

# **Forecasting Energy Load for Efficient Utilization of Battery Energy Storage Systems**

Tim Prause, Kadisatou Fane, Cosima Birkmaier





# Contents

- ❑ Motivation
- ❑ Dataset
- ❑ Baseline Models
- ❑ Models
- ❑ Model Comparison
- ❑ Summary & Outlook
- ❑ References



# Motivation



- more renewable energy sources
- new challenges for power grids
- electricity generation must be flexibly adapted to power consumption
- one solution: **batteries:**
  - Battery Energy Storage Systems (BESS)
  - charge, when more energy is generated than needed
  - discharge to the grid, when demand is higher than (renewable) generation
  - financially: buy energy, when prices are low, sell it, when prices are high
    - supply and demand
- goal of our project: predict energy load for better battery charging and discharging



# Dataset

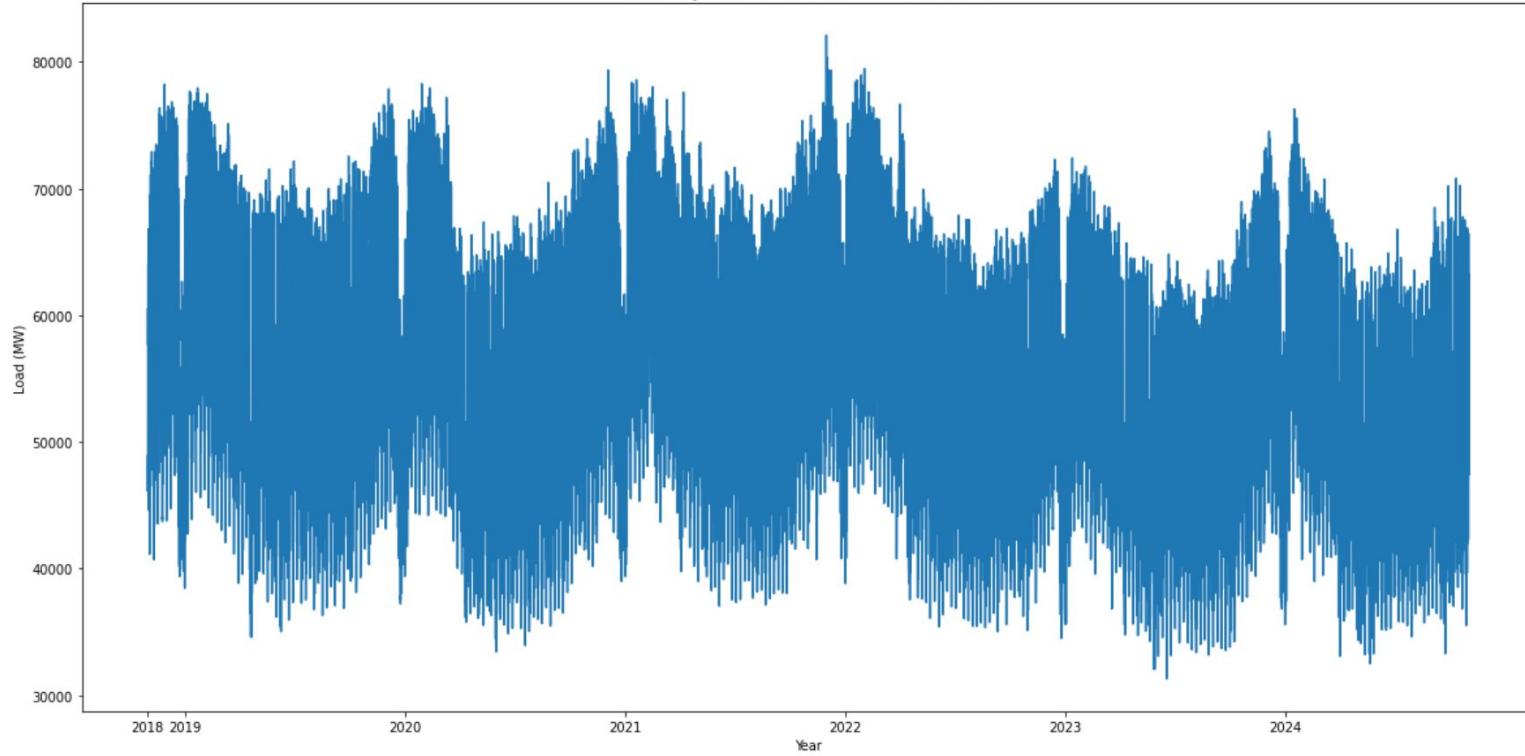


- Dataset obtained from Bundesnetzagentur (Federal Agency for Grids)
- Total energy load from Germany and Luxembourg
- From 01.11.2018 to 31.10.2024
- Energy production per production type (e.g. wind, solar...)
- Hourly data
- Added features:
  - Time Features (hour, weekday, holidays...)
  - Energy Prices in Europe
- Final dataset:
  - 52,620 samples
  - 35 features



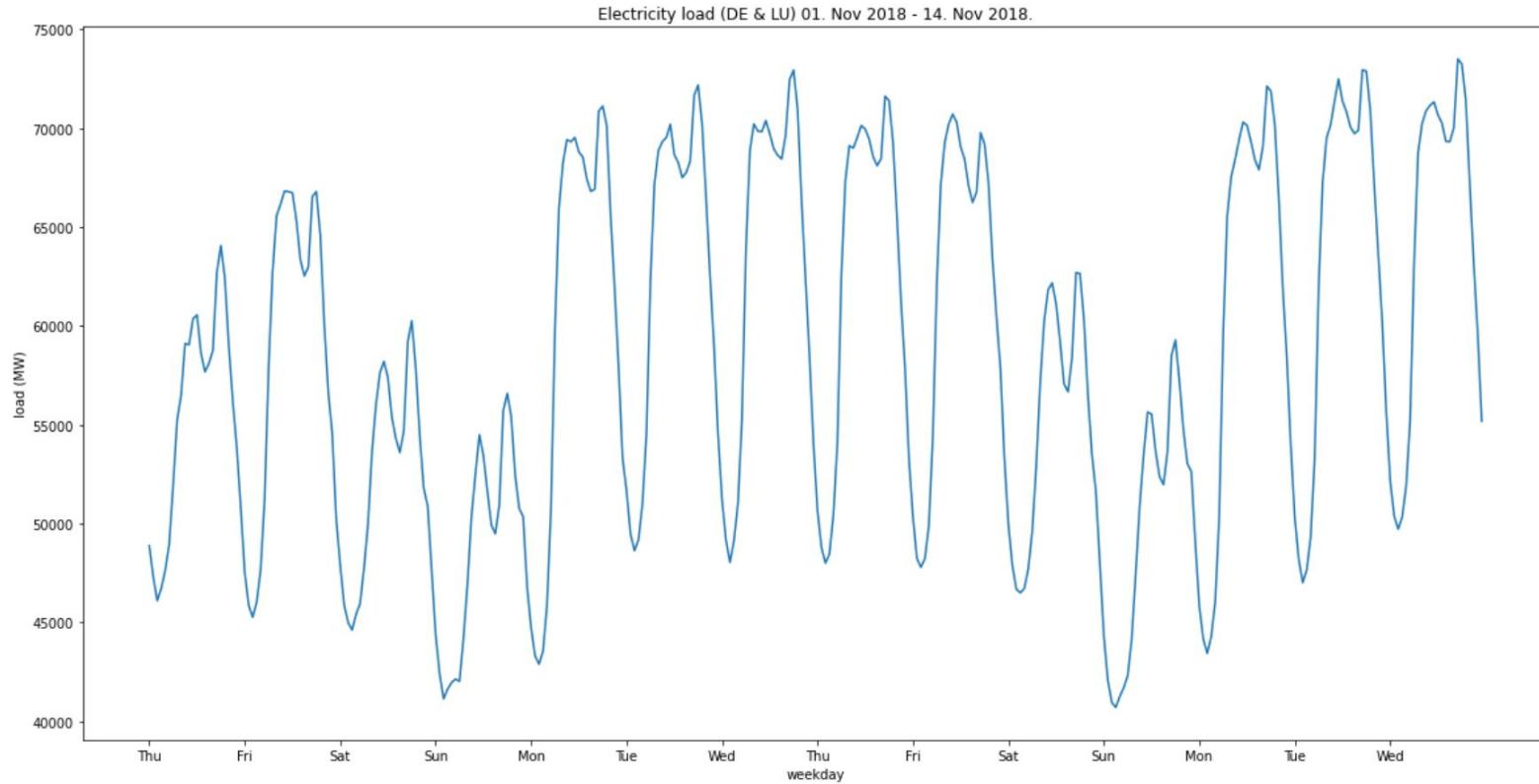
# Data Visualization & Preparation

Electricity load (DE & LU) Nov 2018 - Oct 2024.



