Predicting Electricity Usage: Deliverable 2

Zhiqi Ma
Ruoheng Du
Xiaojiang Wu
Yijian Liu

Agenda

Introduction

Data

Architecture

Models

Results I

Results II

Results III

Summary

Further Improvement




Introduction

- This project focuses on forecasting electricity usage using time series modeling techniques. Given the complex and seasonal nature of electricity consumption, we analyze historical weekly usage data to identify trends and seasonal patterns.

Problem

Electricity usage with complex temporal patterns creates challenges for accurate forecasting.

Objective

Forecast weekly overall electricity consumption with seasonal patterns and structural shifts.

From:

Raw data of electricity consumption of 370 clients

To:

Time series models and weekly overall consumption prediction results

Value Creation

Improved Forecast Accuracy
Enhances electricity consumption forecasting, enabling better resource planning and distribution.

Optimized Energy Utilization
Supports optimized power distribution, reducing energy waste and improving cost efficiency.

Data-Driven Decision Making
Supports informed strategies for infrastructure planning, energy trading, and policy making.

Timeline

Data Preprocessing
Gather raw electricity consumption data from 370 points/clients. Clean, aggregate, and transform data for analysis.

Data Analysis
Identify trends, seasonality, and anomalies. Visualize key patterns using ACF, PACF, PSD, and decomposition.

Model Development
Train different time series models and test different hyperparameters.

Performance Evaluation
Split data into training, validation, and testing sets. Compute MAPE and analyze error distributions.

Final Insights
Summarize key findings and model performance. Provide recommendations for improving forecasting accuracy.




Data

Data

Source: Electricity Load Diagrams 20112014

Type: Time-Series

Instances: 370 clients

Features: 140,256 time-stamped records at 15-minute intervals from 2011 to 2014

Missing Values: None

Feature Type: Real values (electricity consumption in kW)

	MT_001	MT_002	MT_003	MT_004	MT_005	MT_006	MT_007	MT_008	MT_009	MT_010	...	MT_361	MT_362	MT_363	MT_364	MT_365	MT_366	MT_367	MT_368	MT_369	MT_370
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	0.0	0.0	0.0	0.0	0.0
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	0.0	0.0	0.0	0.0	0.0
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	0.0	0.0	0.0	0.0	0.0
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	0.0	0.0	0.0	0.0	0.0
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	0.0	0.0	0.0	0.0	0.0

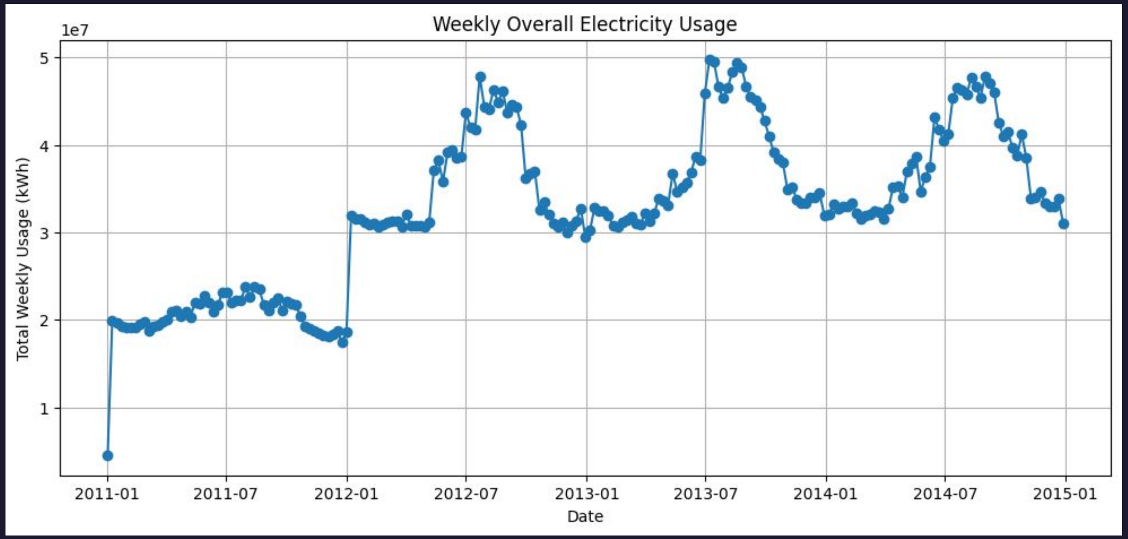
Data Preprocessing

- **Unit Conversion**: Dividing original record in kW at 15-minute intervals by 4 to obtain usage in kWh.
- **Data Aggregation**: Resampling data to a weekly level for model efficiency and long-term trend analysis.
- **Long Table Format**: Converting data into a long table to apply time series models.
- **Overall Electricity Usage**: Calculating the total weekly consumption across all accounts to analyze electricity consumption at a macro level.

