dataset represents the electricity consumption for a specific client at a particular time interval. The dataset contains the following attributes:

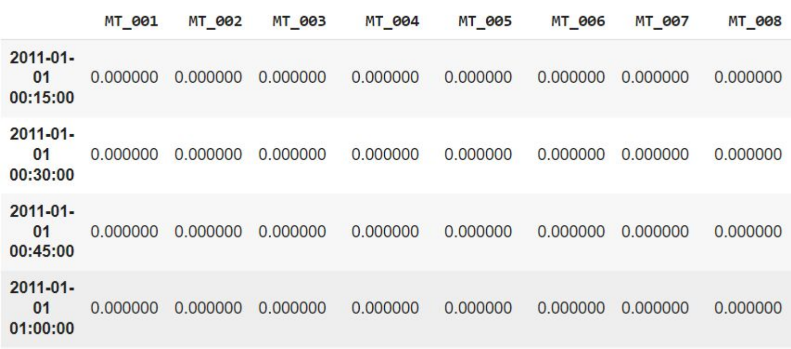
- Timestamp: The date and time at which the electricity consumption reading was recorded. This timestamp allows for the temporal analysis of consumption patterns over the entire duration of the dataset.

- Client ID: A unique identifier assigned to each client, enabling the tracking of consumption behavior for individual clients throughout the dataset.

- Electricity Consumption: The actual electricity consumption value recorded at each timestamp for each client. This is the primary variable of interest and serves as the target variable for forecasting.

Key Observations:

1. Granularity: The data is recorded at a high granularity, with consumption readings captured at 15-minute intervals. This high-resolution data allows for detailed analysis of consumption patterns and trends.



	MT_001	MT_002	MT_003	MT_004	MT_005	MT_006	MT_007	MT_008
2011-01-01 00:15:00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2011-01-01 00:30:00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2011-01-01 00:45:00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2011-01-01 01:00:00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Data Extraction Process

The dataset was extracted from the UCI Machine Learning Repository. It is organized in a text file where data entries are separated by semicolons (;). The first column contains the date and time as strings in the format 'yyyy-mm-dd hh:mm: ss', with subsequent columns representing each client's electricity consumption in kilowatts at 15-minute intervals.

To facilitate hourly consumption analysis, a preprocessing step was designed to aggregate the quarter-hourly measurements into hourly readings. This conversion requires that the kW values be divided by 4 to reflect the quarter-hourly recording intervals, thereby converting them into kilowatt-hours (kWh).

It was claimed on the website that the data has no missing values, but to ensure correctness we also checked for missing values. Additionally, the dataset accounts for clients who were added after the year 2011 by marking their consumption as zero prior to their introduction.

Data Diagnostics

A series of quality checks are performed on the data extract provided. These checks include:

- **Number of Records:**

  The dataset contained the number of records as described, with 140,256 features indicating individual time-interval readings across all 370 clients.

- **Duplicate Records:**

  Duplicate records are allowed to exist as it's possible that the same electronic records exist at different time intervals.

- **Missing Values in Relevant Fields:**

  The dataset is checked to have no missing values as noted.

- **Sum of a Numeric Field:**

  For each client, the sums of the electricity consumption were calculated to provide an aggregate view of the data and to verify the data's integrity across the time span.

- **Data Period Confirmation:**

  The dataset's temporal span was confirmed, ensuring that the records ranged from the beginning of 2011 to the end of 2014. Special emphasis was placed on the Portuguese hour adjustments for daylight saving time changes to affirm that the annual transitions were accurately reflected in the data.

EDA

2. Temporal Patterns: The dataset likely exhibits various temporal patterns, including daily, weekly, and seasonal fluctuations in electricity consumption. These patterns may be influenced by factors such as time of day, day of the week, and seasonal changes in weather.

