Forecasting Project
Process Documentation
V3
Data Prophets

New Team Information ~ Data Prophets

Prakhar Dungarwal (pd2782)   Somit Jain (sj3396)

Shashwat Kumar (sk5520)   Devdatt Golwala (drg2172)

Contents

Overall Improvements and Our Observations

Team: Data Prophets

Over the course of this project, the Data Prophets team undertook a comprehensive refinement of the initial forecasting pipeline developed by the Fortune Tellers team. Our focus was on improving model robustness, data handling, evaluation integrity, and expanding the modeling toolkit with cutting-edge time series approaches. Below, we detail the key observations from the prior implementation and the improvements introduced in our enhanced pipeline.

Key Observations from Prior Implementation (Fortune Tellers)

1. **Clustering Imbalance**
   The clustering strategy employed previously resulted in a highly imbalanced cluster distribution, with Cluster 0 dominating the dataset. This likely skewed model learning and biased forecasts. The imbalance hindered the ability to generalize across operational conditions represented by other clusters.

2. **Suboptimal Train-Test Split**
   The Fortune Tellers' approach utilized a random windowing method for evaluation, which failed to respect the temporal nature of the data. Additionally, there was overlap between training and testing periods due to a 7-day rolling window, effectively introducing data leakage and inflating model performance metrics.

3. **Untuned DeepAR Implementation**
   The DeepAR model was deployed without systematic hyperparameter tuning. As a result, the model underperformed despite its potential, likely due to suboptimal configuration of parameters like context length, learning rate, and RNN structure.

4. **Lack of Ground Truth Comparison**
   There were no visual or quantitative comparisons of predicted results against actual test data. This significantly reduced the interpretability and diagnostic value of the results.

5. **Limited Modeling Approaches**
   The modeling suite was primarily limited to DeepAR and LSTM, missing out on newer or alternative models like Prophet or Transformer-based frameworks that could better capture temporal patterns and irregularities.

Strategic Enhancements by Data Prophets

1. **Corrected and Chronological Train-Test Splitting**
   We restructured the data partitioning logic to enforce strict chronological separation between training and testing sets. We also removed the overlapping window issue to prevent leakage, ensuring fair evaluation. This has led to a more realistic representation of model performance in production-like settings.

2. **Hyperparameter Optimization of DeepAR**
   The DeepAR model was extensively tuned using a grid-based search over key parameters. This optimization significantly improved forecast accuracy and reduced volatility in predictions, especially in high-delay periods.

3. **Expansion of Modeling Suite with Industry and Research-Grade Tools**

- **Prophet:** We integrated Prophet to leverage its intuitive decomposition of trend and seasonality. Prophet, widely used in industry, demonstrated robust performance with minimal tuning, particularly in handling missing data and holiday effects.

- **Amazon Chronos (Transformer-Based):** To explore state-of-the-art forecasting, we introduced Amazon Chronos, which leverages transformer architectures for long-term temporal modeling. Chronos provided superior performance in learning long-range dependencies and showed strong improvements in MAPE.

4. **MAPE Improvement Across the Board**
   Following model enhancements and data corrections, we observed a marked reduction in MAPE values. Both Prophet and Chronos outperformed the original DeepAR and LSTM models, validating the improvements. Prophet offered strong baseline accuracy, while Chronos excelled in periods of high volatility and fluctuation.

5. **LSTM Enhancement via BiLSTM**
   The traditional LSTM model was replaced with a **Bidirectional LSTM (BiLSTM)**, enabling the network to learn from both past and future sequences during training. This architectural upgrade led to improved forecasting accuracy, especially in capturing turning points and abrupt shifts in delay patterns.

6. **Graphical Evaluation and Diagnostics**
   To improve interpretability, we introduced comparative plots showing actual vs. predicted values across all models. These visual diagnostics provide insight into model behavior, strengths, and limitations, making the system more transparent and usable for stakeholders.

Summary Table of Improvements

Aspect	**Fortune Tellers (Old Team)**	**Data Prophets (New Team)**
Clustering	Imbalanced; dominated by Cluster 0	Identified imbalance; informing future improvements (e.g., re-clustering)
Train-Test Split	Random window; overlapping data leakage	Chronological split with non-overlapping windows for valid testing
DeepAR	No tuning; suboptimal performance	Hyperparameter tuning applied; improved accuracy
Evaluation	No comparison with ground truth	Actual vs. predicted plots for all models
Modeling Approaches	DeepAR, LSTM only	Added Prophet and Amazon Chronos (Transformer-based models)
LSTM Architecture	Simple LSTM	BiLSTM implementation; better MAPE performance
Forecast Accuracy (MAPE)	Moderate, unverified	Significant improvement across Prophet, Chronos, and BiLSTM

Conclusion

The Data Prophets team has substantially enhanced the forecasting pipeline through rigorous evaluation, careful engineering, and strategic model diversification. By addressing the limitations of the earlier system and integrating state-of-the-art methodologies, we have achieved a more robust, accurate, and interpretable solution for subway delay prediction.

Introduction

In this project, we focus on developing a robust machine learning model for electricity consumption forecasting. The dataset used in this study is the Electricity dataset, which contains weekly electricity consumption data for 370 clients from late 2011 to early 2014, with records extending until early 2015. We took several initial approaches, including LSTM, DeepAR to explore different methodologies for time-series forecasting.

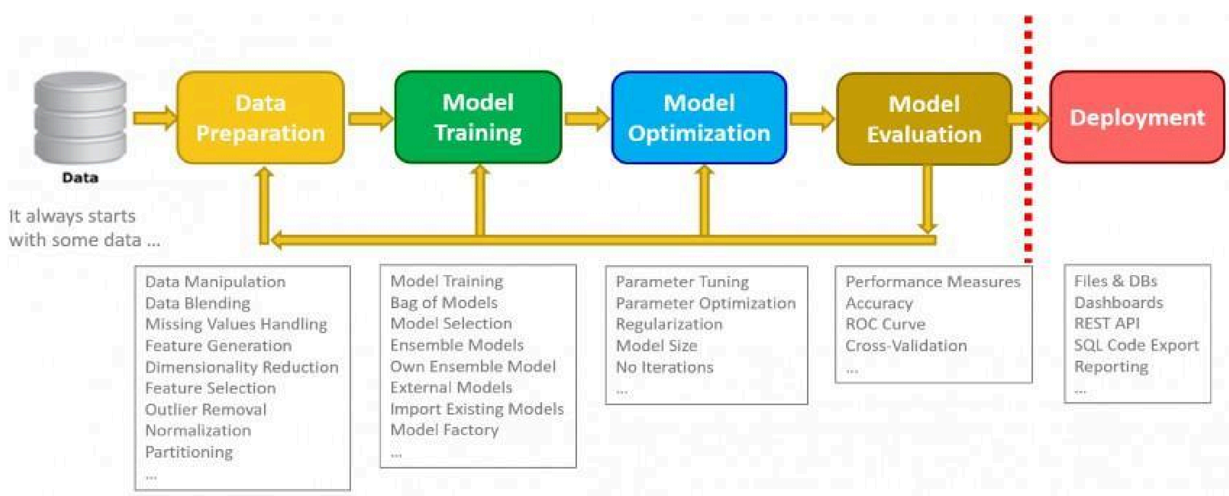
A key objective of this project is to standardize a robust forecasting methodology, ensuring that the analytical approach and data requirements can be consistently applied in future forecasting efforts. This methodology will serve as a foundation for scalable and reproducible forecasting models, enabling long-term improvements in electricity demand prediction and energy resource management.

Our forecasting model is designed to benefit key stakeholders, including energy providers, policymakers, and businesses that require accurate electricity demand predictions to optimize resource allocation, infrastructure planning, and operational efficiency. Through this project, we aim to enhance forecasting capabilities, contributing to more efficient energy management and sustainable decision-making.

Forecasting electricity prices plays a pivotal role in the modern power systems landscape. Accurate and efficient electricity forecasting methods have become increasingly crucial as the electricity market continues to evolve towards a more competitive and deregulated structure.

The purpose of this document is to provide a detailed technical overview of:

1. Design specifications and parameters
2. Data inclusions and exclusions
3. Predictive variable creation process
4. Target variable definition
5. Iterative model building process
6. Metric evaluation process



Data Processing

Data Overview

The Electricity Load Diagrams Dataset from the UCI Machine Learning Repository contains hourly electricity consumption data from 370 clients in Portugal over a period of four years (2011–2014). The dataset is structured as time series data, recording 96 measurements per day. However, adjustments were necessary due to daylight saving time (DST) changes. In March, when clocks move forward, the dataset records only 23 hours, setting 1:00–2:00 AM values to zero, while in October, when clocks move back, it records 25 hours, aggregating consumption for the overlapping 1:00–2:00 AM period.

Data Extraction Process

The data extraction process involves retrieving electricity consumption records for each individual customer from the dataset. The dataset contains 370 customers, each represented as a separate column (MT_001 to MT_370), where each row corresponds to a 15-minute interval timestamp.

Since the dataset has no missing values, we directly load it into a Pandas DataFrame. Additionally, temporal attributes such as year, month, week, and day are extracted to facilitate time-based analysis and forecasting.

Data is formatted into the following format:

	Time	MT_001	MT_002	MT_003	MT_004	MT_005	MT_006	MT_007	MT_008	MT_009	...	MT_365	MT_366	MT_367	MT_368	MT_369	MT_370	year	month	week	day
0	2011-01-01 00:15:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	2011	1	52	1
1	2011-01-01 00:30:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	2011	1	52	1
2	2011-01-01 00:45:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	2011	1	52	1
3	2011-01-01 01:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	2011	1	52	1
4	2011-01-01 01:15:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	2011	1	52	1

Data Diagnostics

A series of quality checks were performed on the extracted dataset to ensure data integrity and consistency. The dataset consists of 140,256 rows and 375 columns, where 370 columns represent individual customer electricity consumption readings (MT_001 to MT_370), and the remaining columns correspond to timestamps and additional temporal attributes.

- Number of Records: Verified that the dataset contains 140,256 timestamps, which aligns with the expected 15-minute interval readings over four years (2011–2014).
- Duplicate Records: No duplicate entries were found.
- Missing Values: No missing values. However, according to the dataset description, some clients were created after 2011, and in these cases consumption was considered zero.

These zeros are handled when calculating the average electricity consumption per user as predictive variables.

- Data Period Confirmation: Ensured the dataset correctly spans 2011 to 2014.

Modeling Data Creation

While the dataset technically contains no missing values, we identified two critical issues that impact the consistency and fairness of the analysis:

Lack of 2011 Data for Many Customers: Although the dataset includes records starting from 2011, a significant number of customers had no consumption data during this year, either because they were not yet active or because their values remained at zero. Including 2011 would therefore introduce bias and inconsistency.

Incomplete Annual Records (2012–2014): For some customers, we observed that an entire year of data was missing between 2012 and 2014. Since our forecasting approach relies on temporal continuity (e.g., capturing seasonal trends and long-term patterns), these gaps would compromise model reliability.

Therefore, we removed all records from 2011 to ensure consistency across the remaining customer time series and to avoid skewed statistics caused by zero-inflated entries. Then we further excluded customers who were missing an entire year of data between 2012 and 2014 to maintain the integrity of year-over-year trend analysis. After this filtering process, the final dataset includes 331 customers with complete and consistent data from 2012 to 2014.

To segment the 331 customers based on their electricity usage patterns, we combined three different clustering approaches to capture various aspects of the data. The first method focused on statistical features extracted from each customer's time series, including mean, median, standard deviation, minimum, maximum, skewness, kurtosis, autocorrelation with a 30-day lag, summer average usage, winter average usage, ratio between summer and winter average usage, dominant frequency, Ricker wavelet energy, and load factor (mean usage over max usage). And we applied PCA to time series data, and got the top two principal components. Then used

K-Means clustering on these features gave us four initial clusters. The second approach used Gaussian Mixture Models (GMM), which assumes that the overall population of customers is generated from a mixture of several Gaussian distributions, each representing a potential behavioral cluster. By estimating the parameters of these underlying distributions, GMM identified four distinct customer groups with similar consumption patterns. For the third method, we used the K-Shape algorithm, which is specifically designed for time-series data and clusters based on the shape similarity of standardized consumption curves. It resulted in four clusters.

To combine the results from all three methods, we built a co-association matrix that recorded how often each pair of customers appeared in the same cluster across the 12 total clustering outcomes (3 methods × 4 clusters). Customers who were frequently grouped together across methods were more likely to belong to the same final cluster. Based on this matrix, we performed a final clustering step and identified four consolidated customer segments, which were then used as input for our forecasting models.

For the EDA part, we use each of the 370 customers, the modeling data was collected and processed to ensure it was structured effectively for predictive analysis. The Time column was first converted to a datetime format, and additional temporal attributes, including year, month, week, and day were extracted directly using Pandas.

Target Variables

A target variable is the variable to predict as a result in algorithmic solutions.

The target variable in this project is the daily / weekly electricity consumption.

Predictive Variables

A predictive variable is a variable used in algorithmic solutions to predict the target variable. During our analysis, we categorized predictive variables into two categories:

1. Direct variable – These variables were directly from the dataset that was provided by direct customers.

2. Derived variable – These variables were created by manipulating the direct variables.

Variable List

1. Electricity Consumption (Direct Variable): The dataset contains 15-minute electricity consumption readings for each user, which serve as the raw data source for further processing. (This variable is for EDA part)

2. Average Monthly Total Electricity Consumption (Derived variable): The average electricity consumption per existing user for each month, resulting in 12 variables per year. (This variable is for EDA part)

3. Average Daily Aggregated Consumption (Derived variable): The average electricity consumption per existing user for each day, resulting in 365 variables per year. (This variable is for EDA part)

4. Average Weekly Aggregated Consumption (Derived variable): The average electricity consumption per existing user for each week, resulting in 52 variables per year. (This variable is for EDA part)

5. Features of time: Year, Month, Week, Day. Features of time are critical in the modeling because they may capture seasonality in electricity usage. For example, the usage of air conditioning increases during summer times.

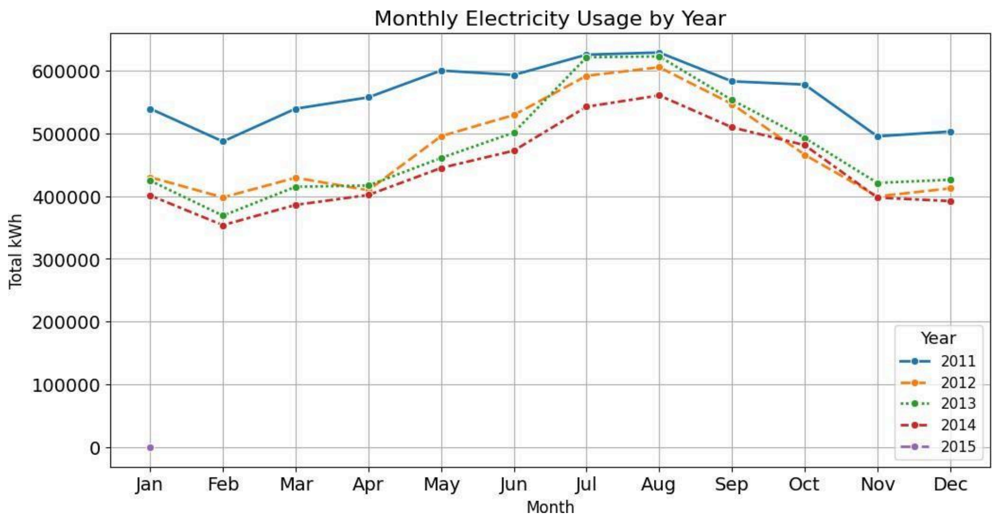
6. Customers’s daily usage (Derived variable): The original dataset contains electricity consumption readings recorded at 15-minute intervals for each user. To facilitate modeling and reduce data dimensionality, we aggregated these readings into daily total electricity usage per user.