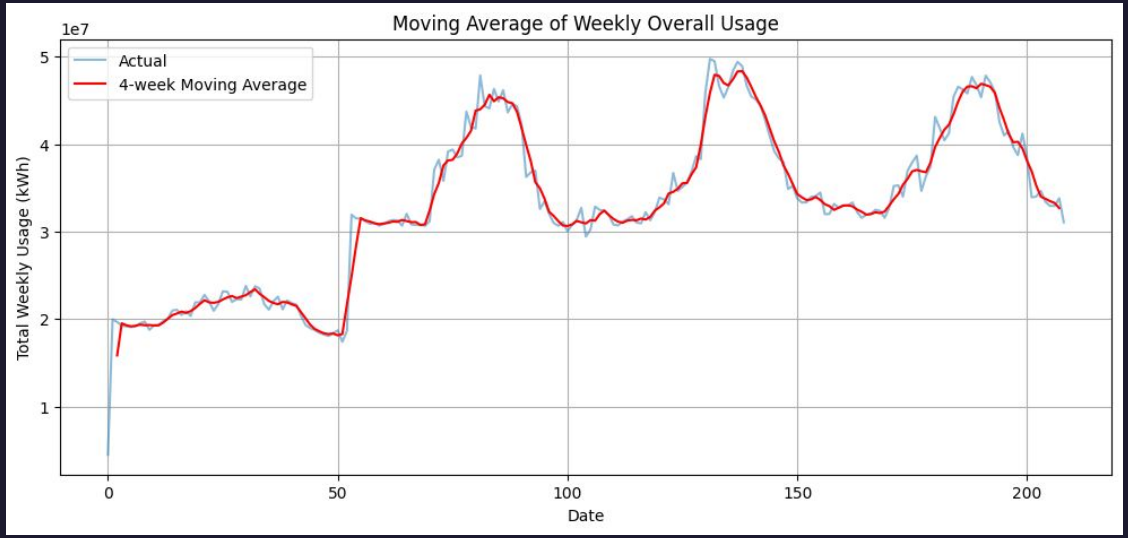
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209 entries, 0 to 208
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Datetime  209 non-null    datetime64[ns]
 1   Usage     209 non-null    float64
dtypes: datetime64[ns](1), float64(1)
```

Exploratory Data Analysis

Long-Term Consumption Patterns



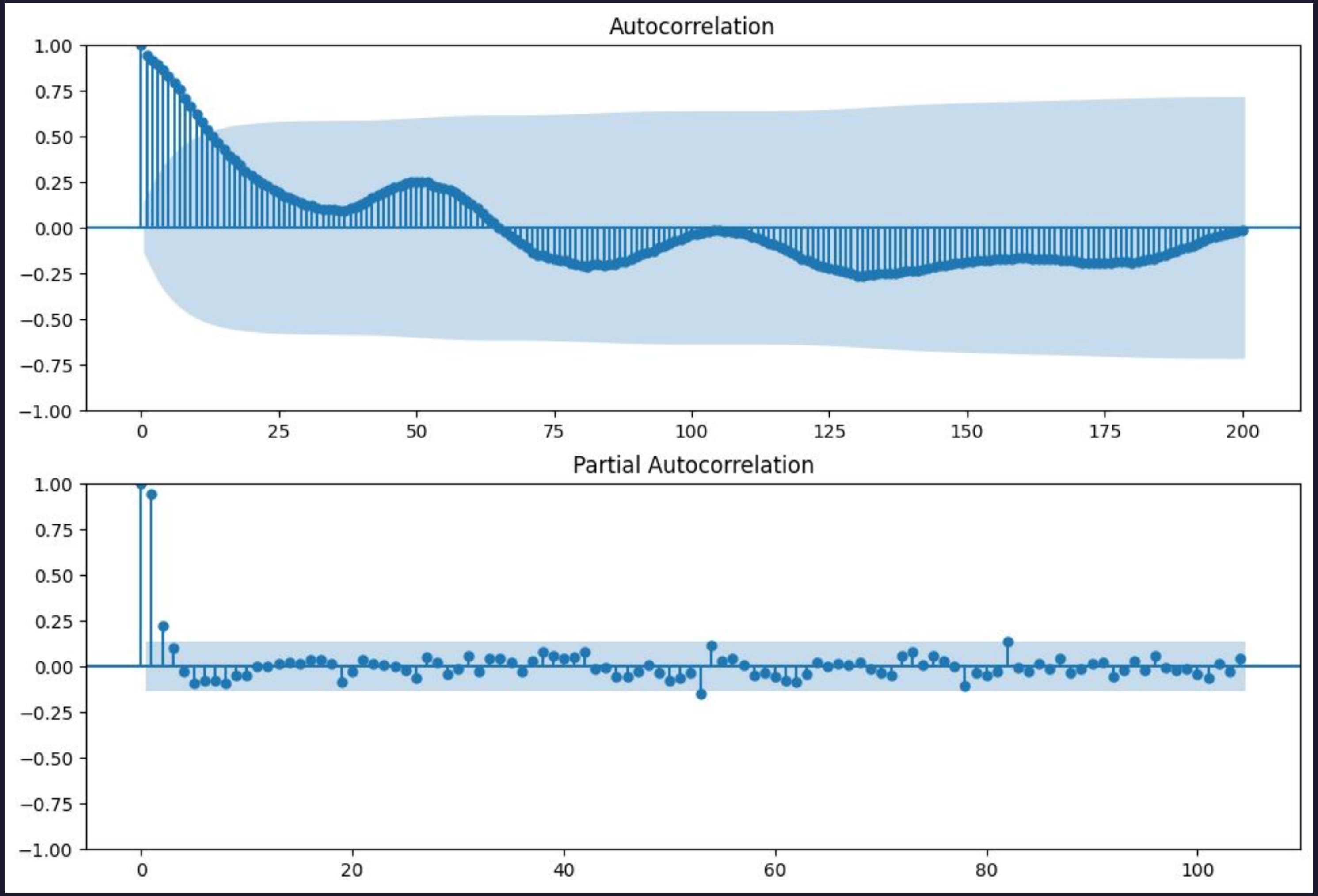
- Weekly overall electricity usage from 2011 to 2015 shows seasonal fluctuations and noticeable trends.



- The 4-week moving average smooths the raw data, revealing long-term trends and seasonal patterns while reducing short-term fluctuations.

Exploratory Data Analysis

Periodic Patterns in Electricity Usage - ACF/PACF

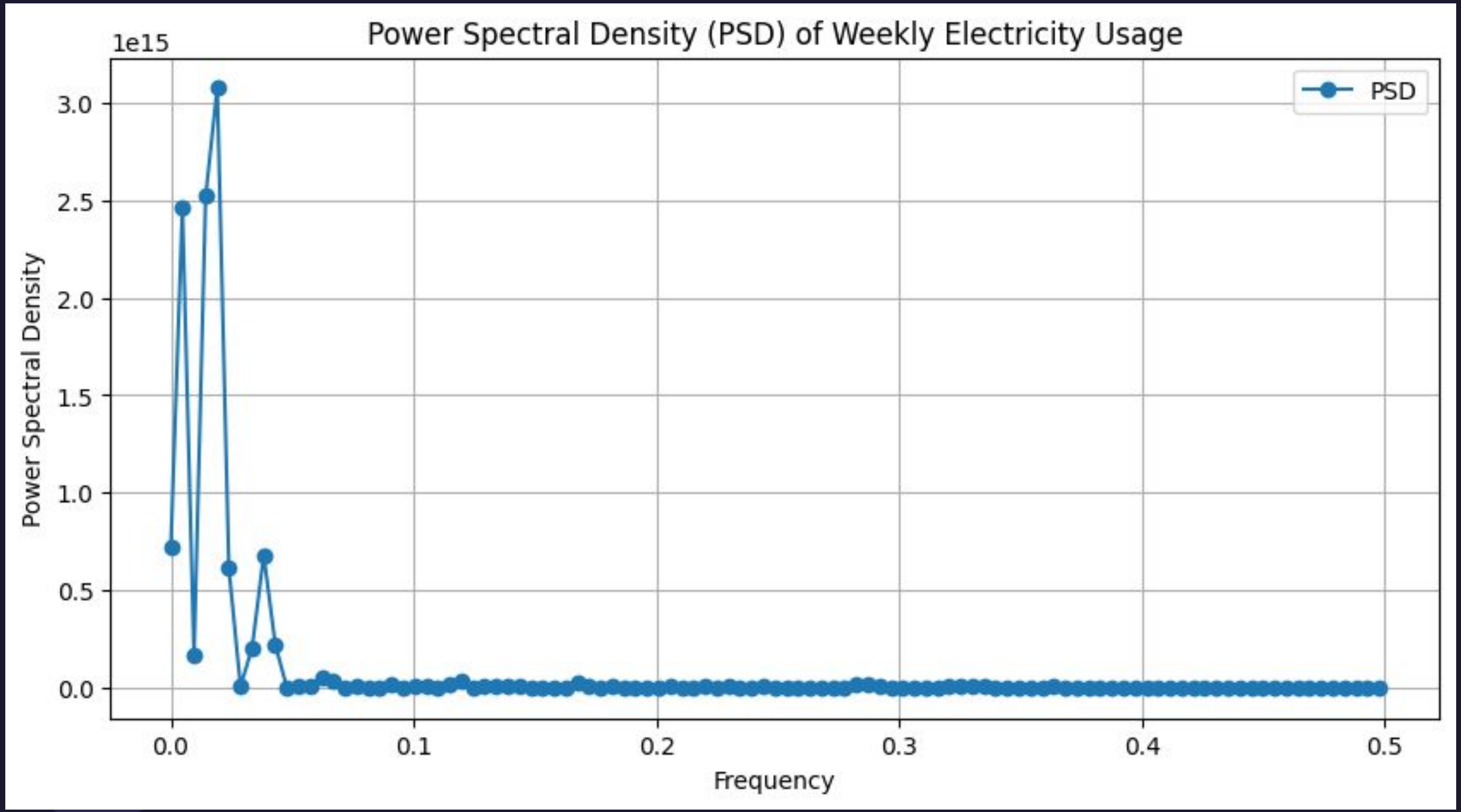


- The autocorrelation plot indicates strong periodicity with significant fluctuations over time, suggesting long-term dependencies.

- The partial autocorrelation plot shows that recent past values have a strong influence, but correlations weaken over longer lags.

Exploratory Data Analysis

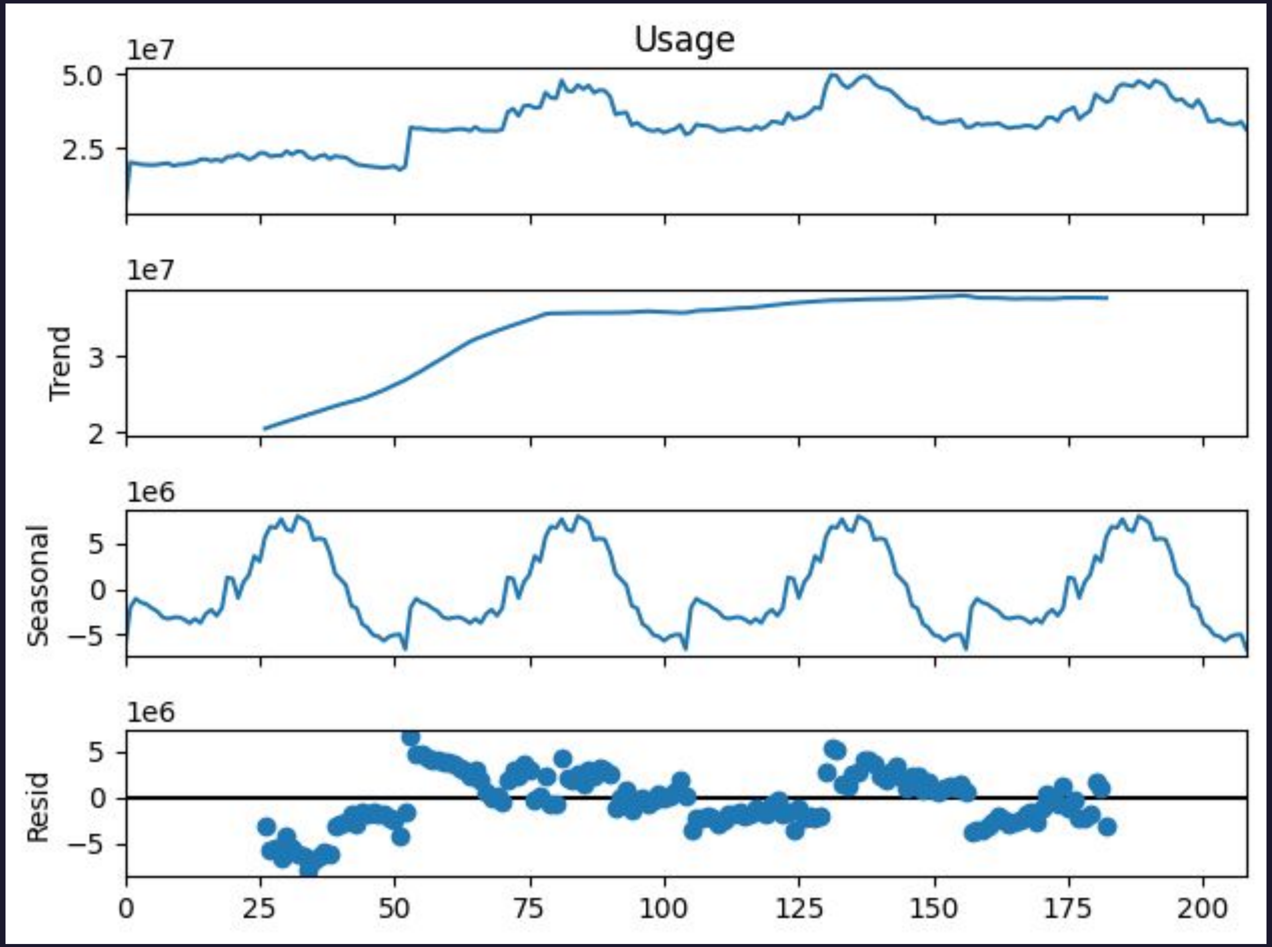
Periodic Patterns in Electricity Usage - PSD



- High power spectral density concentrated at low frequencies, indicating the presence of seasonal cycles, possibly yearly patterns.

Exploratory Data Analysis

Time Series Decomposition

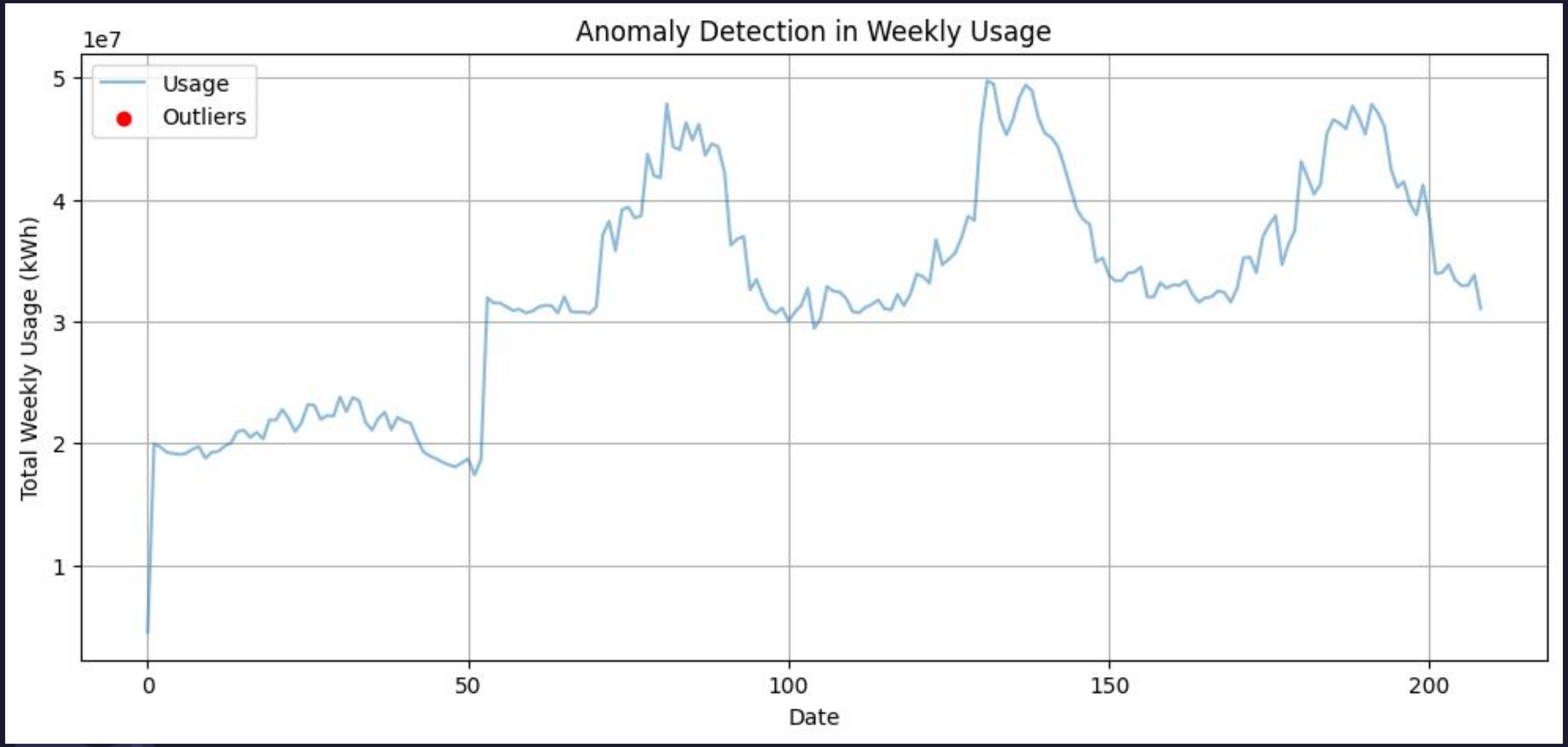


- The time series decomposition separates the data into trend, seasonal, and residual components.

- The trend shows a general upward movement.