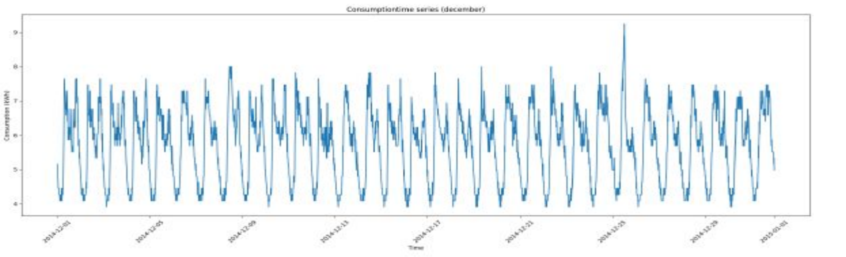


Figure 1: Consumption Trend on December 2014 for MT_002

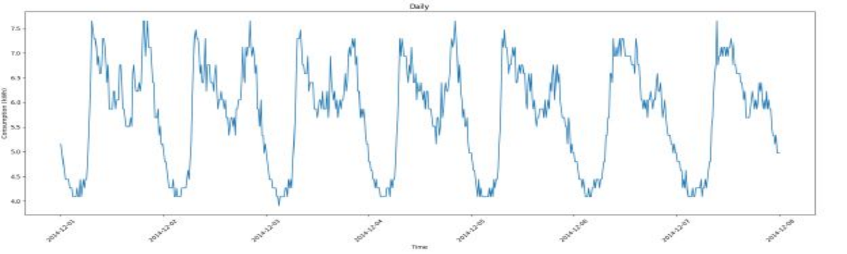


Figure 2: Consumption Trend on 1st Week of December 2014 for MT_002

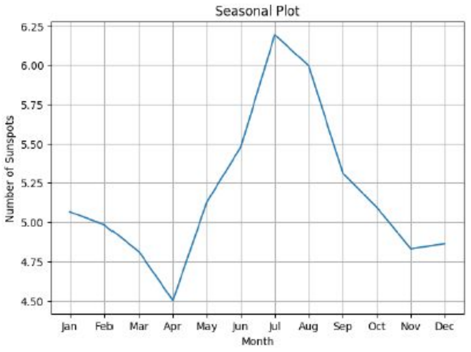


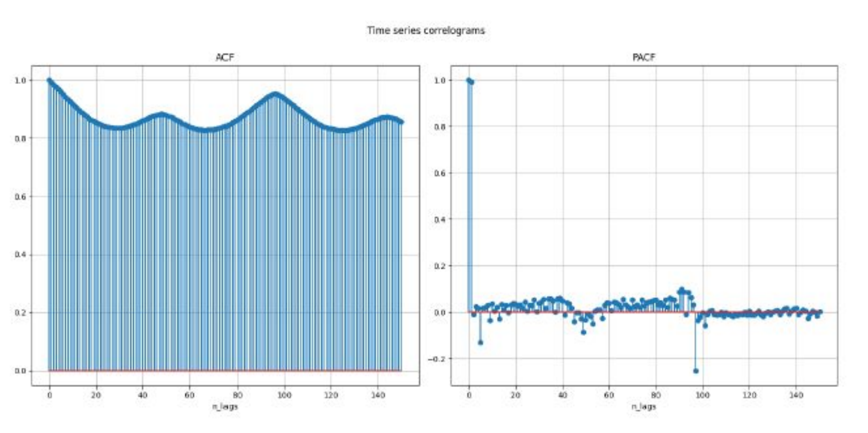
Figure 3:Seasonal Decomposition for MT_002 (1)

3. Seasonality: Given the multi-year duration of the dataset, seasonal patterns in electricity consumption, such as increased usage during summer months for cooling or winter months for heating, are expected to be evident.

EDA

4. Autocorrelation: Autocorrelation, also known as serial correlation, measures the degree of correlation between a time series and a lagged version of itself. In simpler terms, it assesses the relationship between observations at different time points within the same series. A positive autocorrelation indicates that past values influence future values, while a negative autocorrelation suggests an inverse relationship.

Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are commonly used tools to visualize and quantify autocorrelation in time series data. ACF shows the correlation between the series and its lagged values at various lags, while PACF represents the correlation between the series and its lagged values after removing the effects of shorter lags.



Understanding autocorrelation helps in identifying the presence of temporal patterns, seasonality, and trends in the data. It is also essential for selecting appropriate models, such as autoregressive (AR) and moving average (MA) models, for time series forecasting.

EDA

5. Moving Average: Moving Average is a statistical technique used to smooth out fluctuations in a time series by averaging consecutive observations over a specified window or interval. It helps in capturing the underlying trend in the data by reducing noise and random fluctuations.

There are different types of moving averages, including Simple Moving Average (SMA), Weighted Moving Average (WMA), and Exponential Moving Average (EMA). Each type differs in how it assigns weights to observations within the moving window.

Moving Average models are widely used for forecasting time series data, especially when there is evidence of trend or seasonality. They are simple yet effective tools for identifying and modeling patterns in the data, making them valuable components of more complex forecasting methodologies like ARIMA (Autoregressive Integrated Moving Average) and its variants.

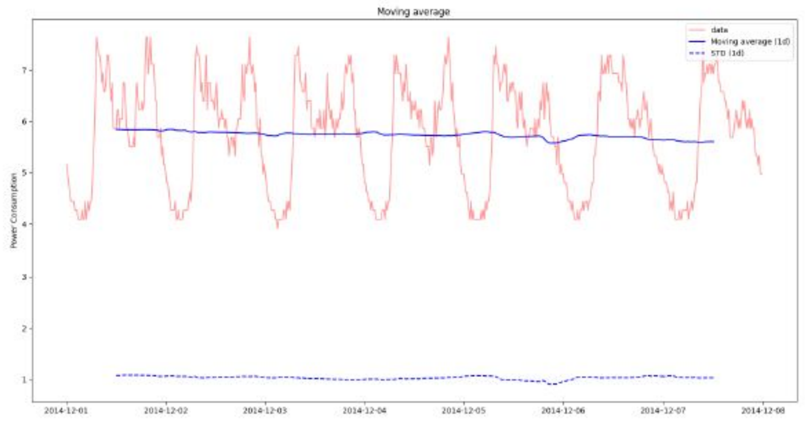


Figure 5: Moving Average(96) for MT_002

Data Processing

For all clients, the modeling data is processed using the following steps.

