esade

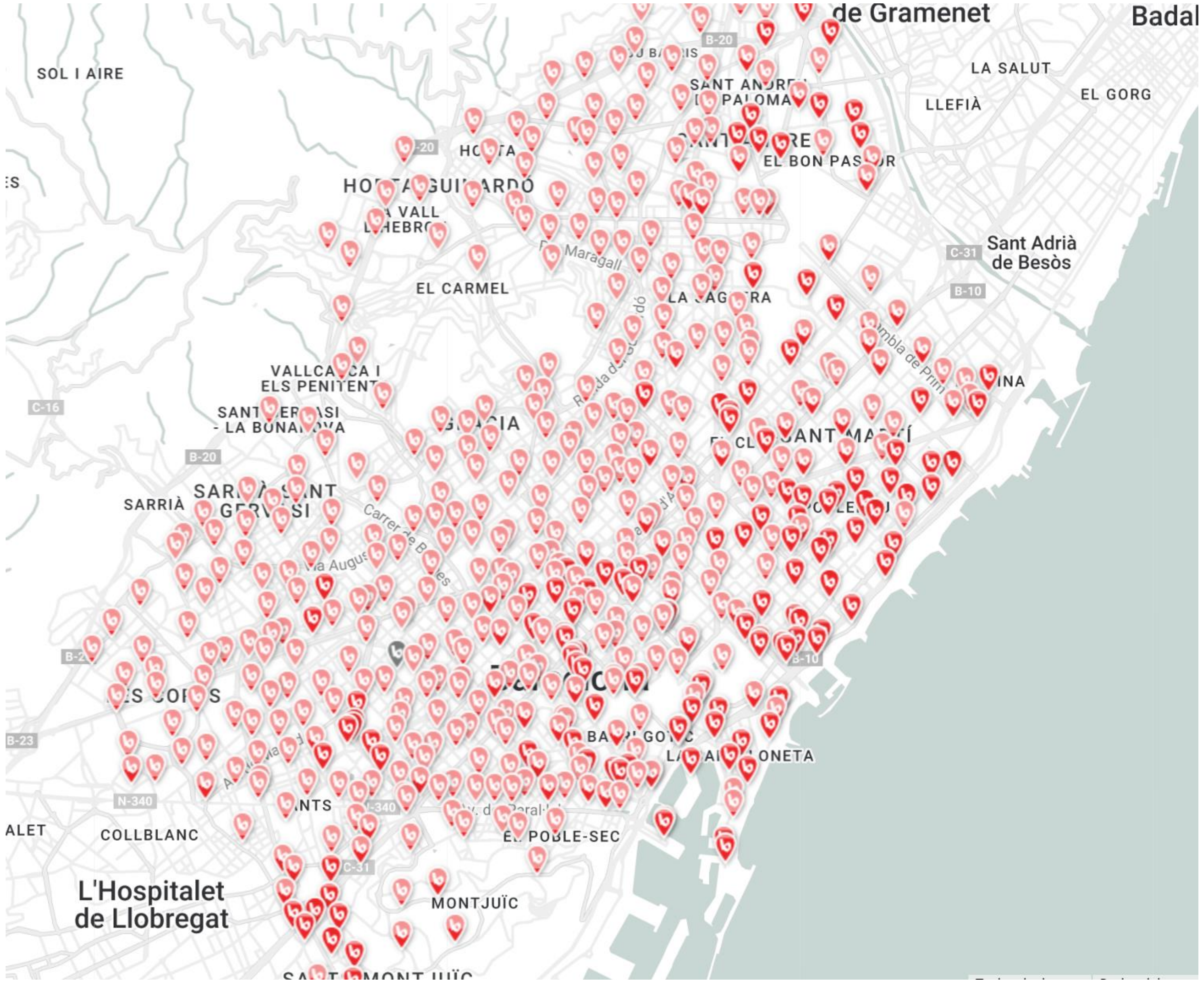
Advanced Python Final Project
Predicting Daily bicing Usage

Barcelona, June 26th 2025

Do Good. Do Better.

Problem
Introduction

Bicing is a bike sharing service with **over 500 stations** spread across the city of Barcelona



bicing

- Availability of mechanical and electrical bikes
- Citizens of Barcelona (with a valid NIE number) can sign up for the service
- Yearly subscription of 35€ or 50€ with free first 30 minutes and subsequent 35ct per half hour
- Users can easily pick up a bike using an QR code scanner within the Smou app

smou

Source: Bicing (2025a), Bicing (2025b)

01 – Problem Introduction

02 – Dataset

03 – Technical Solutions

04 – Model evaluation

05 – Business implications

06 – Struggles & Next Steps

Solving Bicings existing problems by predicting daily usage patterns

Problem

One-sided usage patterns and daily seasonality lead to unequal distribution and availability of bike throughout the cities with too few bikes in some parts and too many in others

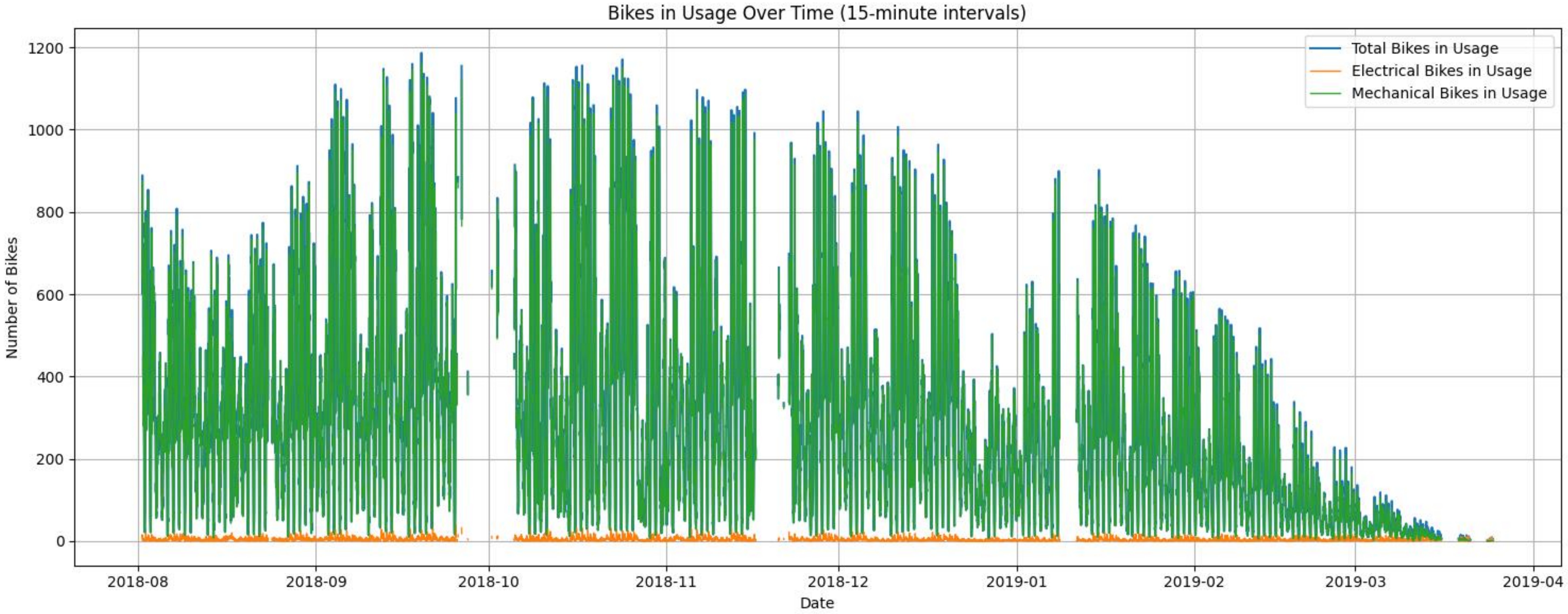
Goal

Improve the **availability and distribution of bikes** throughout the city to increase user satisfaction, determine **optimal maintenance windows** and increase revenues

Solution

Develop a time-series model to predict daily usage patterns and optimize **Bicing** logistics

Data Pre-Processing



Dataset Truncation

1. We observed a **sharp decline in bike usage starting February–March 2019**, breaking the regular seasonality and trend patterns.
2. This drop was likely caused by **external disruptions (COVID-19)**, making the later data **non-representative and unreliable**.
3. We **truncated the dataset from February 1, 2019**, keeping only the stable period from **August 2018 to early 2019**. This step ensures the model is trained on **high-quality, stable and seasonally consistent data**, improving **forecasting reliability and robustness**.