- The seasonal component highlights recurring patterns.
- The residuals capture irregular variations and potential anomalies.

Exploratory Data Analysis

Anomaly Detection

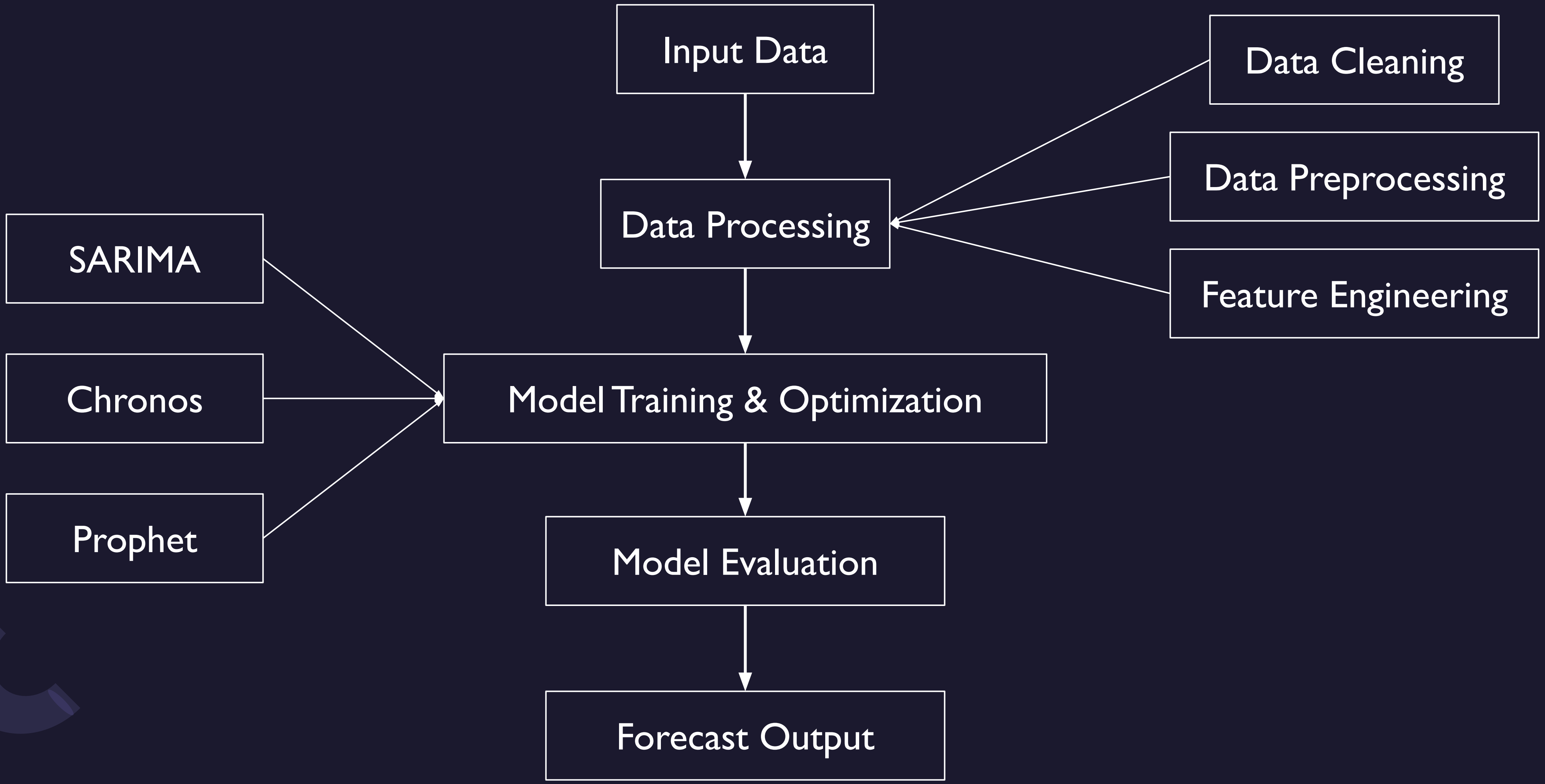


- No significant anomalies were detected, suggesting relatively stable electricity usage patterns.




Architecture

Architecture






Models

Models

SARIMA (p, d, q) x (P, D, Q)^s

SARIMA extends the ARIMA model by incorporating seasonal components, making it suitable for time series data with periodic patterns.

p	AutoRegressive Order - AR	Number of past observations used for regression
d	Differencing Order	Number of times differencing is applied to make the series stationary
q	Moving Average Order - MA	Number of past error terms used for smoothing
P	Seasonal AR Order	Number of past seasonal observations used
D	Seasonal Differencing Order	Number of times seasonal differencing is applied
Q	Seasonal MA Order	Number of past seasonal error terms used
s	Seasonal Periodicity	Length of the seasonal cycle

Models

Amazon Chronos

Chronos is a family of pretrained time series forecasting (foundation) models based on T5 architecture, trained on 84B data.

To use Chronos model, **AutoGluon-TimeSeries (AG-TS)** is the best choice, which provides a robust and easy way to use Chronos through the familiar TimeSeriesPredictor API. Roughly speaking, we can:

- Use Chronos in **zero-shot** mode to make forecasts without any dataset-specific training
  - only prediction length, train data, and model version needed as parameters