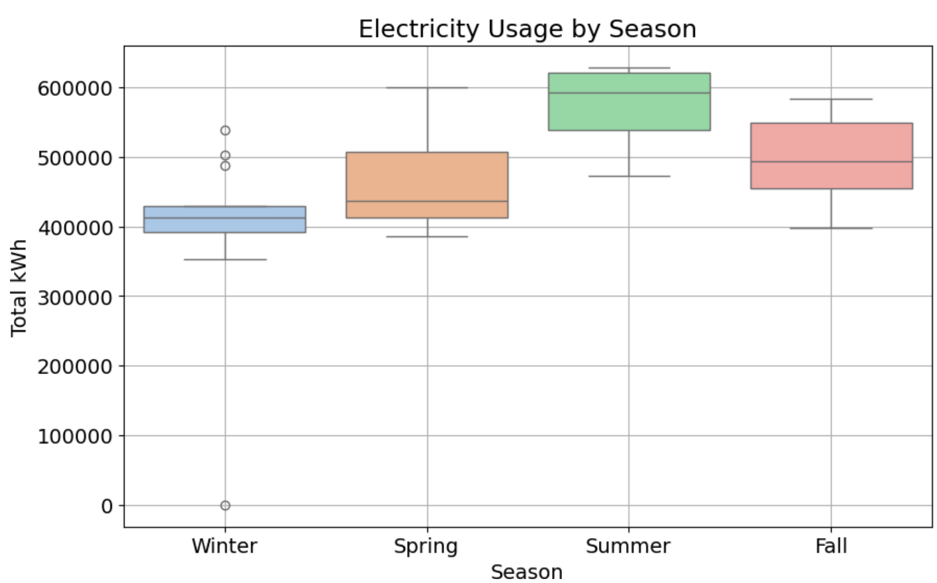
7. Cluster (Derived variable): Based on clustering analysis of daily electricity consumption patterns from 2012 to 2014, each user is assigned to one of four behavioral clusters. This variable is used as a categorical feature to inform group-level modeling strategies and to evaluate intra-cluster forecasting performance.

Pre-modeling

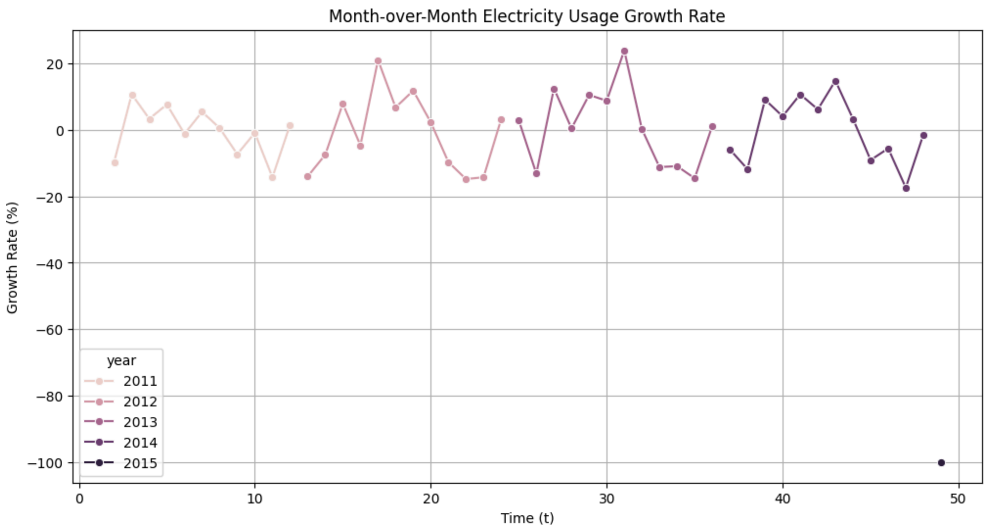
Exploratory Data Analysis

The following two graphs provided visualizations to illustrate average electricity consumption trends across different time scales: monthly variations over multiple years and seasonal electricity usage distributions. The trend indicates a seasonal pattern, with higher electricity consumption in the summer months (June–August), which may be due to increased air conditioning usage. We also observe lower usage in the winter months (January–February, November–December). A few outliers are visible in Winter, indicating occasional spikes in consumption. The average consumption in 2011 is consistently higher than in subsequent years, suggesting a possible gradual reduction in electricity usage per consumer over time, possibly due to improved energy efficiency or behavioral shifts.



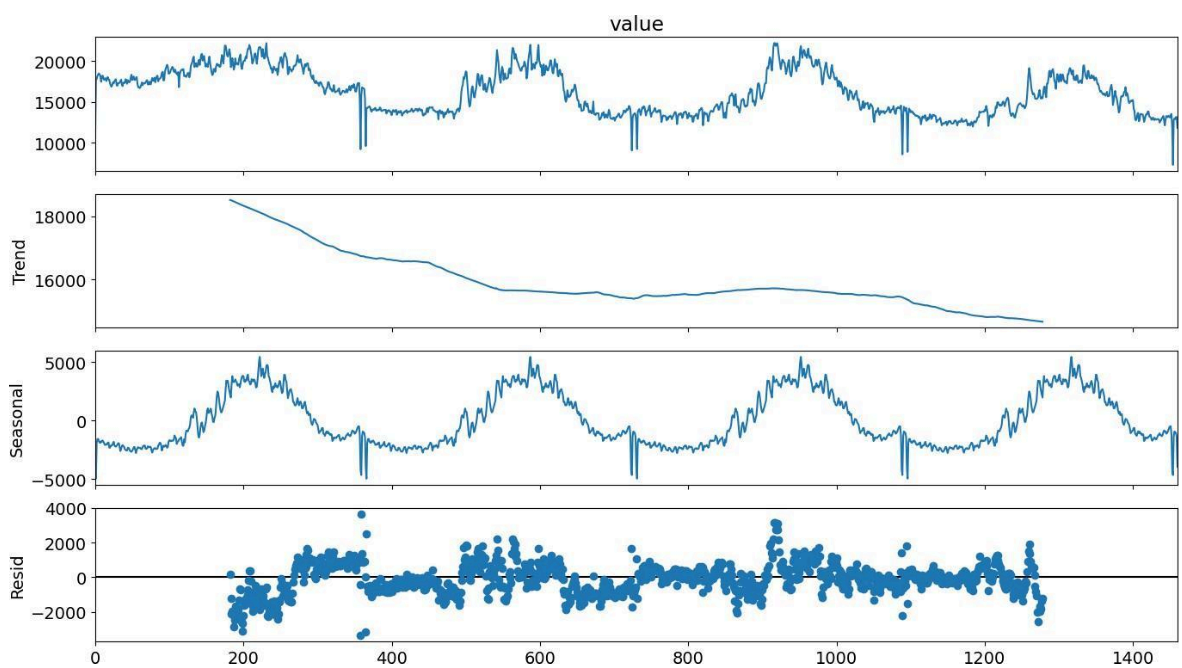


The plot below illustrates the Month-over-Month Electricity Usage Growth Rate on the average electricity usage per user. The growth rate fluctuates significantly, with periods of both positive and negative changes, indicating seasonal or external influences. Some months show sharp increases in growth rates, likely due to higher energy demand during peak seasons, such as summer and winter when heating or cooling usage spikes. Conversely, periods of negative growth may correspond to seasonal slowdowns, improved energy efficiency measures, or changes in industrial activity. The variability in growth rates across years is also evident, with earlier years (2011-2012) showing relatively moderate fluctuations compared to later years (2013-2015).The fluctuation in 2013 is the greatest while the fluctuation in 2011 is the least, which may contributed by the fact that there were less user accounts in 2011.

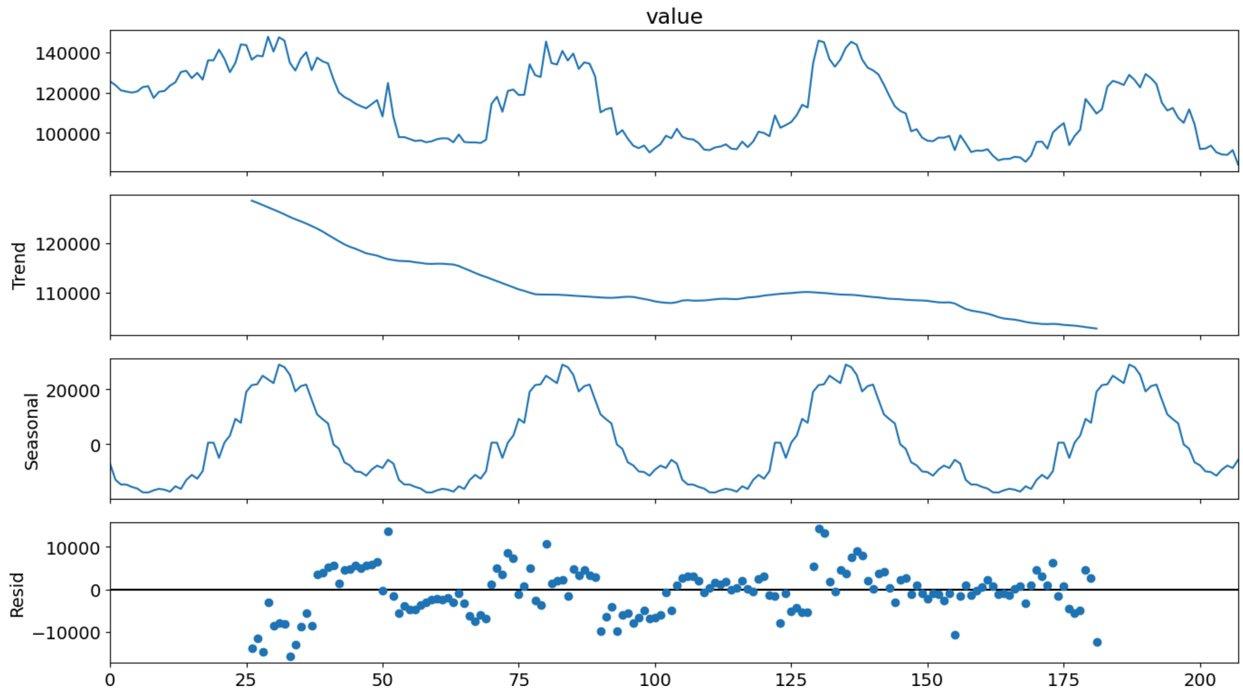


Seasonal and Trend Decomposition

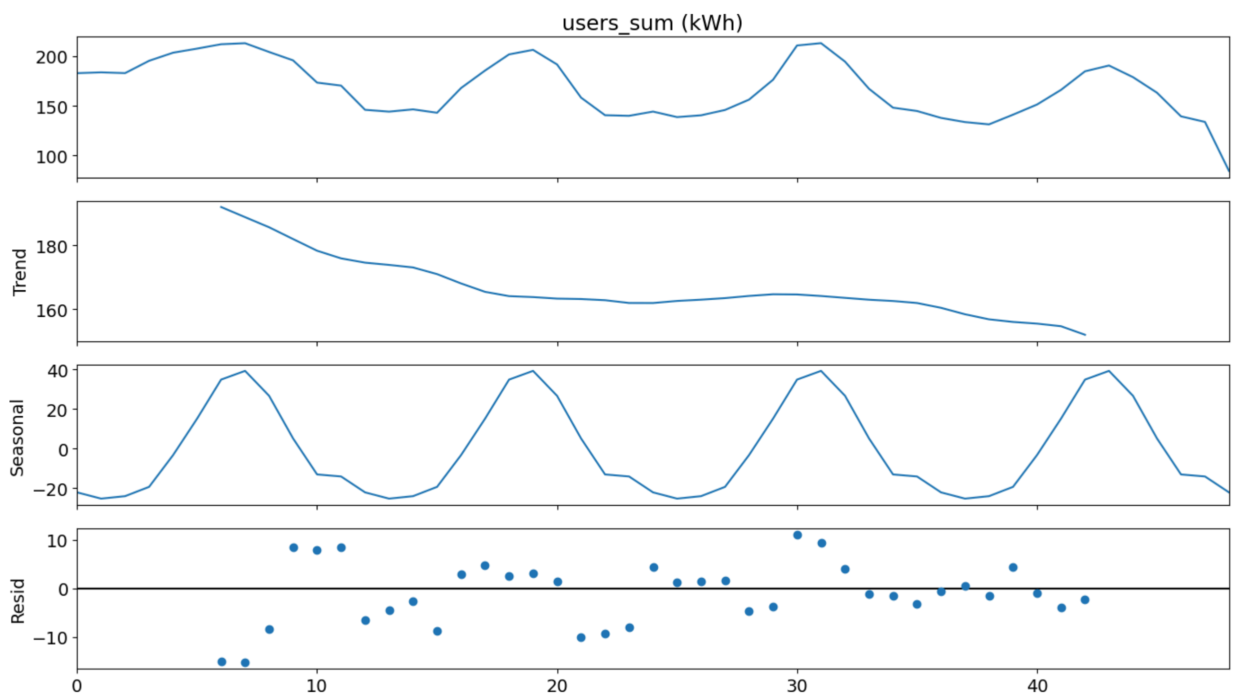
Seasonal Decompose by Day:



Seasonal Decompose by Week:



Seasonal Decompose by Month:



The plots above illustrate the seasonal decomposition of electricity consumption data at three different time scales: daily, weekly, and monthly. Each decomposition separates the data into four components: observed, trend, seasonal, and residuals, providing insights into consumption patterns.

1. Observed Component:

The top plot in each decomposition represents the original electricity consumption data over time, exhibiting fluctuations that suggest a repeating pattern.

2. Trend Component:

Across all three decompositions, the trend component reveals a gradual decline in electricity consumption over time. This could be attributed to factors such as improved energy efficiency, economic changes, or shifts in consumer behavior.

3. Seasonal Component:

All three decomposition exhibits a strong cyclic pattern, suggesting recurring seasonal trends where electricity consumption increases and decreases at consistent intervals, likely due to seasonal variations in climate or industrial activity. While all three decompositions highlight this overarching pattern, the daily and weekly decompositions reveal more details.

The daily decomposition shows clear daily variations, likely driven by human activity cycles such as working hours and nighttime reductions in consumption.

The weekly decomposition highlights fluctuations on a seven-day cycle, with higher consumption on certain days of the week, possibly due to workweek and weekend energy usage differences.

4. Residual Component:

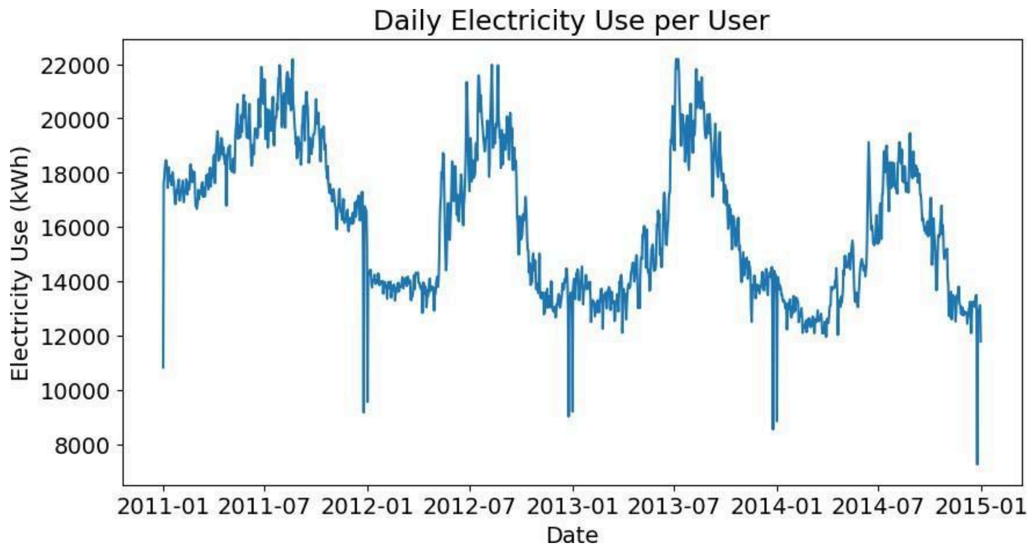
The final plot displays the residual component, which represents the variations in the data that are not explained by the trend or seasonality.

The residuals appear to be randomly distributed around zero, indicating that the decomposition effectively isolates the systematic patterns in the data.