Variable Profiling

1. The dataset aggregates data at **15-minute intervals** and additional **time-based features** were extracted from timestamps: **Hour**, **Minute**, and **Day of Week**.
2. Applied **sinusoidal encoding** to better cyclical nature of time features, i.e. `sin` and `cos` transformations for **hour**, **minute**, and **day.** This allows the model to capture **periodic patterns** smoothly (e.g., 23:00 is close to 00:00).
3. Target variables were **min-max normalized** before training, and inverse-transformed after prediction to interpret results in real units (number of bikes).

Data Pre-Processing

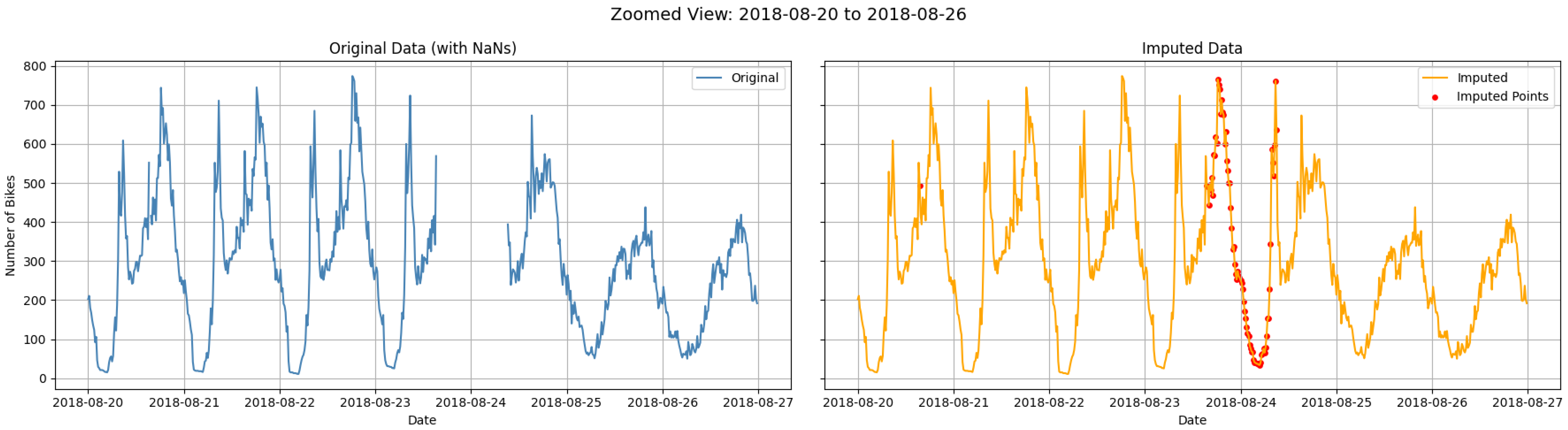
Data Imputation

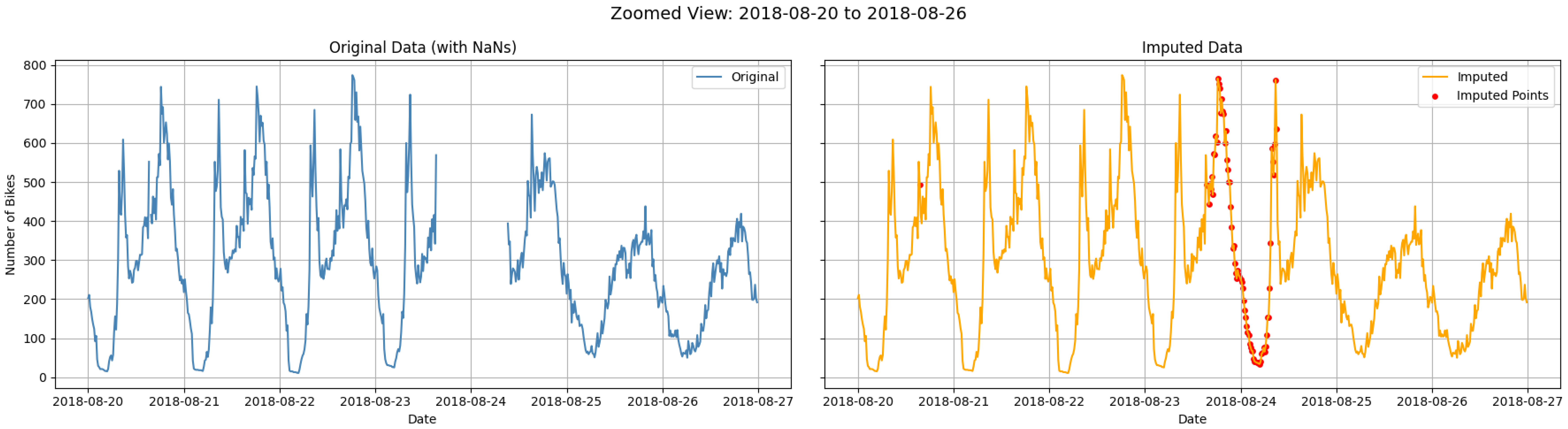
1. Imputation was required due to the presence of NA's in the time series data, which would otherwise break the continuity of the input and degrade model performance.
2. Leveraging the inherent cyclical nature of bike usage, we use a time-aware and seasonality-informed approach.

3. Example of How Imputation Works:

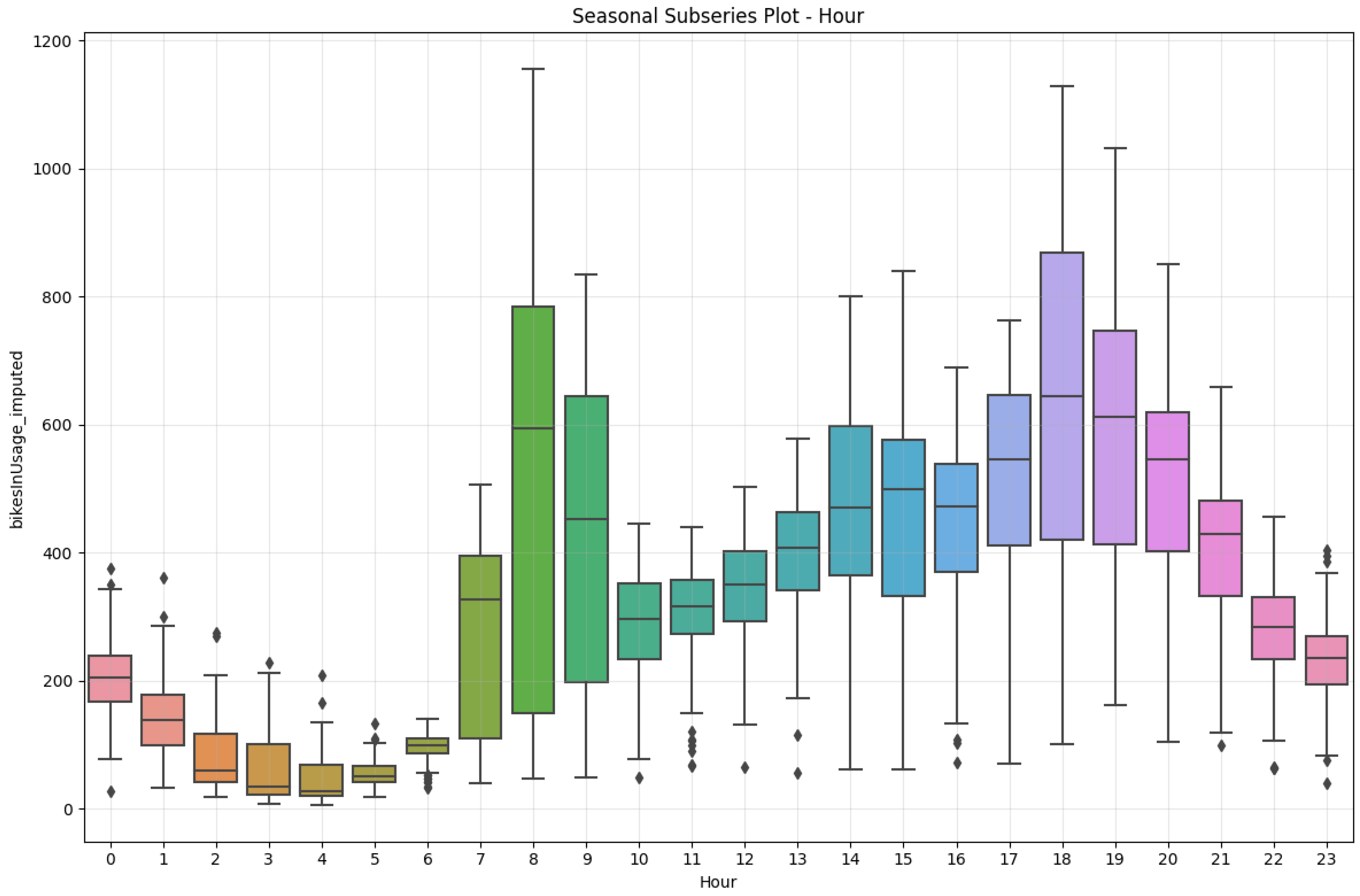
Suppose we have a missing value at this timestamp:

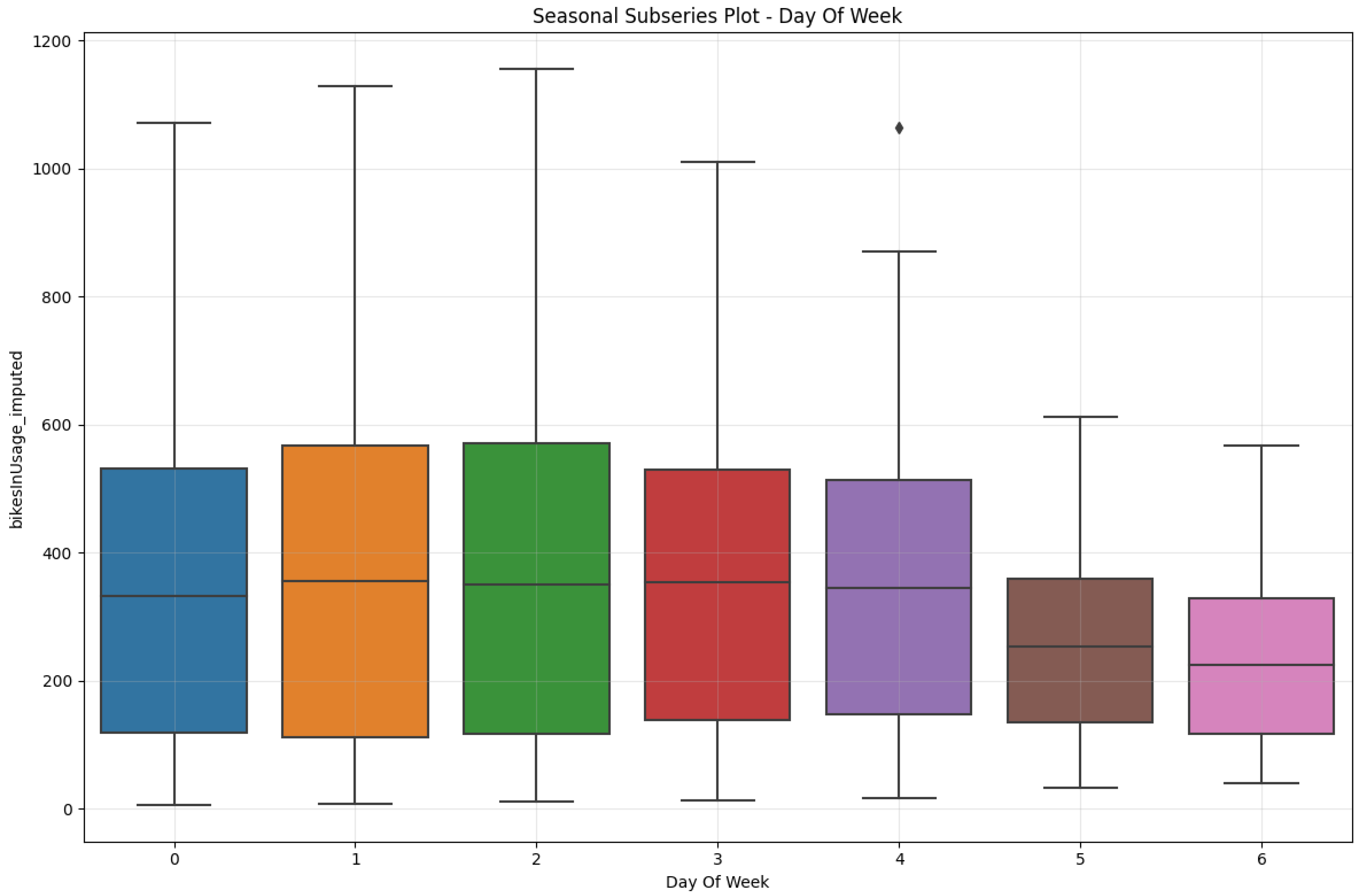
- **DateTime**: 2018-11-14 08:15:00 (which is a **Wednesday**, or day 2 of the week; we start counting at 0)
- We construct the time group identifier from the timestamp: dayofweek = 2 (Wednesday), hour = 08, minute = 15. Time group = "2-08:15“
- **We look up the average usage** at this time group from the training data. **Replace the missing values** using these averages.





Seasonal Subseries Plot (Daily & Weekly)





Strong intra-day (daily) seasonality with a cycle of 24 hours:

1. Distinct hourly seasonality, e.g. Peak usage at 8–9 AM and 5–7 PM; Low usage from 0–5 AM
2. Wide-spread (IQR and outliers) at peak hours → higher variance during commuting hours, possibly due to weekdays vs weekends.

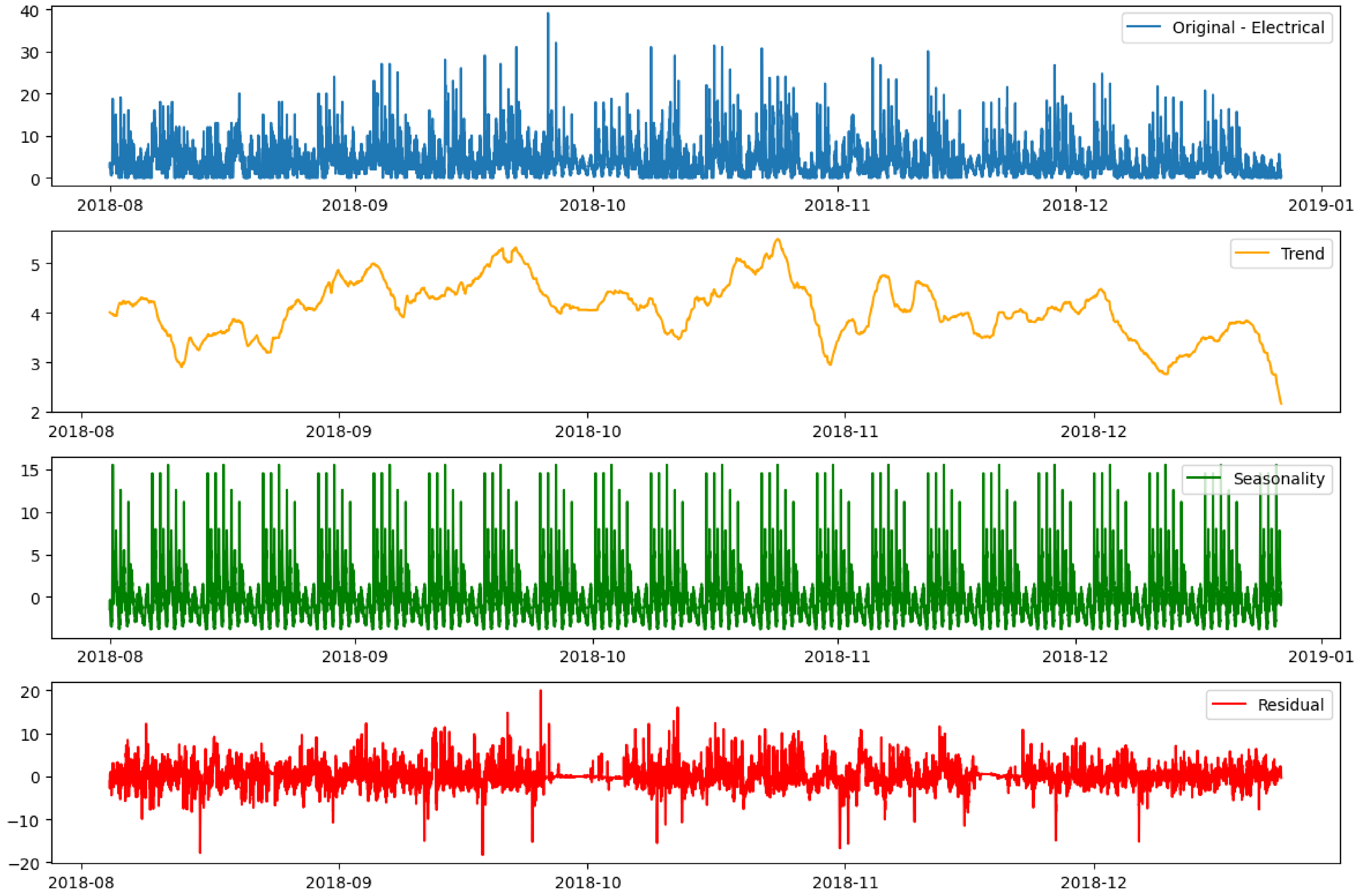
Clear weekly seasonality, cycle of 7 days.

