Energy Load Time Series Prediction

Tim Prause, Kadisatou Fane, Cosima Birkmaier

Contents

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

Dataset

- Load Dataset obtained from Bundesnetzagentur (Federal Agency for Grids)
- Total energy load from Germany and Luxembourg
- From 01.11.2018 to 31.10.2024
- Energy sources (Gas, wind, solar...)
- Hourly data
- Added features:
    - Time Features (hour, weekday, holidays...)
    - Energy Prices in Europe
    - Average Weather Temperature in Germany
- Final dataset:
    - 52,620 samples
    - Over 35 features in total

List of Features Used in the Different Models

📌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

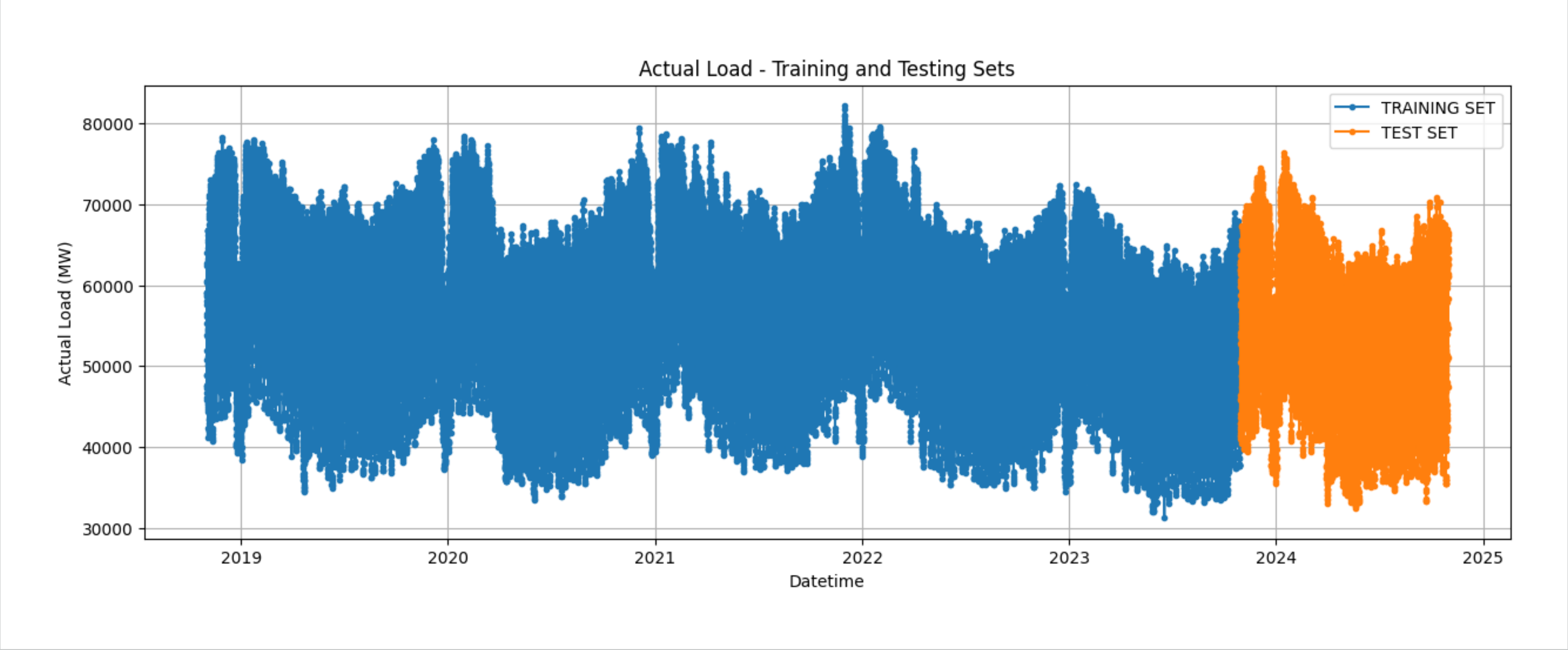
📌Lag-Based, Time-Based, External and Interaction Features:

- ❖ Lag-Based Features from historical data e.g., load_lag_1,..., load_lag_7
- ❖ Holiday and Weekday Features e.g., is_weekend, is_holiday
- ❖ Market and Weather Features e.g., price, temperature, load_time_temp

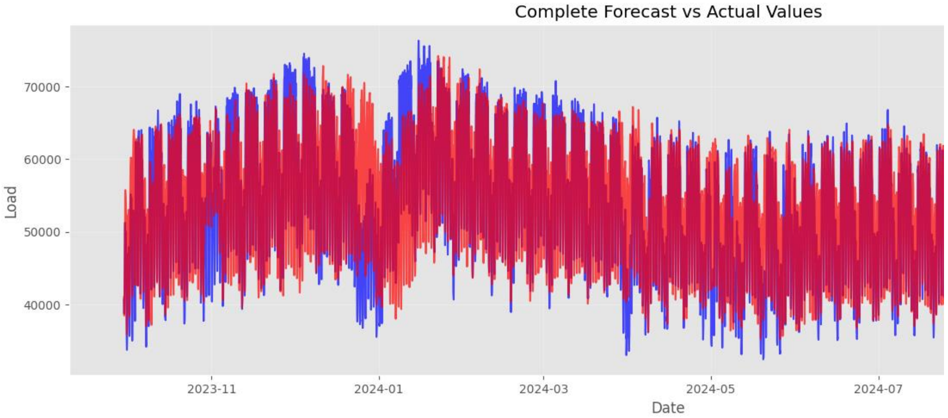
📌 Energy Source-Based Features(e.g.,,):