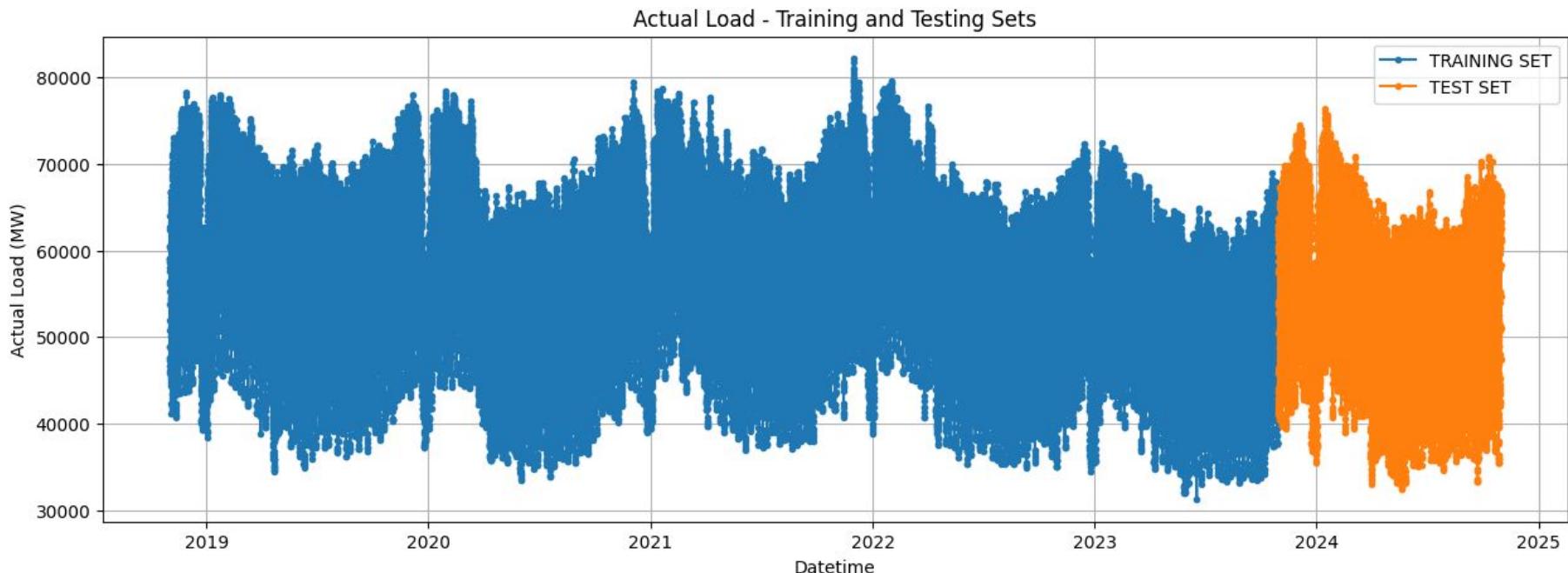


# Data Visualization & Preparation





# Train-Test split for all models:





# Feature Engineering:

## Added features:

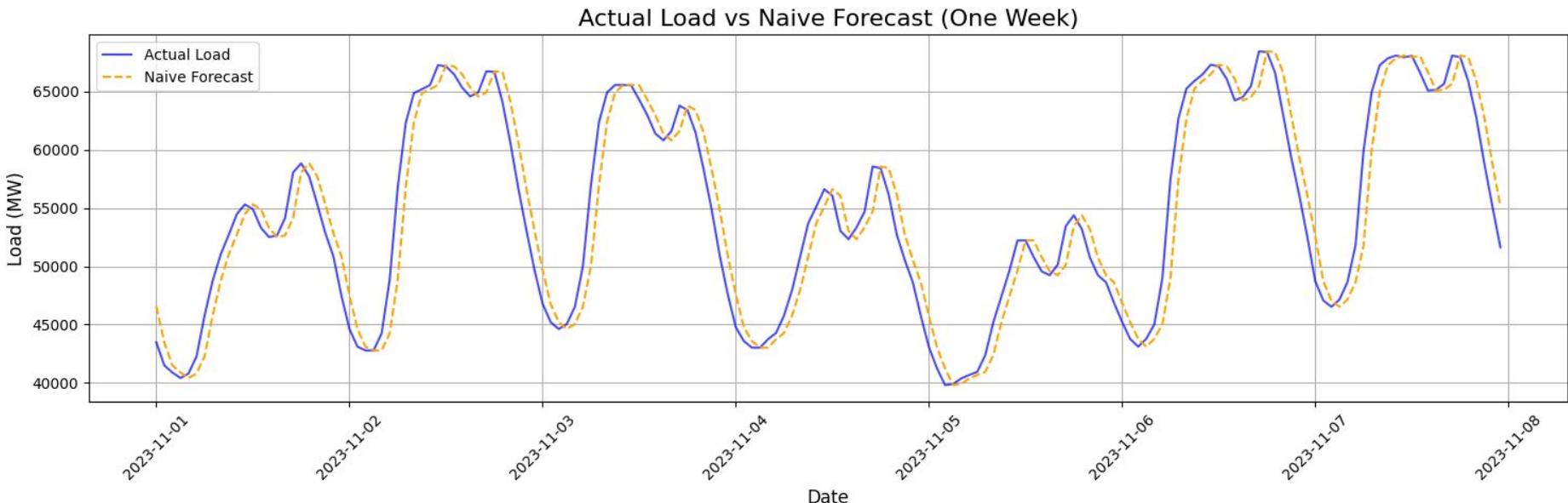
- Time Features
  - hour of day in cos and sin
  - weekday cos and sin
  - day of year cos and sin
  - holiday
  - workday True or False
- Energy Prices in Germany
- Energy Production in Germany
  - Wind, solar, coal, gas...

## Feature selection:

- Random forest Regressor for feature selection: Feature and importance
  - hour: 0.40
  - day of week: 0.086
  - nuclear energy: 0.028
  - holiday: 0.023
  - coal energy: 0.024
  - ...
- Threshold = 0.1 resulted in 13 remaining features



# Baseline Models: Naive Forecast



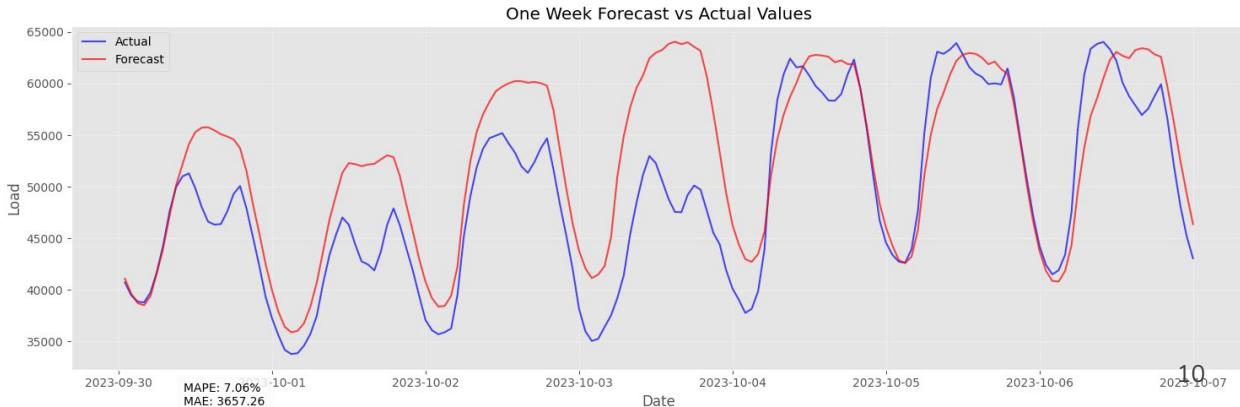
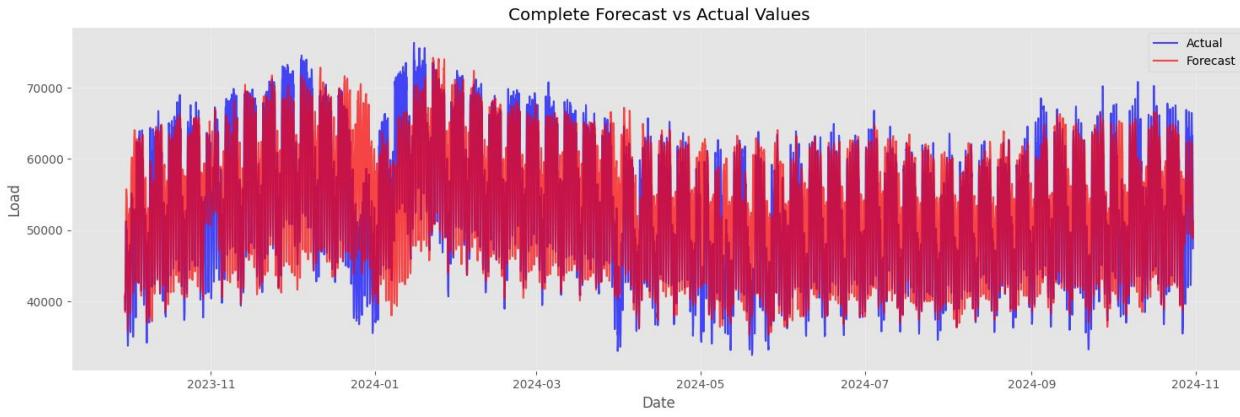
Mean Absolute Percentage Error (MAPE): 3.4592% Mean Absolute Error (MAE): 1807.92



# Baseline Models: Sarimax 1

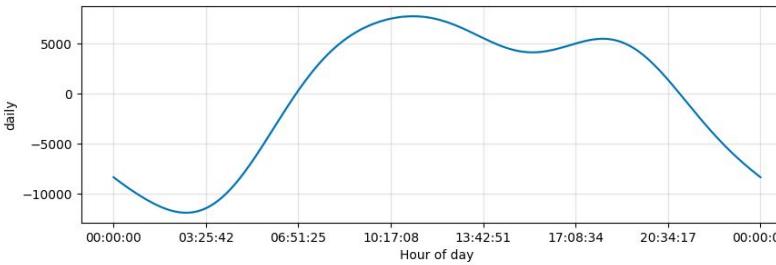
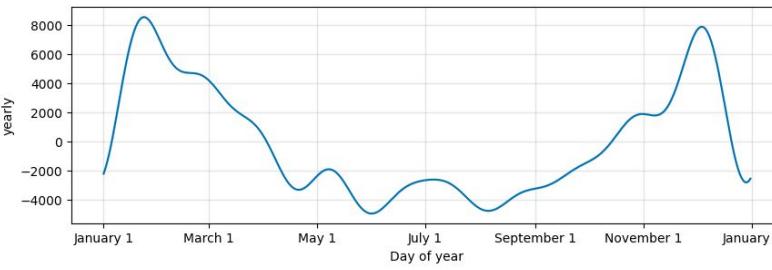
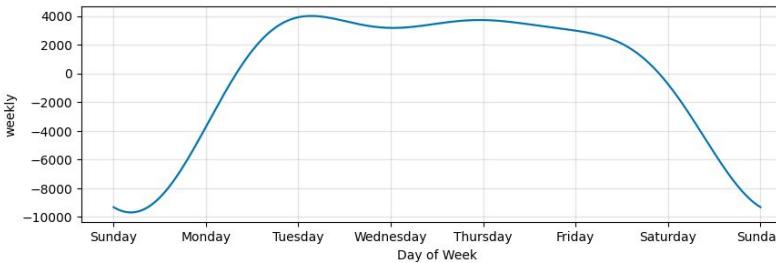
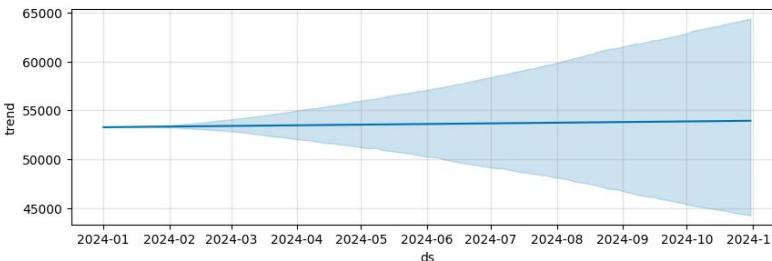
Mean Absolute Percentage  
Error (MAPE): 7.06%