1. Higher median usage on weekdays (0–4) → Monday to Friday. This suggests bike usage is work-related during the week. On weekdays, also higher variability and larger upper whiskers (more extreme highs).
2. Lower usage on weekends (5–6) → Saturday and Sunday.

Time Series Decomposition

Electrical Bikes

Seasonal Decomposition (Additive) - Electrical Bikes

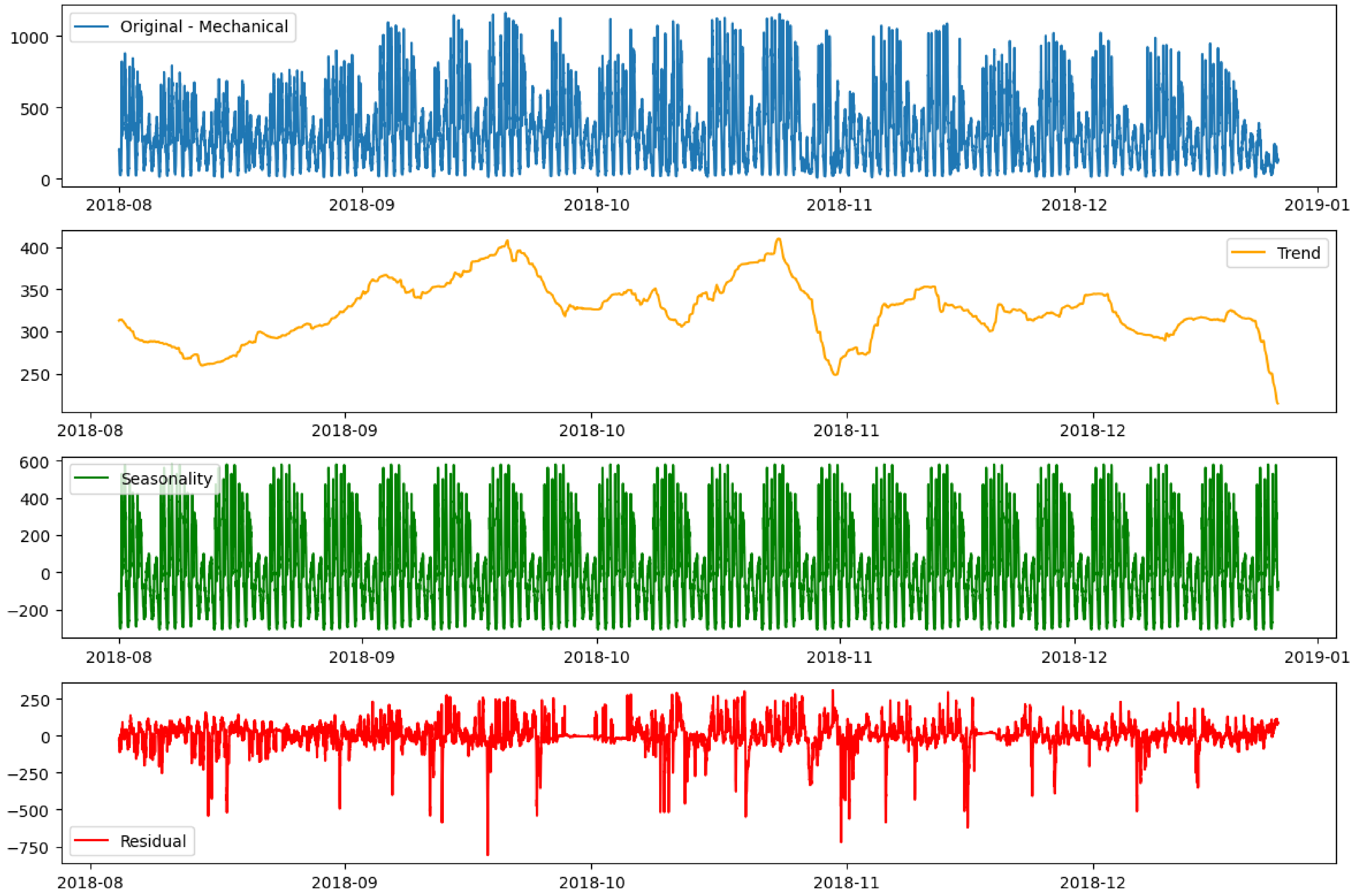


1. **Less stable and noisier trend**, likely due to lower usage volume and greater volatility.
2. Residuals are **highly erratic**, indicating a large portion of the variance is unexplained by trend or seasonality.

Mechanical Bikes

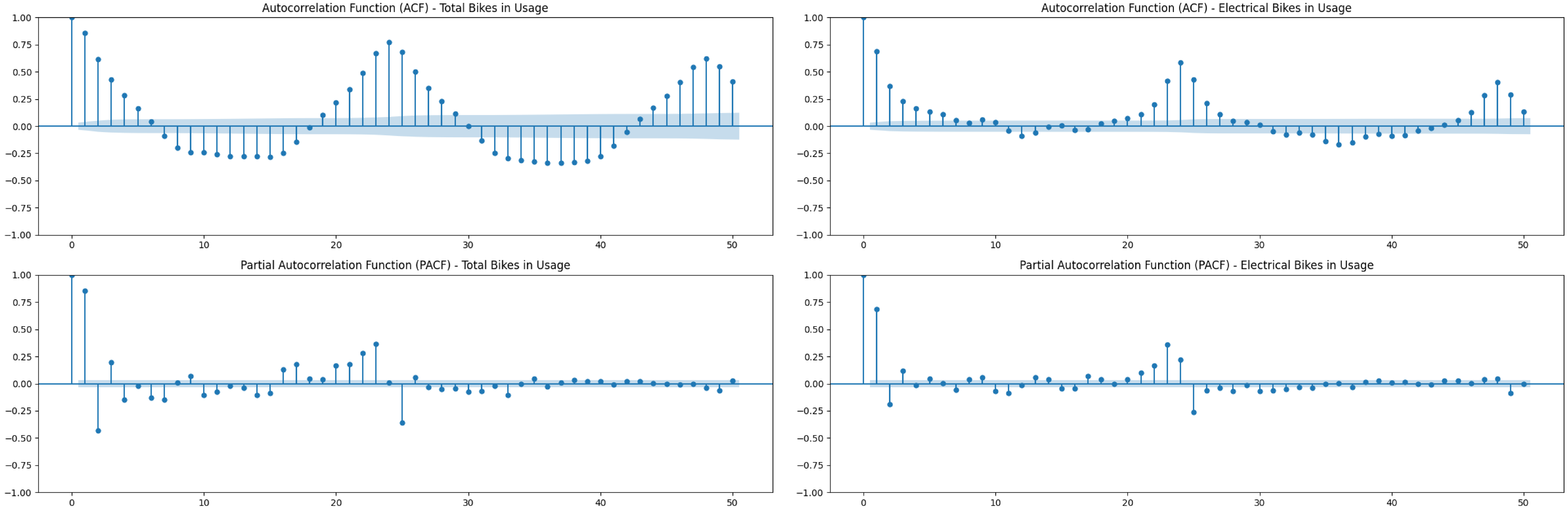
Follows a 7-day cycle

Seasonal Decomposition (Additive) - Mechanical Bikes



1. Seasonality is **much more pronounced** and **regular**
2. Residuals are smaller and more stable, suggesting that **most of the structure is captured by the model** components.