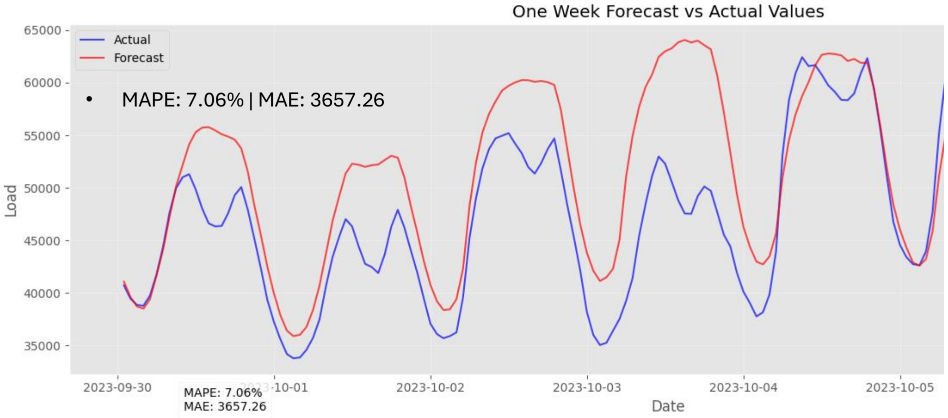
- ❖ Electricity Market Features e..g, Gesamt (Netzlast) [MWh]
- ❖ Energy Generation-Based Features e.g., Biomasse [MWh]
- ❖ Renewable Energy Sources e.g. Biomasse [MWh]
- ❖ Conventional Energy Sources e.g. Kernenergie [MWh]
- ❖ Grid Demand (Net Load) Features

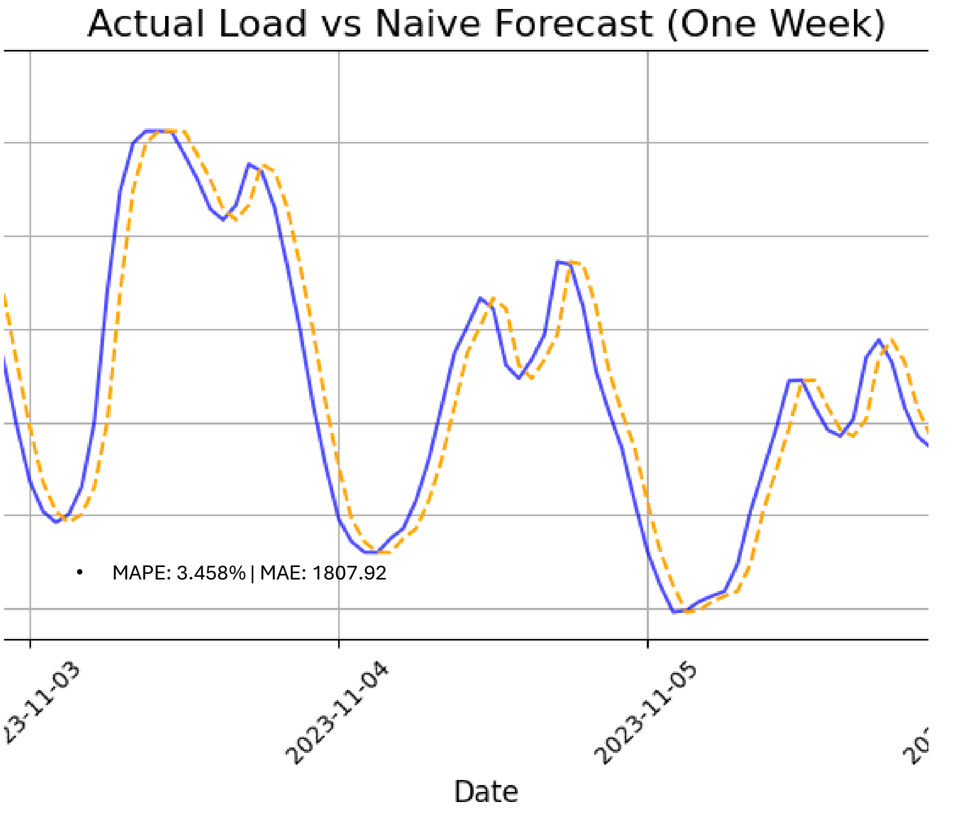
Train-Test split for all models:



Baseline Models: SARIMAX vs Naïve Forecast (one Week)