Mean Absolute Error (MAE):  
3657.26



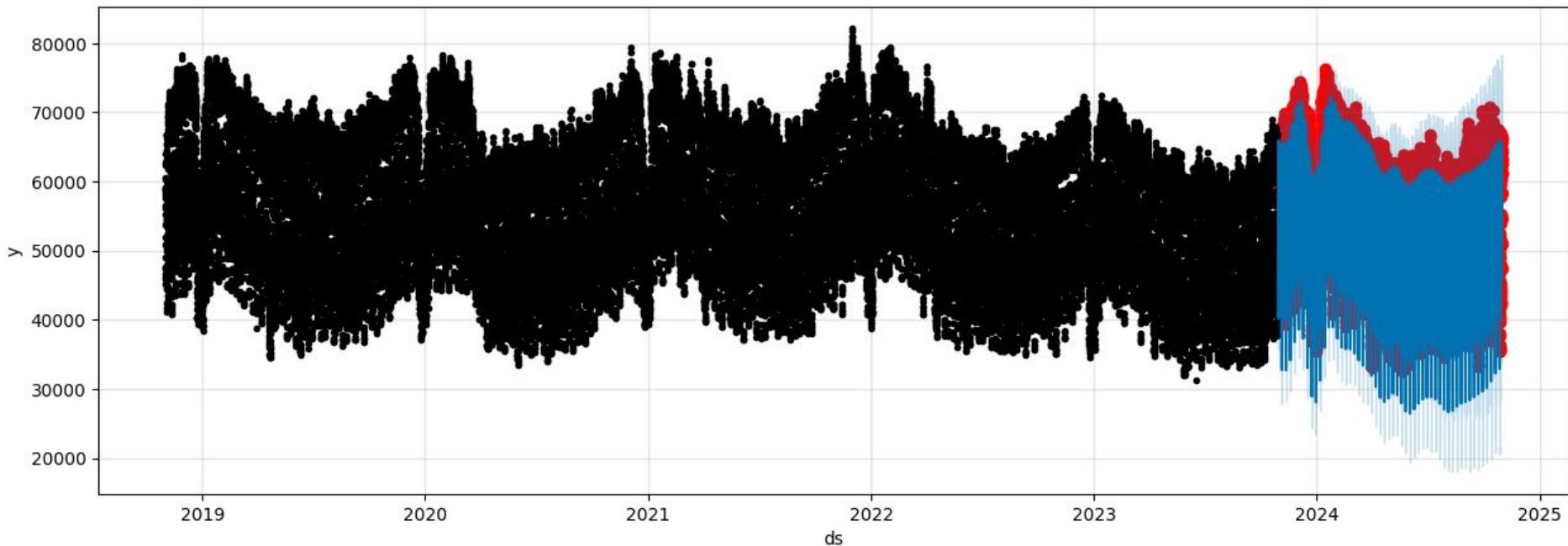


# FB Prophet:





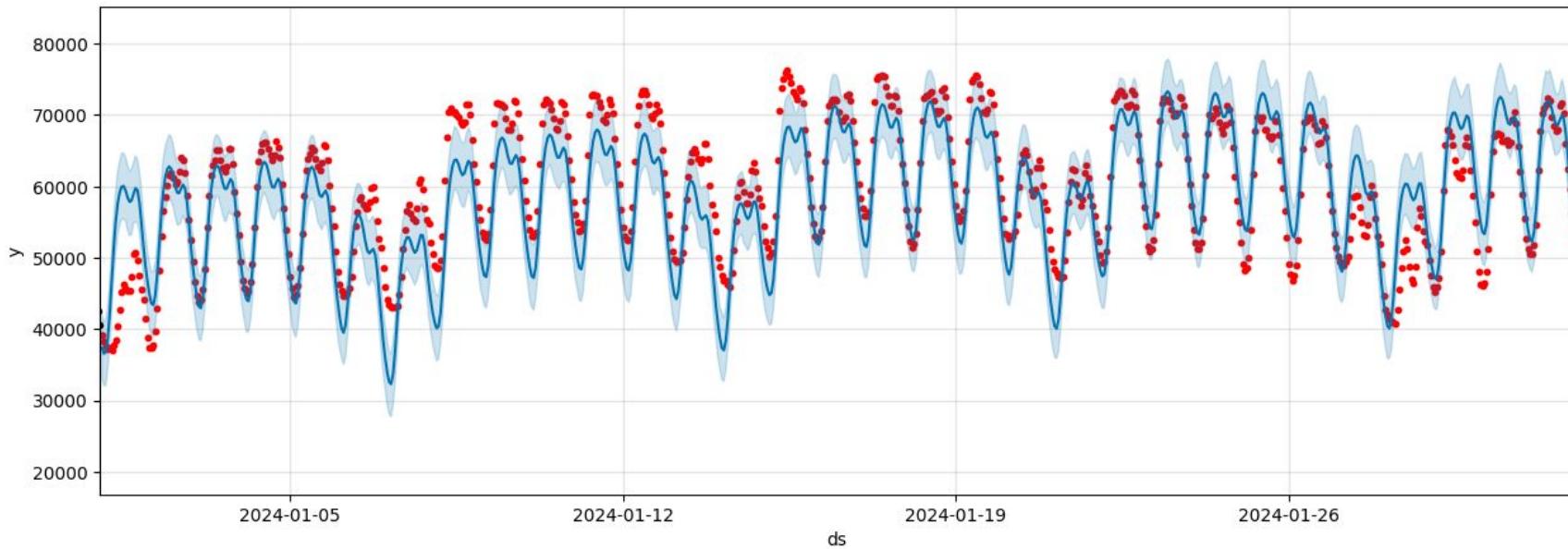
# FB Prophet:





# FB Prophet:

January 2024 Forecast vs Actuals

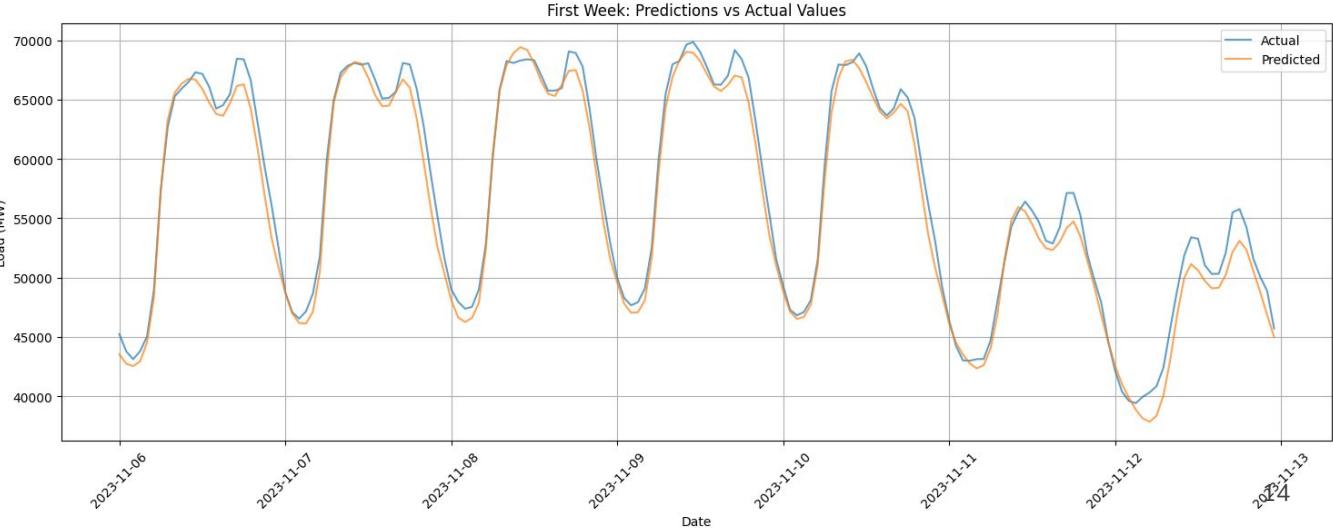
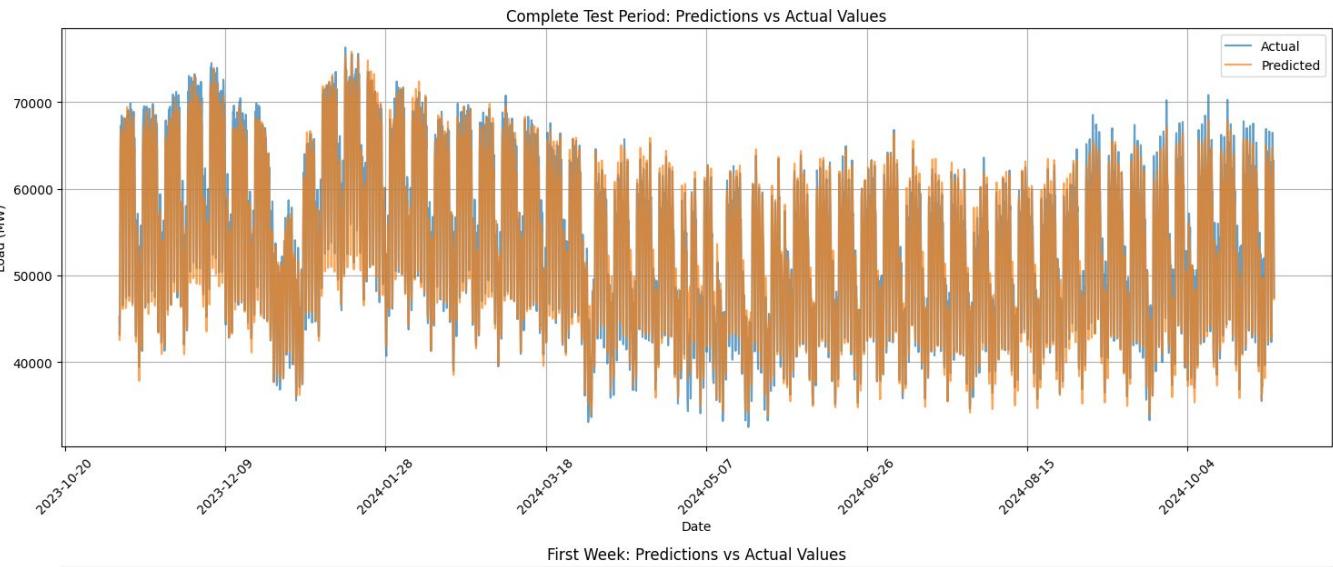


Mean Absolute Percentage Error (MAPE): 5.3319% Mean Absolute Error (MAE): XXX



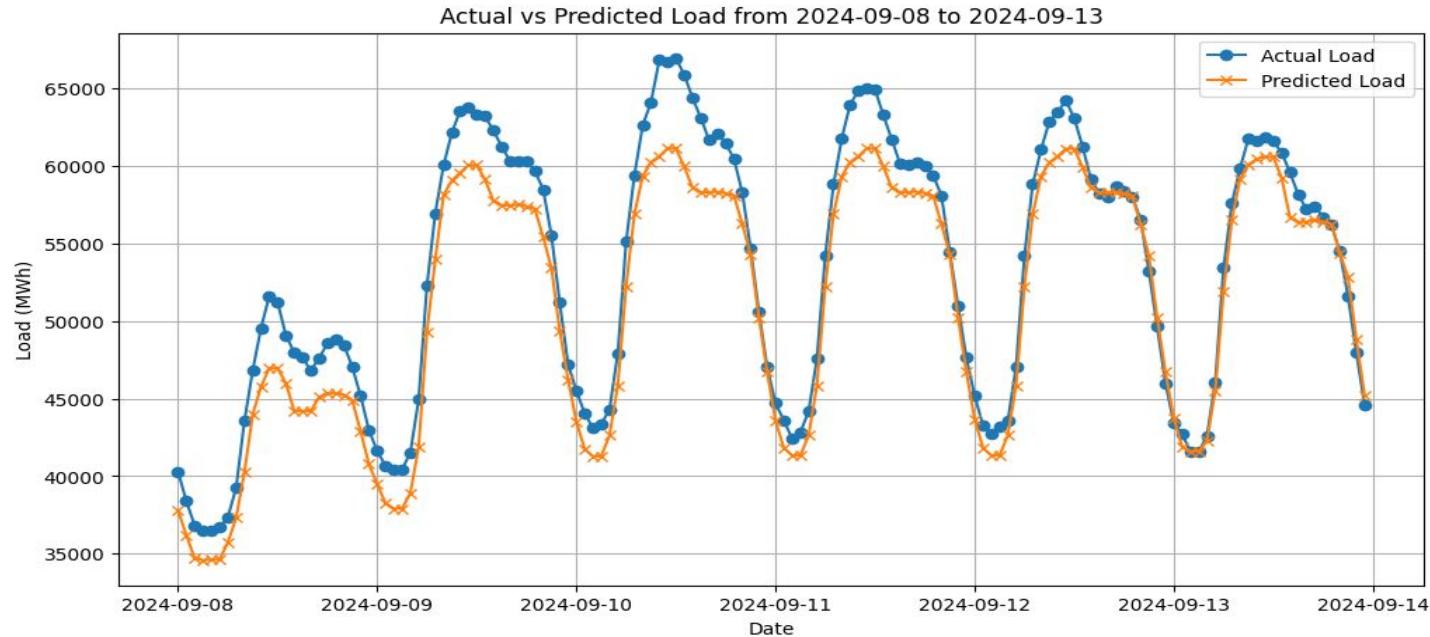
# LSTM:

- Architecture:
  - 3 layers
  - 128 neurons
  - 0.2 dropout layer
  - 48 window size
- Test Set MAPE: 1.85%
- Test Set MAE: 957.33