- **Fine-tune** Chronos models on custom data to improve the accuracy
  - can use default fine tuning setting, or manually set hyperparameters, including learning rate, batch size, fine tune steps, etc.; also allowed to control fine tune time limit
  - can choose to fine-tune using CPU or GPU
- **Handle covariates & static features** by combining Chronos with a tabular regression model
  - can add covariate regressors to be combined with univariate Chronos model to predict target

Evaluation Metric

Mean Absolute Percentage Error (MAPE)

MAPE measures the average percentage error between actual and predicted values, making it useful for evaluating the accuracy of a forecasting model.

Formula

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \times 100\%$$

where:

- Y_t = actual value at time t
- $\hat{Y}_t$ = predicted value at time t
- n = total number of observations




Results I

Results I

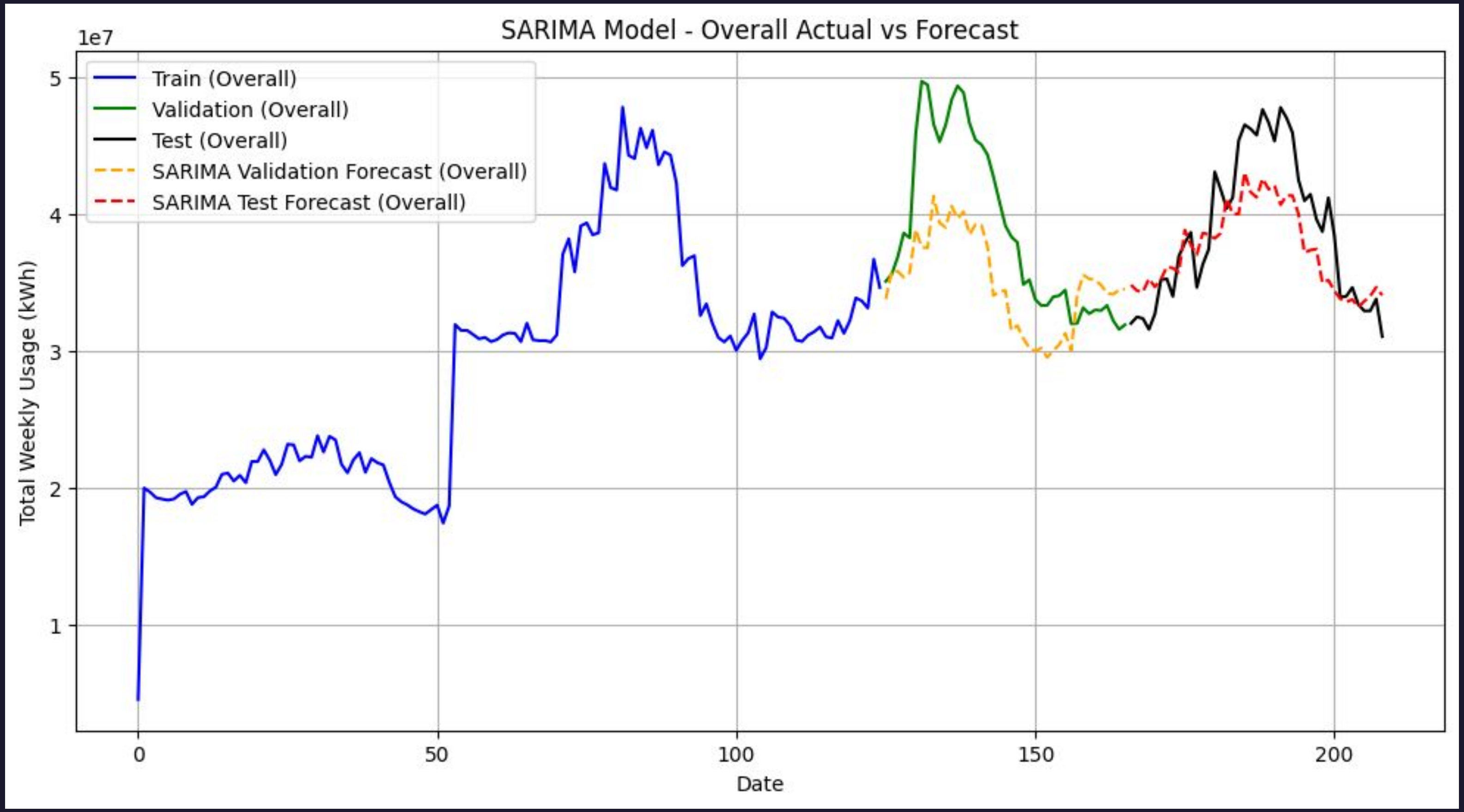
SARIMA (1, 1, 1) x (1, 1, 1) ^{52}

Parameters chosen based on EDA plot analysis.

Train:Validation:Test: 6:2:2

Overall Results:

- Validation MAPE: 11.62%
- Test MAPE: 6.68%



Results I

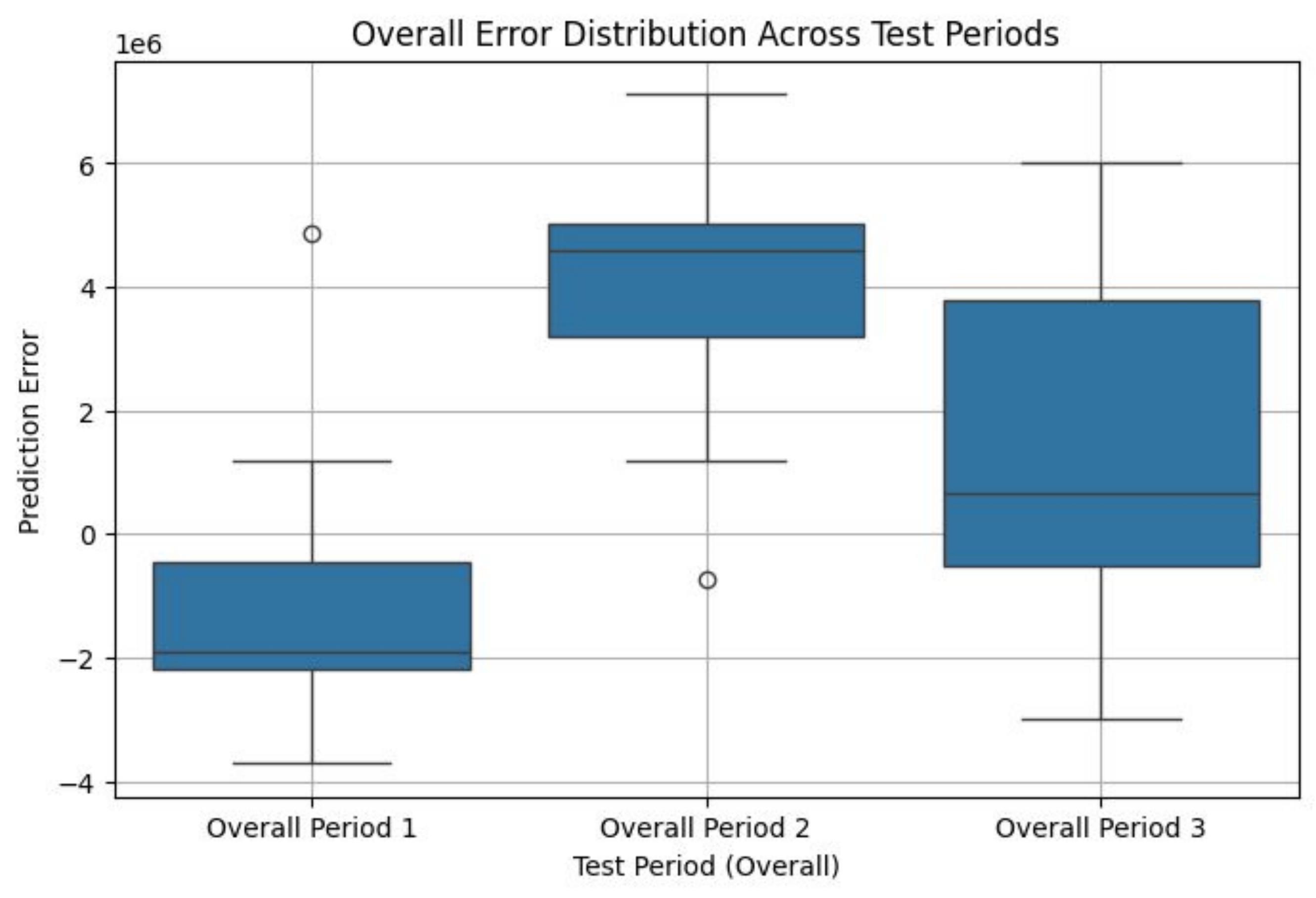
SARIMA (1, 1, 1) x (1, 1, 1) ^{52}

Parameters chosen based on EDA plot analysis.

Train:Validation:Test: 6:2:2

3-Test Period Results:

- Test Period I MAPE: 5.50%
- Test Period II MAPE: 8.76%
- Test Period III MAPE: 5.85%



Results I

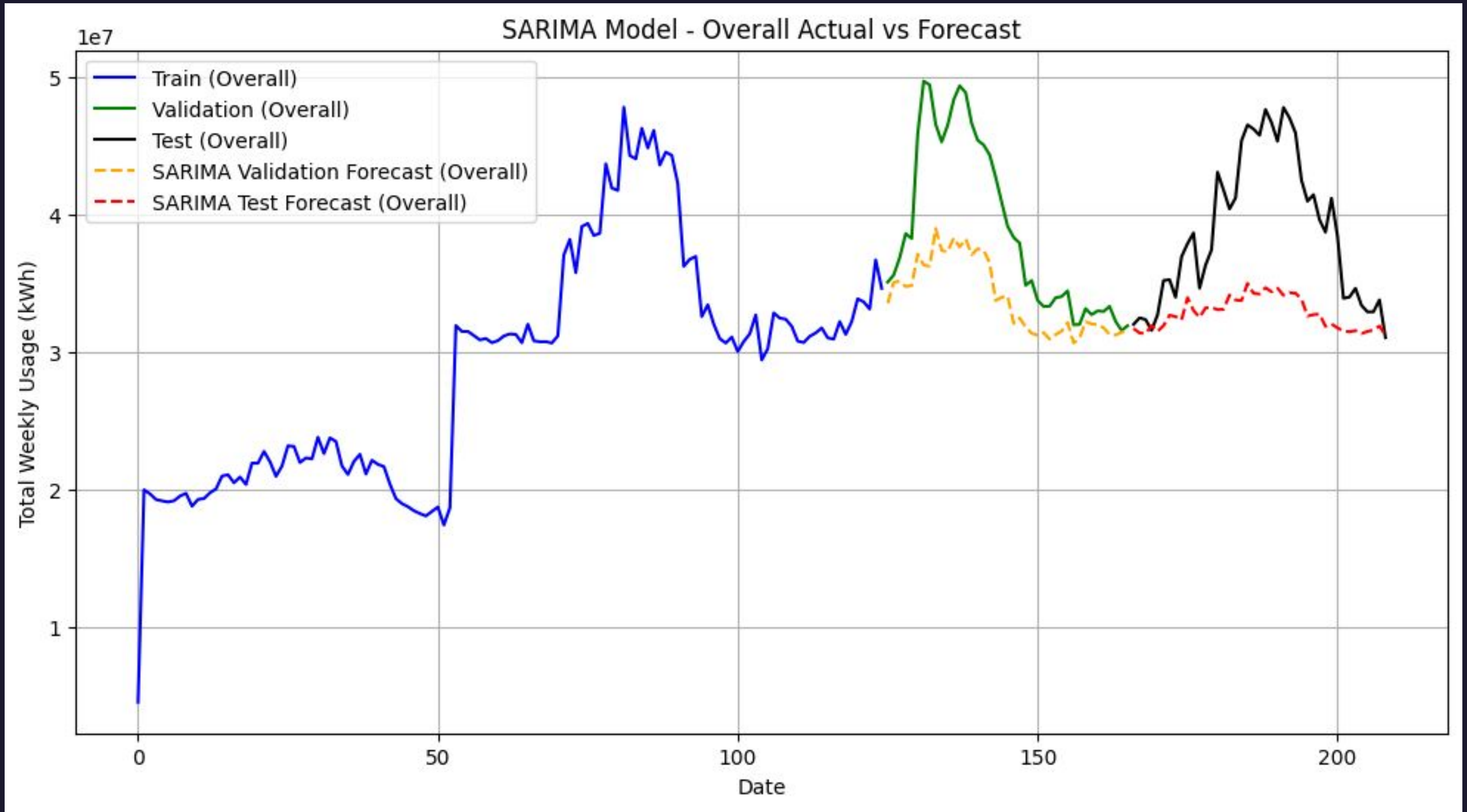
SARIMA (0, 1, 2) x (1, 0, 0) 52