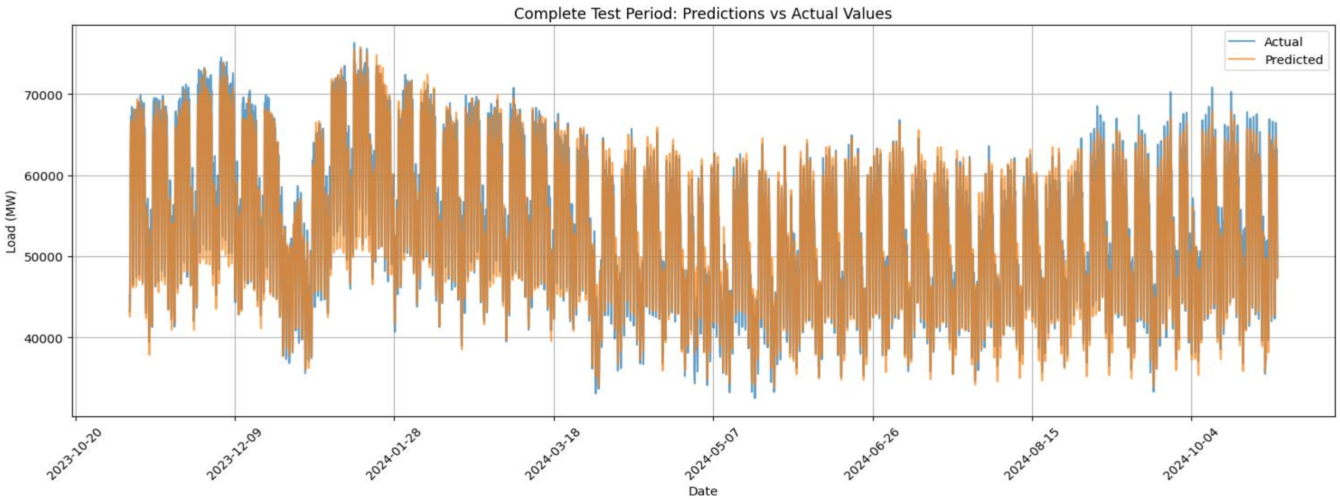


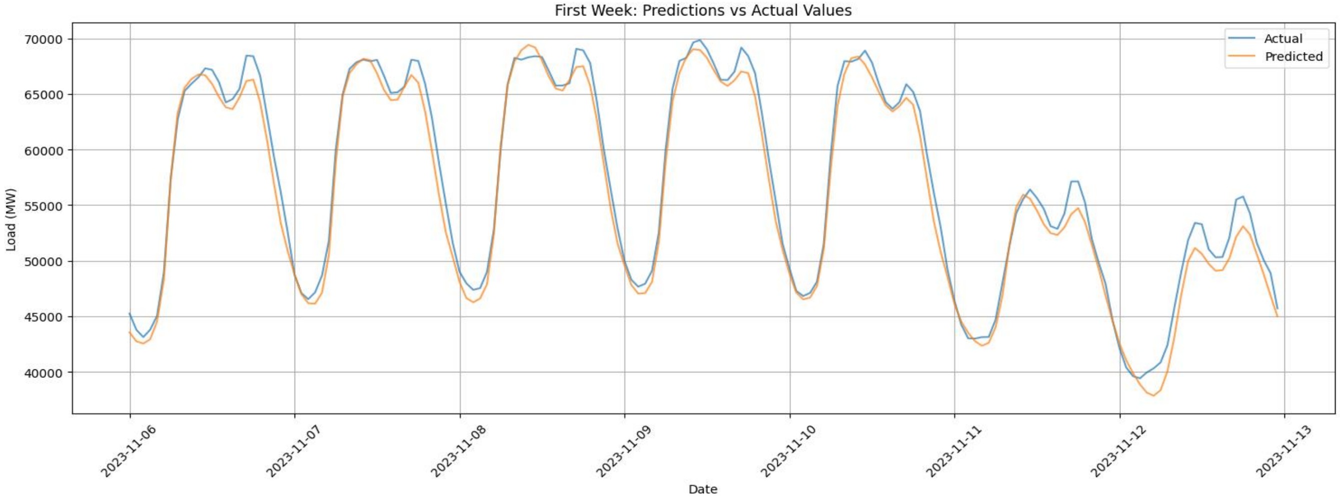




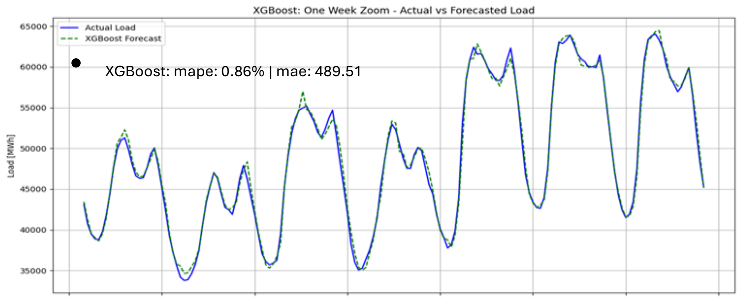
LSTM

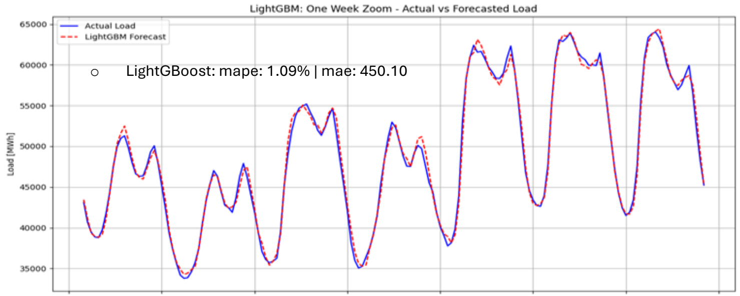
- Architecture:
  - 3 layers
  - 128 neurons
  - 0.2 dropout layer
  - 48 window size
- Test Set MAPE: 1.85%
- Test Set MAE: 957.33