# Univariate XGBoost:

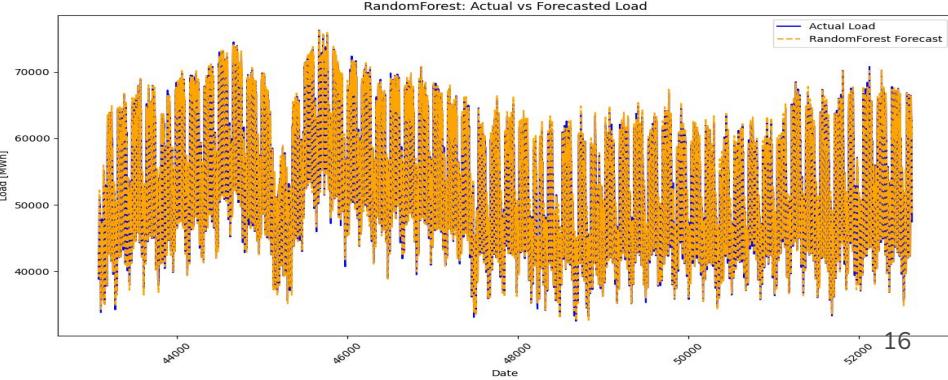
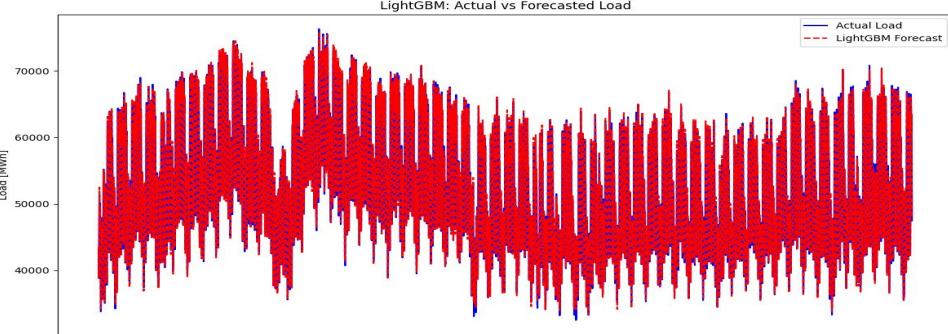
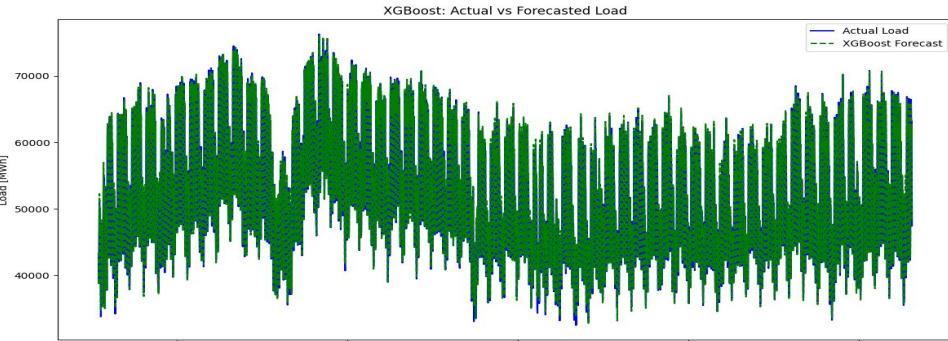


Mean Absolute Percentage Error (MAPE): 3.8462% Mean Absolute Error (MAE): 1988.07



# Multivariate Tree Models:

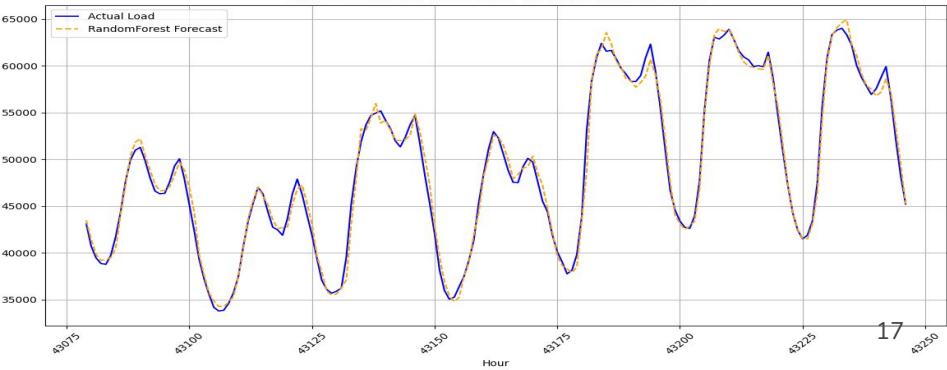
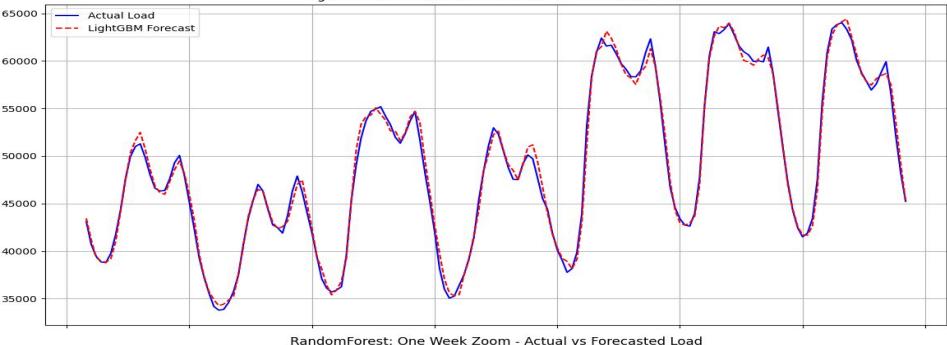
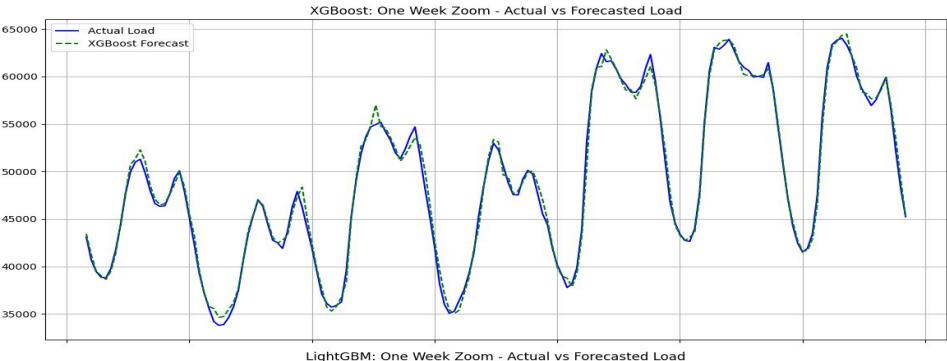
- XGBoost
  - 1000 estimators
  - 30 max depth
  - learning rate 0.01
- LightGBM:
  - 3000 estimators
  - 35 max depth
  - learning rate 0.05
- RandomForest;
  - 30 estimators
  - 25 max depth





# Multivariate Tree Models:

- XGBoost
  - mape: 0.86%
  - mae: 489.51
- LightGBoost:
  - mape: 1.09%
  - mae: 450.10
- RandomForest;
  - mape: 0.89%
  - mae: 515.27