Parameters chosen based on *auto_arima*, which optimizes the model by minimizing information criteria (AIC/BIC).

Train:Validation:Test: 6:2:2

Overall Results:

- Validation MAPE: 11.42%
- Test MAPE: 14.29%



Results I

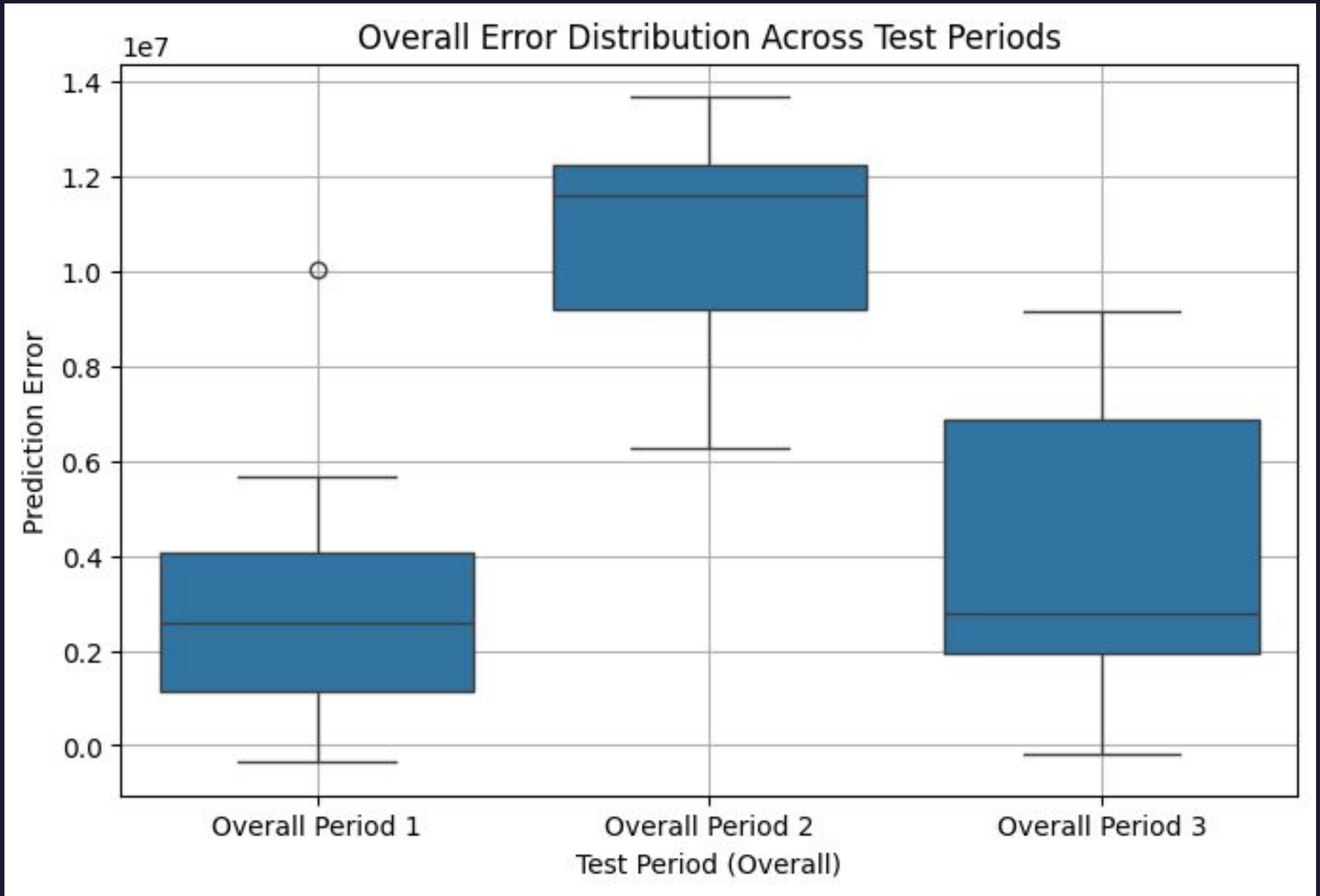
SARIMA (0, 1, 2) x (1, 0, 0) ^{52}

Parameters chosen based on *auto_arima*, which optimizes the model by minimizing information criteria (AIC/BIC).

Train:Validation:Test: 6:2:2

3-Test Period Results:

- Test Period I MAPE: 7.98%
- Test Period II MAPE: 23.89%
- Test Period III MAPE: 11.44%



Results I

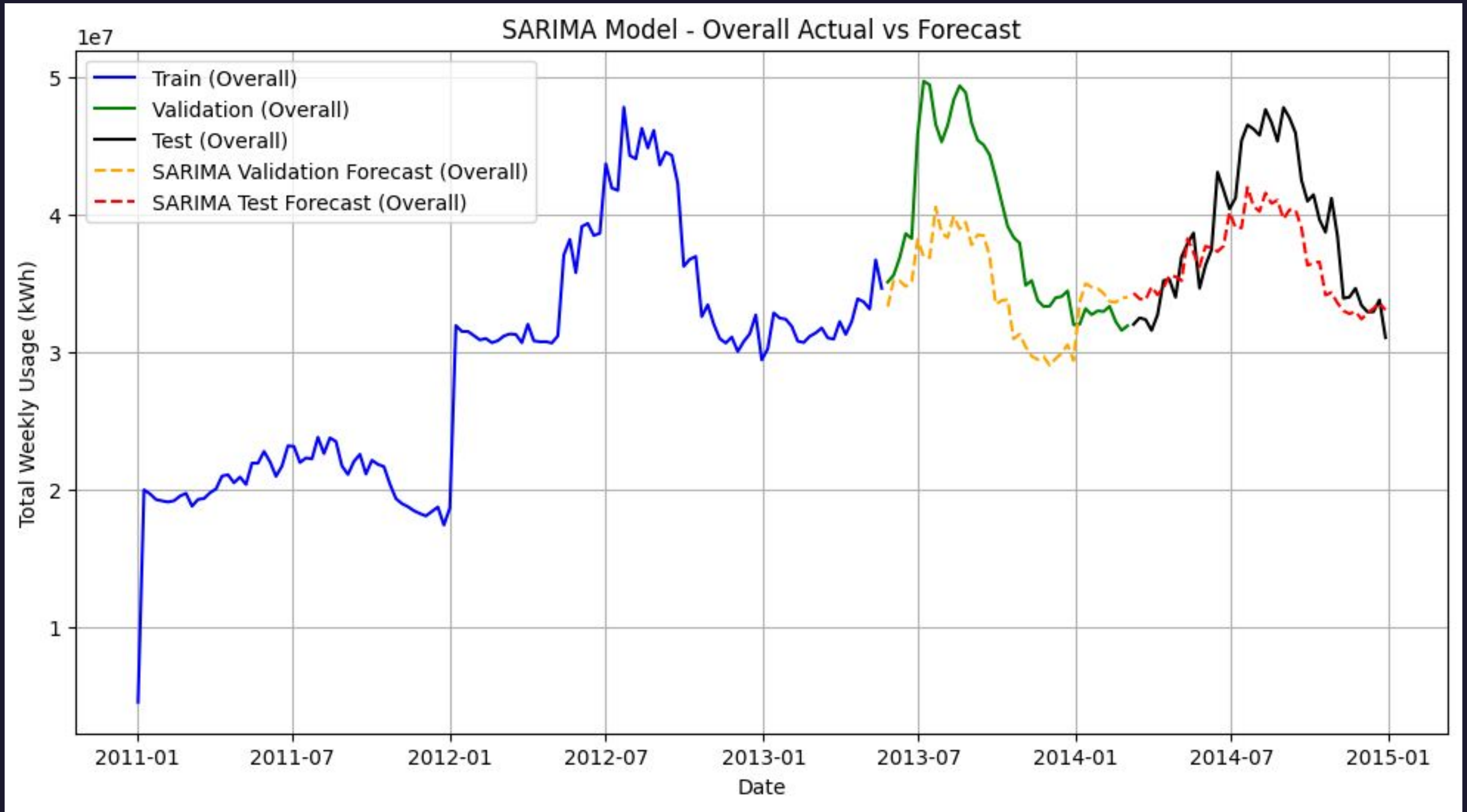
SARIMA (0, 1, 0) x (1, 1, 1) ^{52}

Parameters chosen based on stationarity testing, differencing selection, and grid search.

Train:Validation:Test: 6:2:2

Overall Results:

- Validation MAPE: 12.43%
- Test MAPE: 7.29%



Results I

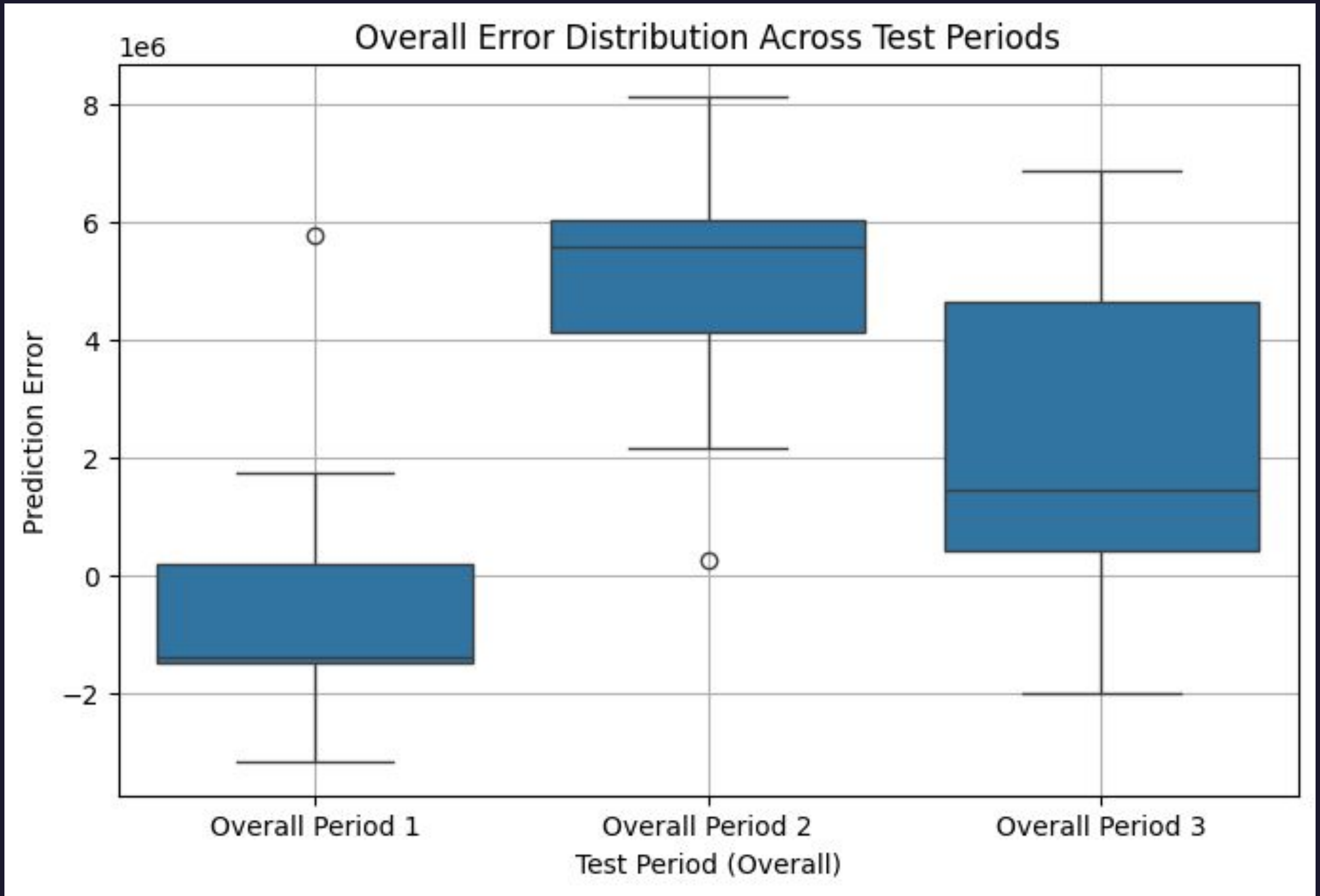
SARIMA (0, 1, 0) x (1, 1, 1) ^{52}

Parameters chosen based on stationarity testing, differencing selection, and grid search.

Train:Validation:Test: 6:2:2

3-Test Period Results:

- Test Period I MAPE: 4.57%
- Test Period II MAPE: 10.71%
- Test Period III MAPE: 6.76%



Results I

SARIMA Summary

SARIMA Models (s=52)	Overall Test MAPE	Test Period I MAPE	Test Period II MAPE	Test Period III MAPE
(1, 1, 1) x (1, 1, 1)	**6.68%**	**5.50%**	**8.76%**	**5.85%**
(0, 1, 2) x (1, 0, 0)	14.29%	7.98%	23.89%	11.44%
(0, 1, 0) x (1, 1, 1)	7.29%	4.57%	10.71%	6.76%

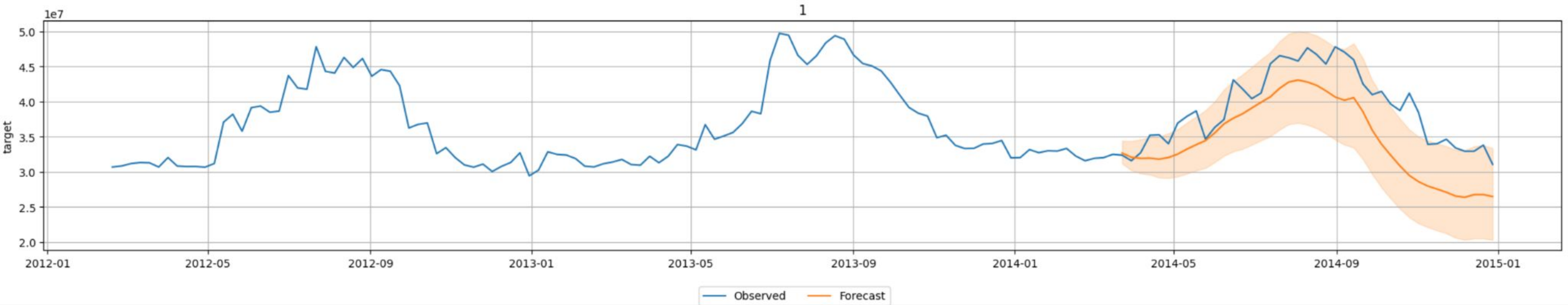