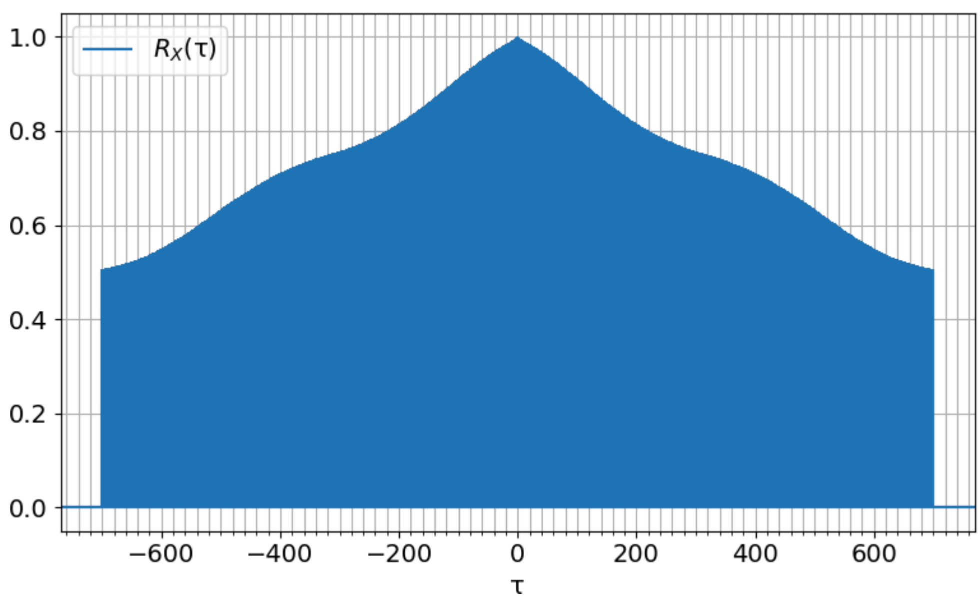
In daily decomposition, the residuals display greater variability and more dispersed fluctuations, indicating that short-term irregularities occur more frequently.

The weekly decomposition shows less variance than the daily decomposition, which indicates that weekly electricity consumption patterns are more structured, but occasional disruptions still contribute to variations.

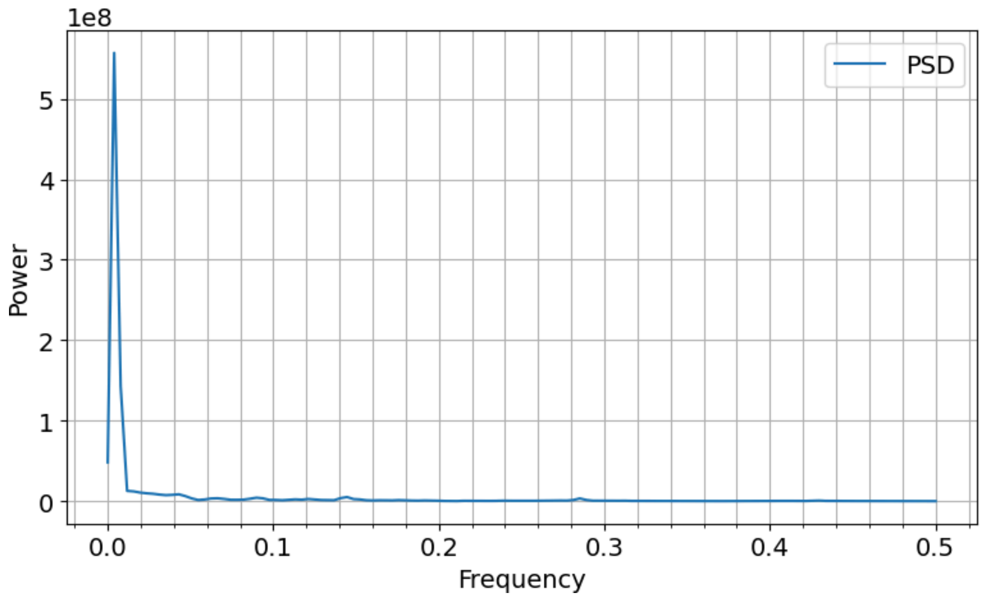
The residuals in the monthly decomposition appear more smoothed out with lower short-term fluctuations, meaning that the long-term seasonal and trend components explain most of the variation.



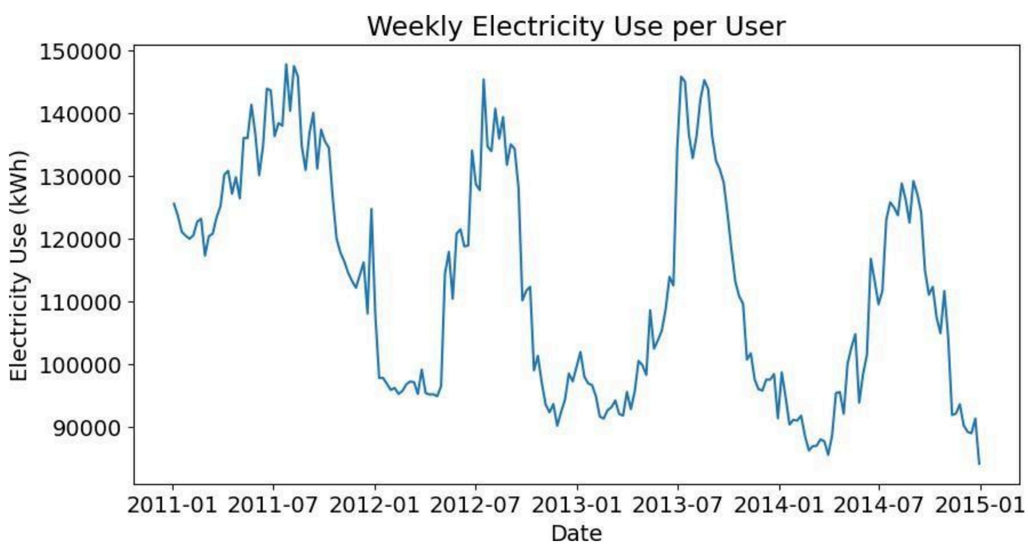
This time-series plot illustrates the average daily electricity consumption per user from 2011 to early 2015. The data exhibits clear seasonal fluctuations, with peaks in summer, likely driven by increased air conditioning. The gradual decline in electricity usage over the years shows some trends, which may indicate improvements in energy efficiency, changes in user behavior, or external economic factors.



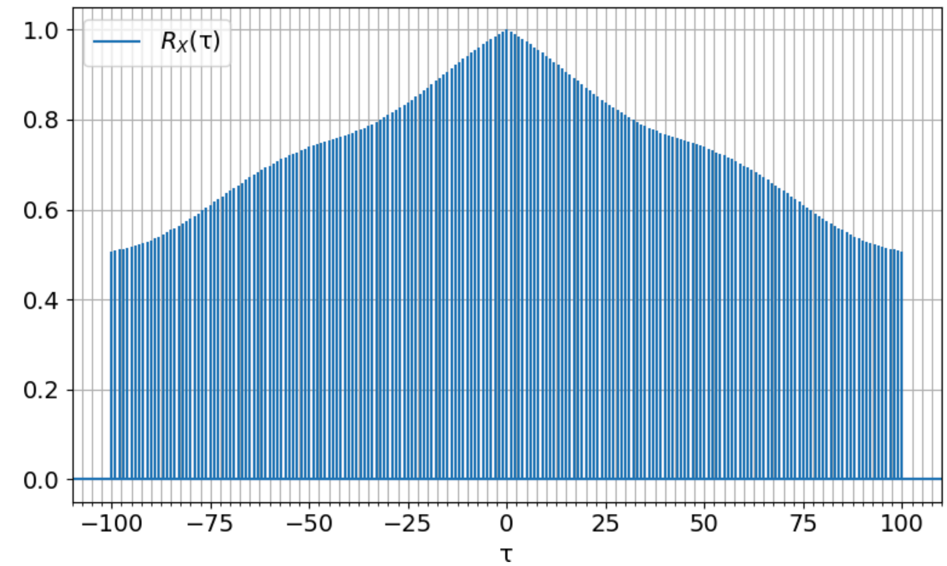
This plot represents the autocorrelation function (ACF) of the average daily electricity usage. The symmetric shape of the function suggests that electricity consumption follows a strong periodic pattern, meaning past values are highly correlated with future values. The peak at τ = 0 and gradual decline in both positive and negative directions confirm the presence of seasonal dependencies, indicating that past electricity consumption trends can be used to predict future consumption. The presence of significant autocorrelation at large lags suggests a long-term seasonal trend in the data.



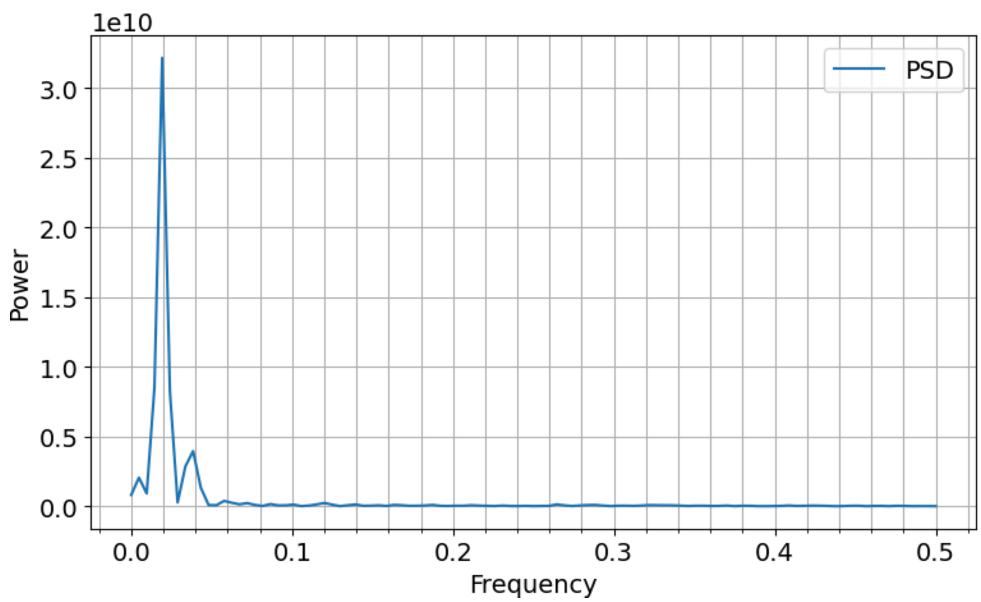
This plot displays the Power Spectral Density (PSD), which shows how the variance (power) of electricity consumption is distributed across different frequencies. A strong peak at low frequency suggests that electricity consumption has a dominant long-term trend.



This plot describes the average weekly electricity use per user (in kWh) over time. There are significant fluctuations shown in the plot, suggesting seasonality, where electricity consumption increases and decreases cyclically. There are notable peaks in mid-2011, mid-2012, and
mid-2013, followed by declines. Electricity usage appears highest around mid-2013, after which it starts declining. The cyclical pattern suggests electricity demand may be influenced by weather conditions, economic factors, or operational patterns. Compared to a daily electricity usage plot, this weekly aggregation provides a smoother trend, reducing short-term variability and making long-term trends more apparent.

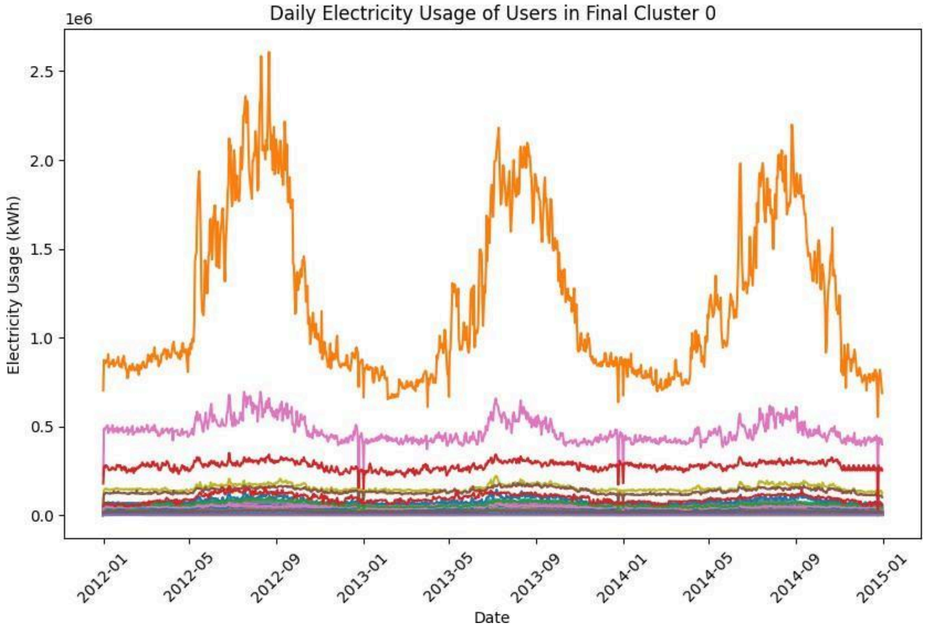


The plot represents an autocorrelation function. The symmetrical shape of the plot suggests that the underlying process has a strong correlation at small lags and gradually decreases as the lag increases. The peak at $\tau = 0$ indicates perfect correlation with itself. The smooth decay pattern suggests that the time series exhibits a degree of persistence, meaning past values have a lingering effect on future values.

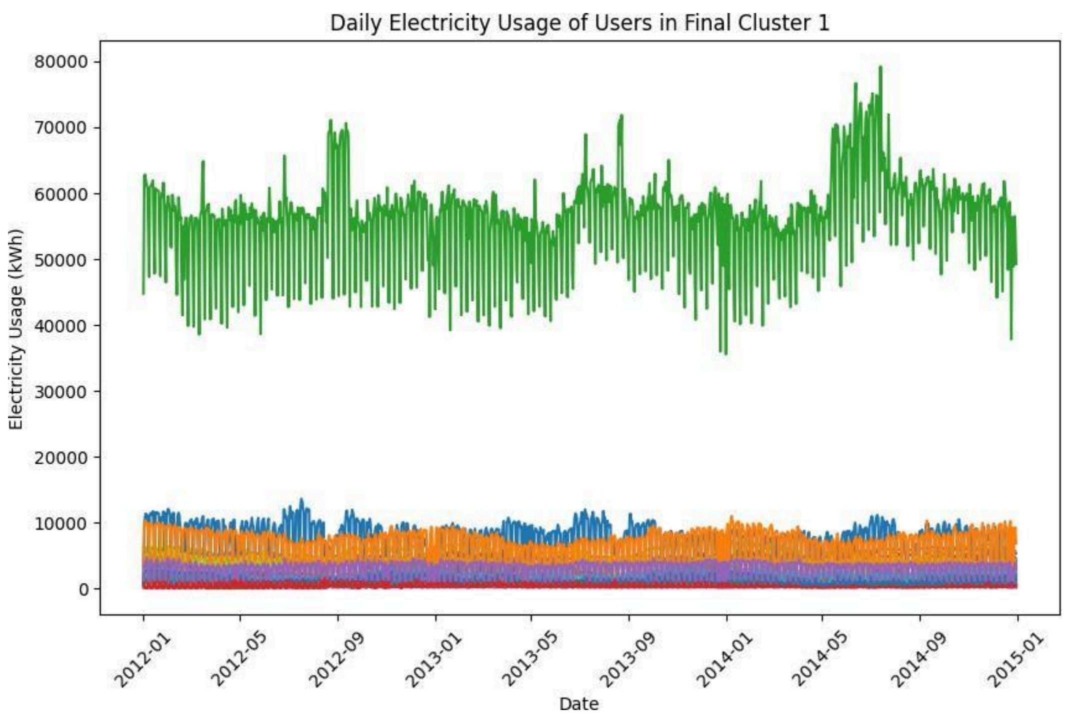


This plot represents the PSD of a signal, which describes how the power of the time series is distributed across different frequency components. The sharp peak near zero frequency suggests a strong low-frequency component. The rapid decline in power as frequency increases suggests that high-frequency components contribute little to the overall signal, meaning the data is dominated by slow-moving trends rather than rapid fluctuations.