- **kWh Conversion:**

  kWh is more general to use in measuring power consumption. Thus, we divided all data points by 4 to convert the kW to kWh.

- **Outliers removal:**

  We used three methods including IQR, Z-Score, and Hampel package to detect the outliers and check whether they give the more reasonable results. To compare the performance, we calculated the numbers and the percentages of the outliers for each client. We also plotted the corrected data with sampled clients to obtain a more clear comparison among the three methods (shown as follows). Overall, Z-Score achieved the most reasonable performance, and we kept this to remove the outliers.

- **Hour adjustments for time-change date:**

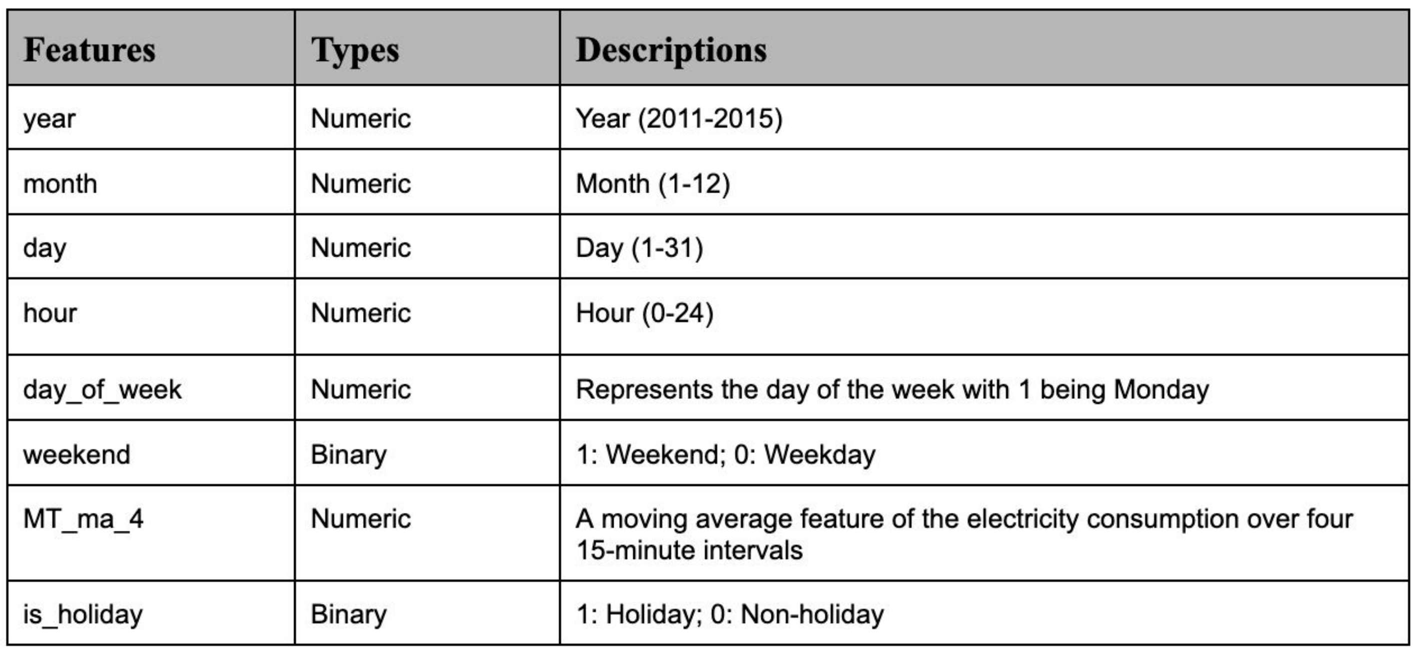
  We are focusing on modeling the time series first, and if the model can pick up the pattern of providing 0 values in March/October for a specific date change day, then we'll not adjust anything. Otherwise, if the model is not able to pick up the pattern, then we will include a manual correction of the generated output to take into account the necessary time correction for March/October.

Target Variable

The dataset contains electricity consumption data from 370 clients recorded between 2011 and 2014. Please note that the previous group selected only the second client as the target variable when fitting models. Hence, the target variable is MT_002. It seems they were only interested in predicting the electricity consumption behavior of a single individual. However, we believe that a better approach would be to average the electricity consumption data of the 370 clients, utilizing the full potential of the data and providing a clearer understanding of the collective electricity consumption behavior. In this project, we will stick to their initial objective and only predict the electricity consumption behavior of the second client.