The best-performing model was SARIMA (1,1,1) × (1,1,1,52), achieving an overall test MAPE of 6.68%.

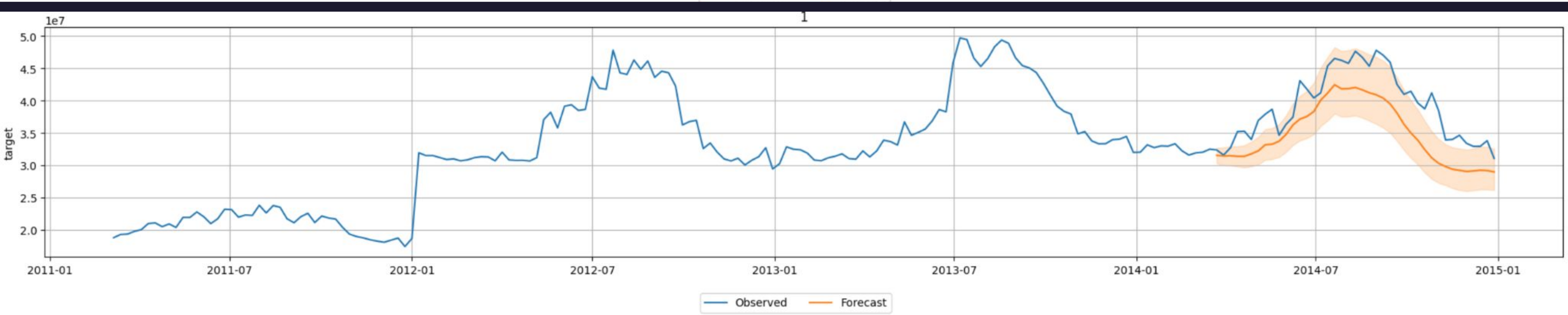



Results II

Results II

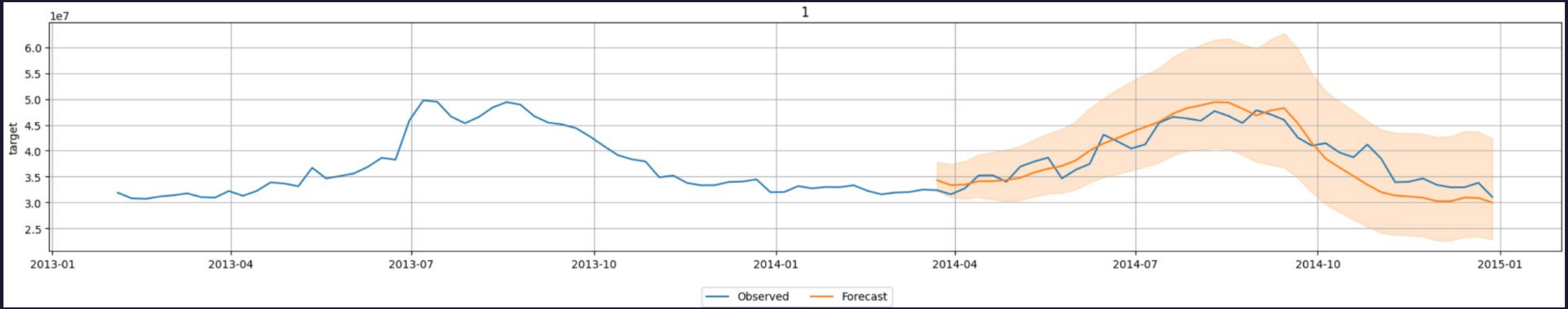
Chronos Zero-shot Forecasting, for bolt_small (above) & bolt_base (below)

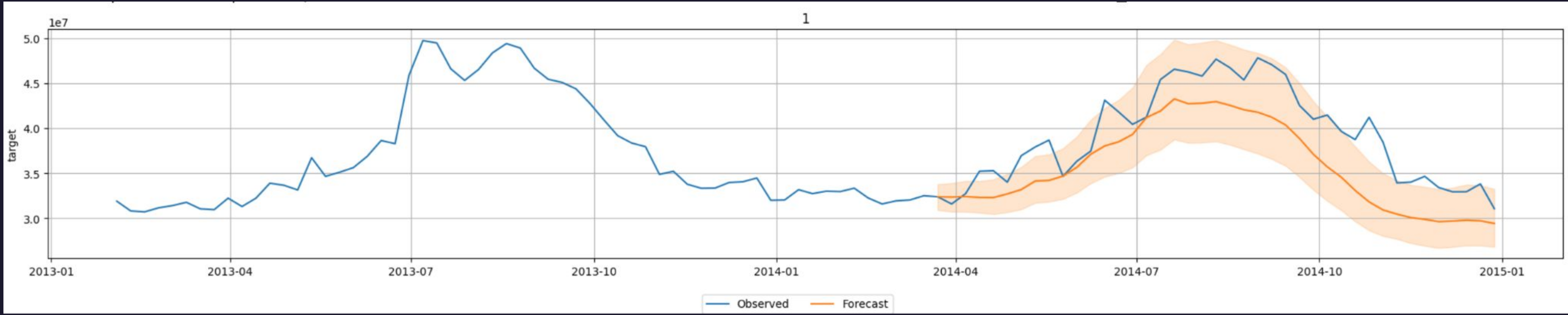




Results II

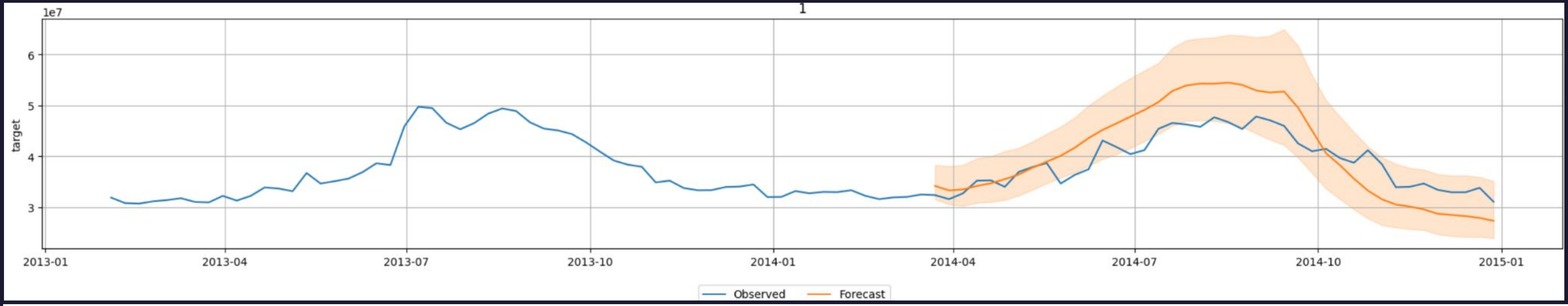
Chronos Default Fine-tune Setting, for bolt_small (above) & bolt_base (below)

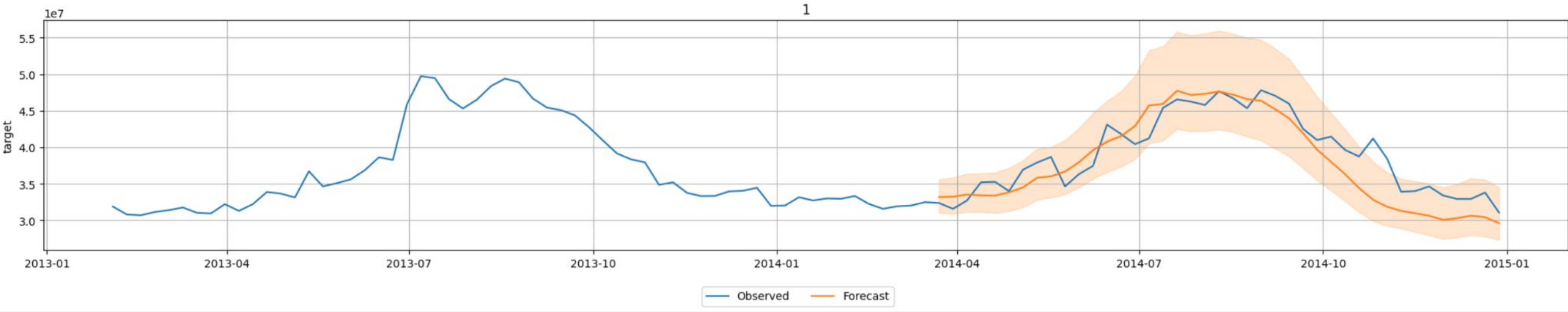




Results II

Chronos Manually Defined Fine-tune Setting,for bolt_small (above) & bolt_base (below)





Results II

Chronos Summary

Model	WQL	MAE	MSE	RMSE	MAPE
Chronos[bolt_small]	-0.085167	-4.630450e+06	-2.847313e+13	-5.336022e+06	-0.118819
Chronos[bolt_base]	-0.083125	-4.242827e+06	-2.231087e+13	-4.723438e+06	-0.106833
ChronosDefaultFineTuned[bolt_small]	-0.053611	-2.292272e+06	-7.352864e+12	-2.711617e+06	-0.059662
ChronosDefaultFineTuned[bolt_base]	-0.065991	-3.488302e+06	-1.642860e+13	-4.053221e+06	-0.087580
ChronosFineTuned[bolt_small]	-0.055450	-2.578221e+06	-9.023399e+12	-3.003897e+06	-0.066755
ChronosFineTuned[bolt_base]	-0.044651	-2.235663e+06	-7.660431e+12	-2.767748e+06	-0.058854

Best result: MAPE 5.89%

Worst result: MAPE 11.88%




Results III

Results III

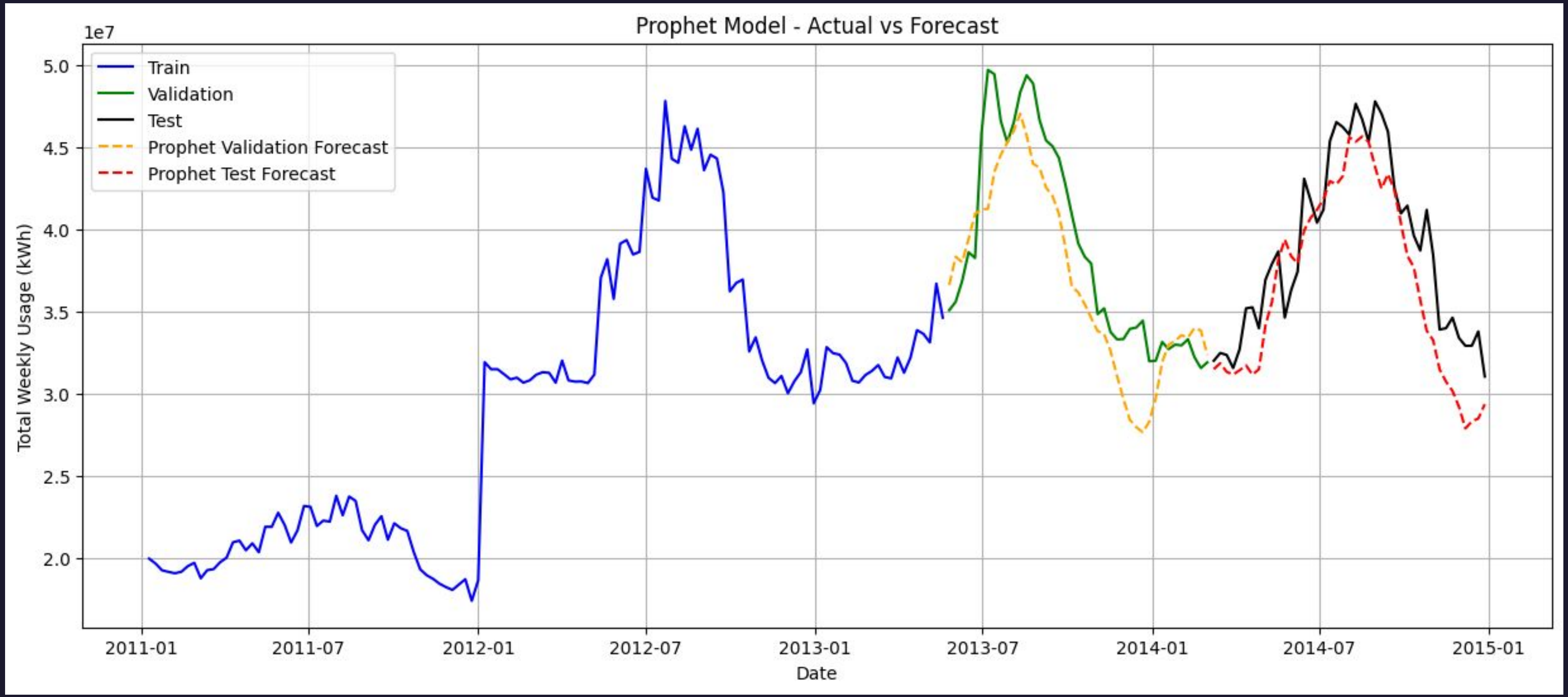
Prophet with Logistic Trends, Customized Weekly Seasonality, Customized Monthly Seasonality, Default Yearly Seasonality

Parameters searched within a predefined range.

Train:Validation:Test: 6:2:2

Overall Results:

- Validation MAPE: 6.88%
- Test MAPE: 6.66%



Results III

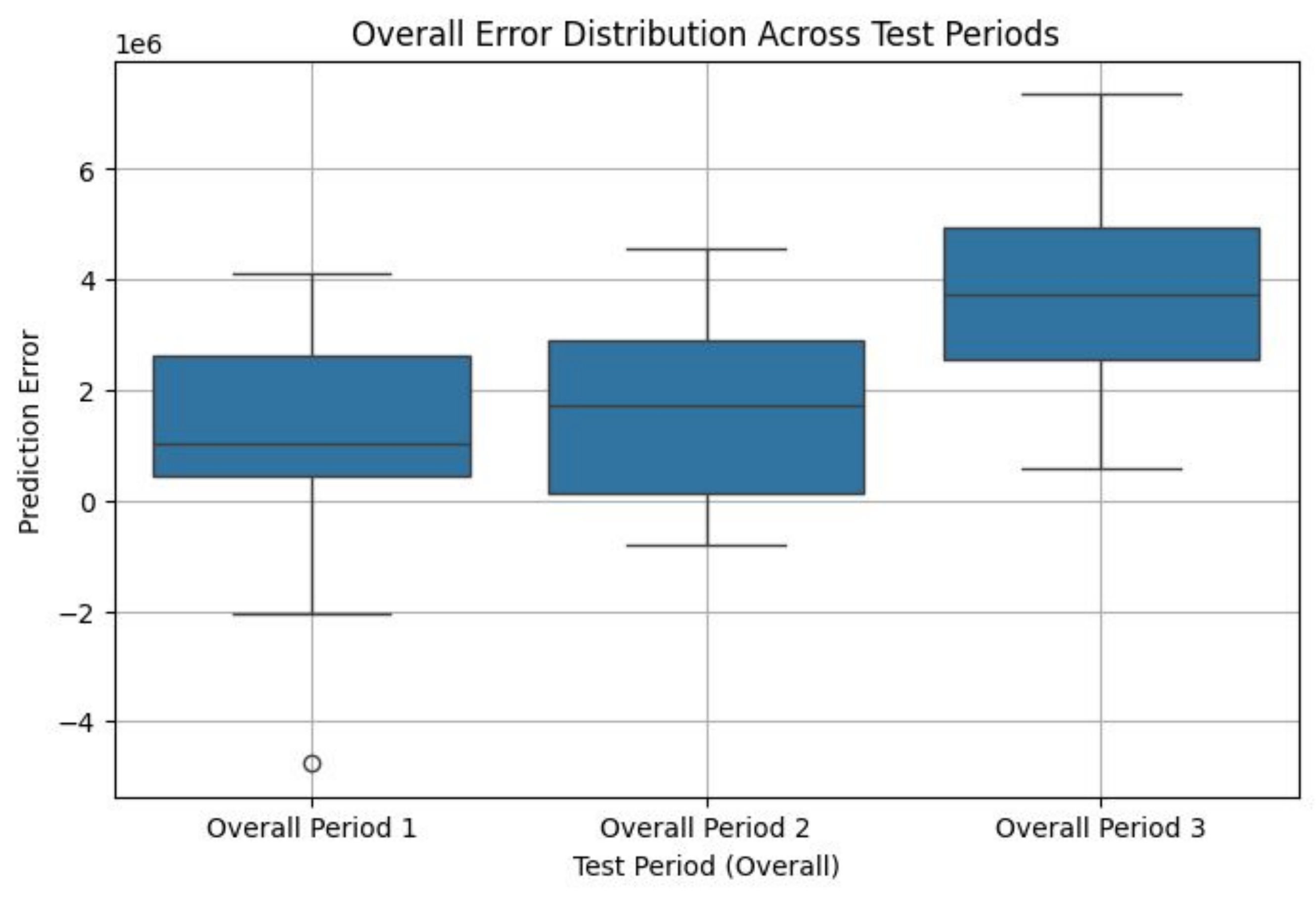
Prophet with Logistic Trends, Customized Weekly Seasonality, Customized Monthly Seasonality, Default Yearly Seasonality

Parameters searched within a predefined range.

Train:Validation:Test: 6:2:2

3-Test Period Results:

- Test Period I MAPE: 5.58%
- Test Period II MAPE: 4.11%
- Test Period III MAPE: 10.38%



Results III

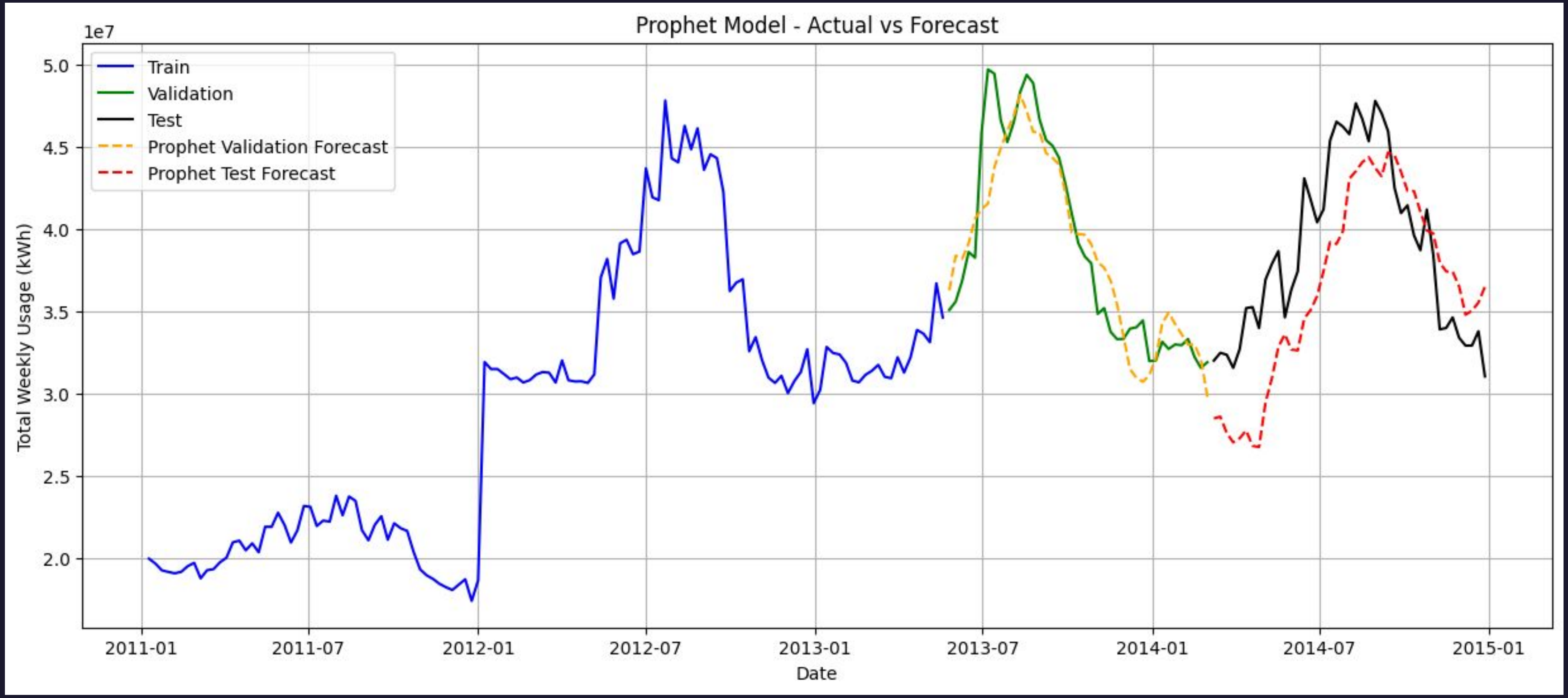
Prophet with Linear Trends, Customized Weekly Seasonality, Customized Monthly Seasonality, Default Yearly Seasonality

Parameters searched within a predefined range.

Train:Validation:Test: 6:2:2

Overall Results:

- Validation MAPE: 4.51%
- Test MAPE: 10.75%



Results III

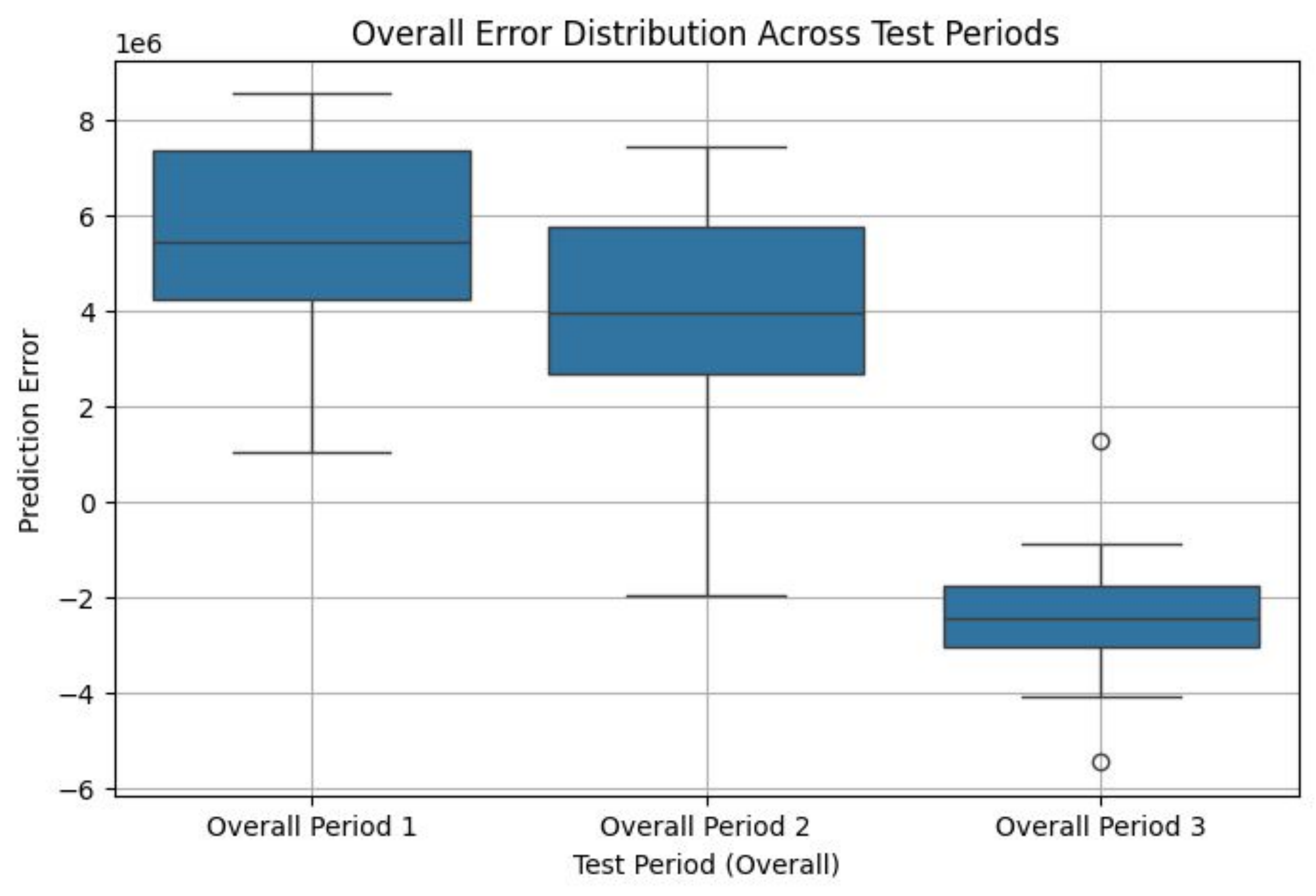
Prophet with Linear Trends, Customized Weekly Seasonality, Customized Monthly Seasonality, Default Yearly Seasonality

Parameters searched within a predefined range.

Train:Validation:Test: 6:2:2

3-Test Period Results:

- Test Period I MAPE: 15.61%
- Test Period II MAPE: 9.00%
- Test Period III MAPE: 7.28%



Results III

Prophet Summary

Prophet Model	Overall Test MAPE	Test Period I MAPE	Test Period II MAPE	Test Period III MAPE
Model 1 (Linear)	10.75%	15.61%	9.00%	7.28%
Model 2 (Logistic)	**6.66%**	**5.58%**	**4.11%**	**10.38%**

Best result: MAPE 6.66%

Worst result: MAPE 10.75%

Summary

Summary

1. Best Model: Chronos
   Achieved the best overall performance with the lowest MAPE, demonstrating the potential of foundation model and deep learning for time series forecasting.