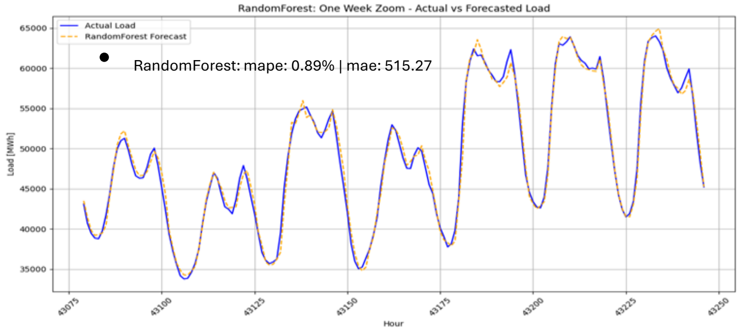


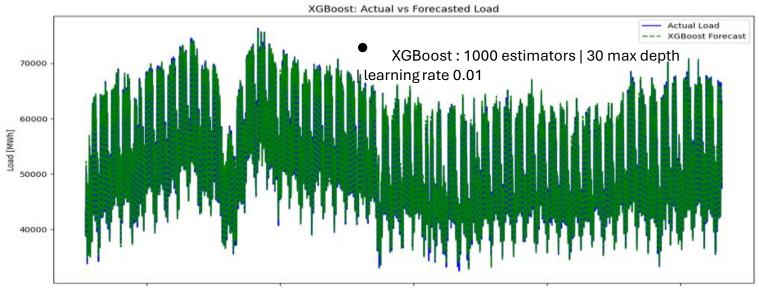


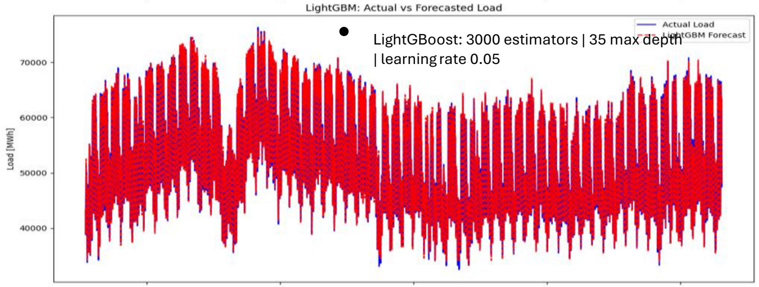
Multivariate Tree Models

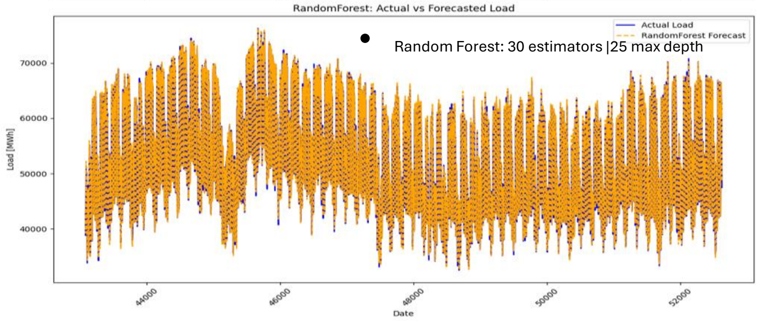






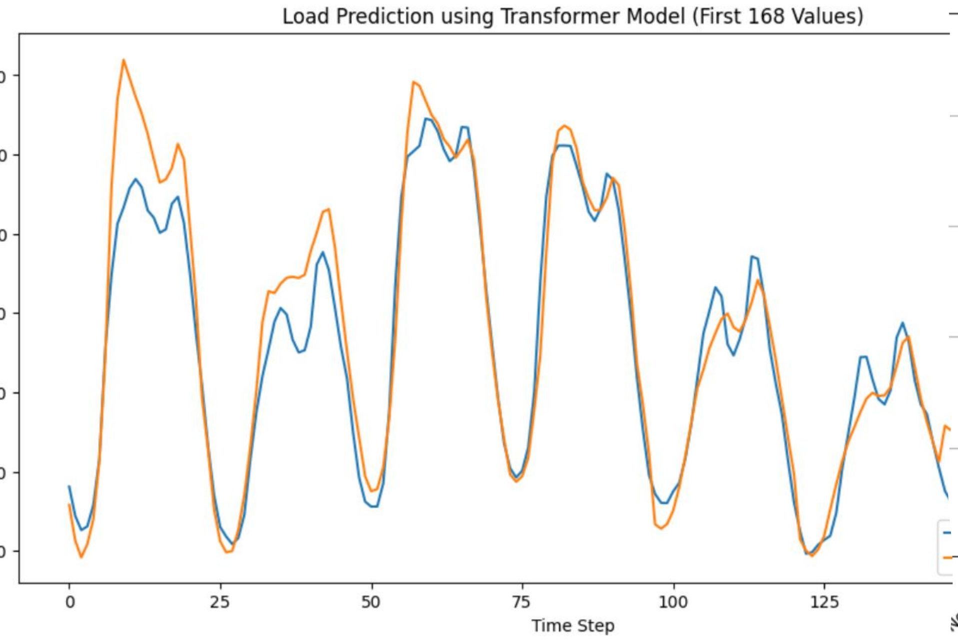


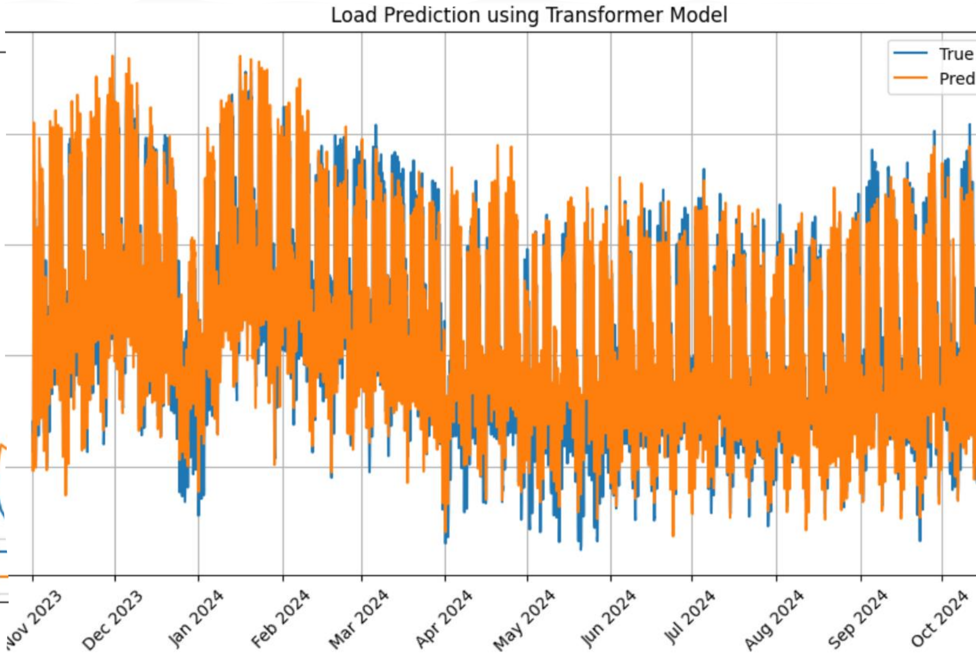