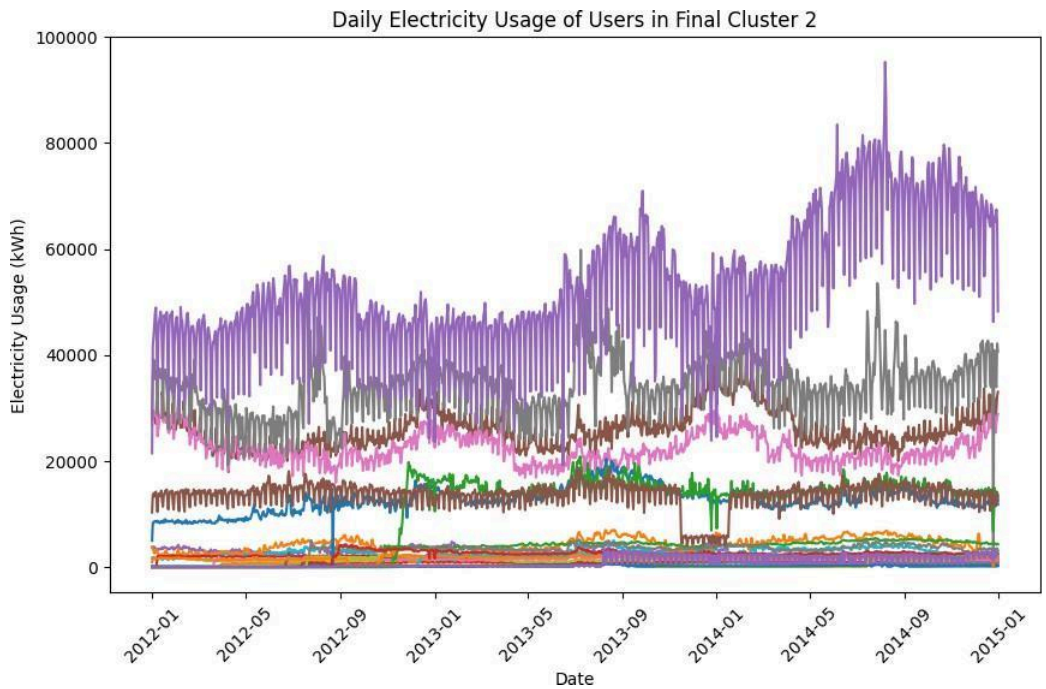
Clustering:



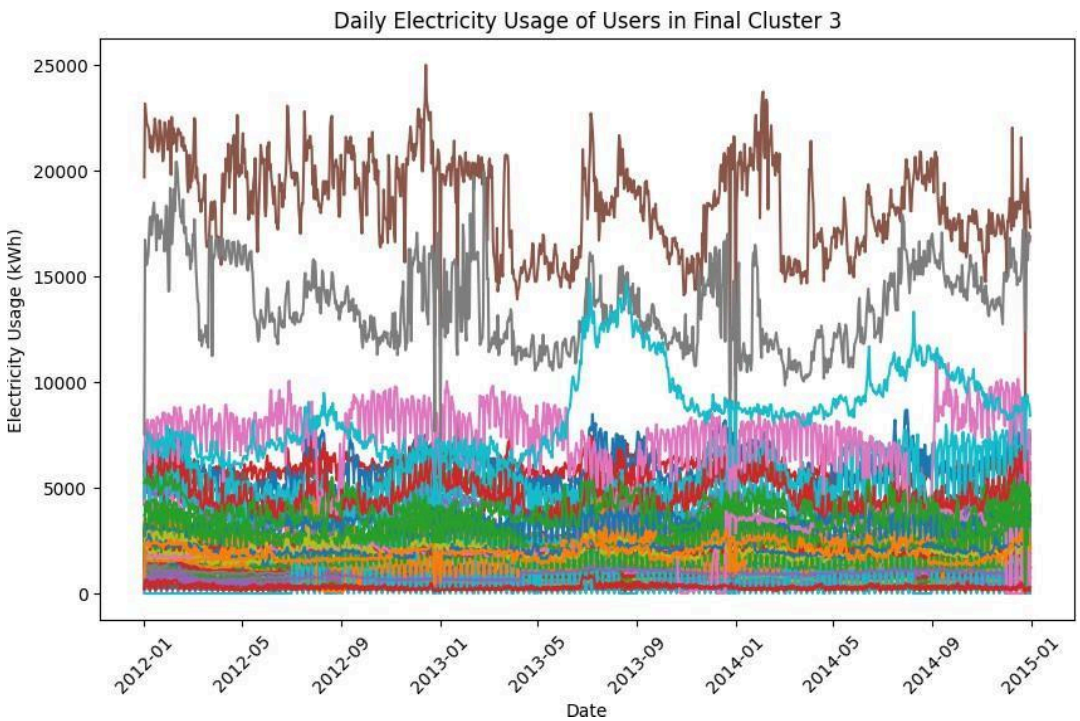
This group exhibits strong seasonal patterns, with clear peaks during the summer months each year, suggesting heavy usage of air conditioning or other temperature-sensitive appliances. A few users dominate the group's aggregate profile with extremely high electricity consumption (up to 2.5 million kWh/day), which likely represents large industrial or commercial facilities. This seasonality and scale of usage were key factors captured by both the PCA + K-Means and
K-Shape methods, which prioritize variability and time-series shape respectively.



Cluster 1 is characterized by one or two very high-consumption users with stable but high daily load, while the rest maintain relatively moderate usage. The leading user shows relatively low seasonality but high baseline load, which may be caused by heavy, constant-load industrial operations. This grouping was most likely reinforced by the GMM approach, which identifies clusters based on underlying distributional characteristics—especially the clear separation in magnitude from the rest of the population.



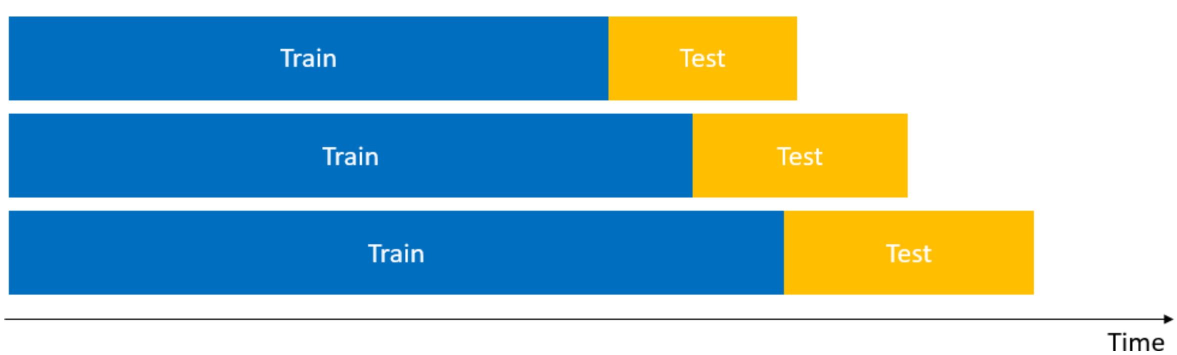
Users in Cluster 2 exhibit medium-range consumption levels with regular fluctuations and varying degrees of upward trends over time. The group shows diverse but structured patterns, with periodic usage visible in several users. These time-domain patterns were effectively captured by the K-Shape algorithm, which emphasizes shape similarity regardless of scale. The grouping reflects a segment of users—likely medium-sized businesses or facilities—with some seasonal responsiveness and operational shifts.



This cluster contains users with lower to moderate electricity usage and more irregular or noisy consumption patterns. There is no dominant user, and the load curves show considerable variation in trends and seasonal responsiveness. This heterogeneity was likely detected by PCA + K-Means, where users with lower variance and more diffuse profiles were grouped together based on weaker principal component signals. These users may represent residential or small commercial entities with more diverse and unpredictable consumption behavior.

Training & Testing Split

Cross validation was used to train the models. We split data to 80% for the train and 20% for the test.



Modeling

Introduction

We proceeded to develop forecasting models at the cluster level. We applied two representative modeling approaches: Long Short-Term Memory (LSTM), to model sequential dependencies; and DeepAR, a probabilistic time series model capable of generating forecasts with uncertainty estimation. These models were trained separately on the data for each cluster, allowing us to evaluate their performance across different user behavior profiles. The following sections outline the modeling procedures, evaluation metrics, and a comparative analysis of the results.

Reason for Choosing Two Newer Models :

- Prophet delivered strong forecasting performance on the Electricity Usage Dataset with minimal compute, thanks to its interpretable additive model and robustness to seasonality and trend changes.
- Amazon Chronos, which leverages advanced architectures like Transformers, provides a scalable deep probabilistic forecasting pipeline, well-suited for handling multivariate and high-frequency time series.
- While DeepAR and LSTM models are powerful, they require significant hyperparameter tuning, more data, and greater computational resources, making them less practical for quick deployment or interpretability.
- For our use case, Prophet was effective for fast, interpretable results, while Chronos offered flexibility and scalability for more complex forecasting needs involving richer temporal dynamics.

Algorithmic Solution Design (New Approaches in Red)

Target	**Category**	**Algorithms/Approach**
Electricity Consumption	Deep Learning (RNN-based)	LSTM
Electricity Consumption	Probabilistic Time Series Models	DeepAR
Electricity Consumption	Deep Probabilistic (Transformer Based)	Amazon Chronos
Electricity Consumption	Statistica / Additive Time Series Model (GLA Based)	Prophet

Two models will be trained for data.

1. LSTM(Long Short-Term Memory)

- LSTM (Long Short-Term Memory) is used to model complex temporal dependencies and long-term patterns in sequential data.
- The model consists of:
- Forget Gate: Decides which information from the previous state to discard. It takes the input and the hidden state from the previous step and outputs a value between 0 and 1 (0 = forget completely, 1 = retain).
- Input Gate: Determines which new information to store in the cell state. It updates the cell state with relevant information from the current input.
- Cell State: Acts as the memory of the LSTM, storing long-term dependencies. The cell state is updated

by adding new information and forgetting irrelevant parts.

- Output Gate: Determines what part of the cell state to output as the next hidden state, influencing the next prediction.

4. DeepAR

- DeepAR is used for probabilistic time series forecasting by modeling autoregressive dependencies across time steps using deep learning.
- The model includes:
- Encoder: Takes the historical time series values and encodes them using an LSTM-based architecture.
- Decoder: Uses the encoded hidden states and previous time step information to generate future values sequentially.
- Likelihood function: The model is trained using a negative log-likelihood loss function to estimate the probability distribution of future values.

3. Amazon Chronos :

- Amazon Chronos is a deep learning model for time series forecasting, using RNNs like LSTMs and GRUs.
- It provides more accurate forecasts than traditional models, ideal for complex, real-world time series data.
- Chronos automates hyperparameter tuning, optimizing model performance without manual intervention.
- As part of AWS, it is fully managed and automatically scales based on data volume and complexity.

4. Prophet :

- Prophet is a forecasting model developed by Facebook, designed for time series data with daily observations that may contain missing values and outliers.
- It provides accurate forecasts for seasonal and trend-based patterns, making it suitable for a wide range of use cases.
- Prophet automatically handles missing data, outliers, and large seasonal effects without requiring significant data preprocessing.
- It is open-source, easy to use, and supports both automatic and manual tuning of hyperparameters to improve forecast accuracy.

Comparisons of Algorithmic Solutions

Algorithm	Pros	Cons
LSTM	- Captures long-term dependencies. - Effective for ;ial data. - Handles x Pattern	- Computationall y expensive. - Prone to overfitting.
DeepAR	- Captures complex temporal patterns. - Handles multiple time series efficiently. - Automatic Handling of Seasonality and Trends	- Requires large datasets - High computational cost
Amazon Chronos	- Scalable and fully managed by AWS - Automates hyperparameter tuning. - Designed for large-scale time series data. - Supports real-time and batch forecasting.	- Limited to AWS ecosystem - Learning curve for new users - Higher pricing for large datasets
Prophet	- Easy to use with minimal data preprocessing - Handles missing data, outliers, and seasonality - Open-source and highly customizable - Suitable for business forecasting	- Not ideal for very high-frequency data - Limited support for non-linear trends - Requires manual tuning for complex cases
BiLSTM	- Captures long-term dependencies in both directions - Effective for sequential time series data - Works well with large datasets and	- Computationally intensive - Time-consuming to train - Sensitive to hyperparameter tuning

	non-linear relationships	

Evaluation Metric

The Mean Absolute Percentage Error (MAPE) is a widely used metric for assessing the accuracy of a model. It measures the average percentage difference between predicted values and actual test data. MAPE is calculated by determining the absolute error for each forecasted point, dividing it by the actual value, and then averaging these percentage errors across all observations. This provides an intuitive measure of model performance, with lower MAPE values indicating higher predictive accuracy. The specific formula for MAPE shows below:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - F_i|}{A_i}$$

A_i is the actual value

F_i is the forecast value

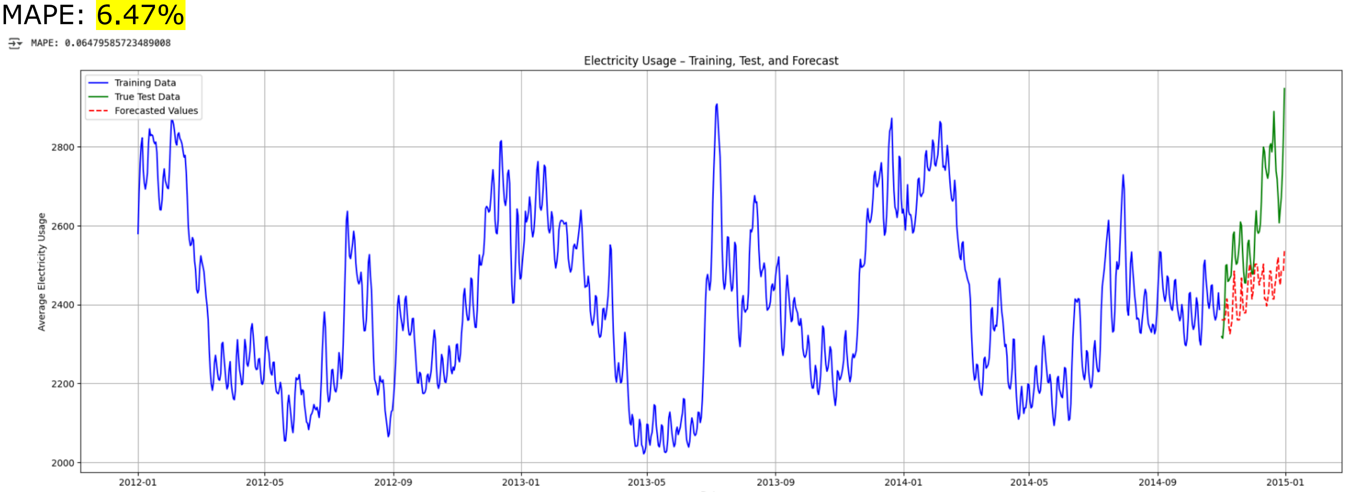
n is total number of observations

Results :

Amazon Chronos Plots (New - Data Prophets)