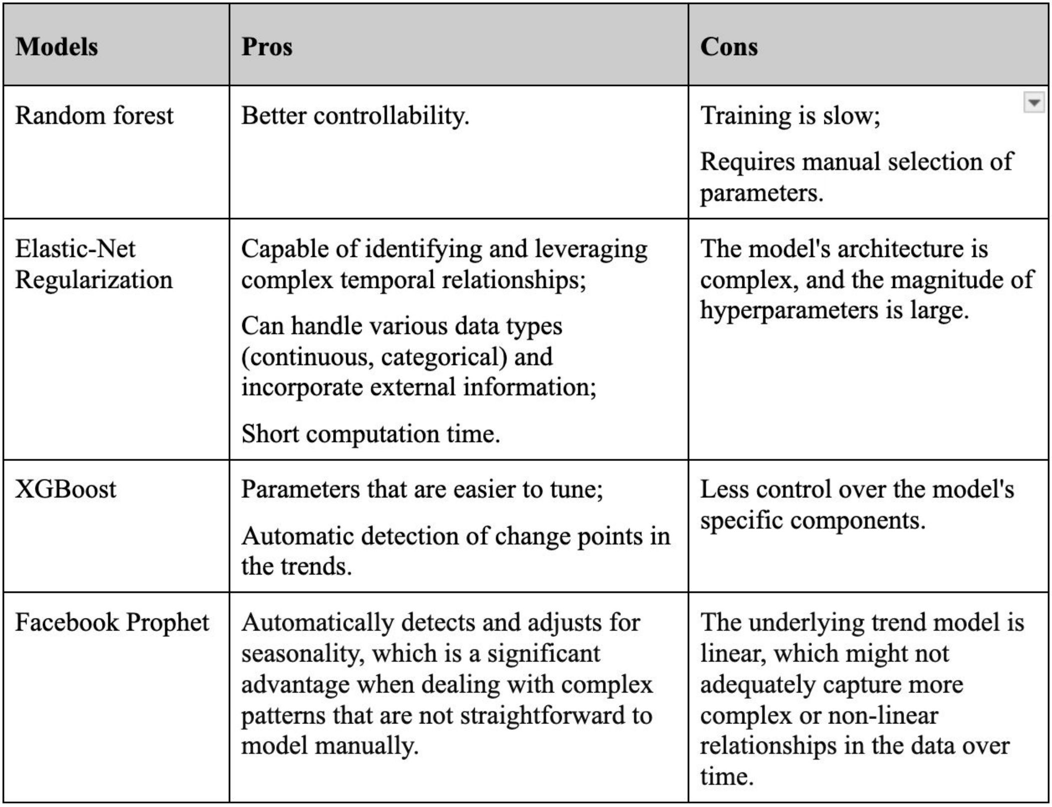
Feature Engineering

While the previous group only used past electricity consumption data from the second client to predict future usage, we enhanced the model's accuracy by incorporating additional predictive variables as shown below:



Features	Types	Descriptions
year	Numeric	Year (2011-2015)
month	Numeric	Month (1-12)
day	Numeric	Day (1-31)
hour	Numeric	Hour (0-24)
day_of_week	Numeric	Represents the day of the week with 1 being Monday
weekend	Binary	1: Weekend; 0: Weekday
MT_ma_4	Numeric	A moving average feature of the electricity consumption over four 15-minute intervals
is_holiday	Binary	1: Holiday; 0: Non-holiday

A Comparison of Models



Models	Pros	Cons
Random forest	Better controllability.	Training is slow; Requires manual selection of parameters.
Elastic-Net Regularization	Capable of identifying and leveraging complex temporal relationships; Can handle various data types (continuous, categorical) and incorporate external information; Short computation time.	The model's architecture is complex, and the magnitude of hyperparameters is large.
XGBoost	Parameters that are easier to tune; Automatic detection of change points in the trends.	Less control over the model's specific components.
Facebook Prophet	Automatically detects and adjusts for seasonality, which is a significant advantage when dealing with complex patterns that are not straightforward to model manually.	The underlying trend model is linear, which might not adequately capture more complex or non-linear relationships in the data over time.

Random Forest - Implementation

To optimize the performance of the Random Forest model, several key hyperparameters were carefully adjusted through a systematic grid search approach:

- Number of Estimators (n_estimators): This parameter defines the number of trees in the forest. We experimented with values of 10, 50, 100, and 300 to find the optimal balance between model accuracy and computational efficiency.

- Maximum Depth of the Trees (max_depth): We tested various depths for the trees to control overfitting. The depths considered were None (allowing trees to grow until all leaves are pure), 10, 20, and 30 levels deep.

- Minimum Samples Split (min_samples_split): This parameter determines the minimum number of samples required to split an internal node. Values tested included 2, 5, and 10, aiming to prevent the model from becoming too detailed and overfitting the data.