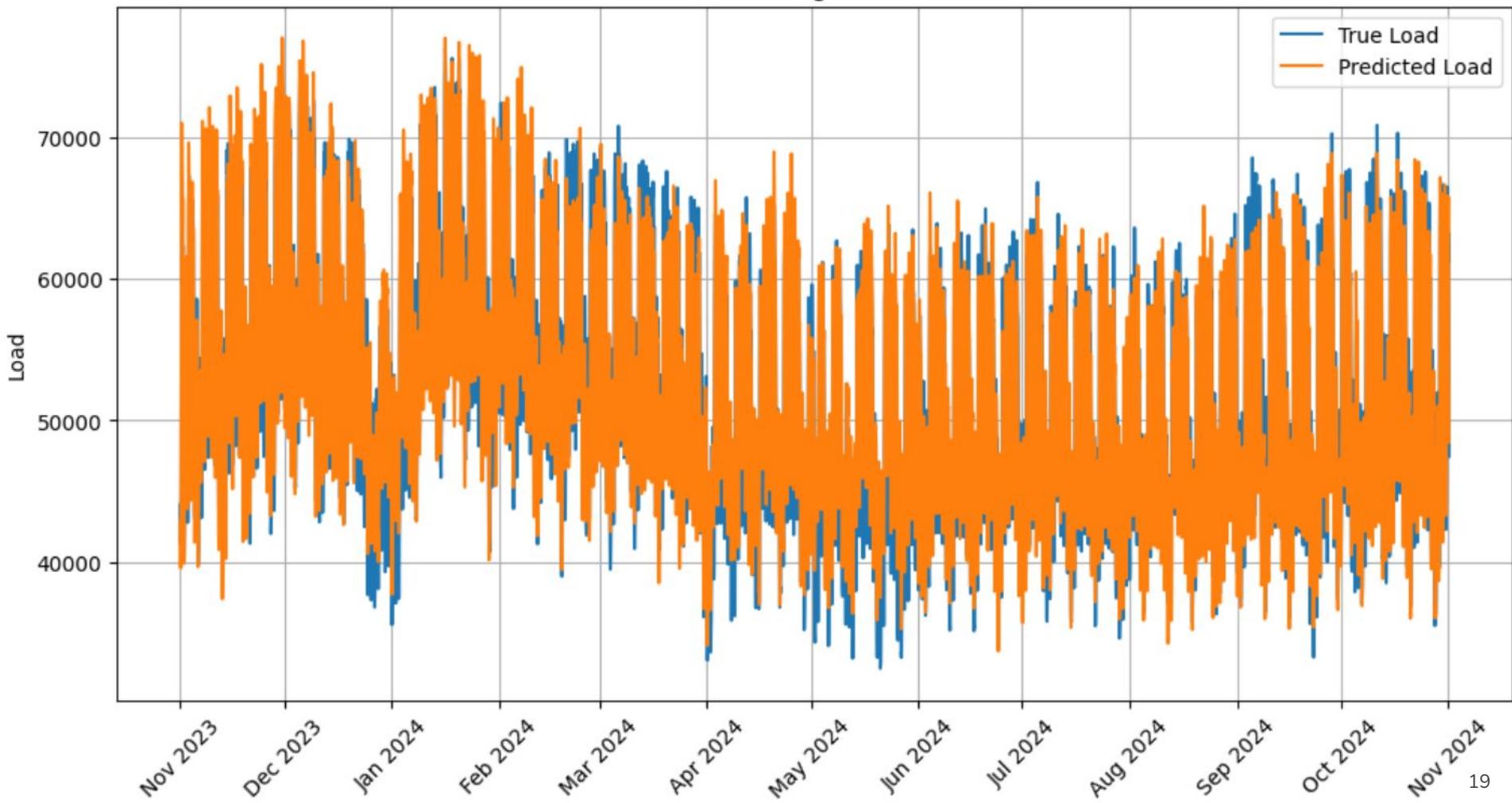




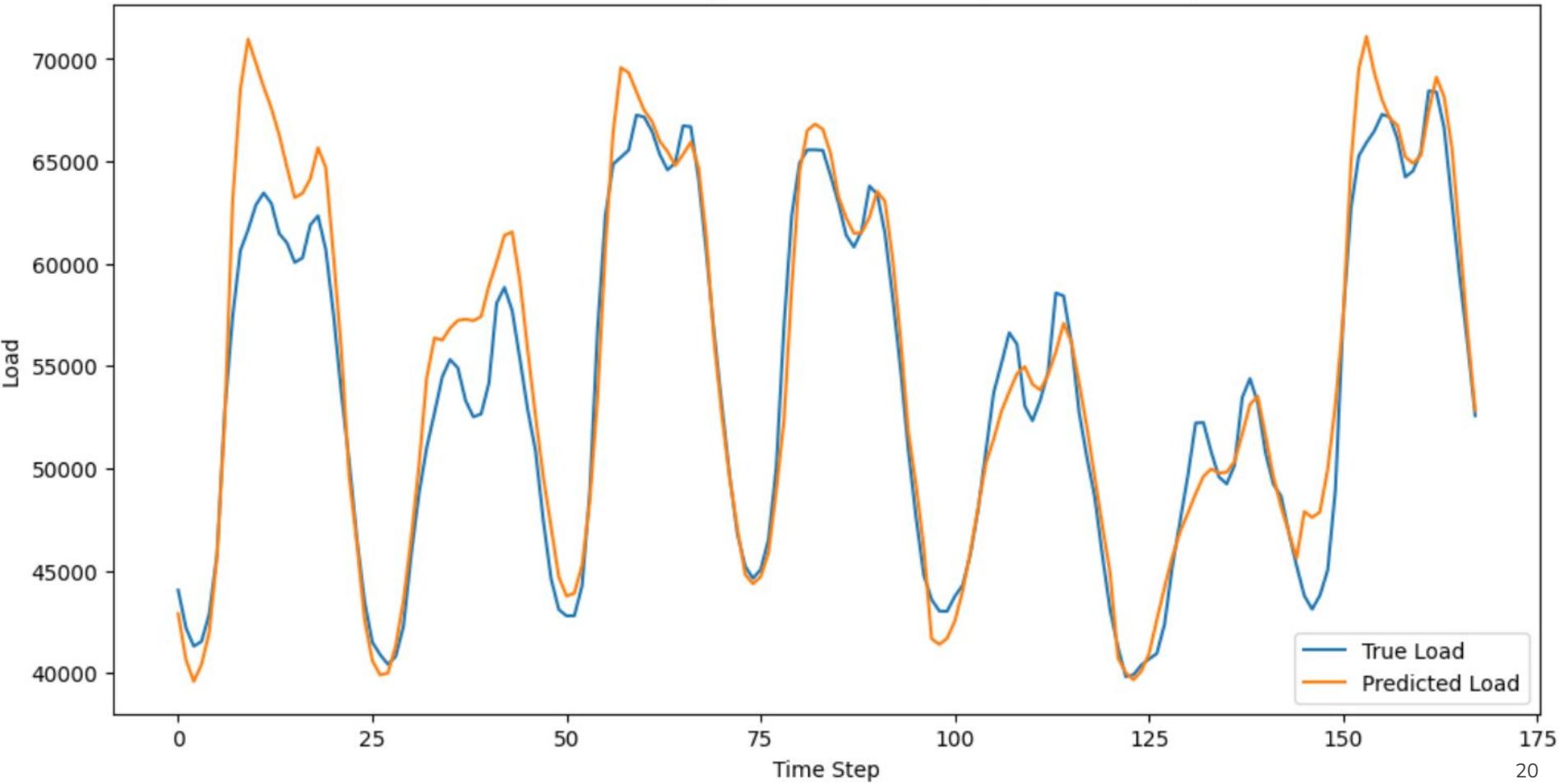
# Transformer Model

- TensorFlow
- 34,718 parameters (all trainable)
- 4 attention heads
- batch size = 1
- 50 epochs
- training time: 1:04 h
- MAPE: 3.69 %
- MAE: 1,898.06

## Load Prediction using Transformer Model



Load Prediction using Transformer Model (First 168 Values)

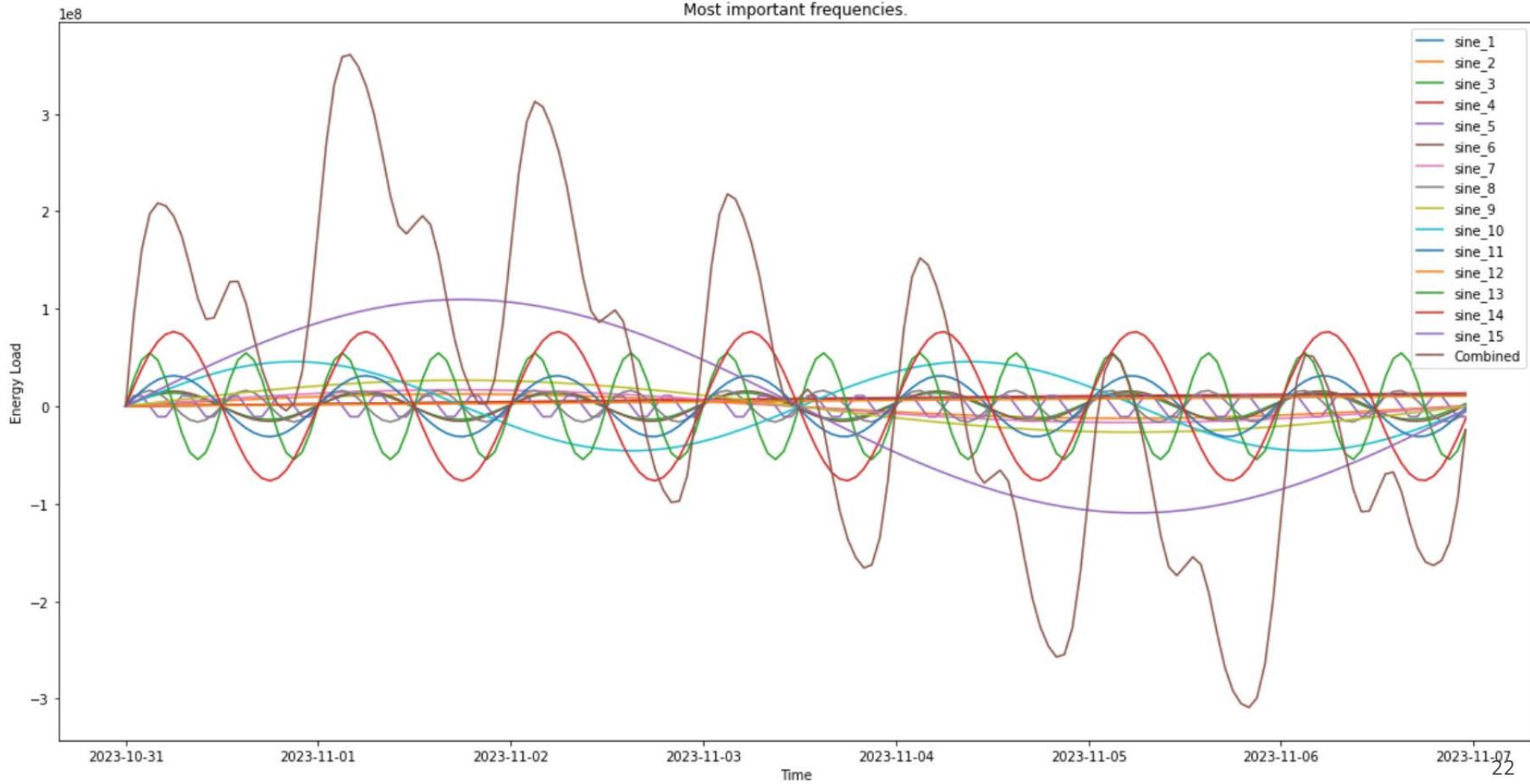




# FFT and Spectral Analysis

- Perform FFT
- Filter results to obtain most relevant frequencies
- Represent those frequencies with their amplitudes (and phases) as sine waves
- Add the sine waves as new columns to the dataset
- Perform SARIMAX with those columns as exogenous factors
- MAPE: 15.78 %
- MAE: 7,634.24
- Quick computation time (a few minutes for FFT, spectral analysis and SARIMAX)

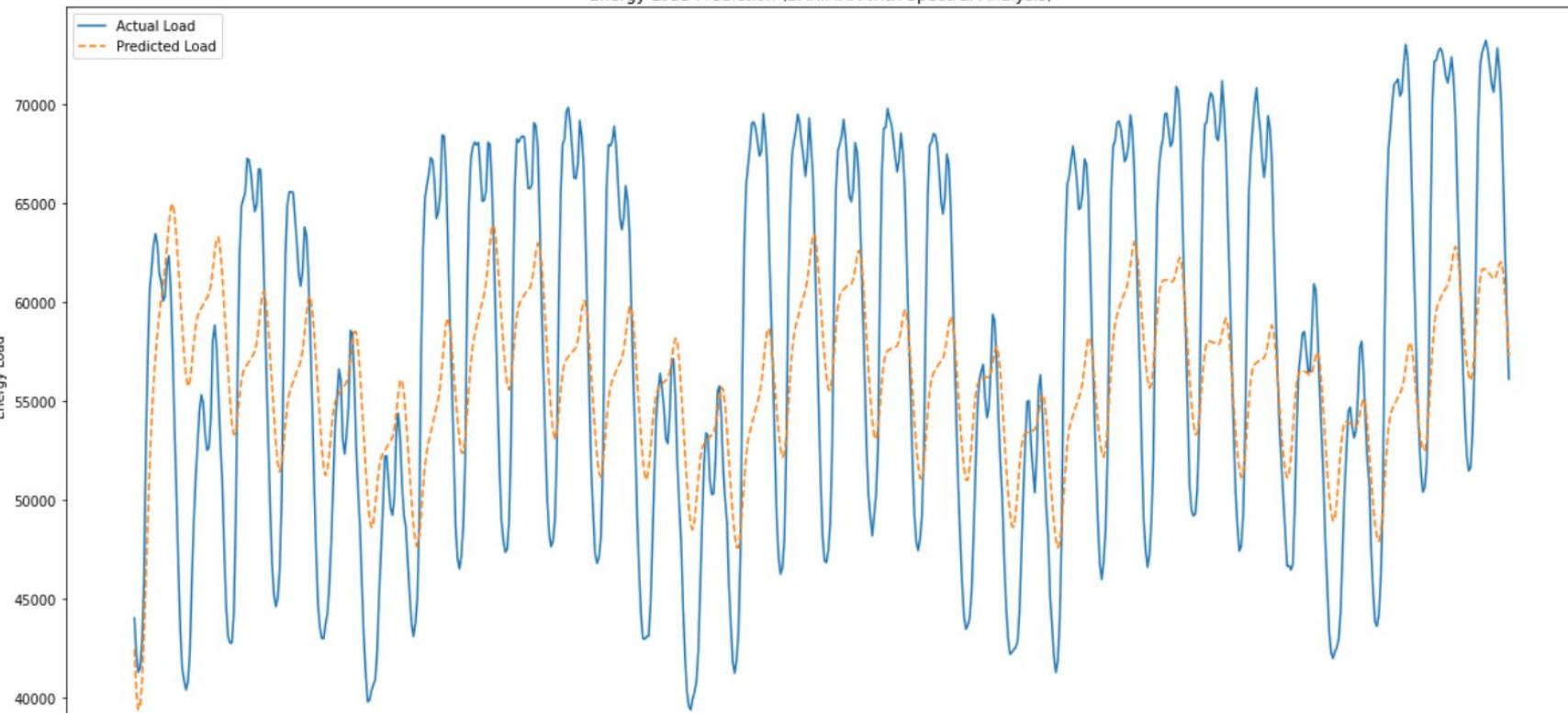
### Most important frequencies.