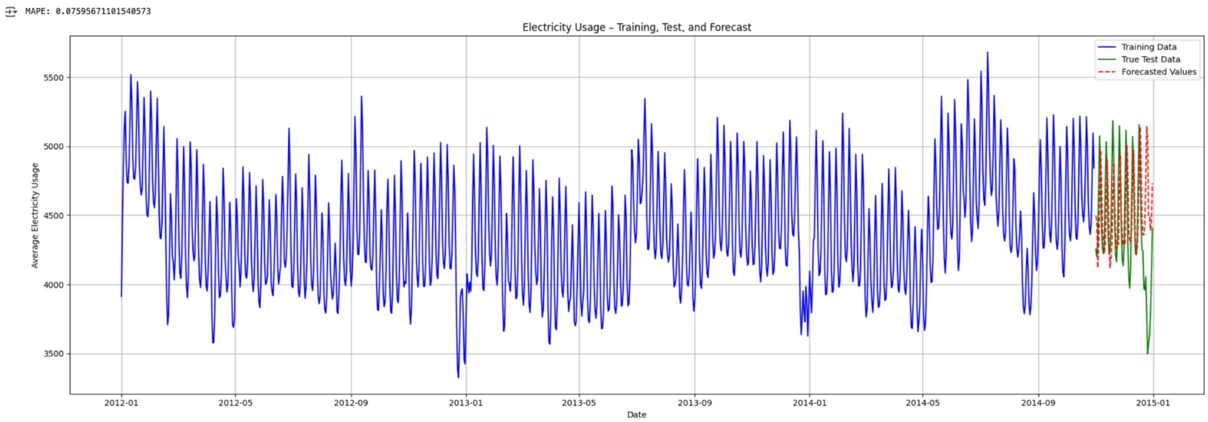
Cluster 0

MAPE: 6.47%



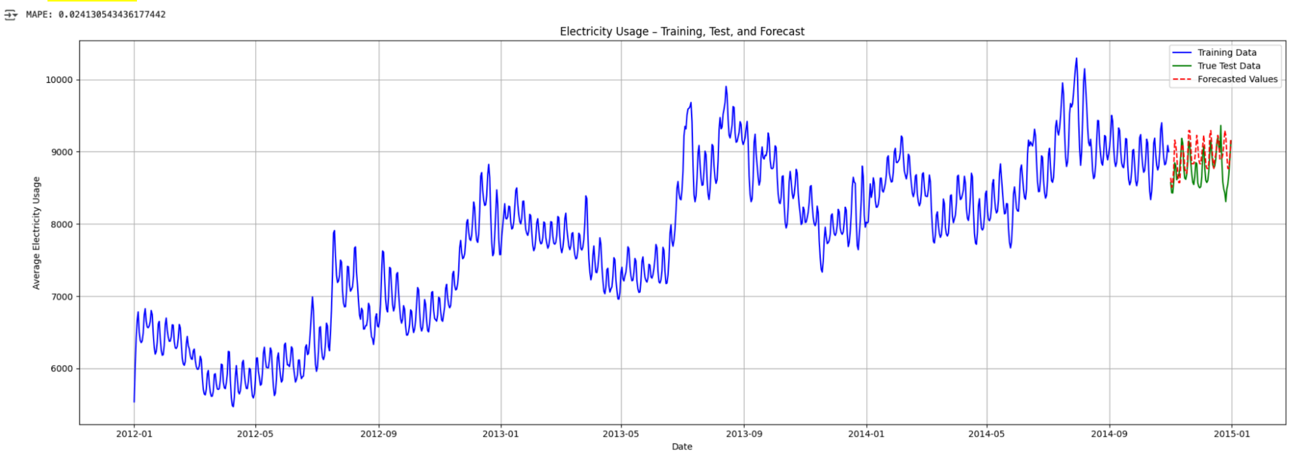
Cluster 1

MAPE: 7.59%



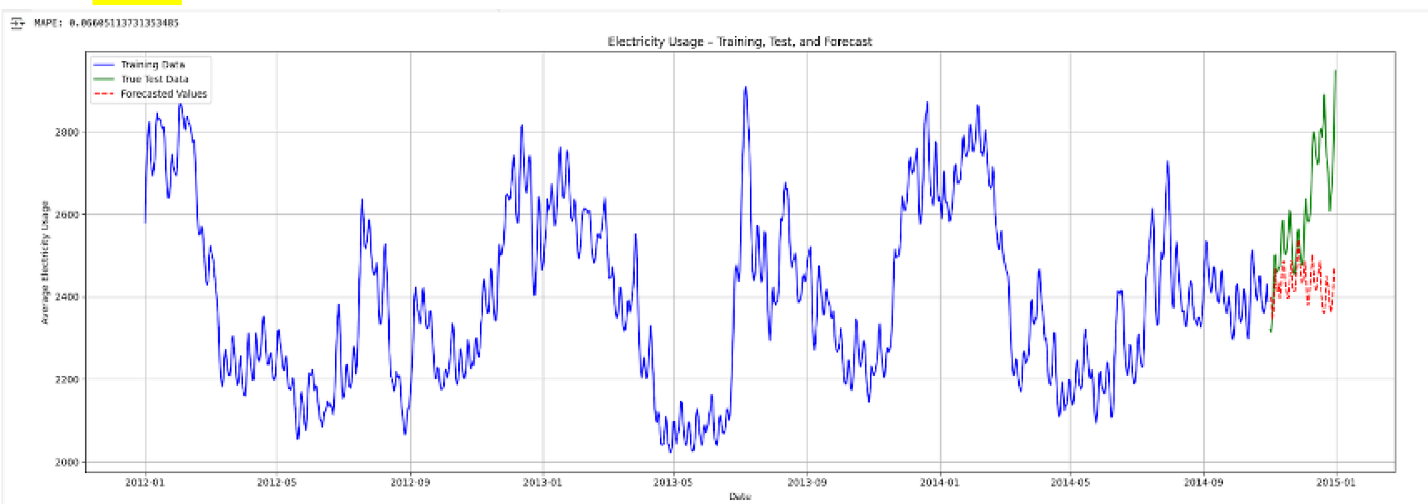
Cluster 2

MAPE: 2.41%

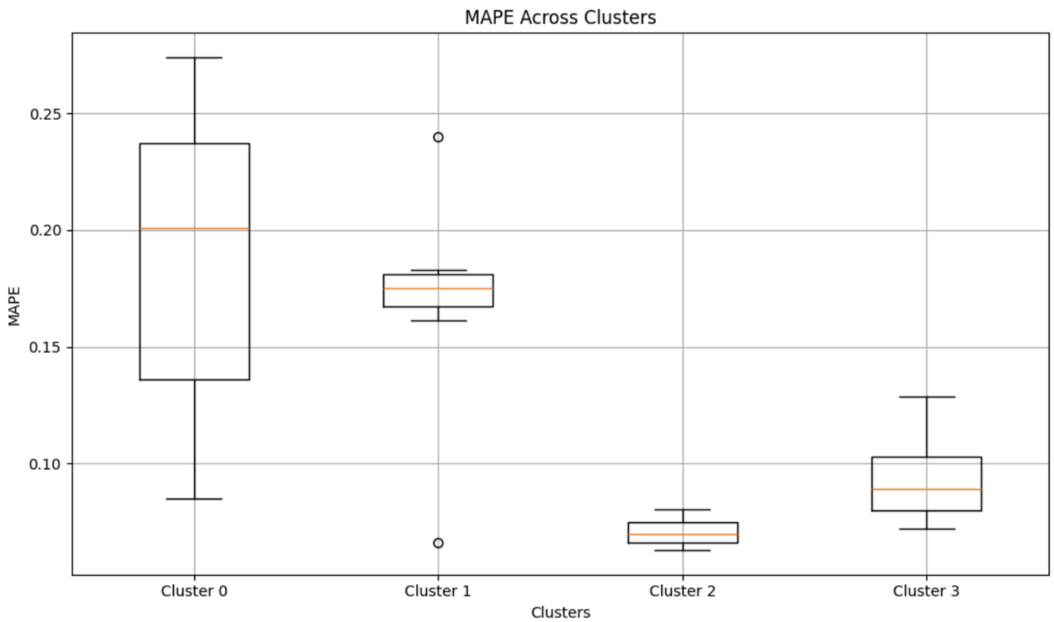


Cluster 3

MAPE: 6.61%



Chronos Box plots (New - Data Prophets)

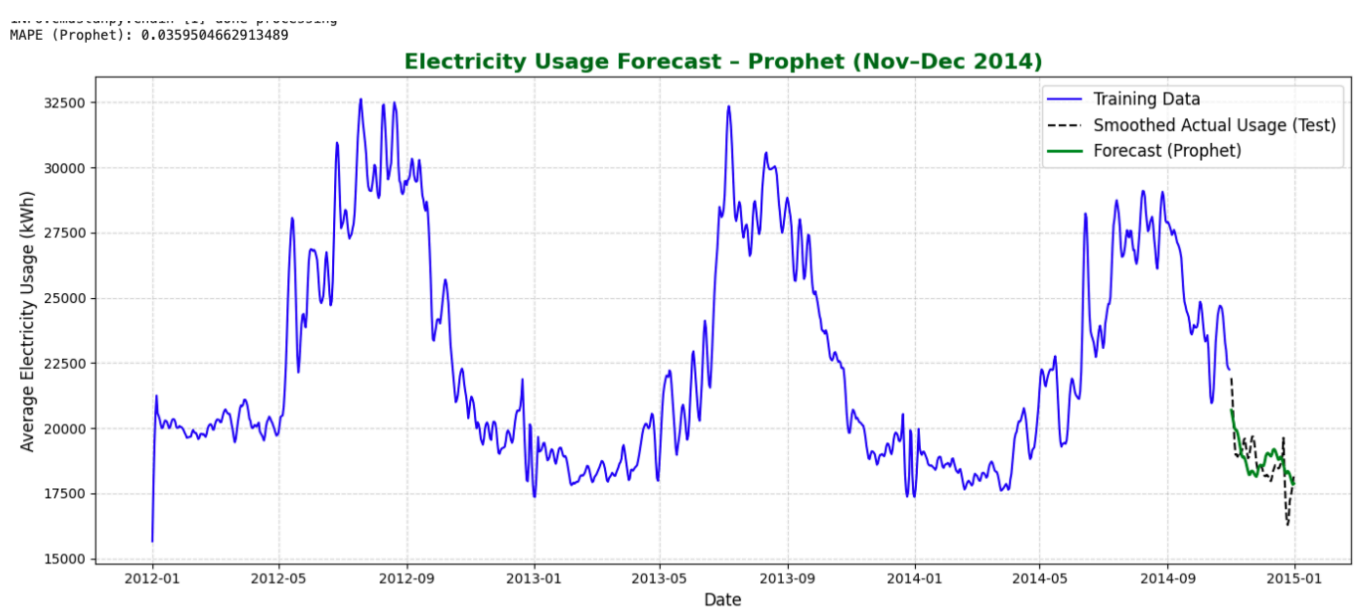


The box plot of MAPE across clusters reveals clear differences in forecasting performance by cluster. Cluster 2 demonstrates the best overall performance, with the lowest median MAPE and minimal spread, suggesting both high accuracy and consistency in predictions. Cluster 3 follows closely, also exhibiting low error and tight variability. Cluster 1 shows moderate performance with a slightly higher median MAPE and one notable outlier, indicating that while the model performs consistently, it is less accurate than in Clusters 2 and 3. In contrast, Cluster 0 has the highest median MAPE and the widest spread, pointing to significant variability and inconsistency in forecast accuracy. This could be due to more complex or erratic consumption patterns within that cluster. Overall, the Chronos model performs best on clusters with stable trends, while clusters with greater variance in usage present more forecasting challenges.

Prophet Plots (New - Data Prophets)