2. Other Models:
   a. SARIMA: provided a solid statistical approach, but its performance could be enhanced by adding exogenous variables in SARIMAX to address seasonal fluctuations and structural changes.
   b. Prophet: offered a balance between interpretability and automation, making it a useful alternative, though its accuracy was less stable across different test periods.

Further Improvement

Further Improvement

1. SARIMA
   a. Incorporate External Factors with SARIMAX: Implement SARIMAX by adding exogenous variables (e.g., holiday indicators, temperature variations) to better capture seasonal effects.
   b. Handling Structural Shift in 2012: Introduce a dummy variable to differentiate pre- and post-2012 data for better model adaptation.
2. Chronos
   a. Consider adding covariate analysis as mentioned when we introduce how to use this model in the model design section.
   b. Switch to using GPU rather than CPU for model fine-tuning, and also change hyperparameters, then check if performance will improve
3. Prophet
   a. Add customized yearly seasonality and daily seasonality and tune the Fourier Order of them to capture more patterns.
   b. Add more necessary holiday components in addition to what we have already modeled here for the big jump. With additional holiday components, the trends and seasonalities can be better learnt.

Team

Zhiqi Ma

MSDS Student

Ruoheng Du

MSDS Student

Xiaojiang Wu

MSDS Student

Yijian Liu

MSDS Student