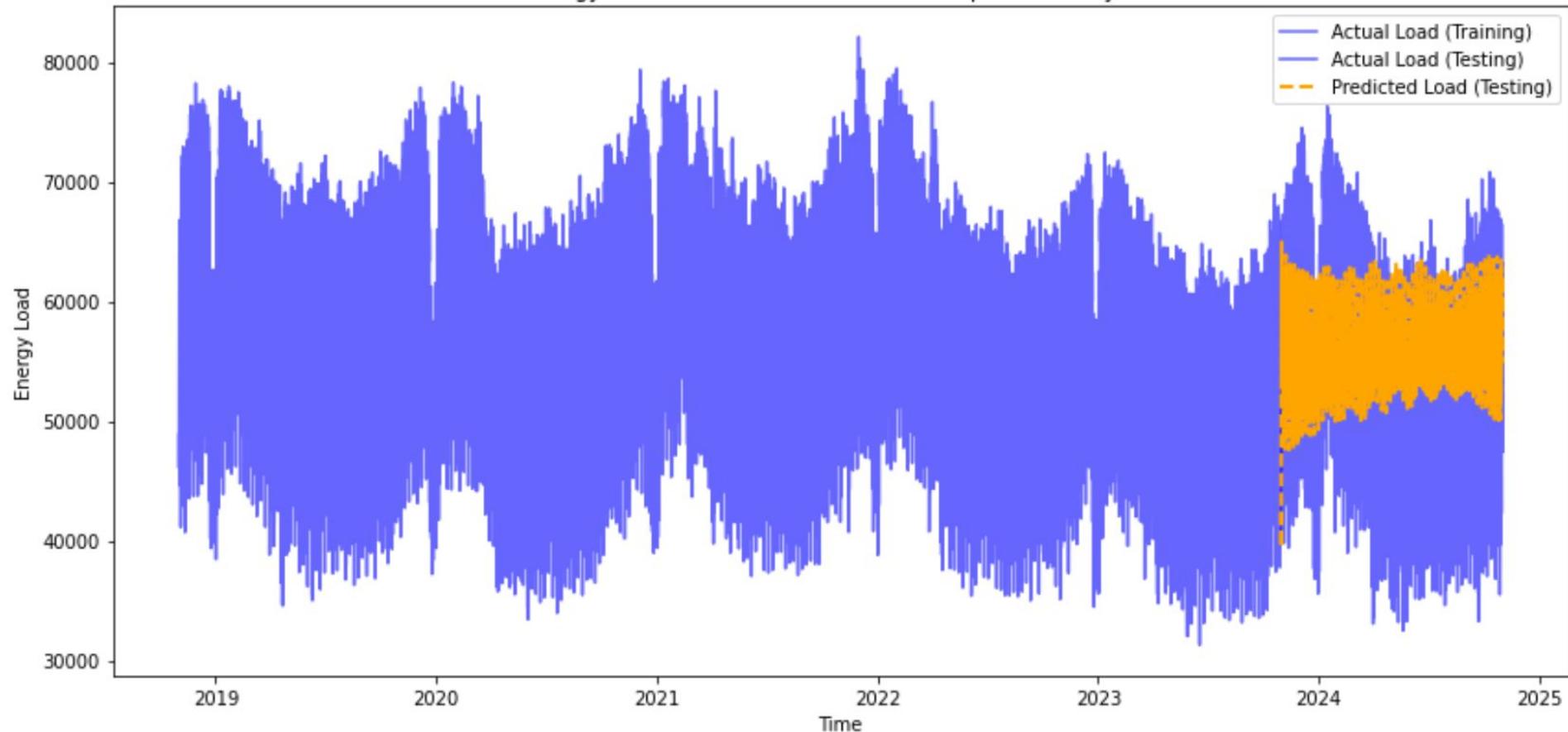


# SARIMAX with FFT

Energy Load Prediction (SARIMAX with Spectral Analysis)



Energy Load Prediction (SARIMAX with Spectral Analysis)





# Recall the main 3 types of Features

📌 Lag-Based Features (e.g., load\_lag\_1, load\_lag\_2, ..., load\_lag\_7):

- ❖ Historical Load Features (Lag Variables)
- ❖ External Market and Weather Features
- ❖ Basic Temporal (Time-Based) Features

📌 Time-Based and Categorical Features:

- ❖ Holiday and Workday Features e.g., is\_workday
- ❖ Expanded Calendar Features (Time Representation) e.g., date, hour, dayofweek
- ❖ Cyclical Features (Fourier Transformed Time Variables) e.g., hour\_sin

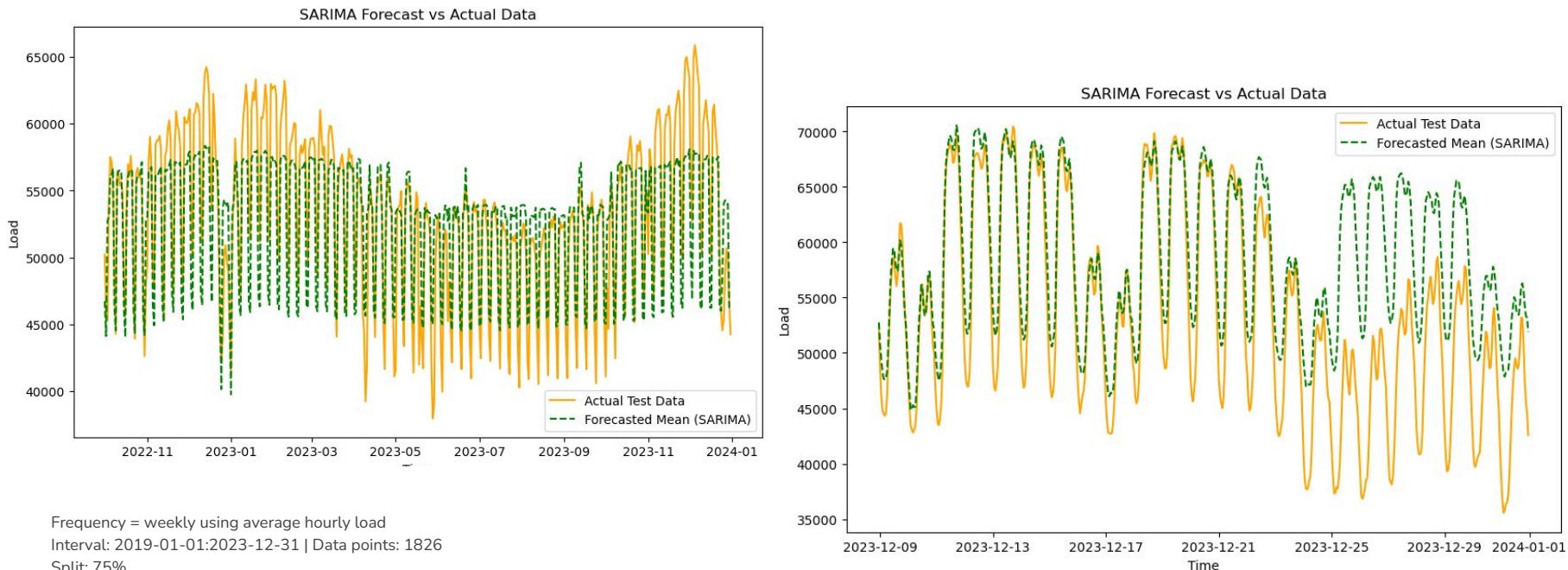
📌 Energy Source-Based Features(e.g.,, ):

- ❖ Electricity Market Features e.g., Gesamt (Netzlast) [MWh]
- ❖ Energy Generation-Based Features e.g., Biomasse [MWh]
- ❖ Renewable Energy Sources e.g. Biomasse [MWh]
- ❖ Conventional Energy Sources e.g. Kernenergie [MWh]
- ❖ Grid Demand (Net Load) Features



## SARIMAX 2: weekly load forecast with hourly and daily data

- ❖ Does model accuracy improve with higher frequency data ?
- ❖ Regressors: hour, is\_weekend, is\_holiday, is\_low, price, temperature, is\_high\_load, load\_time\_temp
- ❖ Hyper parameters: ARIMA components: AR(p), MA(q), d
- ❖ Hyper parameters: Seasonal Components: P, Q, D



Frequency = weekly using average hourly load

Interval: 2019-01-01:2023-12-31 | Data points: 1826

Split: 75%

Model: SARIMAX(4, 1, 0)x(1, 0, [1], 7)

Heteroskedasticity (H): 0.55

AIC: 833.944

MAPE : 0.048

Runtime: 94.254 seconds

Frequency = 24x7

Interval: 2023-10-01:2023-12-31 | Data points: 2160

Split: 75%

Model : SARIMAX(0, 0, 3)x(1, 0, [1], 168)

Heteroskedasticity (H): 0.31

AIC: -2532.929

MAPE: 0.1061

Runtime: 12m59s

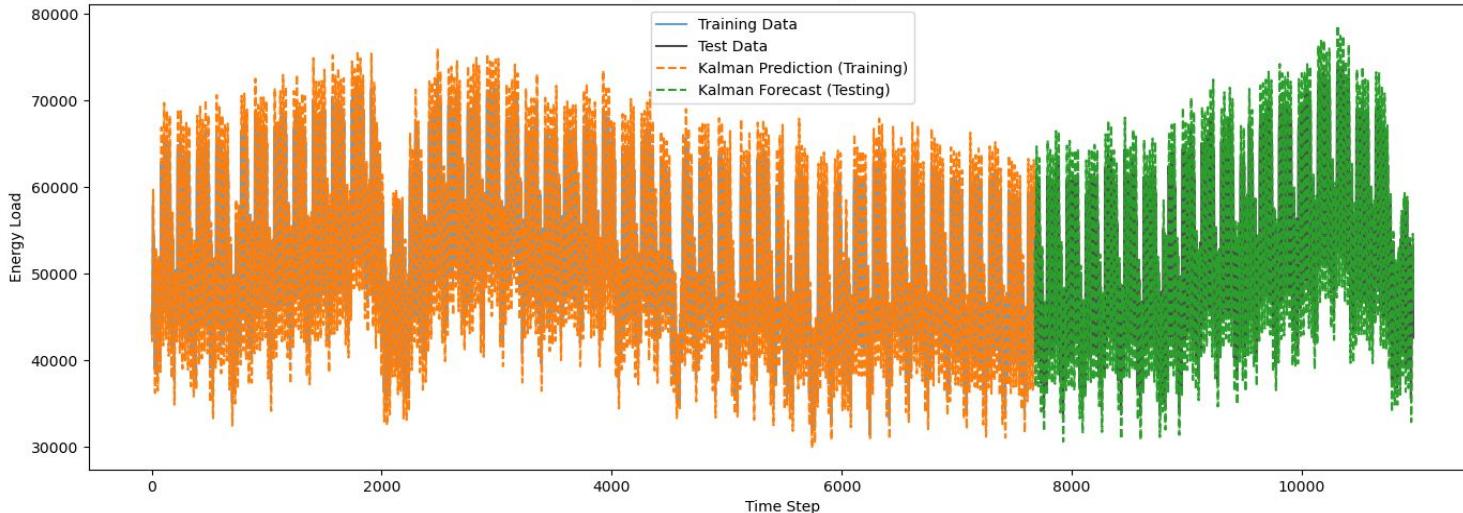


# Kalman Filter with expansion window: Hourly load Forecast

- ❖ Regressors: hour, is\_weekend, is\_holiday
- ❖ State variables: short-term trend & seasonality, long-term trend & seasonality
- ❖ Hyperparameters: state transition matrix (A), Observation matrix (H), Process noise covariance (Q), Observation noise covariance (R ), Initial state vector, Initial state covariance

```
for t in range(forecast_steps):  
    # Predict step  
    state_pred = A @ state_prev # Predict the next state  
    P_pred = A @ P_prev @ A.T + Q # Predict the state covariance  
  
    # Predict energy load (observation)  
    energy_load_pred = H @ state_pred  
    forecasted_energy_load.append(energy_load_pred[0])  
  
    # Optional: Expanding or Rolling window correction  
    if t < len(test_data):  
        # Use expanding window: update with test data as it becomes available  
        observation_residual = test_data[t] - energy_load_pred # Residual  
        S = H @ P_pred @ H.T + R # Innovation covariance  
        K = P_pred @ H.T @ np.linalg.inv(S) # Kalman Gain  
  
        # Update states based on test observation  
        state_updated = state_pred + K @ observation_residual  
        P_updated = (np.eye(n_states) - K @ H) @ P_pred  
    else:  
        # For rolling window: Use the rolling window states  
        state_updated = state_pred  
        P_updated = P_pred
```