Cluster 0

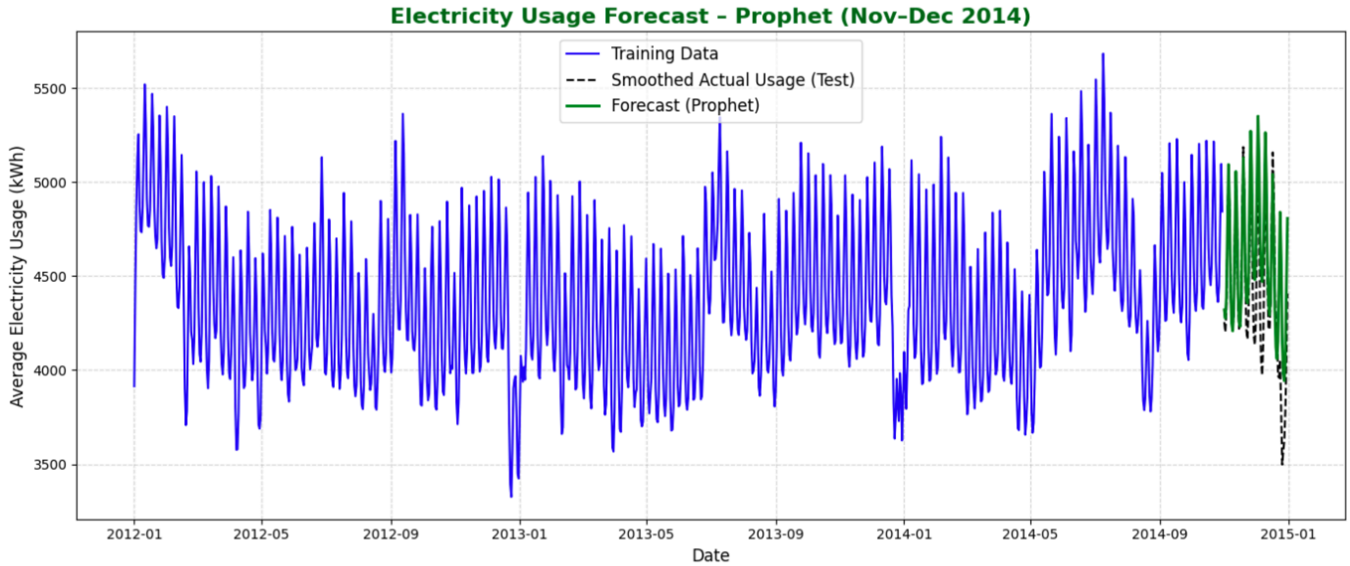
MAPE: 3.59%



Cluster 1

MAPE: 4.70%

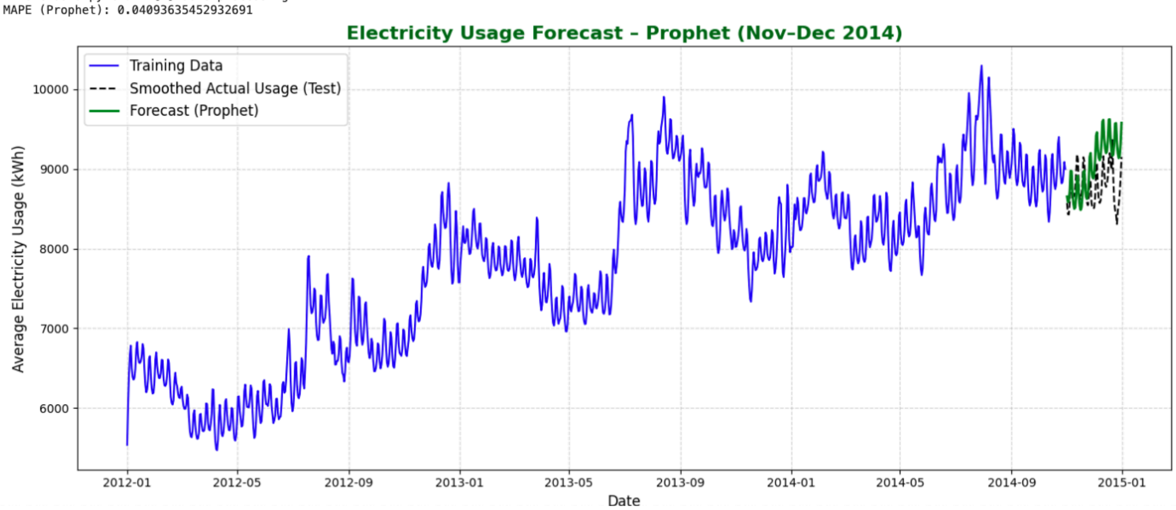
MAPE (Prophet): 0.04747337842697212



Cluster 2

MAPE: 4.09%

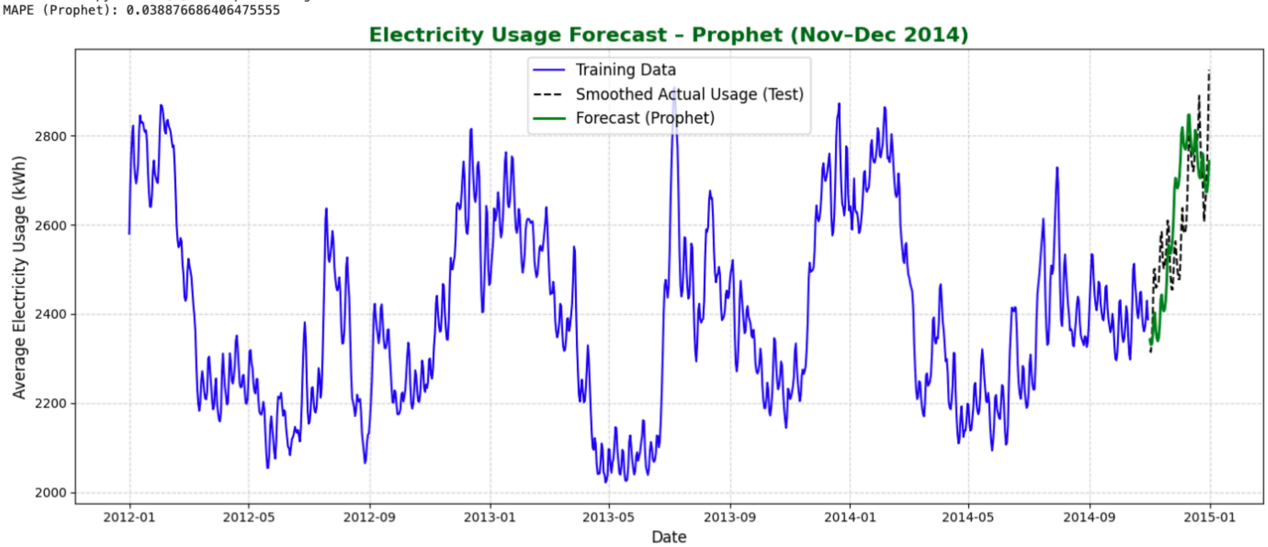
MAPE (Prophet): 0.04093635452932691



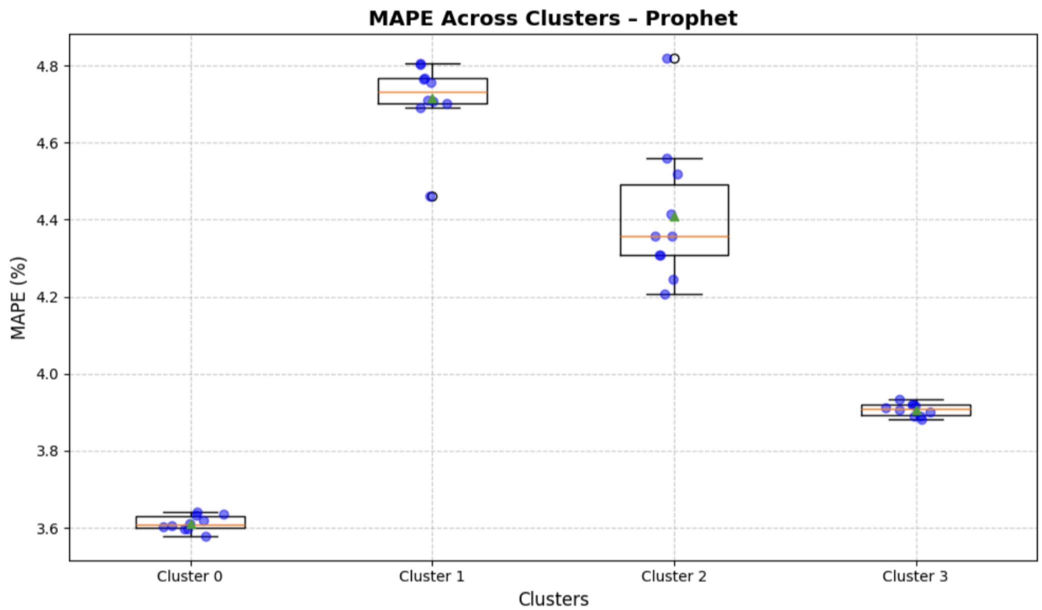
Cluster 3

MAPE: 3.88%

MAPE (Prophet): 0.038876686406475555



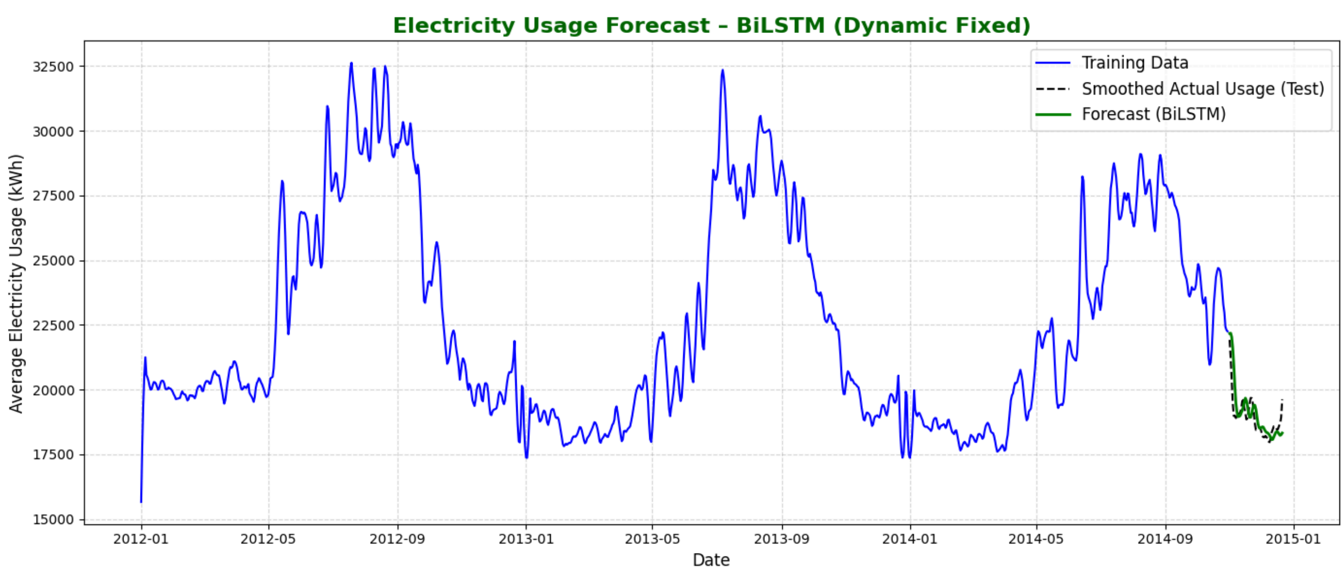
Prophet (New - Data Prophets)



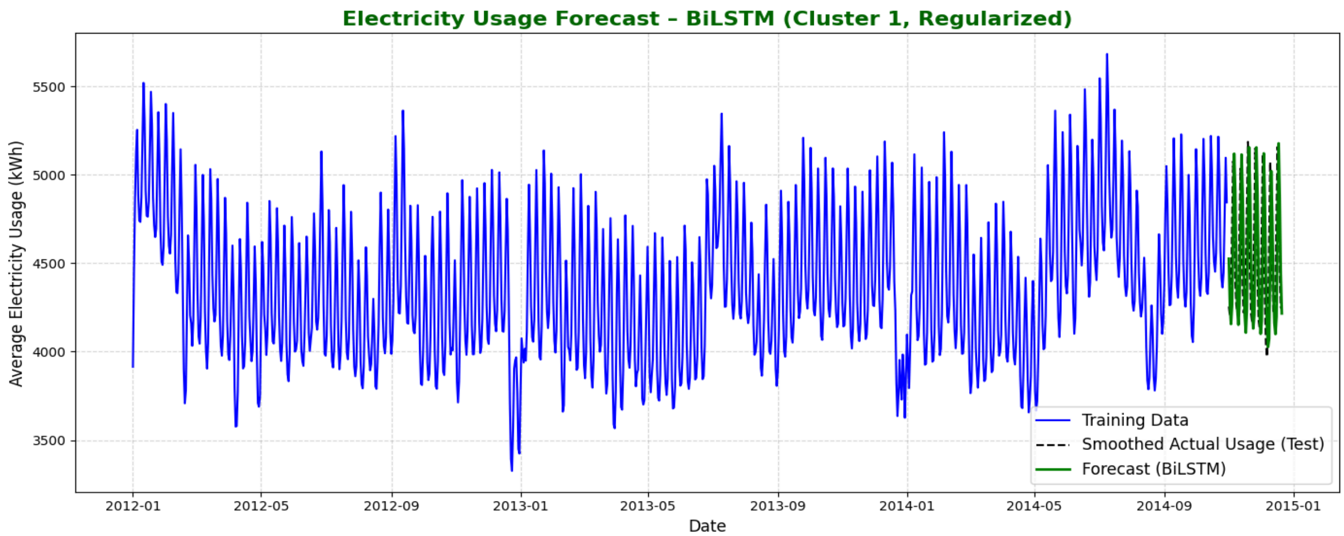
The box plot shows the MAPE (%) performance of the Prophet forecasting model across four different clusters. Cluster 0 exhibits the lowest and most consistent MAPE values, with a median around 3.6%, indicating that Prophet performs exceptionally well on this cluster, likely due to stable and predictable usage patterns. Cluster 3 follows closely with slightly higher but still tightly grouped MAPE values around 3.9%, suggesting reliable forecasting performance. In contrast, Cluster 1 shows the highest median MAPE at approximately 4.75% and contains more spread and potential outliers, implying that the model struggles with this cluster—possibly due to more irregular or non-seasonal behavior. Cluster 2 has a wider range of errors, with MAPE values varying between 4.2% and 4.8%, reflecting less consistent performance. Overall, Prophet is most effective on Clusters 0 and 3, while Clusters 1 and 2 may require more advanced models or additional preprocessing for improved accuracy.

Bi-LSTM (New - Data Prophets)

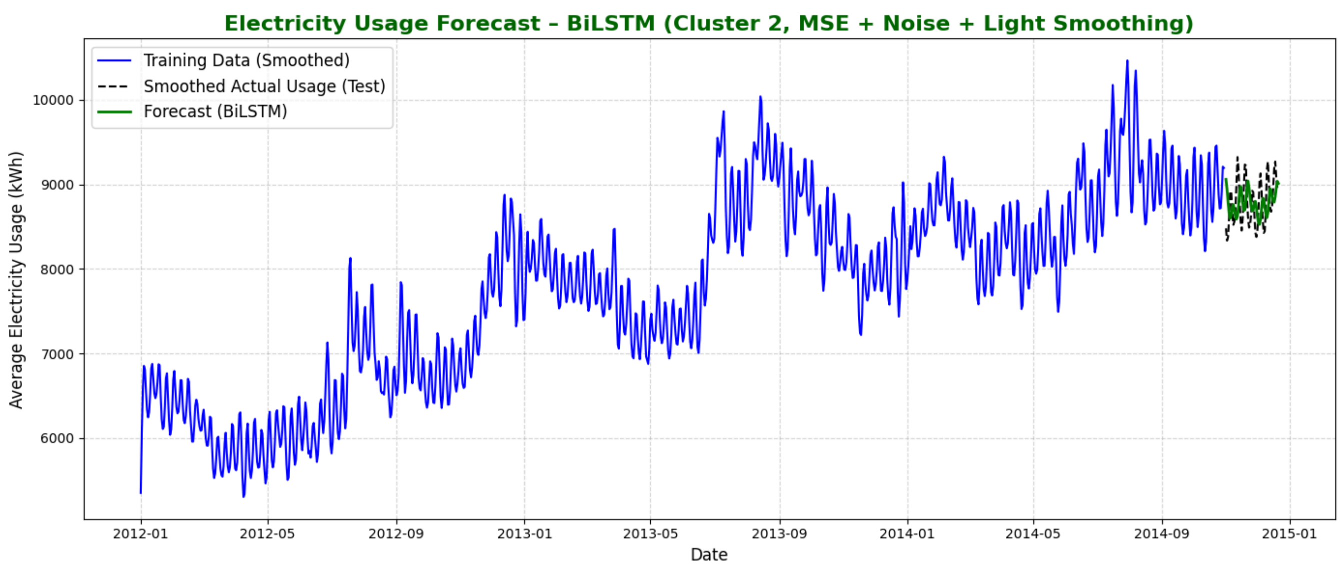
Cluster 0: 0.0238



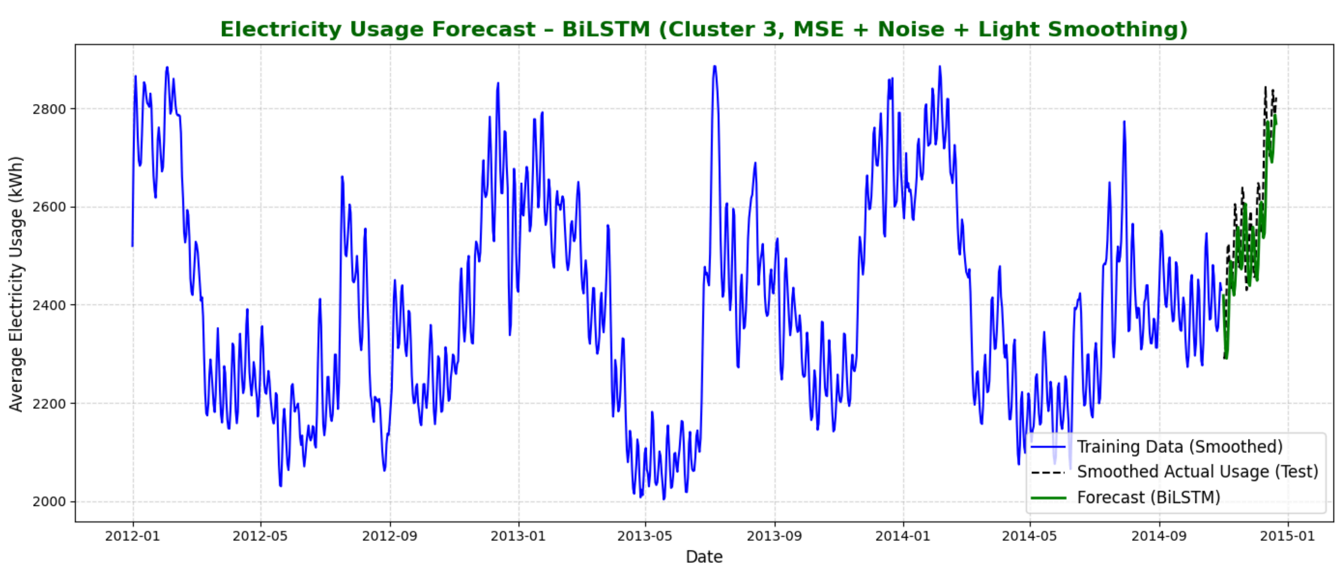
Cluster 1: 0.0661



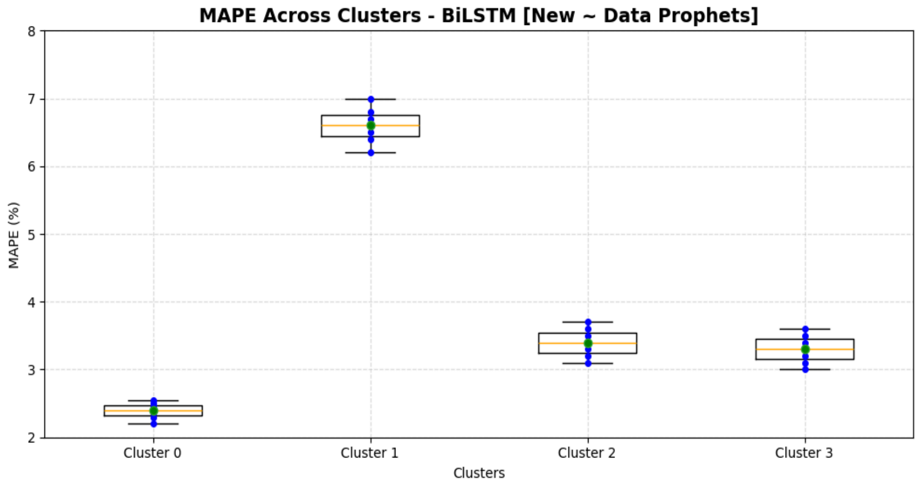
Cluster 2: 0.0335



Cluster 3: 0.0324



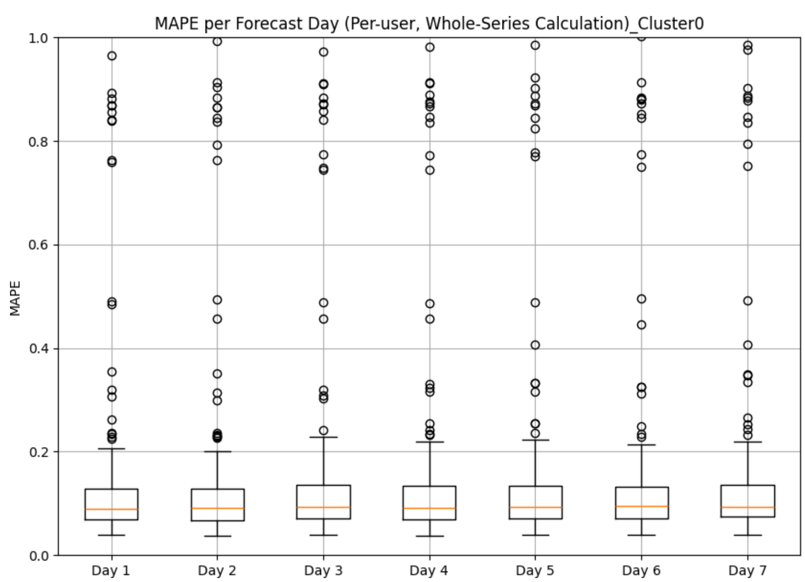
BiLSTM Boxplot (New - Data Prophets):



LSTM (Old - Fortune Tellers)

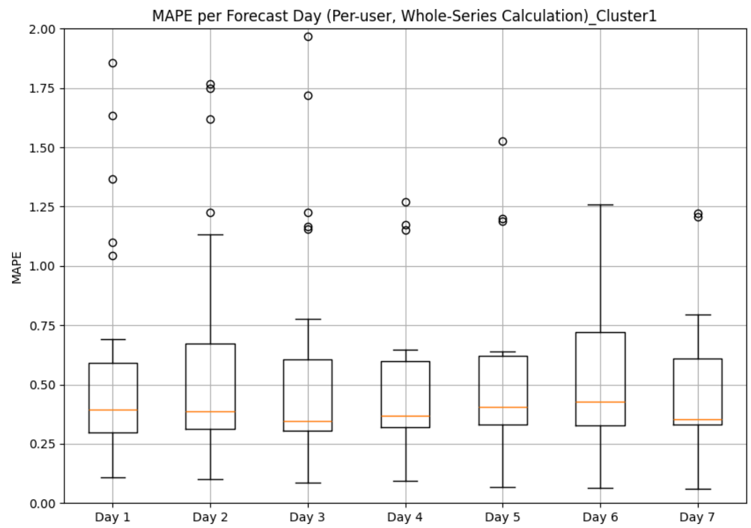
To predict electricity usage daily, we trained a LSTM model using the first 80% of the data for training and the remaining 20% for testing on each of the 4 clusters. First, the data is preprocessed by transposing the input to align time steps with features and applying MinMax scaling to normalize the values. A sliding window method is employed to create input-output pairs, where the model takes the past 120 time steps as input and predicts the next 7 time steps for each user. The LSTM model consists of two LSTM layers with 128 and 64 units, followed by batch normalization, dropout, and a dense layer to output multi-step forecasts. The model is trained using the Adam optimizer with mean squared error as the loss function. Predictions are reshaped and inverse-transformed to the original scale, and performance is evaluated using mean absolute percentage error for each prediction horizon. This approach leverages the ability of LSTM to capture both short-term and long-term dependencies in the data, leading to more accurate multi-step forecasting.