Autocorrelation and Stationarity (for Time Series)



Autocorrelation Function (ACF) - Total Bikes in Usage

Autocorrelation Function (ACF) - Electrical Bikes in Usage

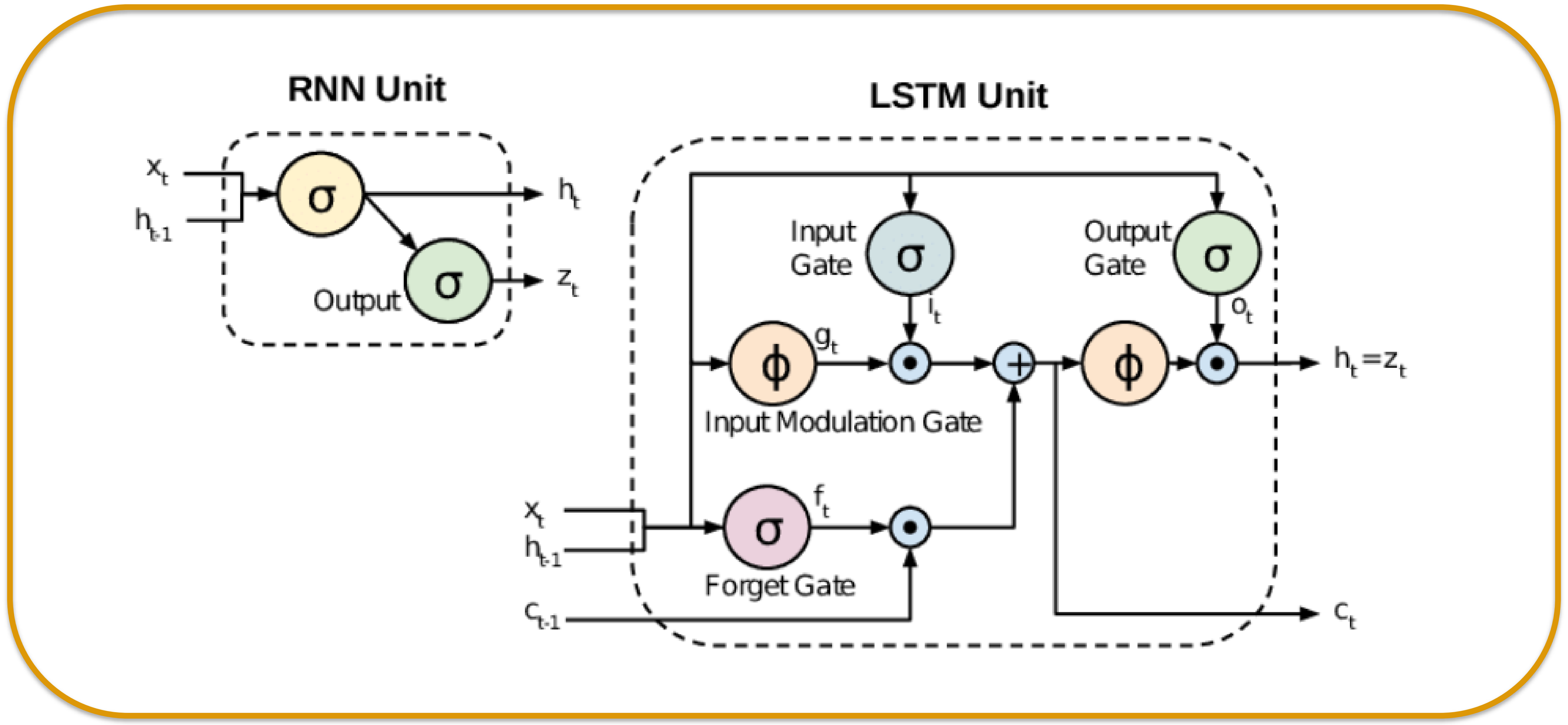
Partial Autocorrelation Function (PACF) - Total Bikes in Usage

Partial Autocorrelation Function (PACF) - Electrical Bikes in Usage

1. Strong Daily Seasonality (Period = 24 hours)
2. ACF/PACF Insights:
   - No quick decay of ACF – suggest non-stationary time-series. If, employing ARIMA/SARIMA would require differencing parameter

Model-LSTM.ipynb
Model_5.ipynb

Our Selected Technical Approach: LSTM



SMA	EMA	Exponential Smoothing
ARIMA	SARIMA	XGBoost

1. **Decided to use Deep Learning Approach after trying 6 different models:**
   - Includes a combination of first Time Series, Statistical Models then Machine Learning to prioritize model simplicity
   - Concluded Deep Learning Approach is best (Covered in our Evaluation and Technical Struggles)
2. **LSTM > RNN:**
   - RNN suitable for time-series as they capture past sequential inputs, modelling temporal dependencies; but there are issues with vanishing gradient to aid learning long-range dependencies
   - LSTM introduces gates to selectively retain information over long-range forward propagation

Model-LSTM.ipynb

LSTM Architecture

Model Configuration

1. LSTM model consist of **2 hidden layers** each having **64 units**, to model temporal dependencies over a week (672 time steps = 24*4*7).
2. The output layer is a **fully connected linear layer** predicting **96 steps ahead (1 day of 15-min interval)**, for **2 targets** (electrical + mechanical).
3. Uses an **encoder-decoder approach**: the encoder processes input sequences, and the decoder recursively generates multi-step predictions.

Training Strategy

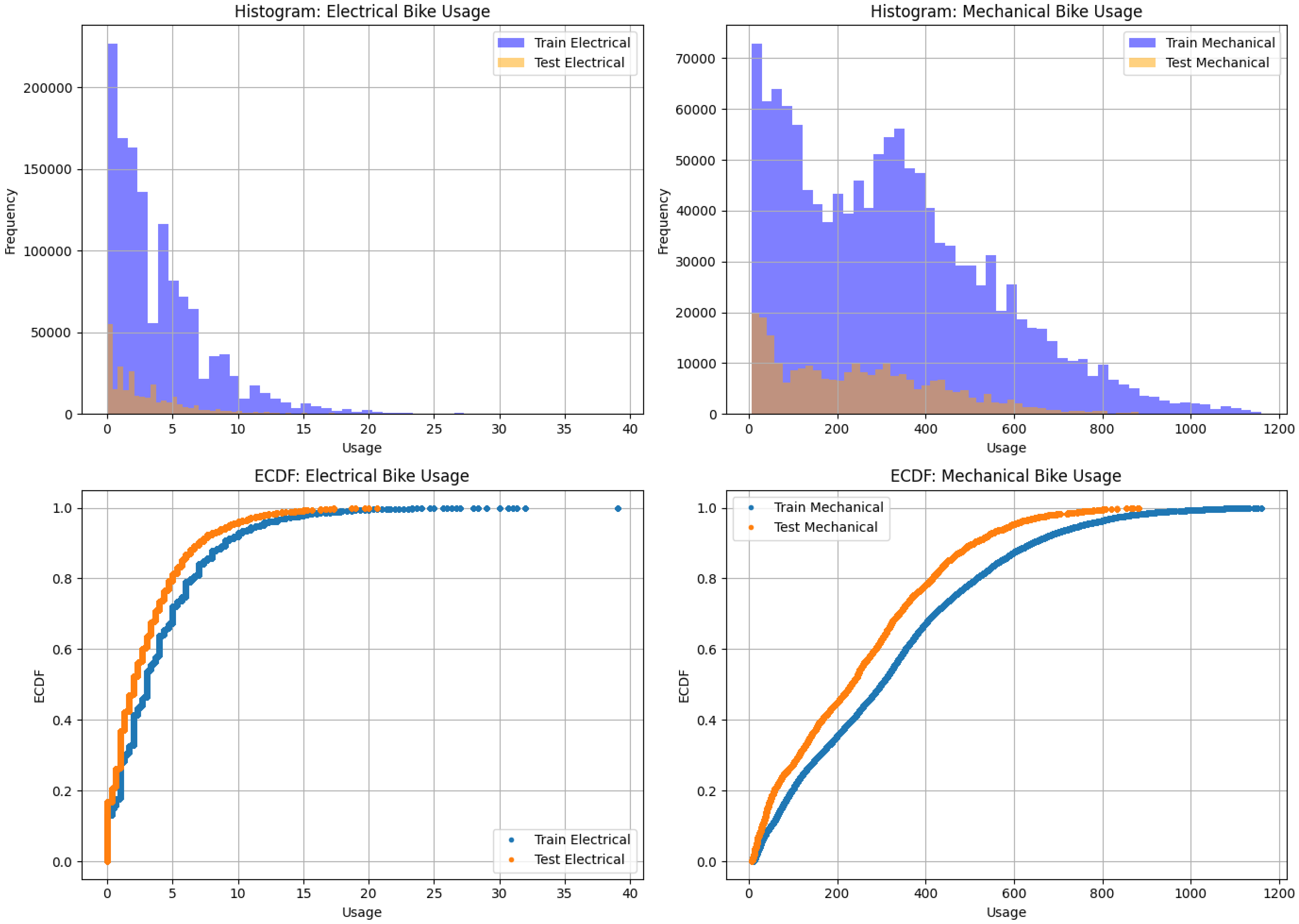
1. Loss function objective: **Mean Squared Error (MSE)** – suitable for continuous regression, penalizes large errors.
2. Used **Early Stopping (patience = 5)** to halt training when validation loss stopped improving, and added Dropout = 0.2 between LSTM layers to reduce overfitting and improve generalization.
3. The forecast horizon is **96 steps** (i.e., 1 full day), predicting both bike types for each 15-minute interval.