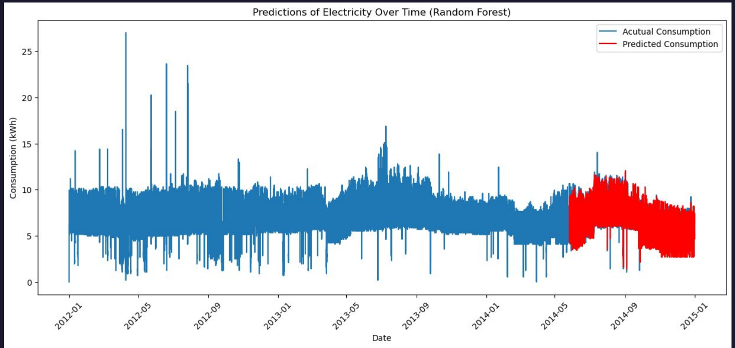
Training Methodology

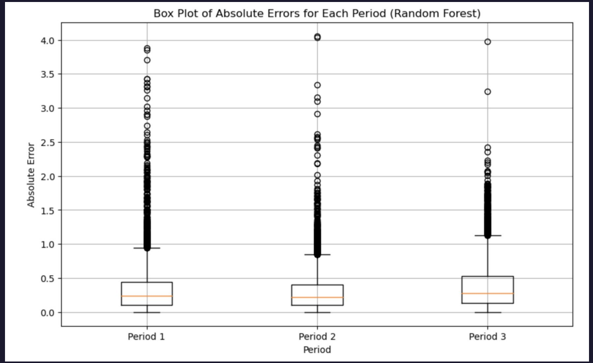
The Random Forest model was trained using a TimeSeriesSplit cross-validation approach, which is suitable for time-series data. This method involves sequentially splitting the data into training and validation sets, ensuring that the validation data always comes after the training data to mimic real-world forecasting scenarios. We employed five splits to provide a robust estimate of the model's performance.

Random Forest - Results

Test Period	Test MAPE
Period 1	4.51%
Period 2	4.31%
Period 3	7.11%

The images show the prediction of electricity consumption over time and the box plot of errors spread in the three periods. From left to right is Periods 1 to 3.





Elastic-Net Regularization - Implementation

The performance of the Elastic-Net model was optimized through a meticulous grid search process, focusing on two primary hyperparameters:

- Alpha (alpha): This parameter represents the overall strength of the penalty, a blend of L1 and L2 regularization. We experimented with values including 0.1 and 1 to find the optimal balance that minimizes overfitting while maintaining predictive power.
- L1 Ratio (l1_ratio): This parameter controls the mix between L1 and L2 regularization. Testing values of 0.1, 0.5, and 0.9 allowed us to determine the most effective ratio for promoting sparsity in the model coefficients (favoring feature selection) while controlling for multicollinearity.

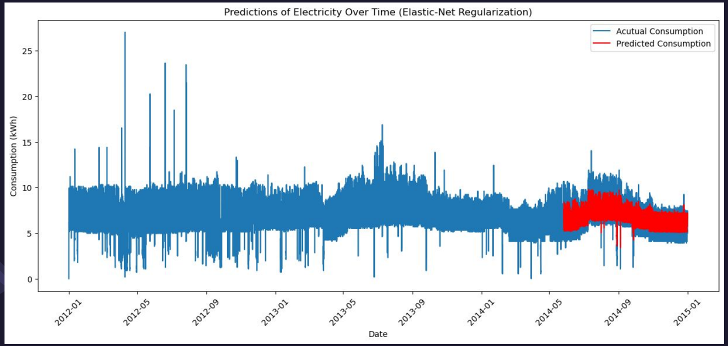
Training Methodology

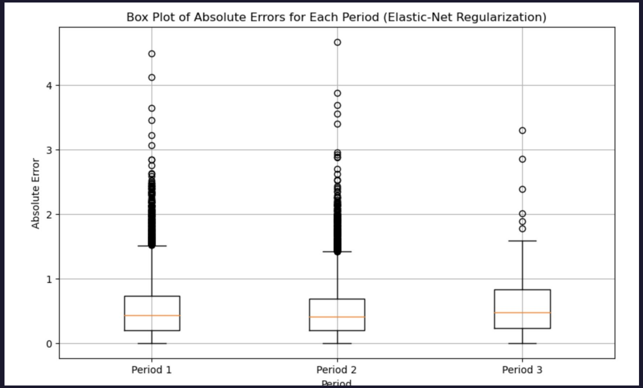
The Random Forest model was trained using a TimeSeriesSplit cross-validation approach, which is suitable for time-series data. This method involves sequentially splitting the data into training and validation sets, ensuring that the validation data always comes after the training data to mimic real-world forecasting scenarios. We employed five splits to provide a robust estimate of the model's performance.

Elastic-Net Regularization - Results

Test Period	Test MAPE
Period 1	7.13%
Period 2	7.38%
Period 3	10.17%

The images show the prediction of electricity consumption over time and the box plot of errors spread in the three periods. From left to right is Periods 1 to 3.





XGBoost - Implementation

To enhance the performance of the XGBoost model, we undertook a rigorous hyperparameter optimization process using grid search:

- Number of Estimators (n_estimators): This parameter defines the number of gradient boosted trees to be used in the model. We experimented with 50, 100, 300, 500, and 1000 estimators to identify the most effective number that provides the best trade-off between prediction accuracy and model training time.
- Maximum Depth of the Trees (max_depth): The depth of each tree, which is a critical factor in controlling model complexity and overfitting, was varied among 2, 4, 6, and 8 levels. Smaller values prevent the model from learning overly specific details of the training data.
- Learning Rate (learning_rate): This parameter shrinks the feature weights after each boosting step to prevent overfitting. Values tested included 0.02, 0.05, and 0.1, to manage the speed at which the model learns patterns in data.