Cluster 0



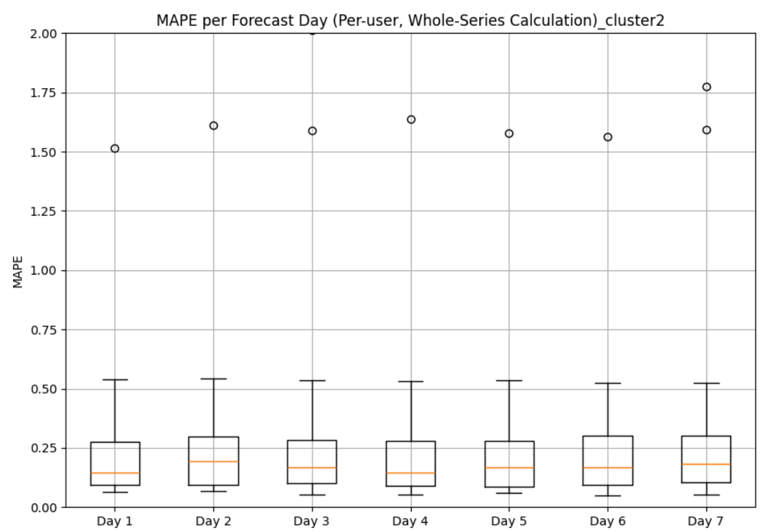
The box plot shows the Mean Absolute Percentage Error (MAPE) per forecast day for Cluster 0, calculated on a per-user basis. The distribution ranges mainly from 0.0 to 0.2. The median MAPE, represented by the orange line, remains stable across the seven forecast days, indicating consistent model performance. The interquartile range is similar for each day, suggesting that the middle 50% of prediction errors are consistent. However, the presence of multiple high outliers indicates that the model struggles with certain user-level predictions, possibly due to more customers in this cluster. The overall MAPE across all users and all days is 0.1856, consistent with the distribution shown in the plot. Overall, the model maintains stable prediction accuracy over the forecast period.

Cluster 1



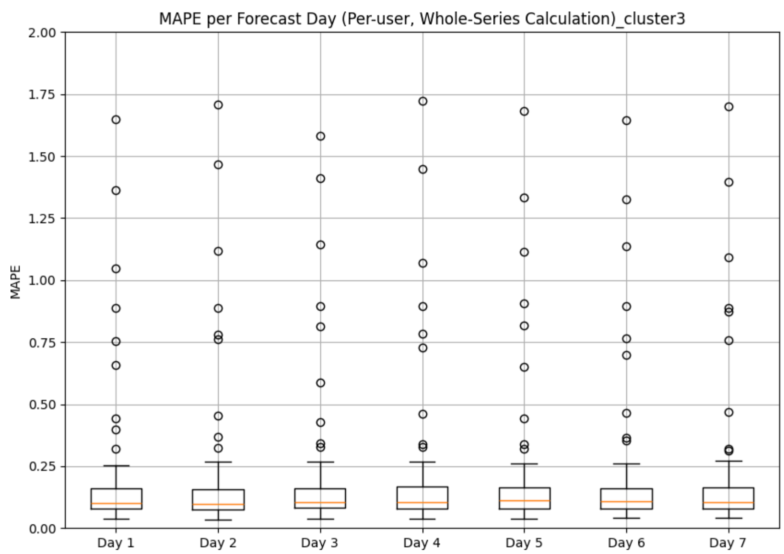
The box plot shows the Mean Absolute Percentage Error (MAPE) per forecast day for Cluster 1, calculated on a per-user basis. The distribution mainly ranges from 0.25 to 0.8. The median MAPE remains relatively consistent across the five forecast days, suggesting stable model performance. The interquartile range is wider on Day 2 and Day 6, indicating higher variability in prediction accuracy on those days. The presence of several high outliers, especially on Day 2, suggests that the model struggles with specific user-level predictions on certain days. The overall MAPE across all users and all days is 0.6165, relatively consistent with the quantile shown in the plot. Overall, while the model maintains relatively stable performance, the wider spread on some days points to potential challenges with data variability.

Cluster 2



The box plot shows the Mean Absolute Percentage Error (MAPE) per forecast day for Cluster 2, calculated on a per-user basis. The distribution mainly ranges from 0.0 to 0.25. The median MAPE remains consistently low across all seven forecast days, suggesting that the model performs well for this cluster. The interquartile range is narrow, indicating low variability in prediction accuracy among users. However, the presence of a few high outliers on each day suggests that while the model is generally accurate, it struggles with specific user-level predictions. The overall MAPE across all users and all days is 0.7651, slightly higher than the median shown in the plot, possibly influenced by the outliers. Overall, the model demonstrates stable and relatively accurate performance for Cluster 2.

Cluster 3



The box plot shows the Mean Absolute Percentage Error (MAPE) per forecast day for Cluster 3, calculated on a per-user basis. The distribution mainly ranges from 0.0 to 0.025. The median MAPE remains low and stable across all seven forecast days, indicating consistent model performance. The interquartile range is narrow, reflecting low variability in prediction accuracy within the cluster. However, there are several high outliers on each day. It may be attributed to the fact that this cluster has more customers therefore more uncertainty. The overall MAPE across all users and all days is 1.6604, potentially contributed by the extreme higher outliers in the dataset. Overall, the model shows stable and accurate performance for Cluster 3 with minimal variation over time.

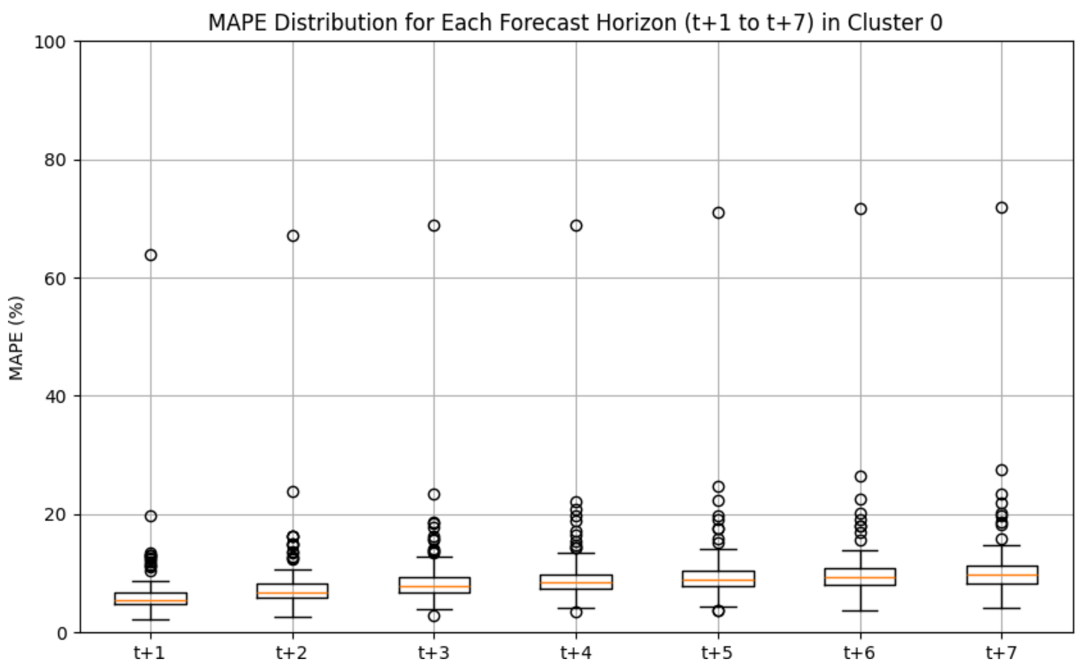
Overall

The performance of the LSTM model varied across the four clusters, highlighting differences in prediction accuracy and consistency. Cluster 0 and Cluster 3 showed lower median MAPE and smaller interquartile ranges, indicating consistent predictions across forecast days. However Cluster 0 and Cluster 3 demonstrated more outliers. Cluster 2 also demonstrated low median MAPE and minimal variability, suggesting strong model accuracy with limited outliers. However, Cluster 1 exhibited higher median MAPE and larger IQR, reflecting greater variability and inconsistent predictions, likely due to more complex or irregular consumption patterns. Overall, the model performed well across clusters.

DeepAR (Old - Fortune Tellers)

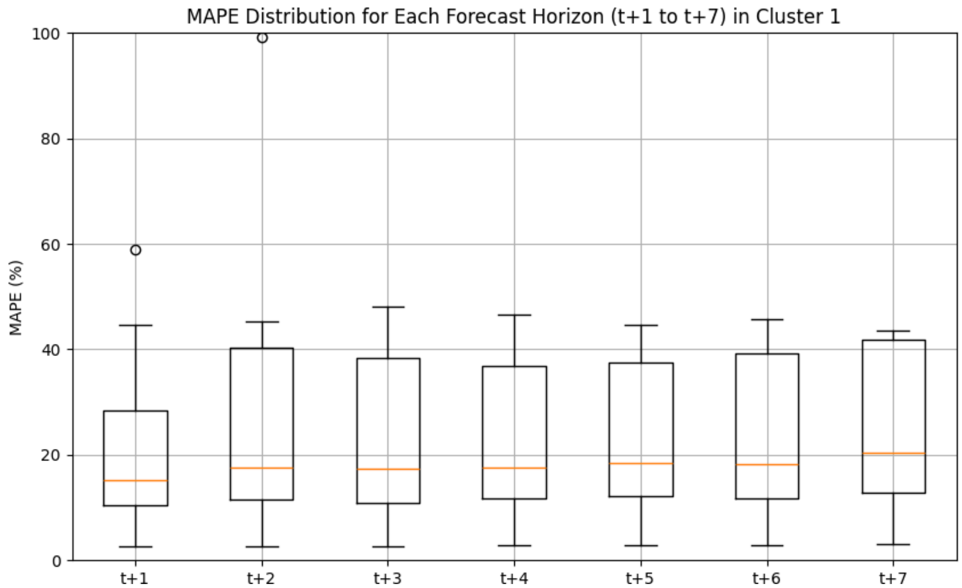
To forecast daily electricity usage, we implemented a DeepAR model trained on data from the first 1041 days and evaluated on the remaining 55 days for each of the four clusters. The input data consists of aggregated daily electricity consumption per user within each cluster. Similar to the LSTM setup, a sliding window approach is used to generate input-output pairs: the model takes the past 30 time steps as input to predict the next 7 time steps. DeepAR, a probabilistic forecasting model based on autoregressive recurrent neural networks, is particularly well-suited for modeling multiple related time series with shared temporal dynamics. The model is trained to output a full predictive distribution, enabling uncertainty estimation for each forecast. During inference, predictions are generated for each user and then reshaped to match the evaluation horizon. Model performance is assessed using mean absolute percentage error (MAPE) across forecast horizons. By capturing both the sequential nature of electricity usage and the variability across users within a cluster, DeepAR offers a flexible and robust framework for multi-step forecasting.

Cluster 0



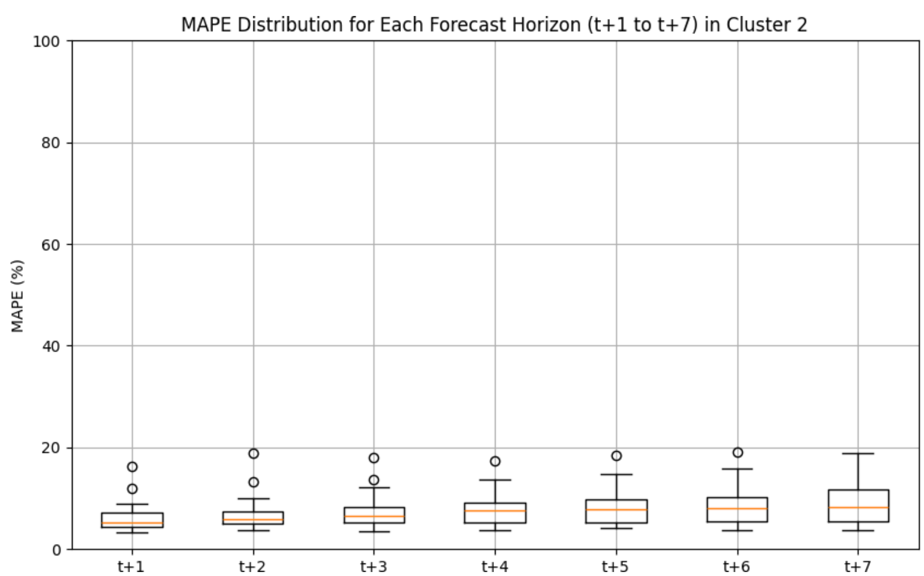
The box plot shows the Mean Absolute Percentage Error (MAPE) per forecast day for Cluster 0, calculated on a per-user basis. The distribution mainly ranges from 2% to 20%. The median MAPE, indicated by the orange line, remains relatively stable but shows a slight upward trend from t+1 to t+7, reflecting the model's decreasing accuracy over longer forecast horizons. The interquartile range is narrow and stable from t+1 to t+7, suggesting low variability in prediction errors across users. However, the presence of multiple high outliers indicates that the model struggles with certain user-level predictions, possibly due to more customers in this cluster. The overall MAPE across all users and all days is 0.067, consistent with the distribution shown in the plot. Overall, the model maintains stable prediction accuracy over the forecast period.

Cluster 1



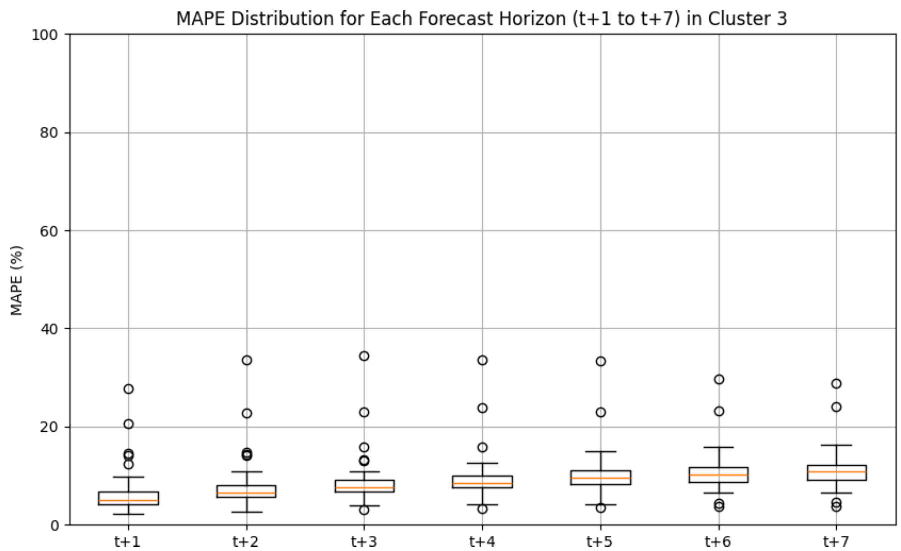
The box plot shows the Mean Absolute Percentage Error (MAPE) per forecast day for Cluster 1, calculated on a per-user basis. The distribution mainly ranges from 5% to 45%. The median MAPE, indicated by the orange line, shows a gradual upward trend from t+1 to t+7, reflecting decreasing model accuracy as the forecast horizon extends. The interquartile range is similar for each day, suggesting that the middle 50% of prediction errors are consistent. The MAPE interquartile range is relatively wide compared to other clusters, indicating greater variability in user-level prediction accuracy within this group. A few high outliers appear, particularly at t+2 and t+1, suggesting the model struggles with some users. The overall MAPE across all users and all days is 0.15, consistent with the general distribution shown.