Optimization & Tuning

1. Learning rate chosen using a **learning rate finder** → optimal LR ≈ 1e-2 based on steepest loss descent.
2. Optimizer: **Adam** – for sparse and noisy time series.
3. Target values were **Min-Max scaled** between 0 and 1 before training, and **inversely transformed** after prediction for evaluation.

Model-LSTM.ipynb

LSTM Train-Test Results



Train-Test Data Inspection:
The **ECDF curves** and **histograms** highlight clear differences in the statistical distributions of training and testing data, suggesting a general distributional shift

Electrical Bikes:

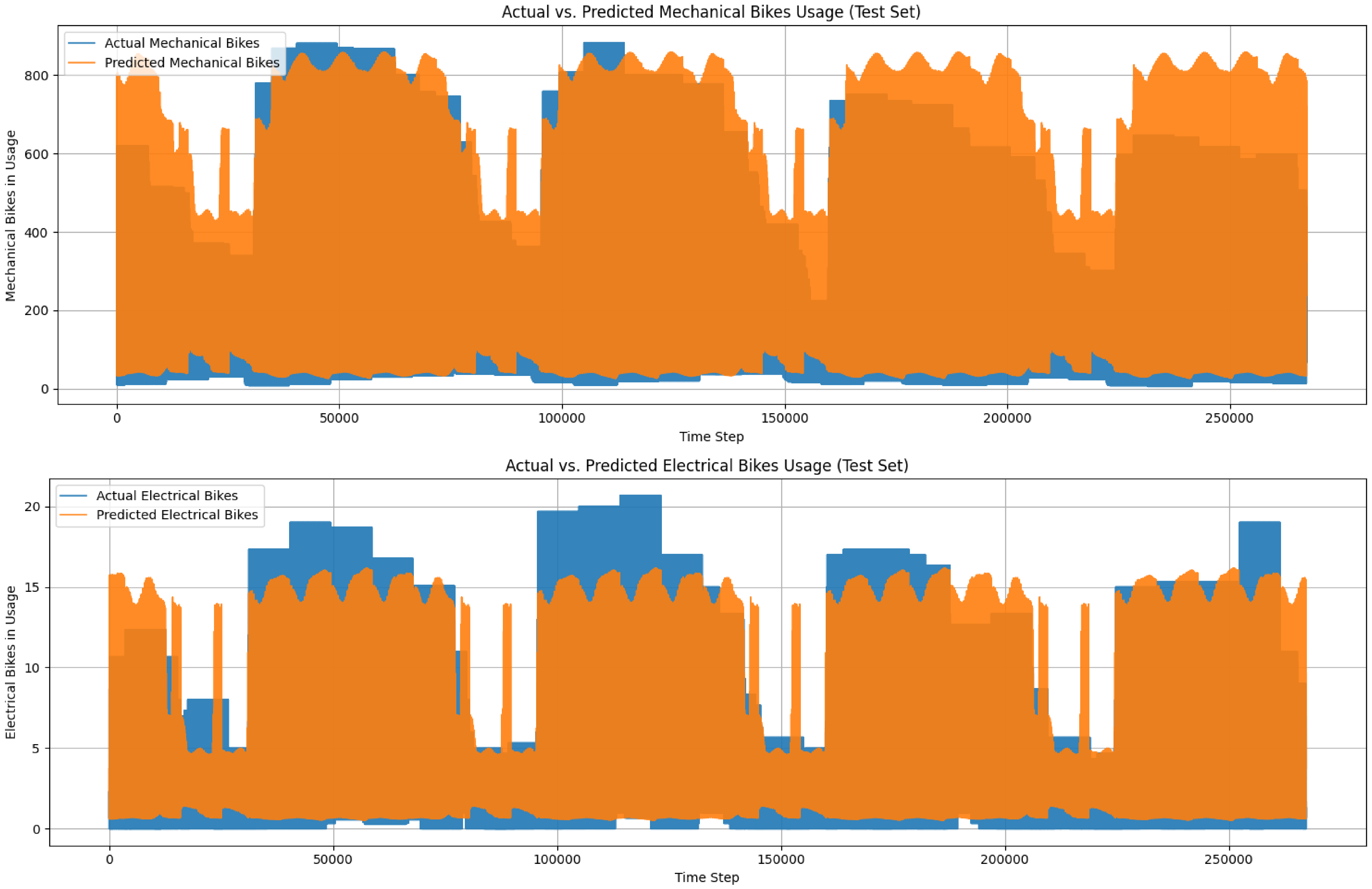
- Highly **sparse and skewed** usage, concentrated near zero.
- Test set has even **fewer high-usage values.**

Mechanical Bikes:

- The **test set shows a greater concentration of low-usage values** compared to the training set, suggesting a structural shift in demand or availability.
- This limits the **generalizability of the LSTM model** on the test set (next slide).

Model-LSTM.ipynb

Forecast to Mechanical vs Electrical Bikes:



Insights:

1. For mechanical bikes, **clearer trend-seasonality structure** and **less noise**.

2. For electrical bikes, the model shows **minimal variance in predictions** → orange area does not reproduce the sharp drops and spikes in blue area.

3. This indicates that the electric bikes model has **limited responsiveness to abrupt fluctuations**, especially with sparse & low-volume signals.

Next Steps &
Recommendations

Recommendations for Business Case:

Business Recommendations

1. To reduce customer disappointment as business objective, it might be good to ensure the availability of bikes by having predictions for smaller time-steps (i.e. 15 minute intervals)
2. However, this means (1) volume of data collected per 15-minute is less vs a longer interval, (2) any model struggles if data is too sparse → like for the usage model electrical bikes.
3. **Practical and useful to implement stable forecasting model for mechanical bikes only** – affects more customers.
4. Provide incentives to switch from electric to mechanical bikes in case of unavailability → Mechanical bikes are greener for the environment

Technical Recommendations

1. Sharp drop in bike usage after 2019 → Retrain model on updated post-COVID usage patterns once new reliable data is available
2. Aggregate data from 15-minute interval to 1-hourly interval to have more data to work with in a timeframe of time-series modelling
3. Try hybrid models (e.g., Prophet + LSTM) to combine trend/seasonality detection with deep learning
4. Deploy anomaly detection tools to better handle spiky behaviors in electrical bikes