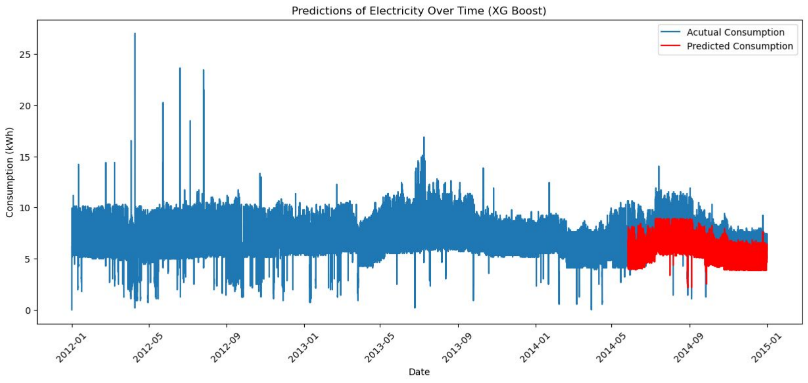
Training Methodology

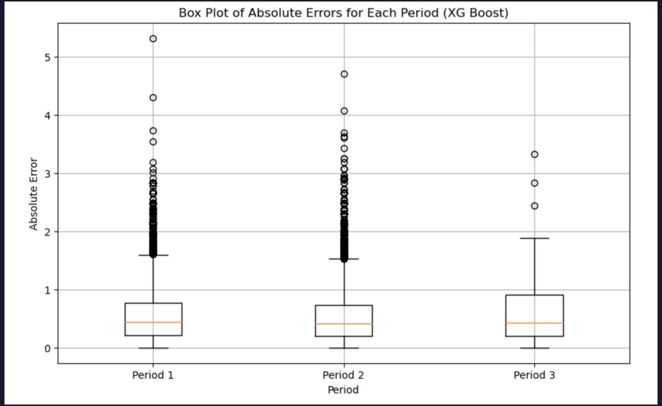
The XGBoost model was trained using a TimeSeriesSplit cross-validation approach, similar to that used for Random Forest. This approach is particularly effective for time-series data as it respects the temporal order of observations. We utilized five splits in our cross-validation to ensure a thorough evaluation across different subsets of the data.

XGBoost - Results

Test Period	Test MAPE
Period 1	7.53%
Period 2	7.90%
Period 3	10.90%

The images show the prediction of electricity consumption over time and the box plot of errors spread in the three periods. From left to right is Periods 1 to 3.





Facebook Prophet- Implementation

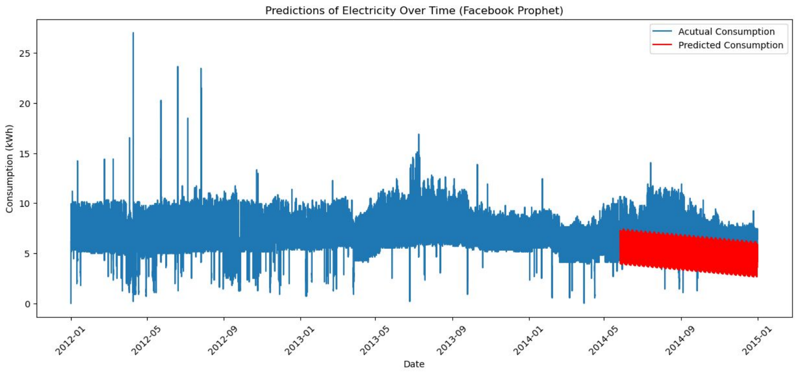
Model Configuration

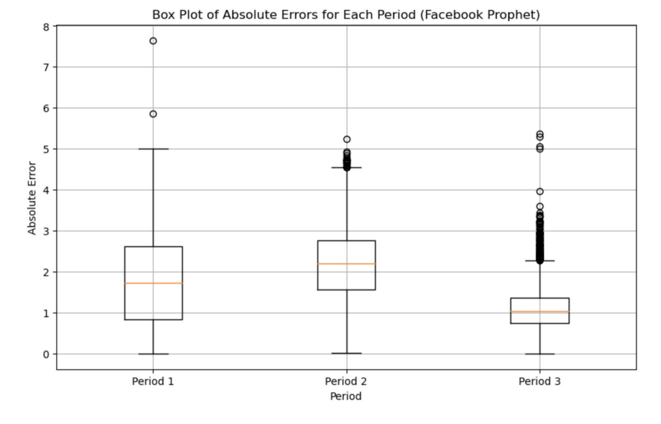
- **Growth:** Linear, assuming a steady trend that does not change direction.
- **Seasonality Mode:** Additive, which is suitable for most scenarios where seasonality does not increase with the level of the time series.
- **Yearly Seasonality:** Disabled, focusing the model on capturing daily and weekly patterns rather than annual changes.
- **Weekly Seasonality:** Enabled, allowing the model to adjust for fluctuations within the week.
- **Daily Seasonality:** Enabled, to account for variations within each day.
- **Changepoint Prior Scale:** Set to 0.05 to moderate the model's sensitivity to changes in the trend, balancing flexibility and overfitting.