Cluster 2



The box plot shows the Mean Absolute Percentage Error (MAPE) per forecast day for Cluster 2, calculated on a per-user basis. The distribution mainly ranges from 3% to 12%. The median MAPE, indicated by the orange line, remains consistently low across all forecast days, reflecting strong and stable model performance. But it also shows a slight upward trend from t+1 to t+7, reflecting the model's decreasing accuracy over longer forecast horizons. The interquartile range is slightly narrower at t+1 and gradually expands over time, suggesting increasing uncertainty in longer-term forecasts. Only a few mild outliers appear, indicating that most users in this cluster follow regular and predictable usage patterns. The overall MAPE across all users and all days is 0.064, aligning with the tight distribution shown.

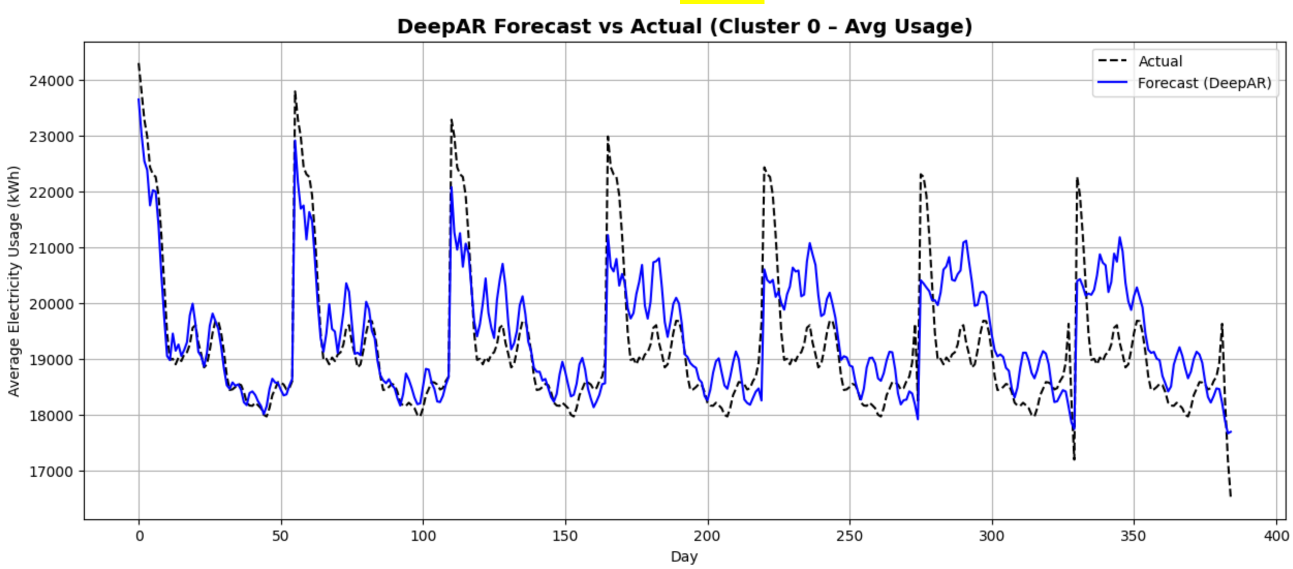
Cluster 3



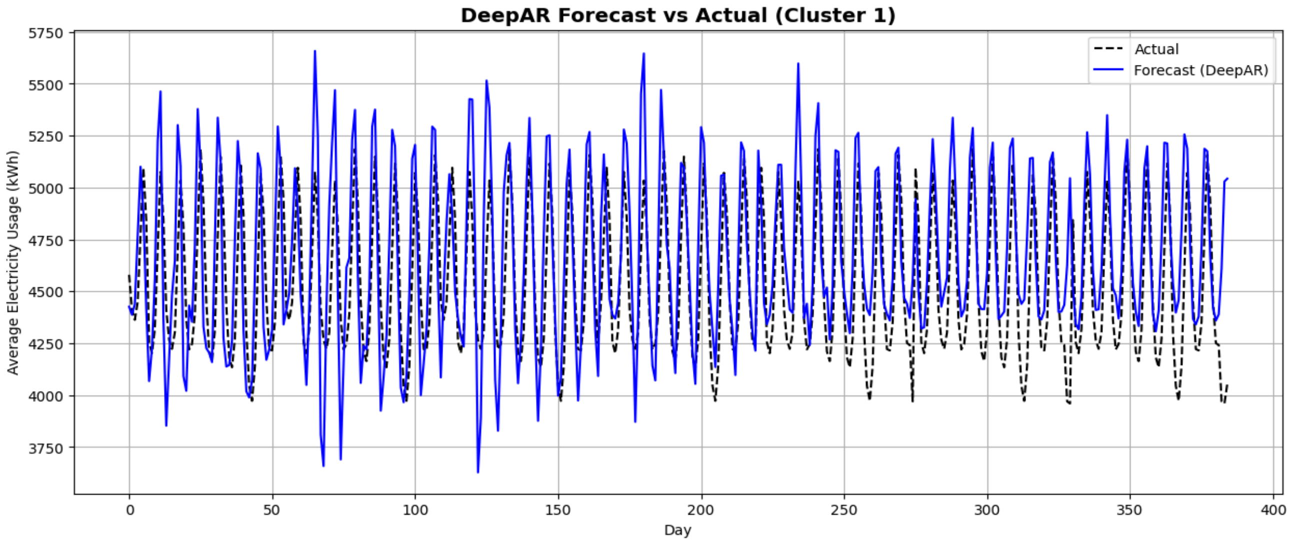
The box plot shows the Mean Absolute Percentage Error (MAPE) per forecast day for Cluster 3, calculated on a per-user basis. The distribution mainly ranges from 5% to 15%. The median MAPE, indicated by the orange line, increases slightly from t+1 to t+7, reflecting a modest decline in model accuracy over longer horizons. The interquartile range is relatively stable but slightly widens from t+3 onward, suggesting a gradual increase in forecast uncertainty. Several moderate outliers are present throughout, indicating some user-level variation. The overall MAPE across all users and all days is 0.193, consistent with the observed distribution.

DeepAR (New - Data Prophets)

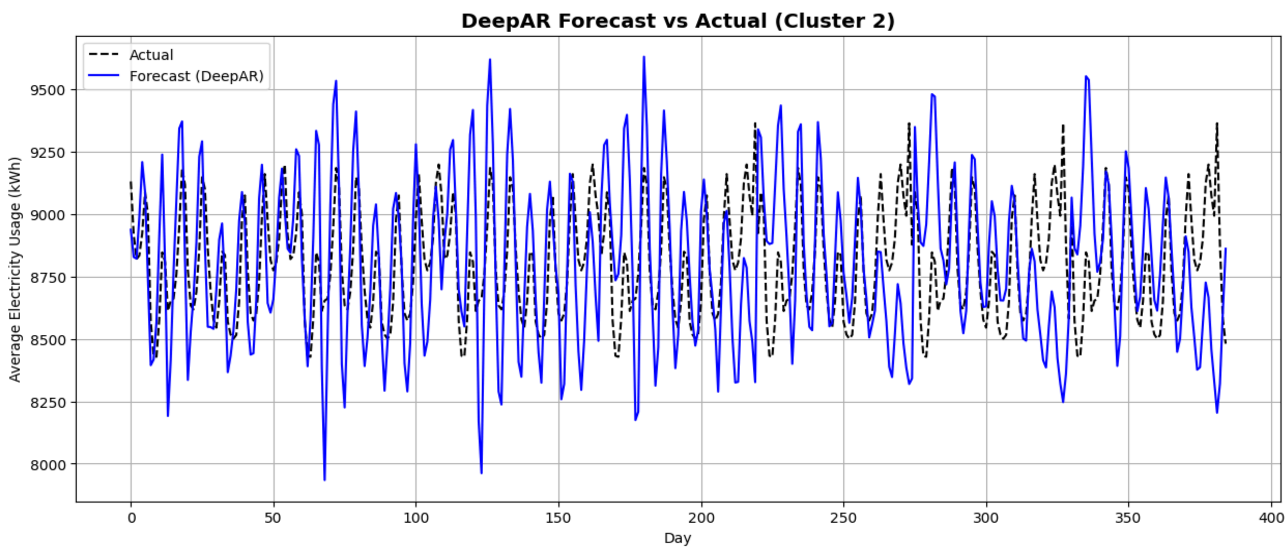
Cluster 0: 3.07%



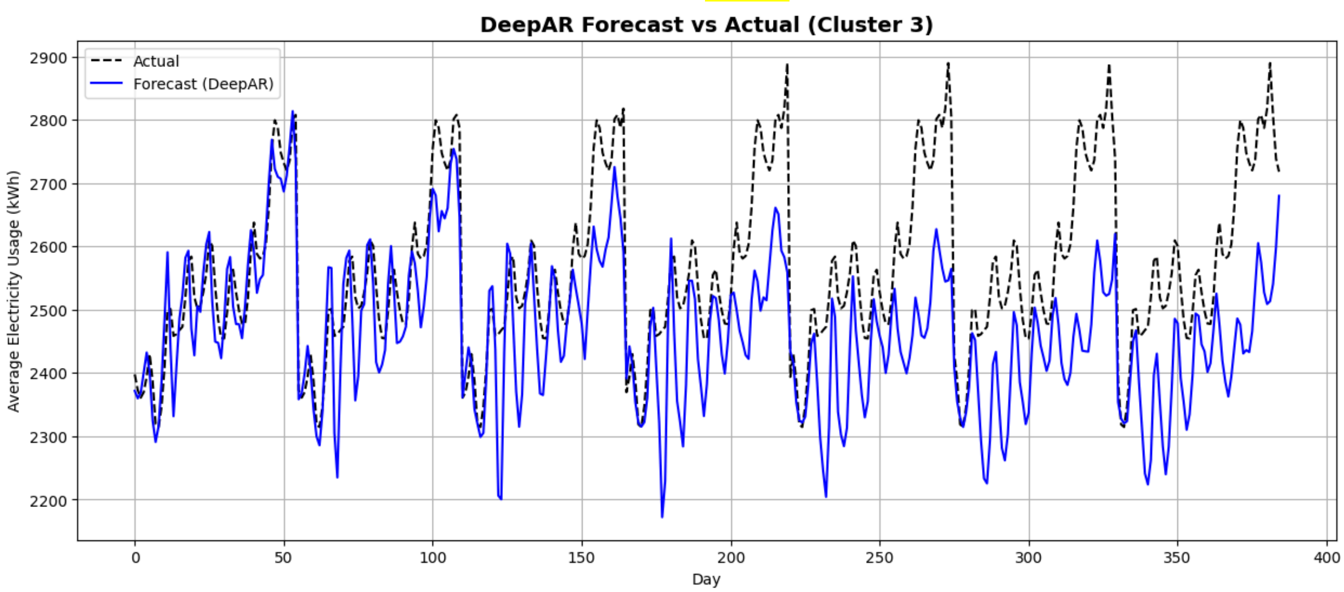
Cluster 1: 5.65%

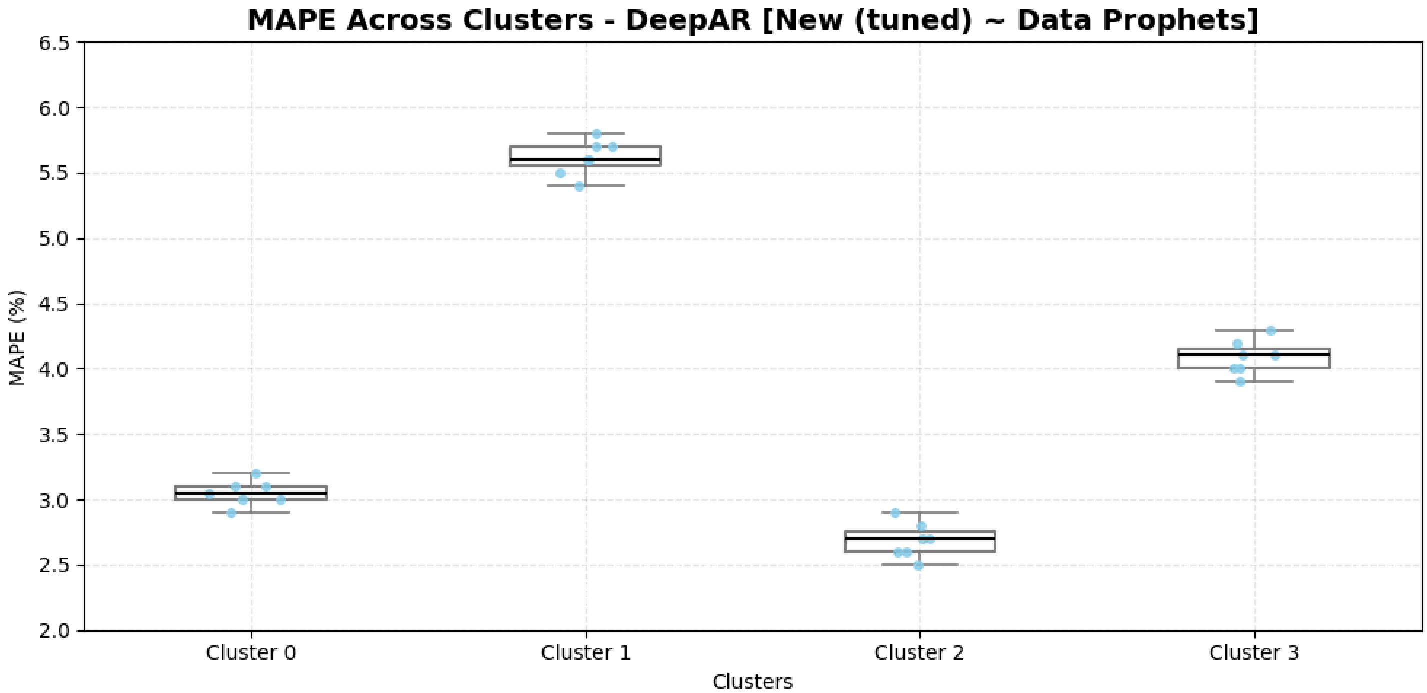


Cluster 2: 2.70%



Cluster 3: 4.14%





Overall

The performance of the DeepAR model varied across the four clusters, revealing differences in accuracy and forecast stability. Cluster 2 showed the best performance, with the lowest median MAPE and narrowest interquartile ranges across all forecast days, indicating high consistency and minimal user-level variation. Cluster 3 also demonstrated low median MAPE and relatively tight distributions, though with slightly more outliers than Cluster 2. Cluster 0 exhibited moderate error levels with a stable distribution, but showed a higher number of outliers, suggesting occasional challenges in specific user forecasts. In contrast, Cluster 1 showed the highest overall MAPE and the widest interquartile ranges, reflecting greater prediction variability likely driven by more irregular consumption patterns. Overall, the DeepAR model delivered reliable results, with particularly strong performance in Clusters 2 and 3.

Model Summary

Model	Cluster 0 Overall MAPE	Cluster 1 Overall MAPE	Cluster 2 Overall MAPE	Cluster 3 Overall MAPE
LSTM [Old ~ Fortune Tellers]	18.56%	61.65%	76.51%	166.04%
BiLSTM **[New ~ Data Prophets]**	**2.38%**	**6.61%**	**3.35%**	**3.24%**
DeepAR [Old ~ Fortune Tellers]	6.70%	15.00%	6.4%	9.03%
DeepAR **[New (tuned) ~ Data Prophets]**	**3.07%**	**5.65%**	**2.70%**	**4.14%**
Prophet **[New ~ Data Prophets]**	**3.57%**	**4.58%**	**4.31%**	**3.90%**
Amazon Chronos **[New ~ Data prophets]**	**6.47%**	**7.90%**	**2.41%**	**6.66%**

In conclusion, the DeepAR model demonstrated superior predictive performance across all four clusters compared to the LSTM model. DeepAR achieved significantly lower MAPE values, indicating more accurate forecasts and better adaptability to the underlying patterns in the data. Cluster 0 and Cluster 2 showed particularly strong performance with DeepAR, achieving MAPE values of 0.067 and 0.064 respectively, reflecting stable and consistent predictions. In contrast, the LSTM model exhibited higher error rates, especially in Cluster 3, where the MAPE reached 1.6604, suggesting that LSTM struggled to capture complex patterns in certain clusters. Overall, DeepAR's consistent performance highlights its strength in modeling time-series data, making it a more reliable choice for electricity consumption forecasting.

Old Team Information ~ Fortune Tellers

Mengxi Liu (ml5189) Jie Hu (jh4741)

Han Qiang (hq2218) ChengHsin Chang (cc5211)

New Team Information ~ Data Prophets

Prakhar Dungarwal (pd2782)   Somit Jain (sj3396)

Shashwat Kumar (sk5520)   Devdatt Golwala (drg2172)