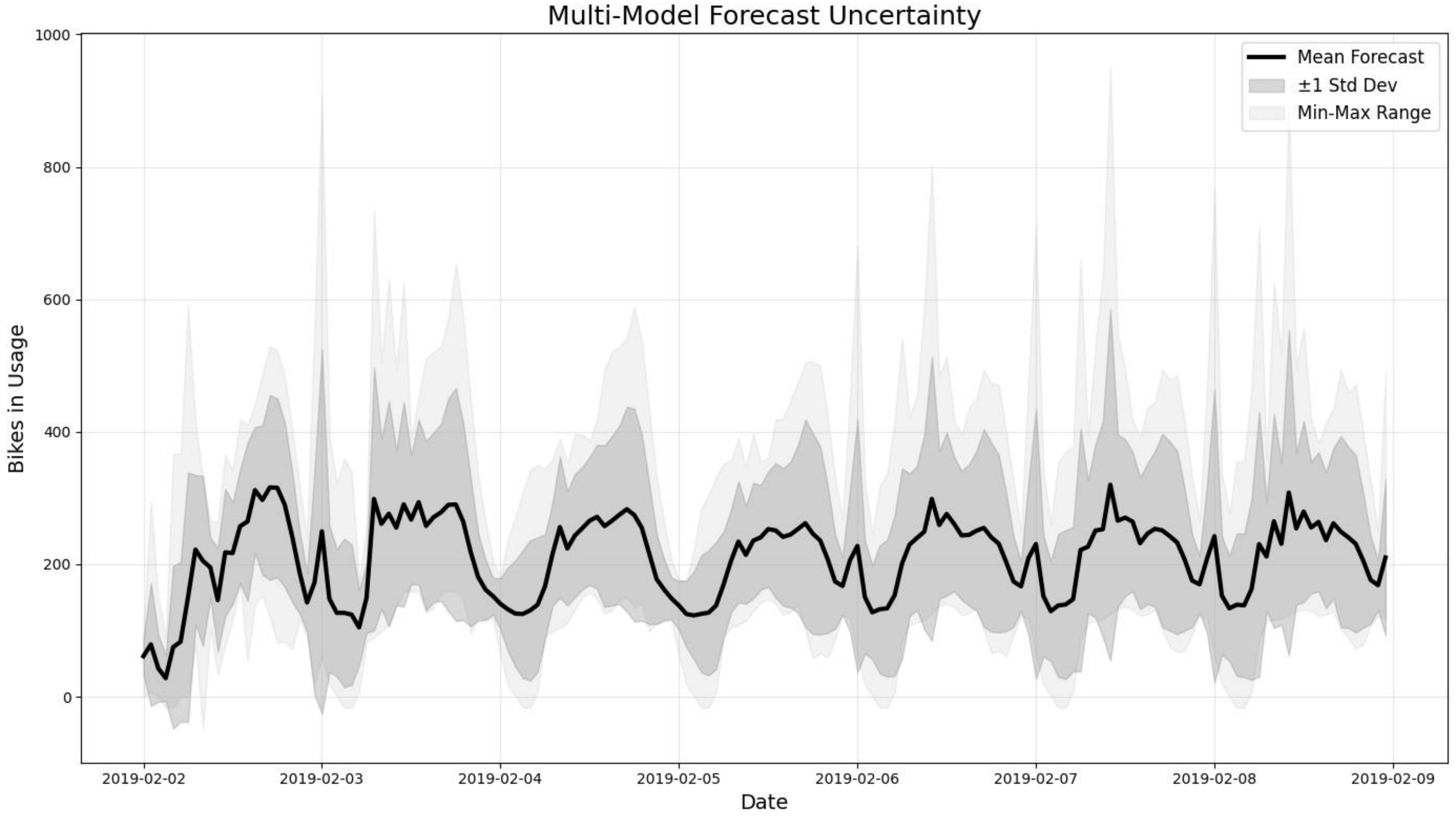
Model_5.ipynb

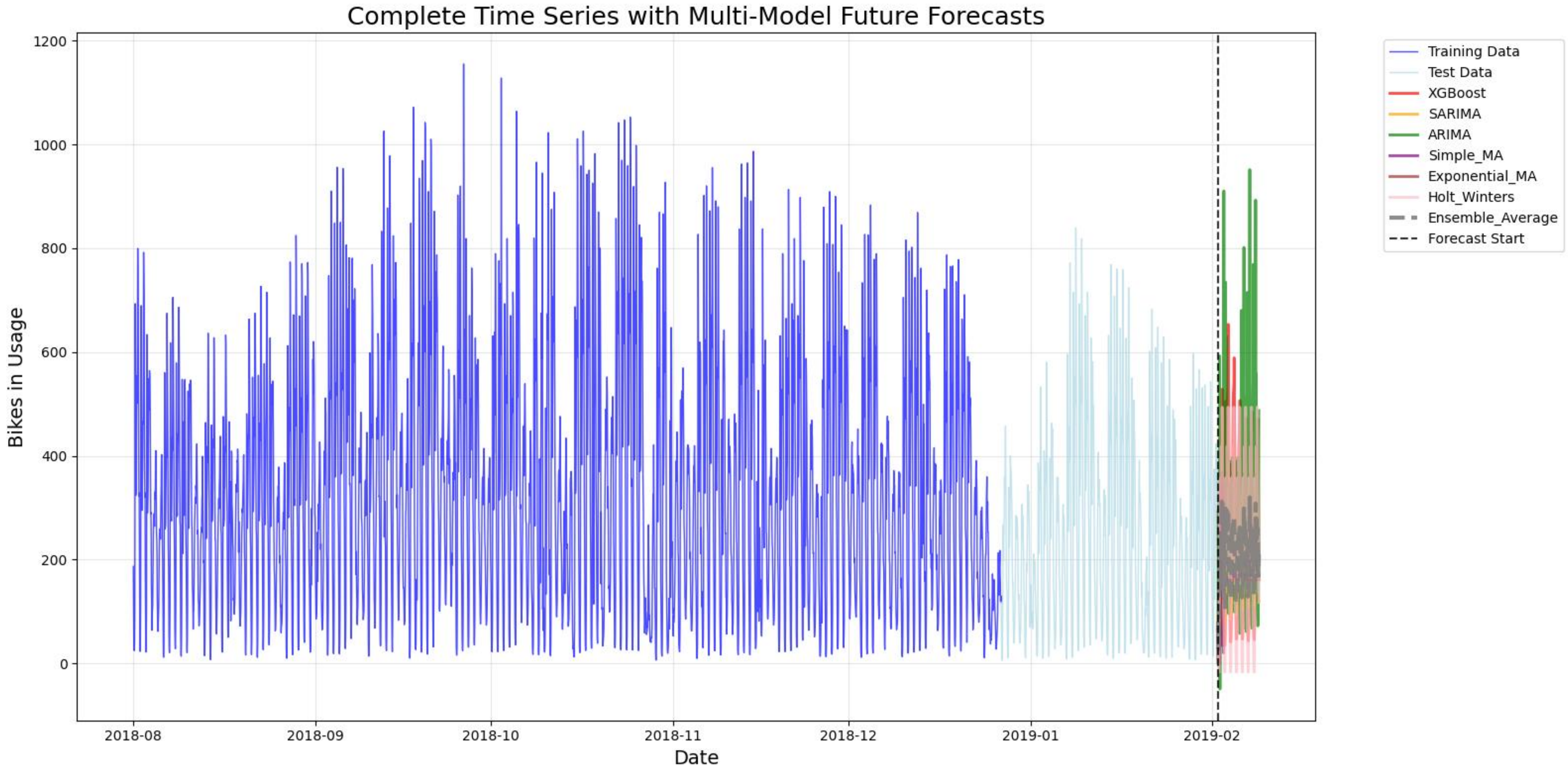
Other 6 Models tried:

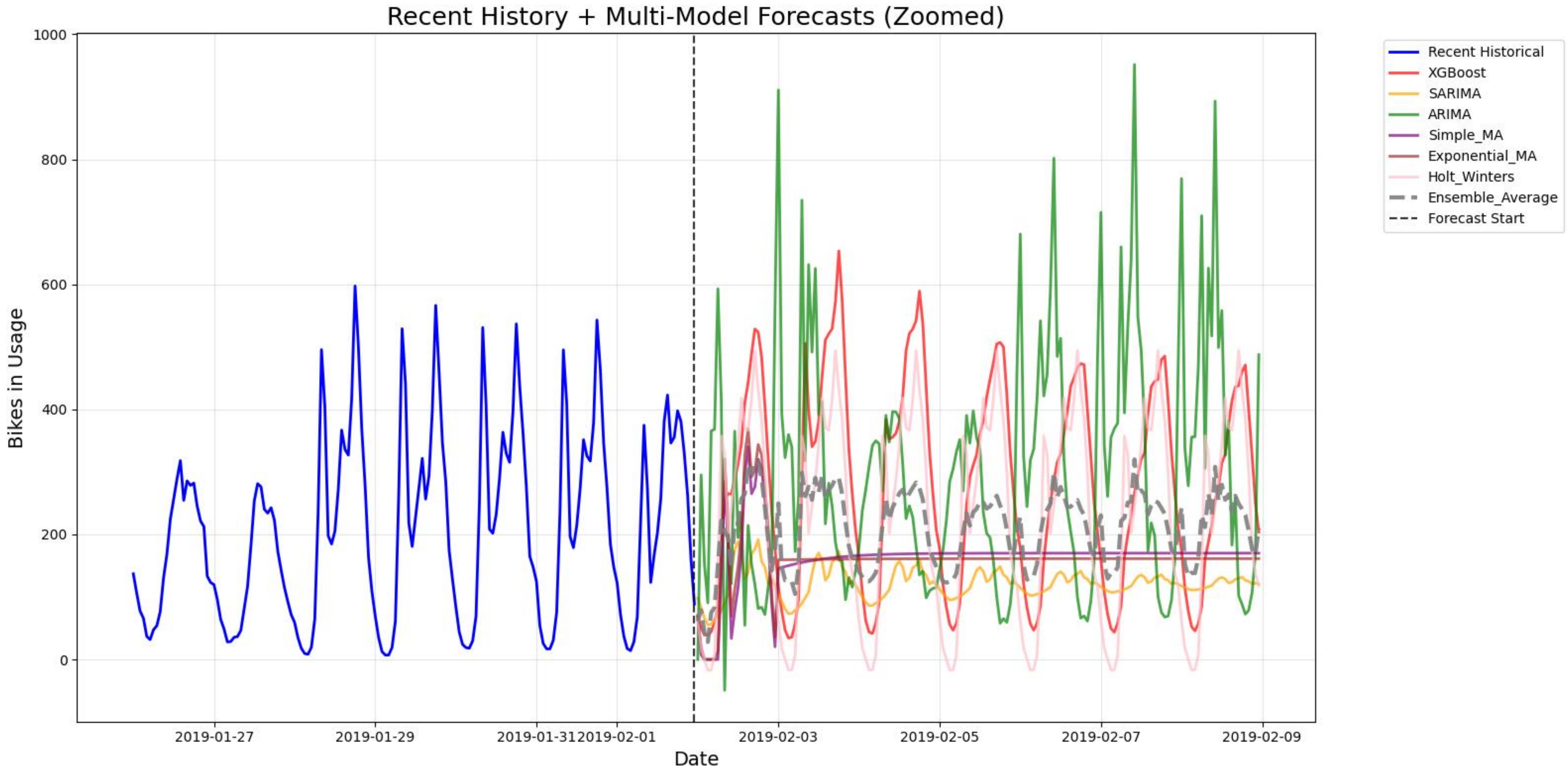
Model	MAPE	MAE	RMSE	R2	Goodness of Fit	(Limited) Configuration of Parameters
XGBoost	81.03	104.31	144.68	0.3099		• Augmented approach that included hybrid forecasting approach • Technicalities include: Lag features, weighted sampling with recency bias, a trend coefficient and damping factor
SARIMA	115.66	157.82	208.84	-0.4378	AIC: 5480.240	• p,d,q = (2,1,1) • P,D,Q,S = (1,0,0,24)
SMA	119.05	159.43	209.14	-0.4419		3-period
Exp Smoothing	120.18	162.05	212.49	-0.4885		Holt-winters with optimized parameters: • Alpha (Levels) • Beta (Trend) • Gamma (Seasonal) • Phi (damping factor)
EMA	122.71	157.76	206.39	-0.4043		3-period
ARIMA	276.69	152.55	182.10	-0.0932	AIC: 11819.53	p,d,q = (2,1,2)

Model_5.ipynb

6 Models Forecasts and Uncertainty:







Model-LSTM.ipynb
Model_5.ipynb

Evaluation: LSTM as the best-performing model

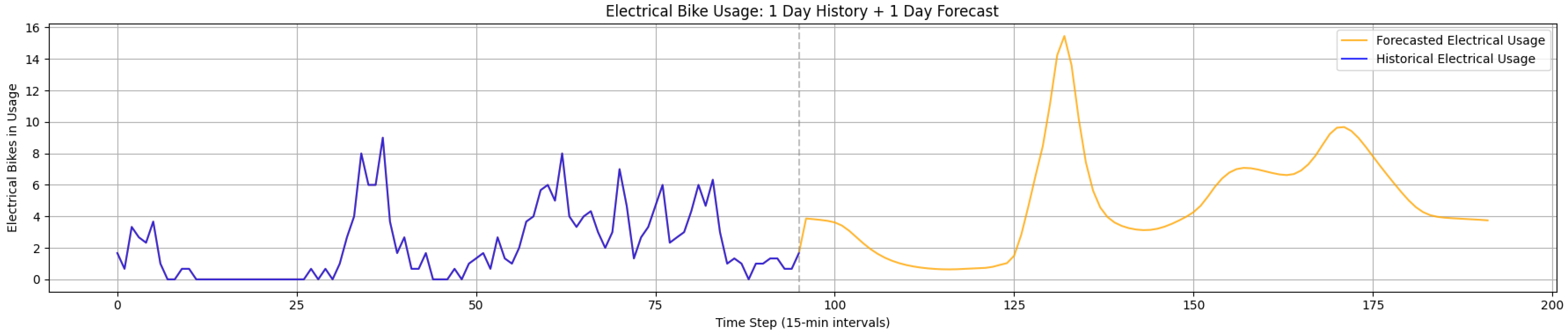
MAPE %	Train	Test
Electrical	76.27%	108.2%
Mechanical	41.44%	75.91%

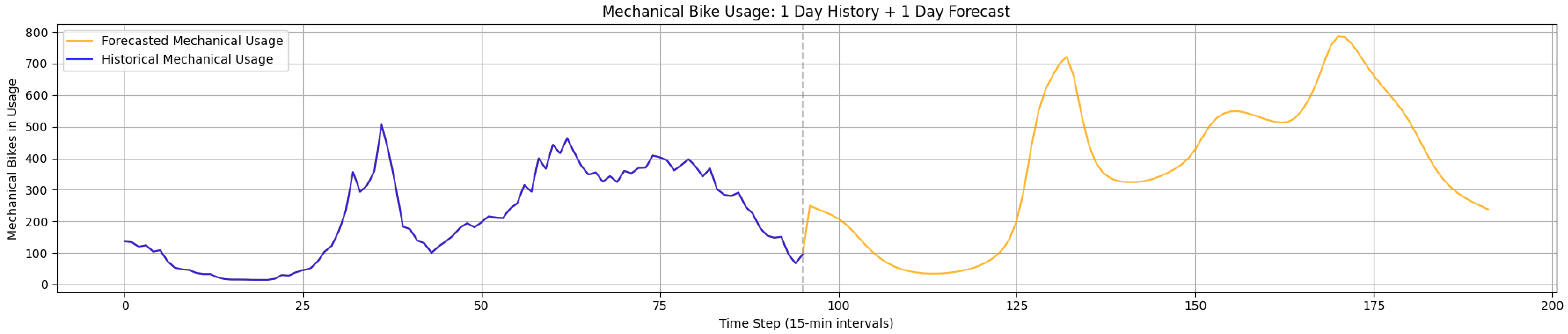
Future Improvements:

1. Apply **log-transformation** to highly skewed targets
2. Consider **feature engineering** to include external covariates (weather, calendar, events)
3. Explore **alternative architectures** (e.g., attention-based models or ensemble approaches) for more stable and generalized predictions.

1. Despite the high Test MAPE values, **LSTM has the lowest MAPE** amongst the 6 other models we have attempted,
2. In comparison with other models (next slide), we have **had to combine both electrical + mechanical bikes data** and **increase the time-interval from 15 minutes to 1-hour interval** so that more data can be collected for the model to work.
3. The primary cause of high error is still the **combination of non-stationary data, evaluation metric sensitivity, and distributional shift patterns** between train and test sets.

Generating Forecast based on LSTM:





Mechanical Bikes:

1. The model forecast generalizable cyclical patterns smoothing the line
2. But we must consider while the trend pattern can be observed, it may not reflect accuracy
3. **LSTM also struggles with erratic, low-volume data**

Electric Bikes:

1. Very low and sparse usage (values mostly under 10) also makes it difficult for model to forecast meaningfully

Evaluation: Technical Issues and Struggles

Temporal Consistency of Dataset

1. **Severe drop in bike usage** post-Feb 2019 → this required truncation of dataset, otherwise the predictions tend to 0
2. **Distributional Shift of Data** across time, causing many model results to have MAPE > 100% and R2 to be negative
3. Less complex models (SMA/EMA) provide mere naïve forecasting

Workarounds for Model Generalizability

1. Created complex XGboost to include lag features, weighted sampling with recency bias etc, to still get a slightly higher MAPE than LSTM
2. LSTM still best approach overall

Computational Complexities

1. Unable to parallelize computation for Machine Learning: (1) Rolling window train-test split is sequential in nature. (2) Memory constraints
2. Unable to converge computations in ARIMA/SARIMA for 15-minute interval data: (1) More lags required, (2) Larger matrices for model fitting, (3) More complex parameter search space

Workarounds for Computations

1. Aggregated 15-minute intervals to 1-hour intervals
2. Separated Deep Learning training and 6 other models on 2 different notebooks and 2 machines with separate CPU and GPUs
3. Use of already available libraries for ARIMA/SARIMA

Thank you!

APPENDIX

Sources

- Bicing (2025a): Bicing Map, https://www.bicing.barcelona/es/mapa-de-disponibilidad
- Bicing (2025b): Servicio Bicing, https://www.bicing.barcelona/es/servicio-bicing