Energy Load Prediction Using Kalman Filter



Frequency = hourly

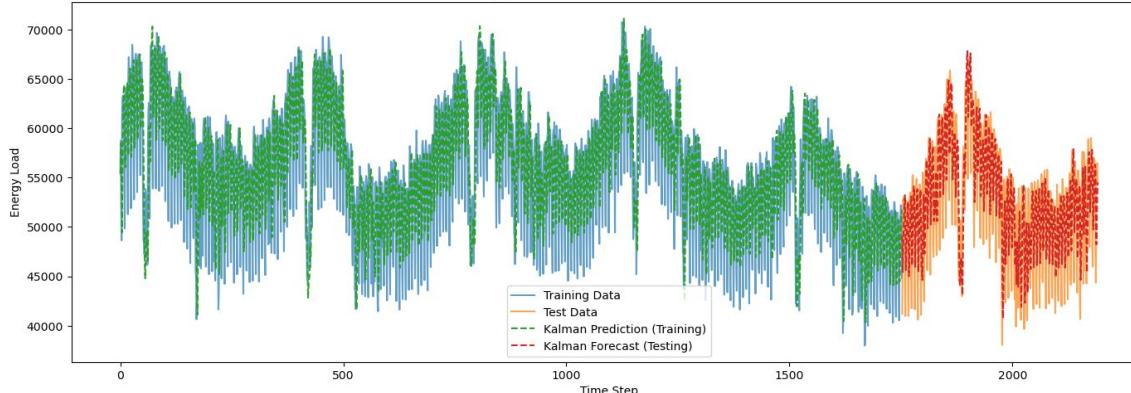
Interval: 2022-10-01:  
2023-12-31 | Data points:  
10969

split: 80%

MAPE: 0.0841

Runtime: 4 seconds

Energy Load Prediction Using Kalman Filter



Frequency = daily

Interval: 2018-11-01:  
2024-10-31 | Data points:  
2192

split: 80%

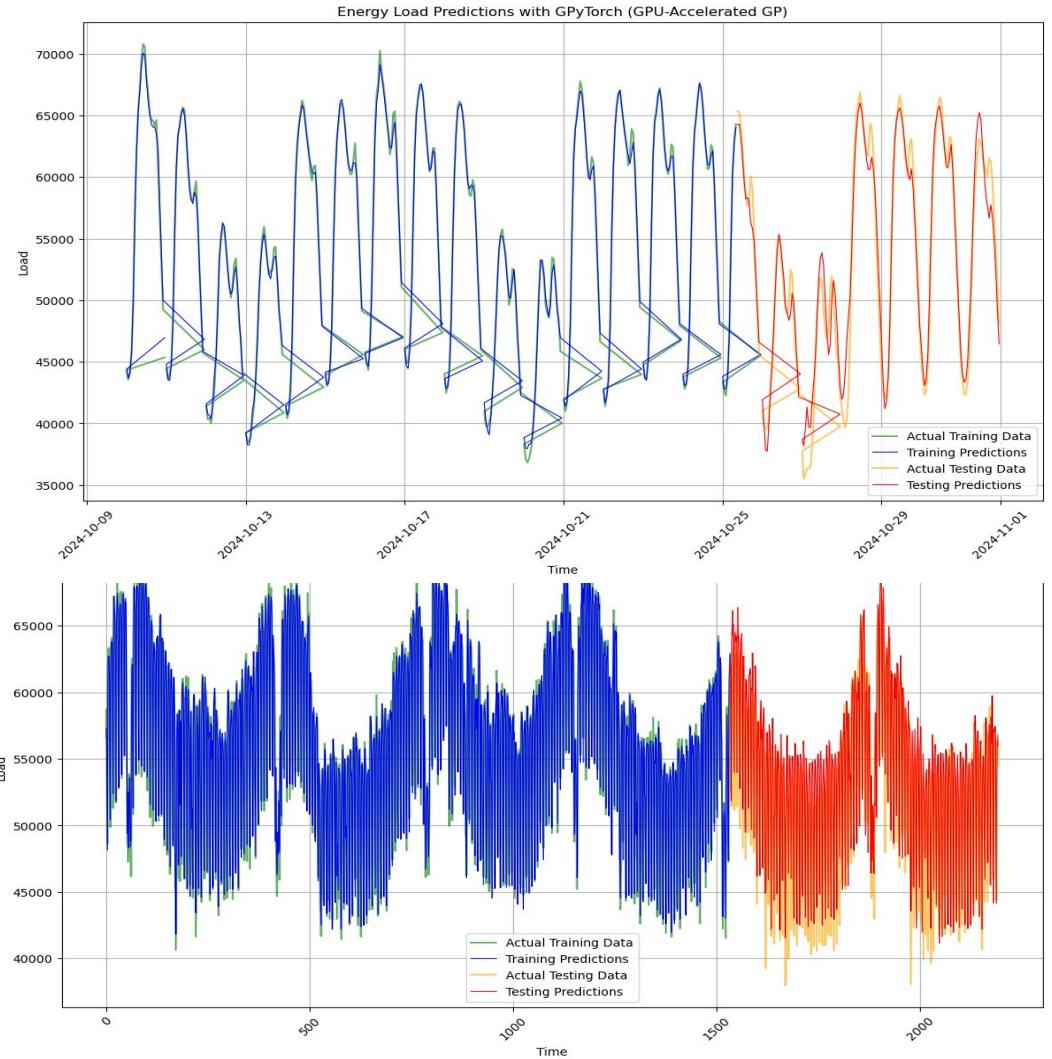
MAPE: 0.1007

Runtime: 4 seconds



# GP model: another advanced architecture

- ❖ Features: load\_lag\_1, load\_lag\_2, load\_lag\_3, load\_lag\_4, load\_lag\_5, load\_lag\_6, load\_lag\_7, price, temperature, day\_of\_week, month
- ❖ Hyperparameters:
  - Out layer: **Mean function**, **RBF kernel**, **Periodic Kernel**, **Matern Kernel**, **Gaussian Likelihood noise variance**, Learning rate, Number of iterations
  - In layer: length scale, periodic length, output scale
  - RBF Kernel models smooth trends over multiple features.
  - Periodic Kernels captures yearly and weekly seasonality (based on time-related features).
  - Matern Kernel adds flexibility for irregular variations.



Frequency = hourly

Interval: 2023-10-10:  
2024-10-31| Data points:  
8640

split: 70%

MAPE: [0.0068, 0.02795]

Runtime: < 1m

Loss: [1.562, 0.445]

Frequency = daily

Interval: 2019-01-01:  
2024-10-31| Data points:  
20100

split: 70%

MAPE: [0.0103, 0.0349]

Runtime: < 1m

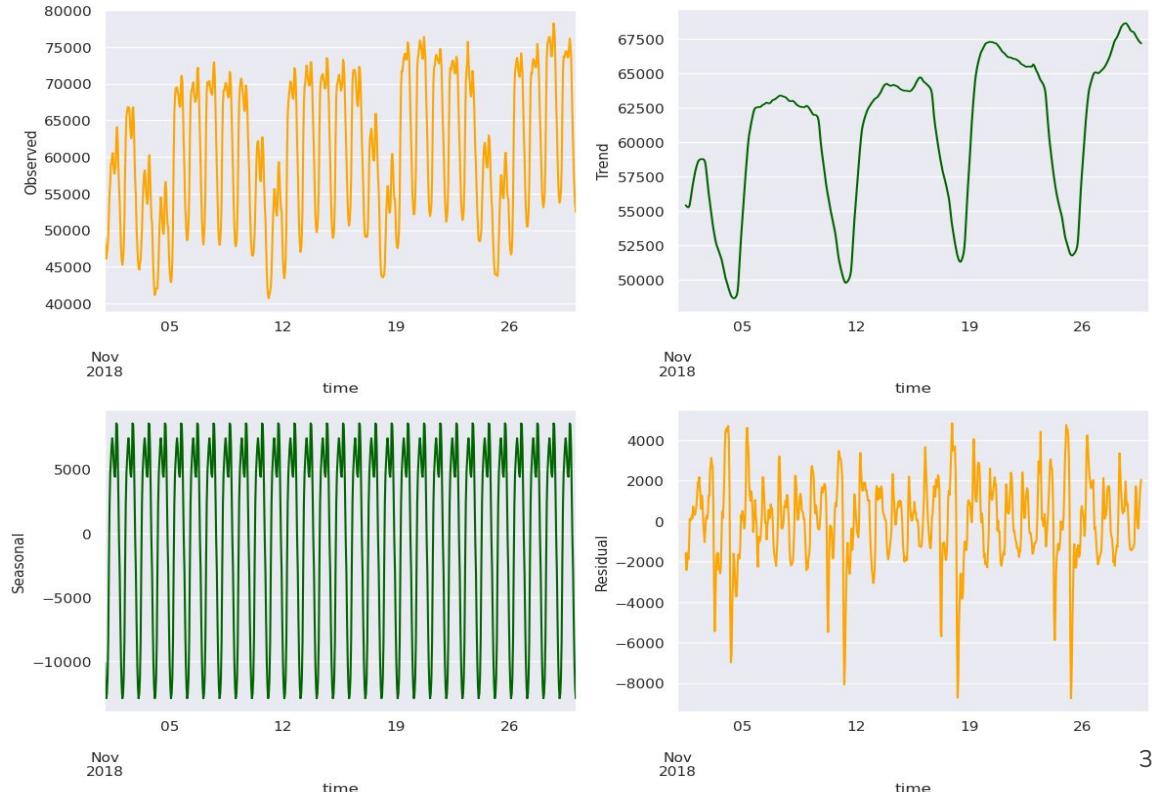
Loss: [1.552, 0.490]