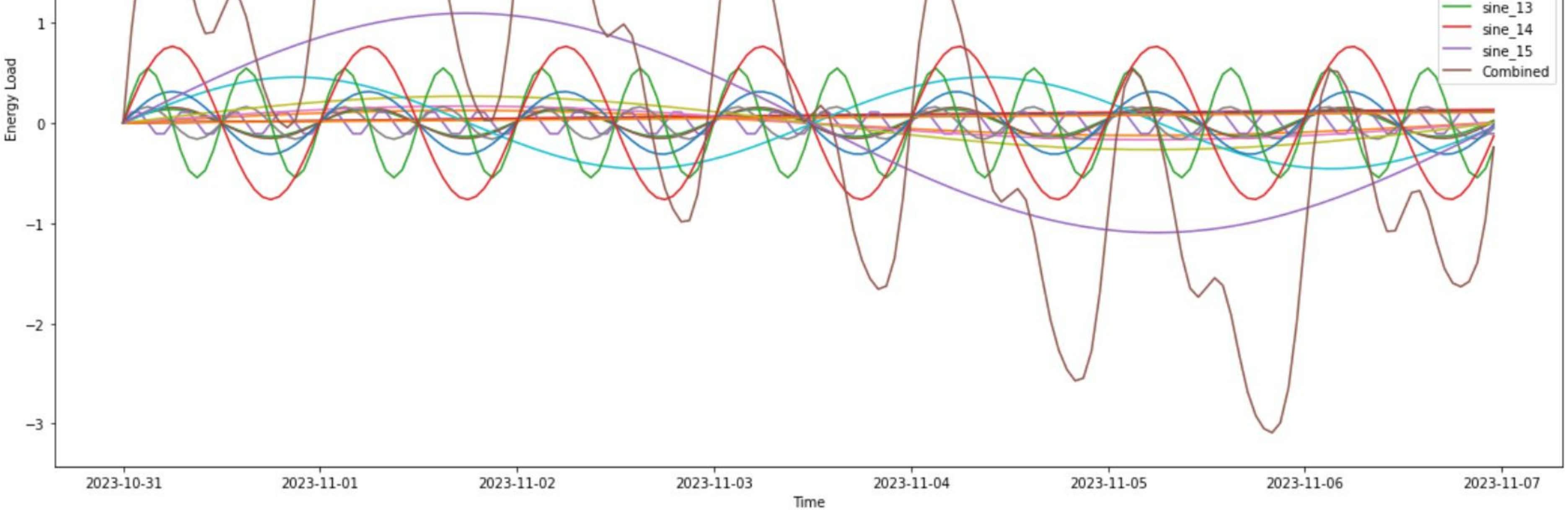


The Transformer Model

- 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

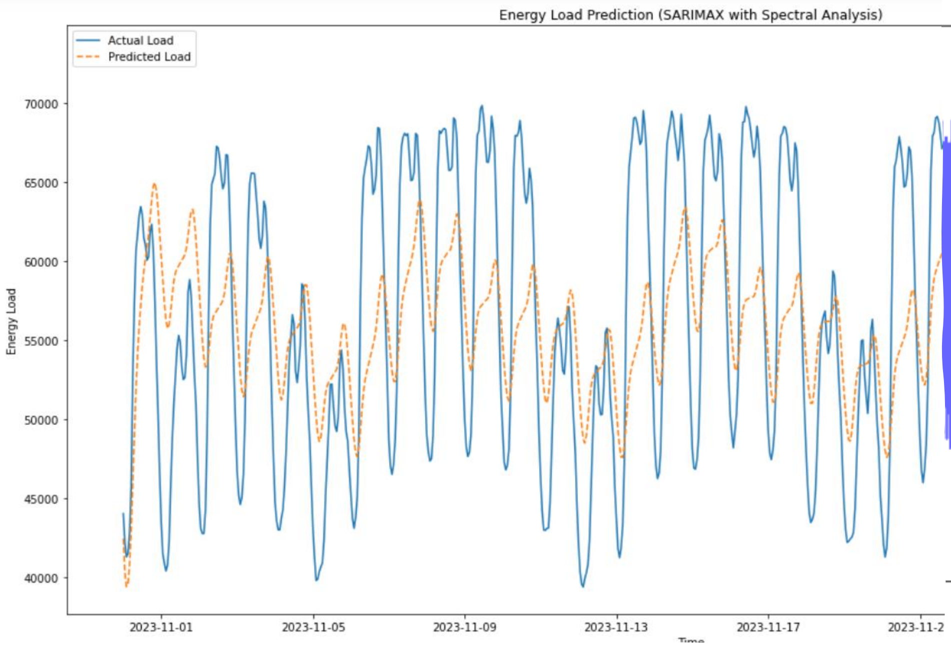


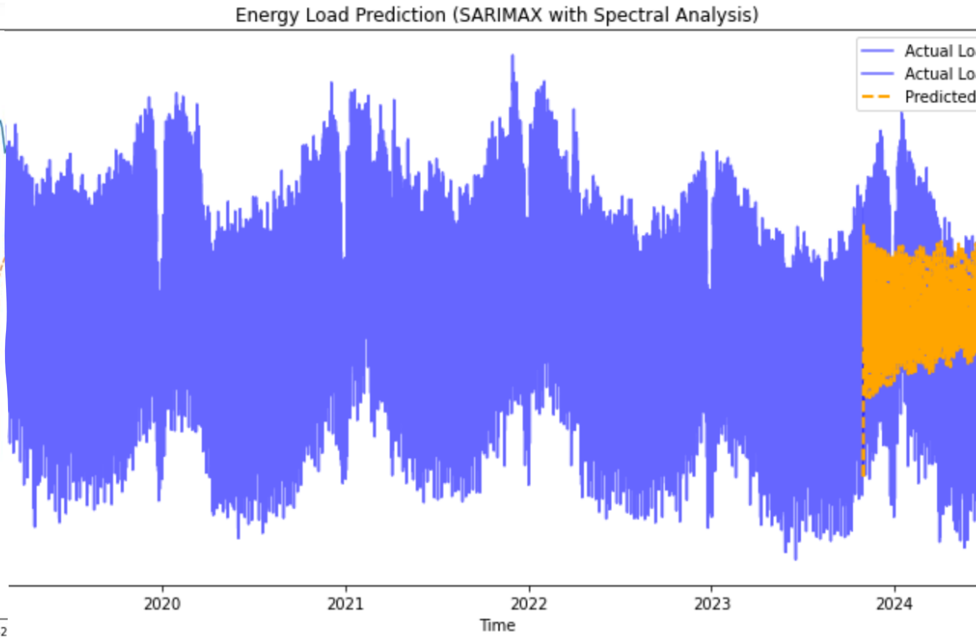




FFT and Spectral Analysis

- 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)

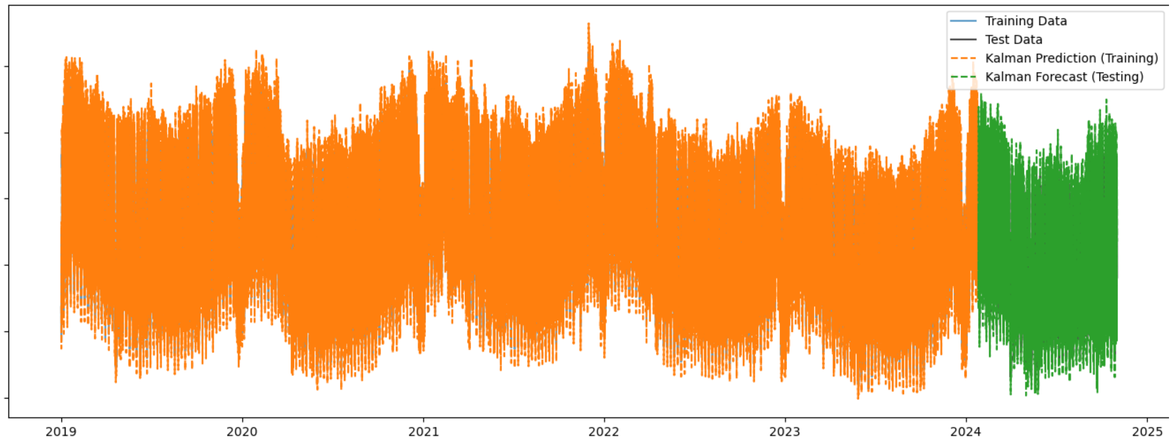




SARIMAX with FFT

- MAPE: 15.780% | MAE: 7634.2403

Kalman Filter with Expansion window

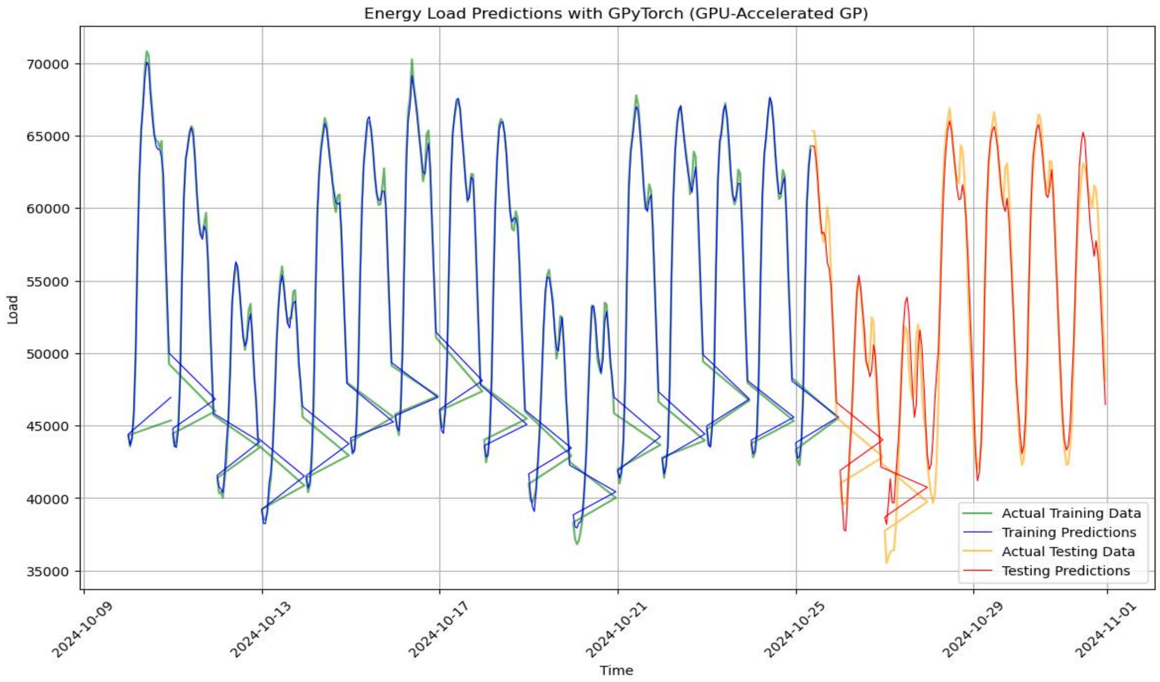


- Frequency: hourly | split: 80% | MAPE: 0.0817 | Runtime: 4 seconds

```
# 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
```

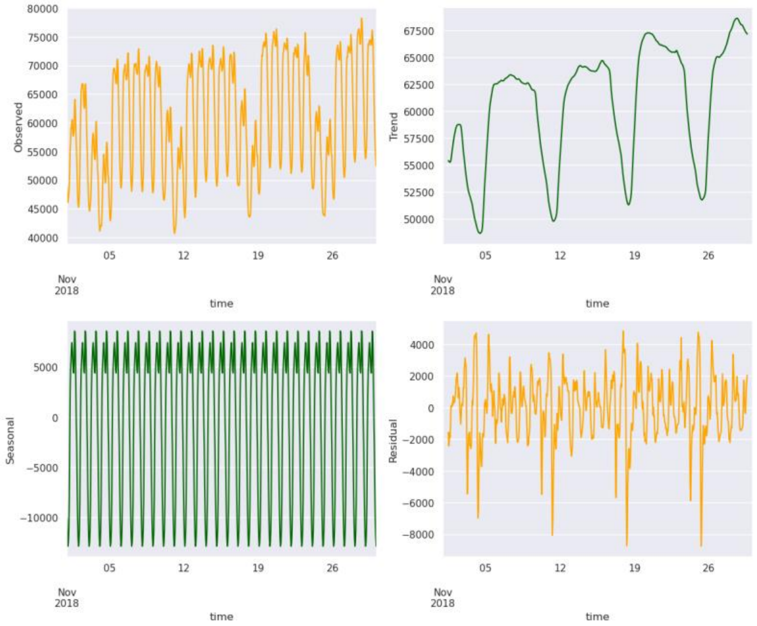
Gaussian Process: another advanced architecture

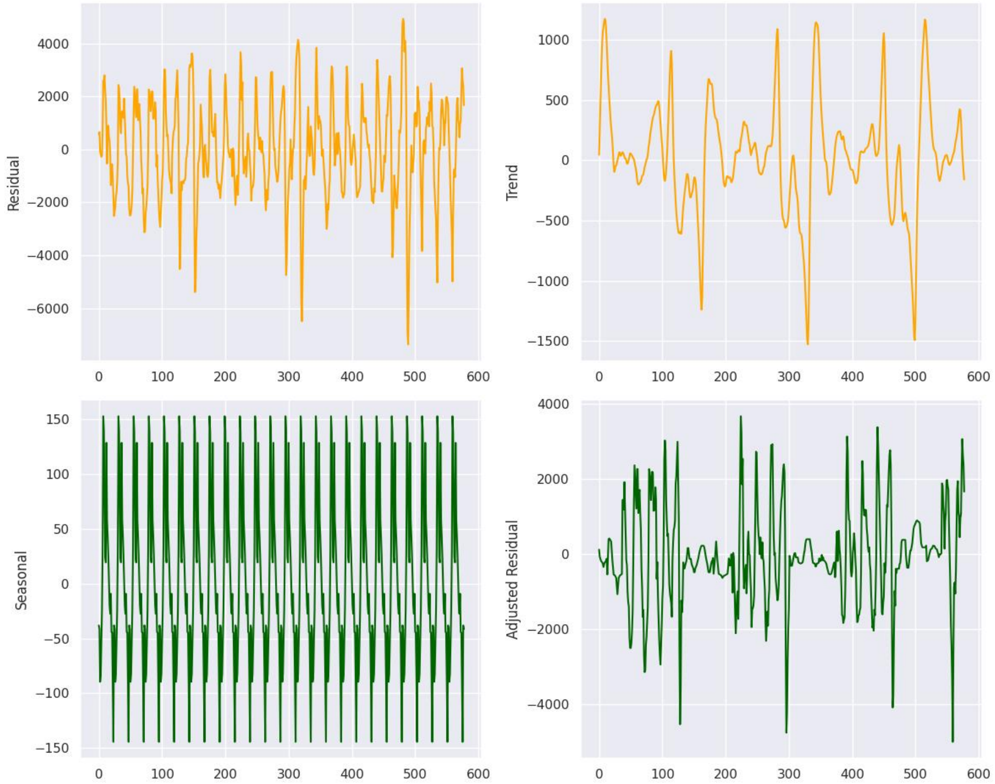
Frequency: Hourly | split: 70%
Loss: [1.562, 0.445] | MAPE: [0.0068, 0.02795]
N= 810 | Runtime: < 1m



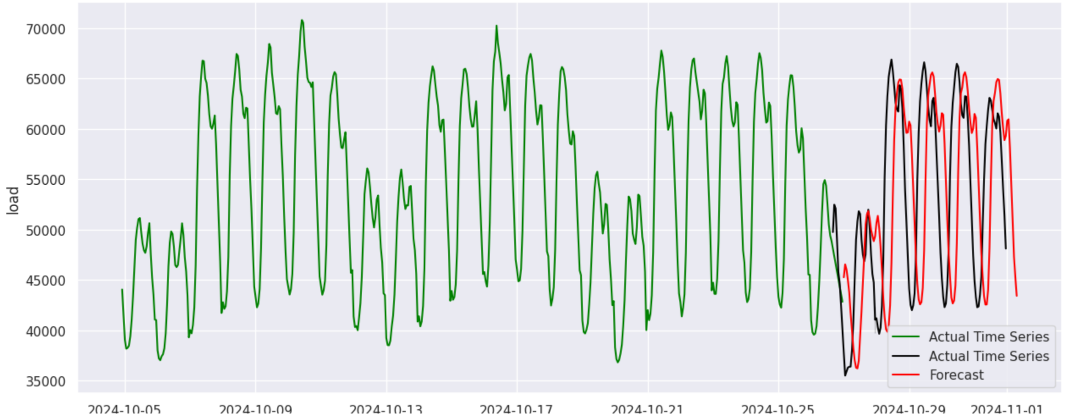
Load Decomposition with adjusted residual

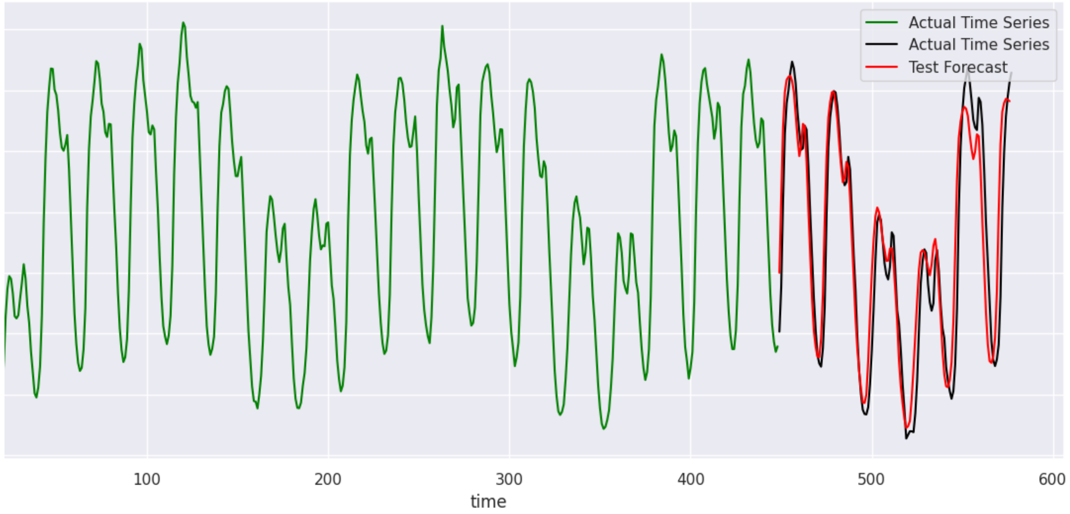
- Features: no external features during forecast
- Hyperparameters: context_len, horizon_len, imput_patch_len, output_patch_len, num_layers, model_dims, backend





Times FM: Load Forecast vs. Load Forecast with Partially Adjusted Residual





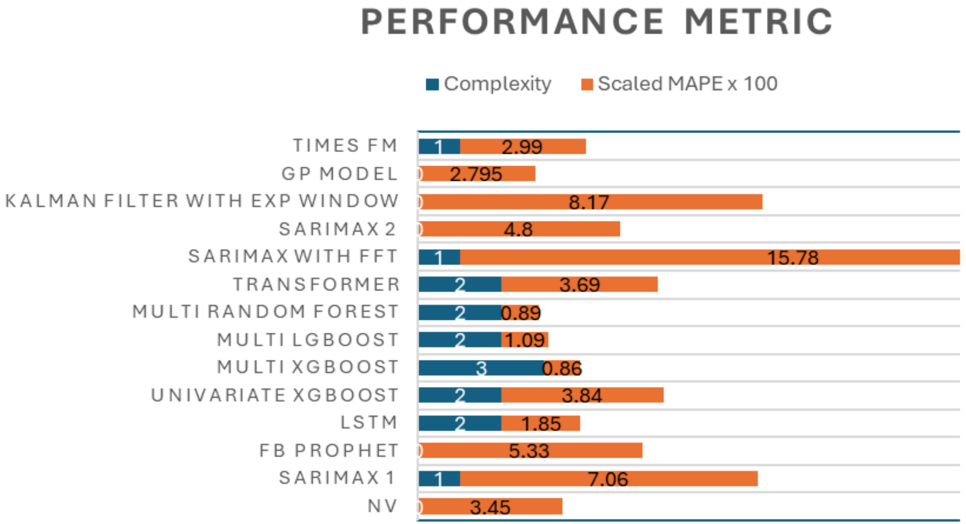
- Frequency : hourly | Interval: 20-11-1: 2028-11-29|
- Data points: 640 | Split :80%
- MAPE: [0.02991, 0.0785] | Runtime: < 1m

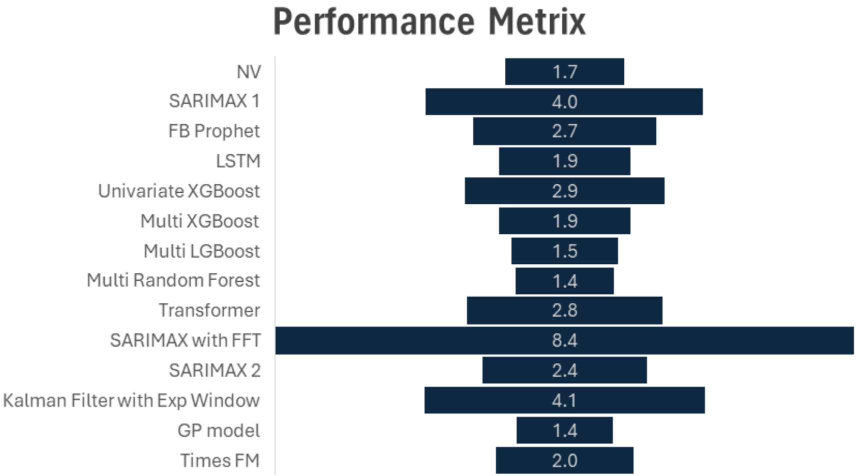
Model Comparison

Model	MAPE	MAE	Training + testing 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	0.0086	489.51	10 min
Multi LGBoost	0.0109	450.10	3 min
Multi Random Forest	0.0089	515.27	7 min
Transformer	0.0369	1898.06	1 hour
Sarimax with FFT & Sine Waves	0.1578	7634.2403	2 min
Sarimax 2 Daily load forecast	0.048	—	1m34s
Kalman Filter with expansion window	0.0817	—	4s
Gaussian Process	[0.0068, 0.02795]	—	<1 min
Times FM	[0.0795, 0.0299]	__	< 1 min

Best Performing Models at a glance

- 3 basic criterions: Runtime, Number of hyperparameters, Number of features.
- Runtime criterion: takes {0,1} based on whether model runtime is larger or smaller than the weighted average of runtime of the model
- complexity: the sum of the 3 criterion for each model {0,1,2,3}
- Performance: average of MAPE x 100, and complexity





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

References

- Energy load data from 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/
- “EEX-Hourly Spot Price E/Mwh” Data from Refinitiv Eikon
- Temperature data from https://open-meteo.com/en/docs/historical-weather-api#latitude=51.5&longitude=10.5&start_date=2018-01-01&end_date=2025-01-01&hourly=temperature_2m&daily=&models=

Photo credit

- Slide 3, top center:
- https://unsplash.com/de/fotos/luftaufnahme-des-rasenplatzes-mit-blauen-sonnenkollektoren-Ilpf2eUPpUE
- 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
- Slide 4, right:
- https://unsplash.com/de/fotos/weisse-windmuhle-auf-gelbblattrigem-blumenfeld-tagsuber-E56cTF65xFw