Facebook Prophet - Results

Test Period	Test MAPE
Period 1	22.78%
Period 2	30.01%
Period 3	18.37%

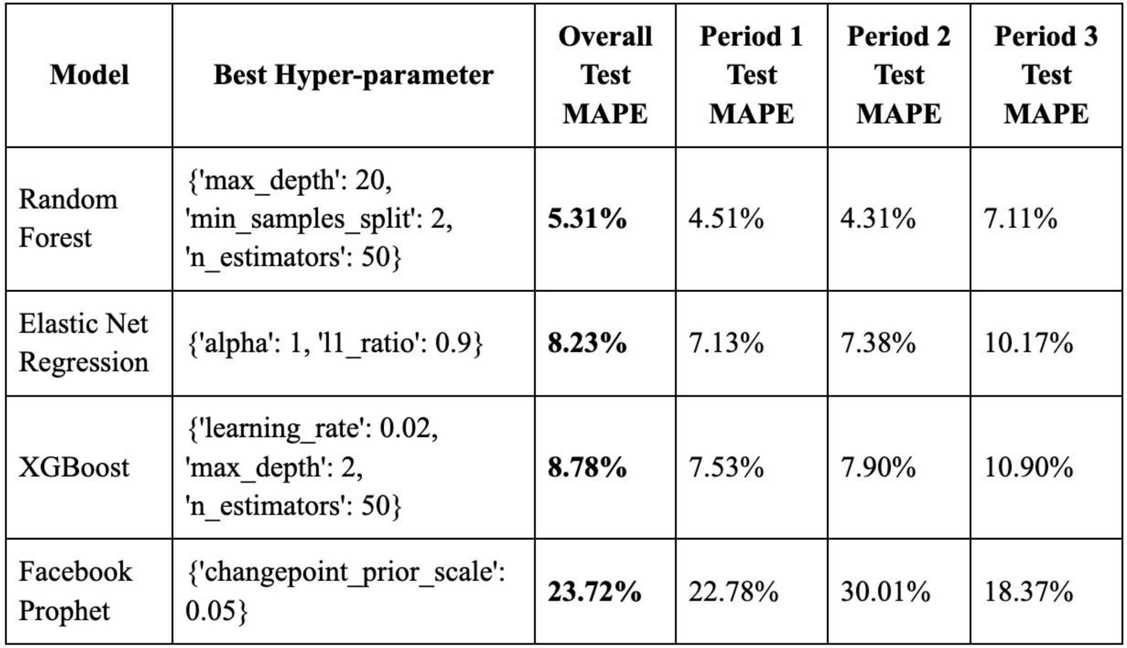
The images show the prediction of electricity consumption over time and the box plot of errors spread in the three periods. From left to right is Periods 1 to 3.





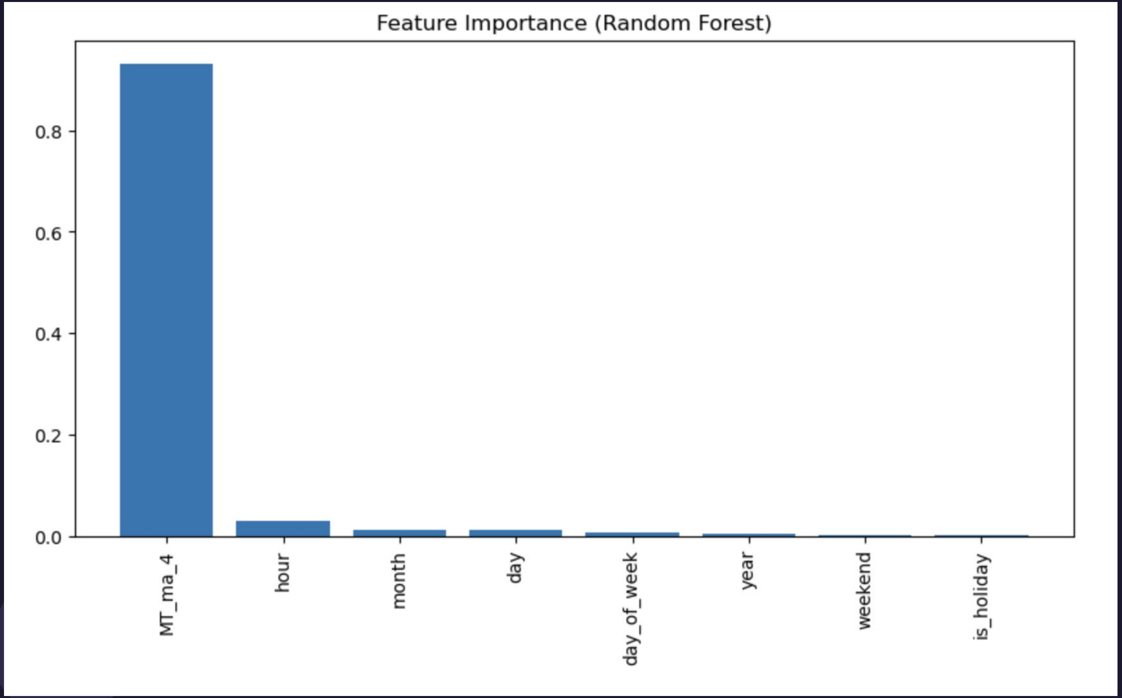
Comparative Analysis

According to the table below, we can see the Random Forest Model not only have the lowest overall test MAPE but also lowest MAPEs in all three different periods. Therefore, the Random Forest model stands out as the most effective model for forecasting electricity consumption due to its accuracy and consistency



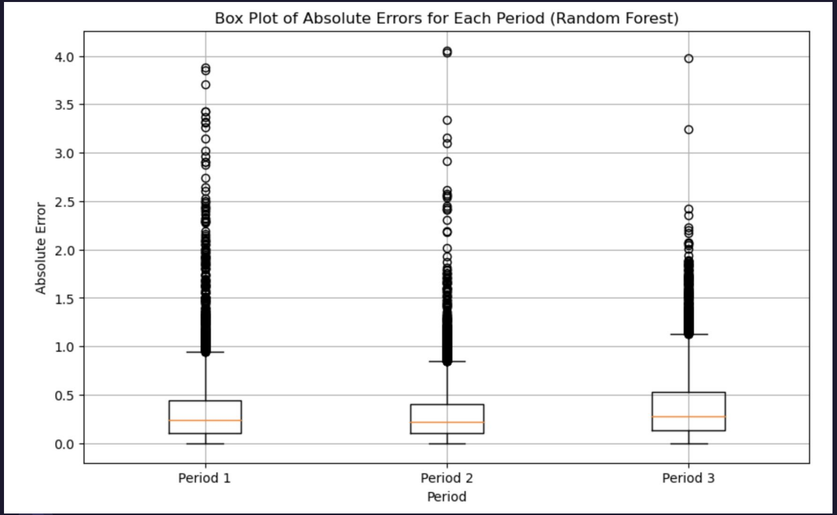
Model	Best Hyper-parameter	Overall Test MAPE	Period 1 Test MAPE	Period 2 Test MAPE	Period 3 Test MAPE
Random Forest	{'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 50}	**5.31%**	4.51%	4.31%	7.11%
Elastic Net Regression	{'alpha': 1, 'l1_ratio': 0.9}	**8.23%**	7.13%	7.38%	10.17%
XGBoost	{'learning_rate': 0.02, 'max_depth': 2, 'n_estimators': 50}	**8.78%**	7.53%	7.90%	10.90%
Facebook Prophet	{'changepoint_prior_scale': 0.05}	**23.72%**	22.78%	30.01%	18.37%

Best Performing Model (RF) Analysis



This feature importance plot from the Random Forest model highlights the relative significance of various features used to predict electricity consumption. The most influential feature, by a considerable margin, is MT_ma_4, which represents the moving average of electricity consumption over the past hour. This indicates that short-term historical consumption is a critical predictor, reflecting immediate past conditions as a strong indicator of near-future usage.

Best Performing Model (RF) Analysis



This box plot represents the distribution of absolute errors in electricity consumption predictions made by the Random Forest model across three different periods.

Period 1:
The median error in Period 1 is observed to be lower than 0.5, indicating that at least 50% of the prediction errors are below this value.
Period 2:
Maintains a level similar to Period 1, reinforcing the model's consistent predictive performance.
Period 3:
The median error is slightly higher but still quite similar to the previous periods.

Overall, the box plot suggests that the Random Forest model's prediction accuracy remains stable across the three different periods. In the meantime, across all periods, a significant portion of the data points exhibits prediction errors below 0.5, signifying a high level of precision in the model's ability to forecast electricity consumption.

Further Improvements

Incorporation of Additional Data Types:

Weather Data Integration: Beyond basic temperature and precipitation, including solar radiation levels, wind speed, and humidity might enhance the model's ability to predict usage patterns, especially in areas reliant on heating and cooling.

Including Additional Time-Based Features:

Expanding Moving Averages: Given the substantial influence of the moving average of the previous hour (MA_4) on model performance, we propose adding longer-term moving averages to capture broader consumption trends. Specifically, incorporating a moving average for the previous 24 hours (MA_24) could help the model account for daily consumption cycles, enhancing its ability to forecast based on daily peaks and troughs in electricity usage.

Adoption of Advanced Modeling Techniques:

- Deep Learning Models: Implementing deep learning approaches such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformer models to capture the complex non-linear relationships typical in electricity data.
- Hybrid Models: Combining machine learning models with traditional statistical methods (such as SARIMA) might yield better results by leveraging the strengths of both approaches.