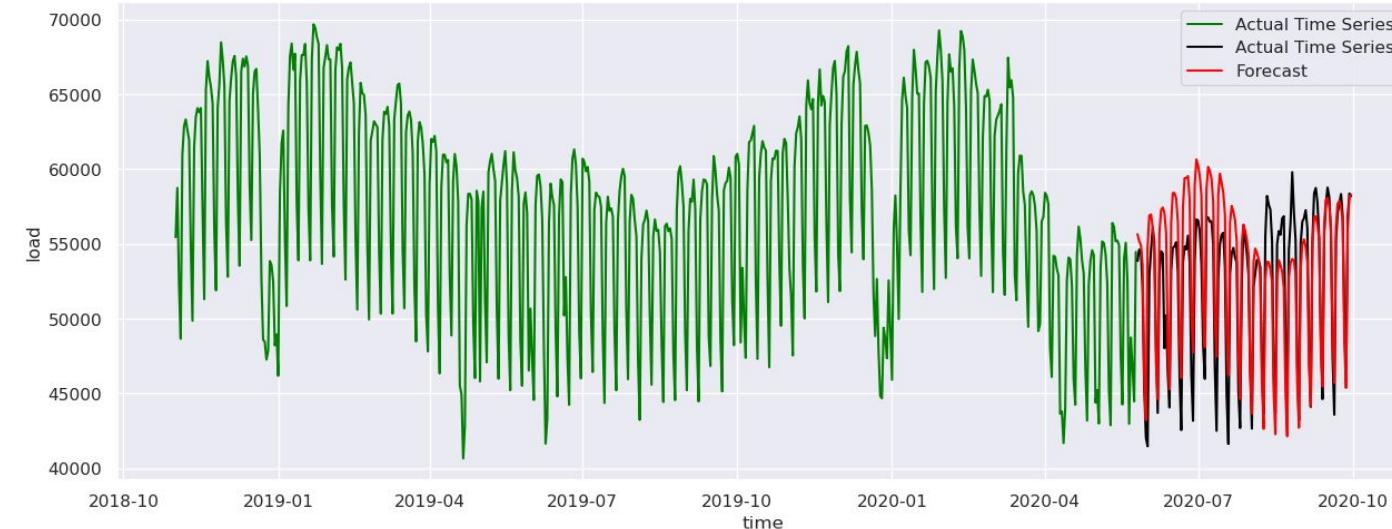
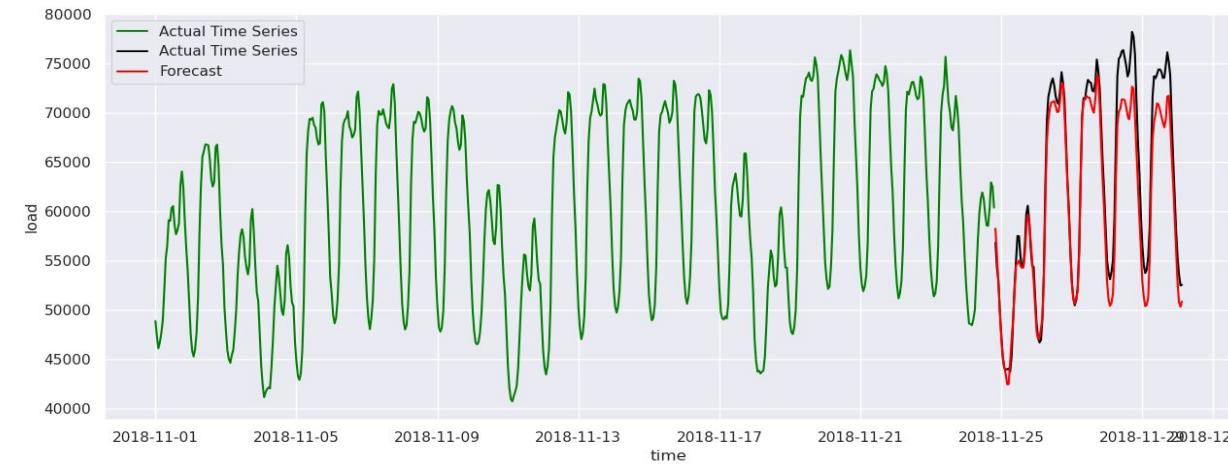


# Times FM at the edge

Hourly load decomposition into state components

- ❖ Features: no external features during forecast
- ❖ Hyperparameters:  
context\_len, horizon\_len,  
input\_patch\_len,  
output\_patch\_len,  
num\_layers, model\_dims,  
backend

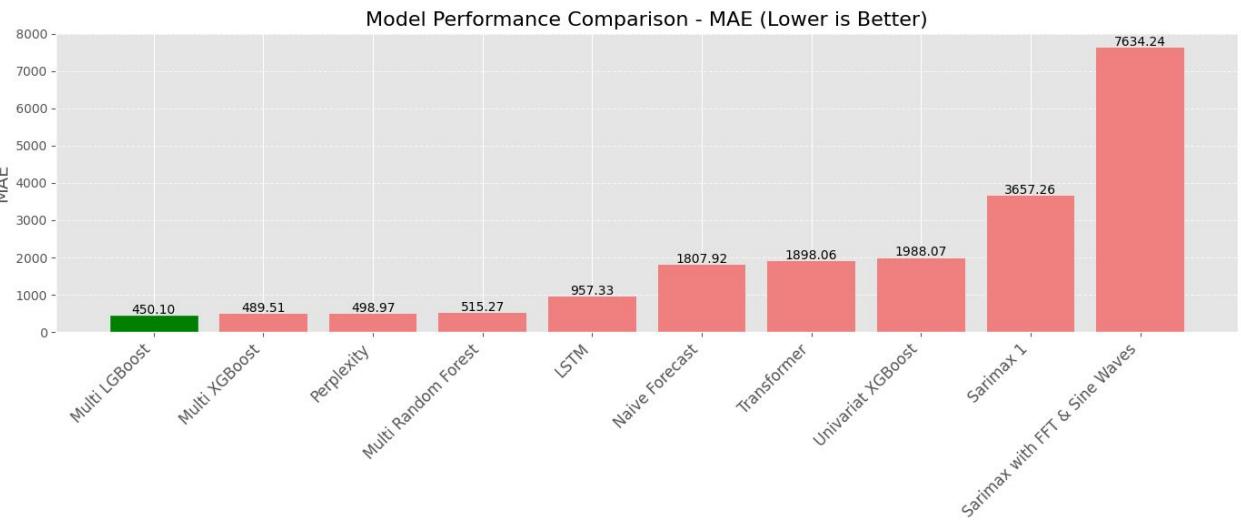
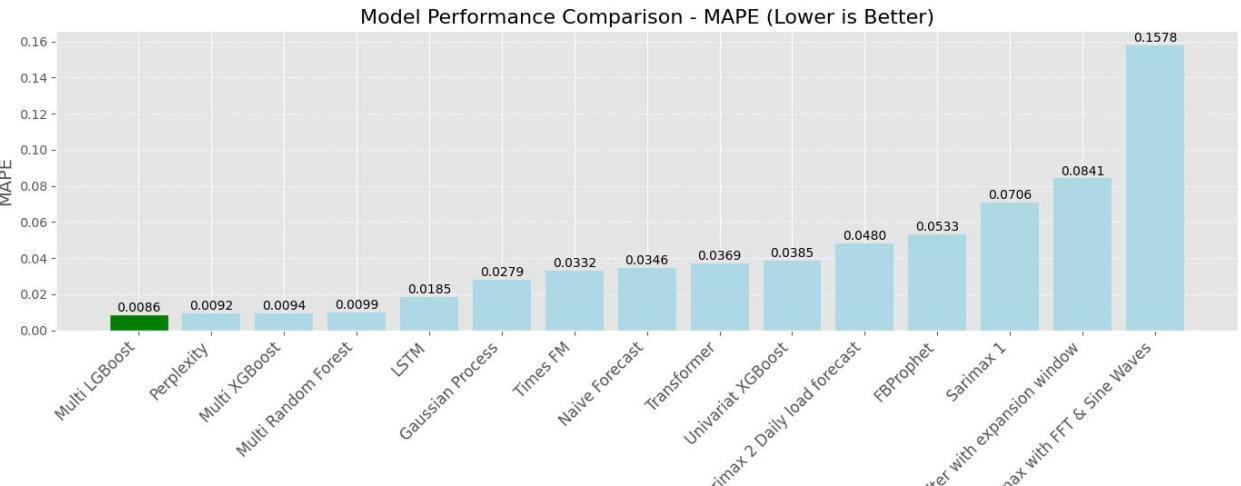




# Model Comparison

<u>Model</u>	<u>MAPE</u>	<u>MAE</u>	training time (estimated)
Naive Forecast	0.034592	1807.92	—
Sarimax 1	0.0706	3657.26	10 min
FBProphet	0.053319	---	4 min
LSTM	0.0185	957.33	20 min
Univariat XGBoost	0.038462	1988.07	4 min
Multi XGBoost	0.0094	489.51	10 min
Multi LGBoost	<b>0.0086</b>	<b>450.10</b>	3 min
Multi Random Forest	0.0099	515.27	7 min
Transformer	0.0369	1898.06	1 hour
Sarimax with FFT & Sine Waves	0.1578	7634.24	2 min
Sarimax 2 Daily load forecast	0.048	—	1m34s
Kalman Filter with expansion window	0.0841	—	<b>4s</b>
Gaussian Process	(0.02795)	—	<1 min
Times FM	0.0332	—	< 1 min
Perplexity	0.0092	498.97	< 1 min

# Final Model Comparison





# Outlook:

- Add more Features
- More fine tuning for each model
- Use models optimized for rolling forecasts
- Forecast Energy price as well
- Forecast different regions
- Consider official and unofficial regional holidays (carnival) and the regions affected
- Consider long “bridge” weekends
- Reproduce the models with alternative set of features
- Develop an algorithm to charge/discharge Batteries efficiently



# References

- <https://www.smard.de/home/downloadcenter/download-marktdaten/?downloadAttributes=%7B%22selectedCategory%22:1,%22selectedSubCategory%22:false,%22selectedRegion%22:false,%22selectedFileType%22:false%7D>
- International Energy Agency (2024): Batteries and Secure Energy Transitions.  
<https://iea.blob.core.windows.net/assets/cb39c1bf-d2b3-446d-8c35-aae6b1f3a4a0/BatteriesandSecureEnergyTransitions.pdf>
- Lewinson, E. (2022): Three Approaches to Encoding Time Information as Features for ML Models.  
<https://developer.nvidia.com/blog/three-approaches-to-encoding-time-information-as-features-for-ml-models/>



# Photo credit

Slide 3, top center:

<https://unsplash.com/de/fotos/luftaufnahme-des-rasenplatzes-mit-blauen-sonnenkollektoren-llpf2eUPpUE>

Slide 3, right:

<https://pixabay.com/de/photos/windrad-energie-windkraft-5267130/>

Slide 4, top center:

[https://www.bundesnetzagentur.de/SharedDocs/Bilder/\\_init\\_bild\\_stromleitung.png?\\_\\_blob=normal&v=3](https://www.bundesnetzagentur.de/SharedDocs/Bilder/_init_bild_stromleitung.png?__blob=normal&v=3)

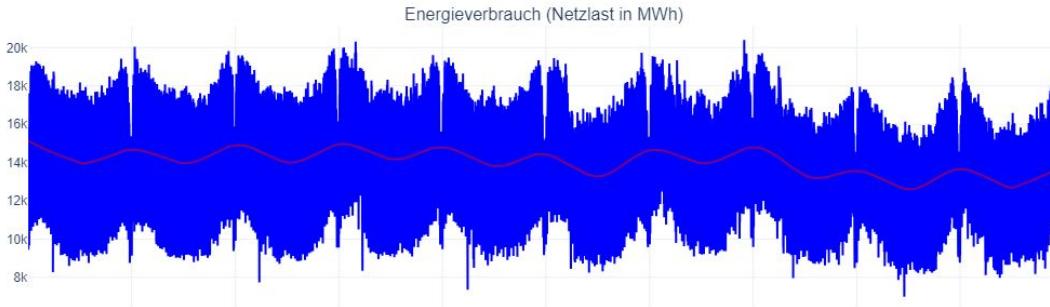
Slide 4, right:

<https://unsplash.com/de/fotos/weisse-windmuhle-auf-gelbblattrigem-blumenfeld-tagsuber-E56cTF65xFw>

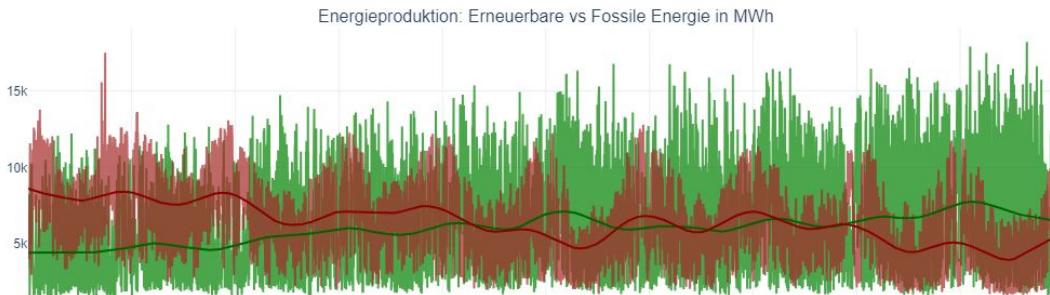
Abb. 1: Energieverbrauch, Erzeugung und Preisentwicklung in Deutschland (2015-2025)



Load



Production



Price:

