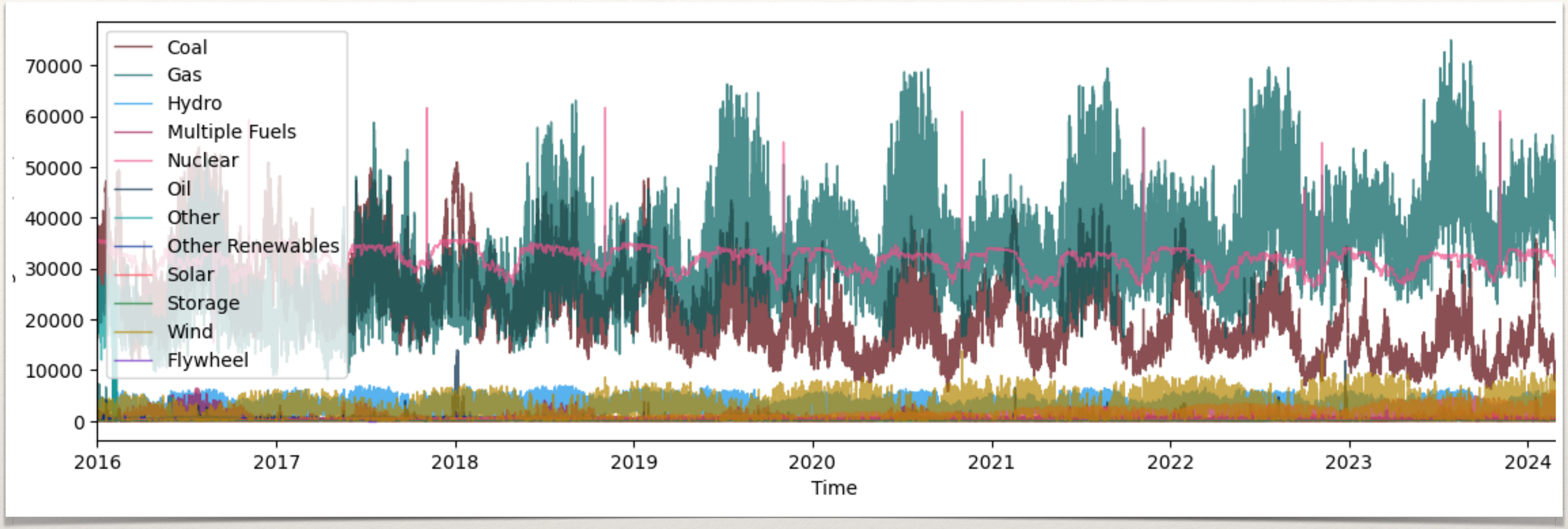


Project Presentation:

Energy Grid Forecasting

Comparing forecasting models utilizing electrical load data.

Executive Summary

The Problem:

In an increasingly complex energy market, predictive forecasting is paramount.

Different prediction lengths, or forecast horizons, lend themselves to different real-world decisions and forecasting models.

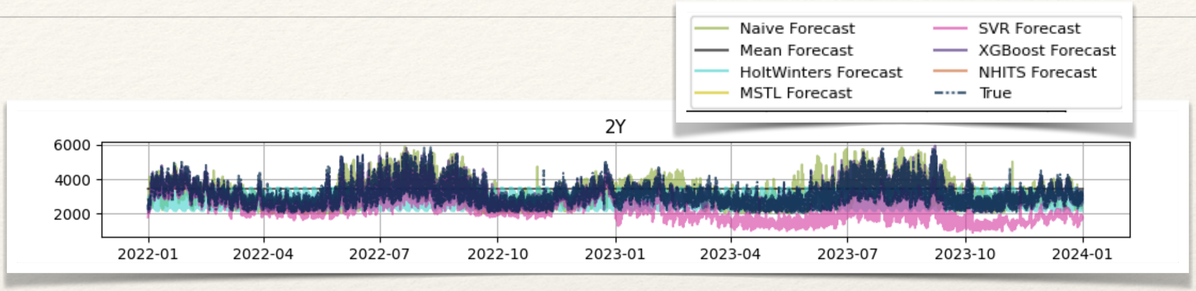
Objective:

Acquire and process a large dataset of electrical load data and covariate features.

Feature engineer a broad covariate matrix.

Specify and tune several models from different method families.

- Statistical, Machine Learning, Deep Learning
- Match models to their best performing forecast horizons.



Key Findings:

Accuracy and efficiency tradeoff is more important in their use case.

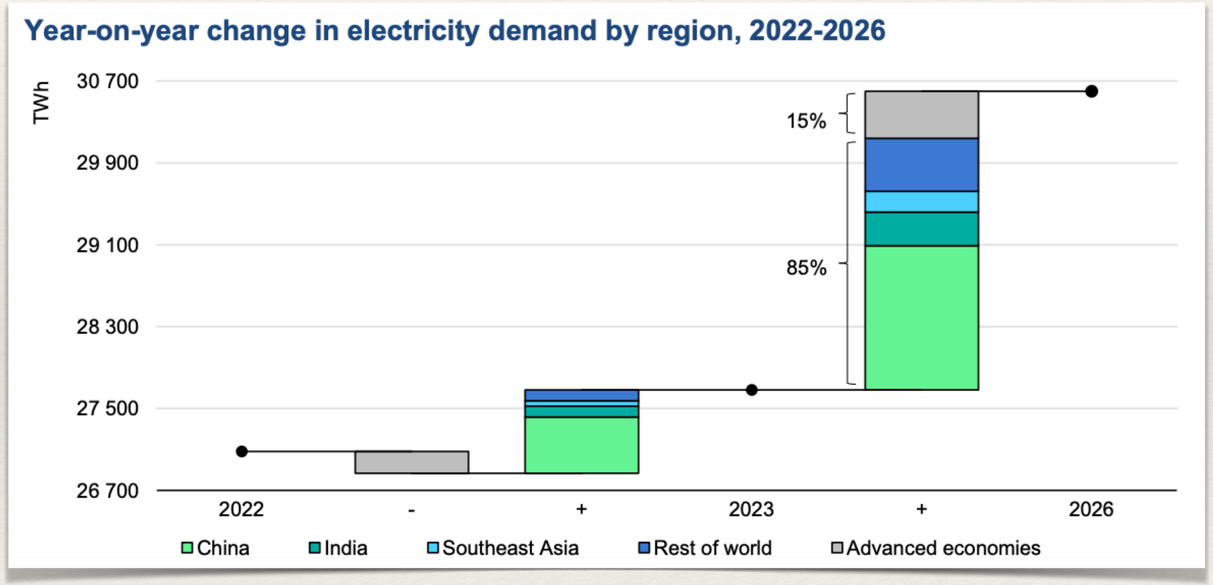
- XGBoost offers strong performance across horizons with low computational demands.
- Holt-Winters and MSTL excel in short-term forecasts.
- SVR evaluates well across all short and medium horizons with a few task-specific exceptions.
- N-HiTS performs well on short-term horizons despite computational costs, likely excelling on longer horizons with full training data and resources.

A Changing Marketplace and Technology

Three things are happening simultaneously:

1. Energy generation is becoming more diversified.
2. Energy demands are changing.
3. Advances in Data Science and Machine-Learning.

More accurate and efficient **energy forecasting** methods directly translates a more efficient, resilient, and **future-ready energy grid**.



Source: International Energy Agency (IEA) - Electricity 2024 - Analysis and forecast to 2026 Report, page 15

Energy forecasting is a vital part of today's energy markets and operations.

- Energy Forecasting is Used To:
  - Set the cost of electricity.
  - Power generation operations.
  - Power distribution.
  - Infrastructure planning.
  - Buy and sell on the energy marketplace.

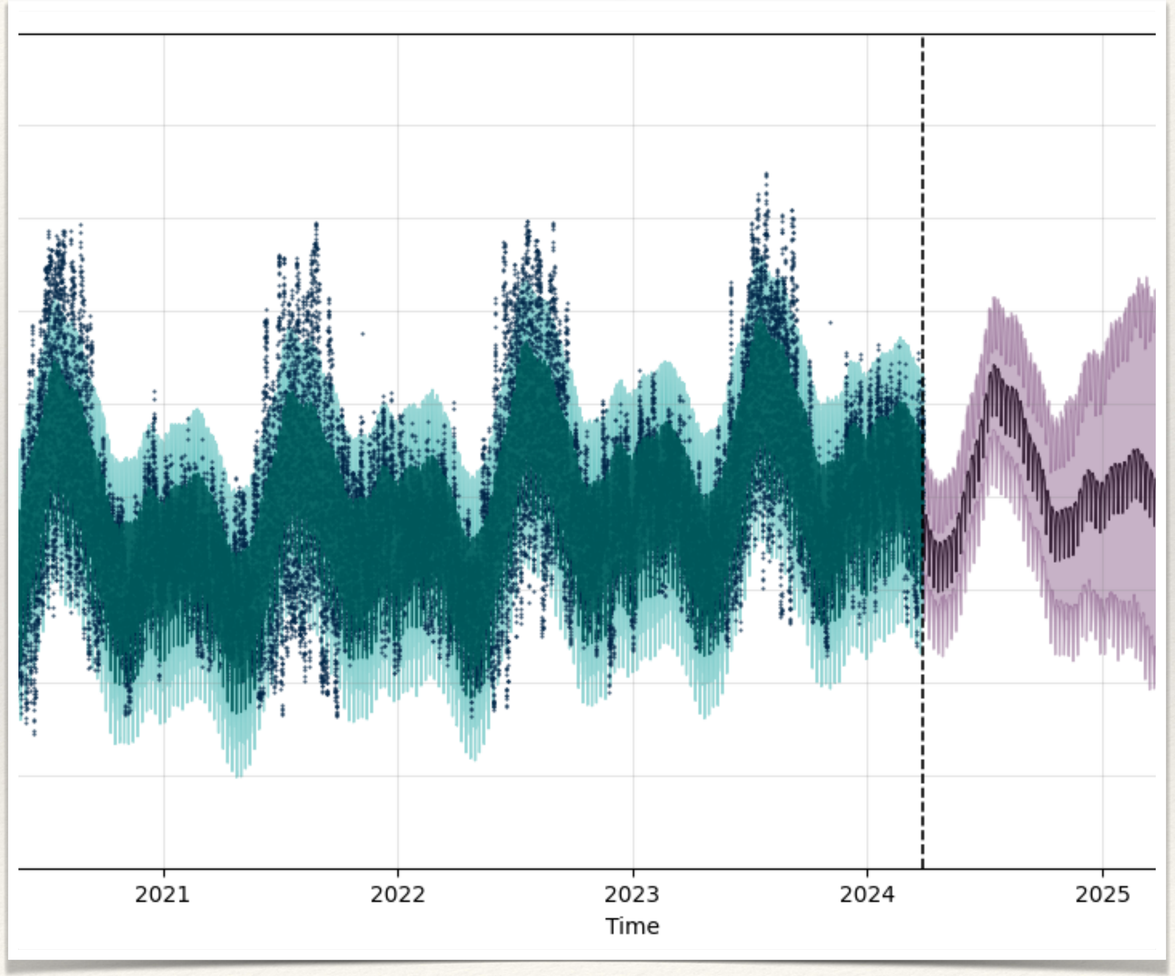
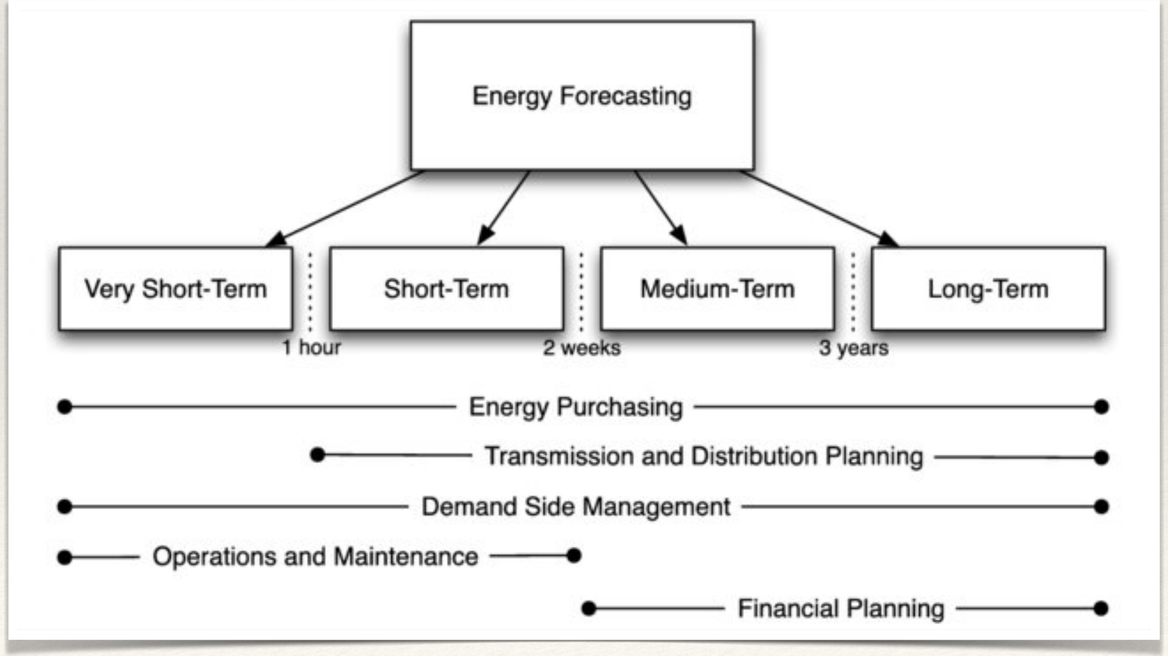


Table 1: Defining prediction forecast horizon intervals. The forecast horizon determines how far into the future a forecast model predicts.

No.	Forecast Horizon Intervals	Duration
1	Very Short-Term	$X < 1$ Hour
2	Short-Term	1 Hour $\leq X <$ 1 Week
3	Medium-Term	1 Week $< X \leq 1$ Year
4	Long-Term	$X > 1$ Year



Forecast Horizons determine how for into the future a model predicts and each have different applications.

The performance of forecasting models depends on:

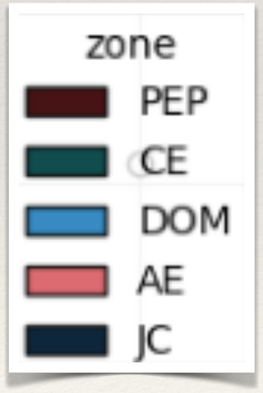
- Forecast horizon interval
- The time series data
  - Periodicity / Seasonality
  - Granularity

2 Datasets | with hourly and daily sample rates.

Dataset Overview:

Table 2: Dataset size and date ranges.

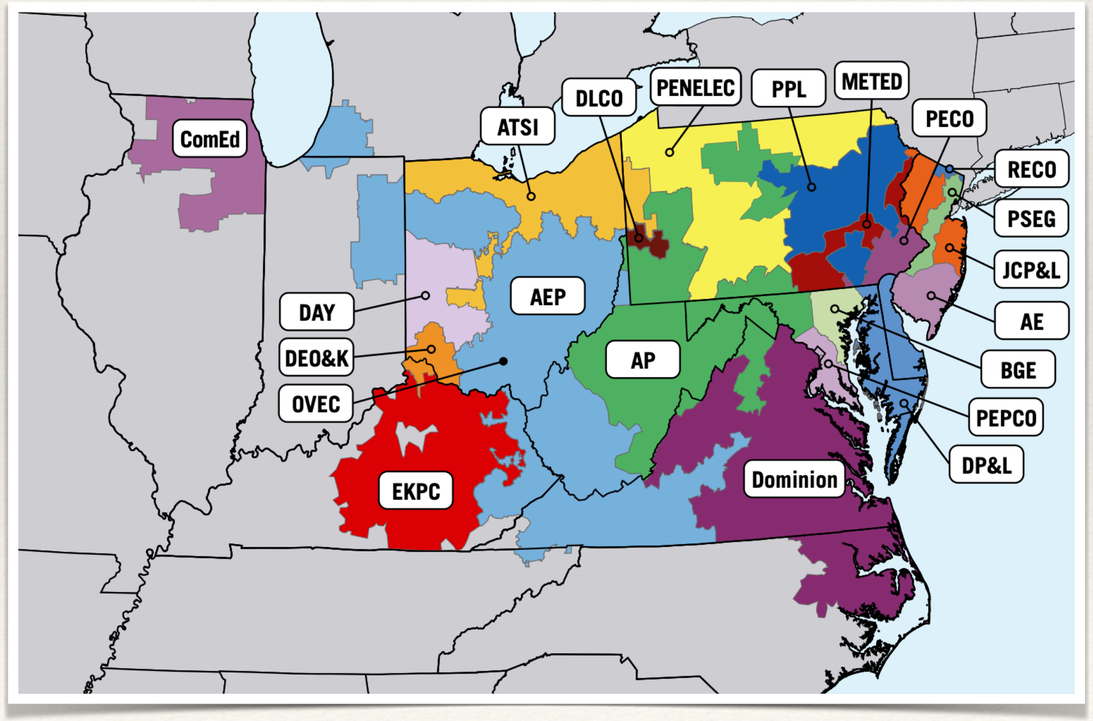
Dataset	Size (Rows)	Date Range
Hourly Load	5,446,243	1993/01/01 - 2024/03/27
Daily Weather	32,400	1993/01/01 - 2024/04/09



Energy Load & Weather organized by region.

Selected Zones

Data collection regions.



Energy Load Dataset | Hourly

	Date	zone	mw
363660	2014-09-19 04:00:00	PEP	2324.799
83764	2002-07-23 04:00:00	PEP	3807.000
243118	2010-02-17 23:00:00	CE	11594.762
165525	2007-03-08 07:00:00	DOM	14532.000
394880	2015-07-17 20:00:00	JC	3492.333
531249	2018-08-27 06:00:00	JC	2510.793
510885	2018-03-10 13:00:00	CE	10181.509
697011	2022-06-08 15:00:00	AE	1743.157
132498	2005-12-04 14:00:00	CE	10599.000
613926	2020-07-16 06:00:00	AE	1108.564
486077	2017-08-15 20:00:00	JC	3409.412
474959	2017-05-15 04:00:00	PEP	2202.951
309091	2012-08-22 06:00:00	DOM	9378.000
615745	2020-07-31 09:00:00	PEP	3833.365
18355	1995-02-04 19:00:00	PEP	3578.000
653974	2021-06-14 23:00:00	JC	2481.762
95201	2003-11-11 17:00:00	PEP	3704.426
550499	2019-02-03 16:00:00	JC	2466.809

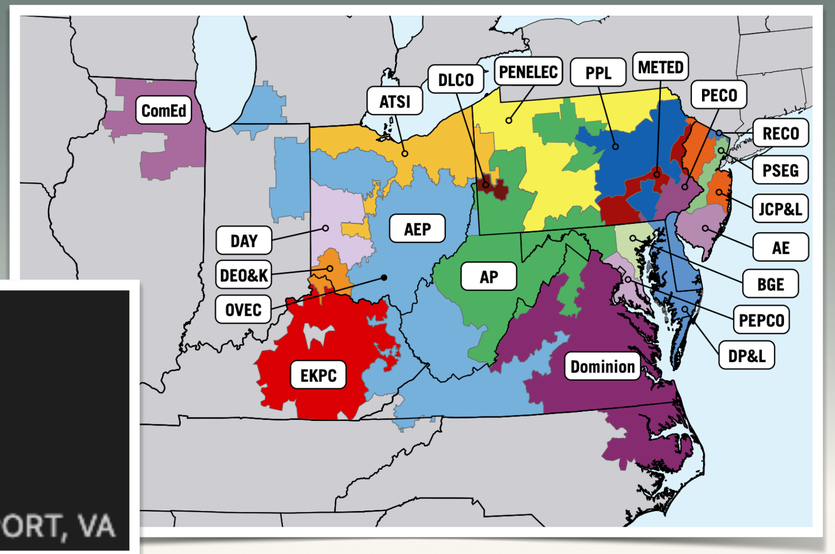
Feature Overview:

- **Date:** Hourly date and time in EPT.
- **Transmission Zone:** Zone where the electricity transmission occurs. Flagged by a string of the energy company(s)' name.
  - `AE`: Atlantic City Electric Co.
  - `CE`: ComEd
  - `DOM`: Dominion
  - `JC`: Jersey Central Power & Light
  - `PEP`: Potomac Electric Power Co.
- **MW:** Electrical load measured in Megawatts (MW).

Feature Overview: Weather Dataset | Daily

- **Date:** Daily sample rate.
- **MaxTemperature**
- **MinTemperature**
- **AvgTemperature**
- **Precipitation:** measured in inches.
- **Zone:** Electric distributor matched from other dataset.

	Date	MaxTemperature	MinTemperature	AvgTemperature	Precipitation	zone
4241	2006-09-02	77	57	67.0	0.00	CE
22851	1998-02-17	56	41	48.5	0.86	PEP
24950	2003-11-17	64	47	55.5	0.00	PEP
3074	2023-06-02	87	48	67.5	0.01	AE
23726	2000-07-11	79	68	73.5	0.01	PEP
11418	2007-05-18	61	53	57.0	0.42	DOM
22923	1998-04-30	70	57	63.5	0.00	PEP
25150	2004-06-04	69	62	65.5	0.12	PEP
10842	2005-10-19	81	49	65.0	0.00	DOM
5454	2009-12-28	30	14	22.0	0.00	CE
25977				75.5	0.00	PEP
15555				77.0	0.09	DOM
20096				51.0	0.00	JC
27267				58.0	0.53	PEP
6591				32.5	1.07	CE
				64.0	0.00	CE
				52.5	0.03	PEP
				75.5	0.00	PEP



Weather Stations Chosen:

- AE: MILLVILLE MUNICIPAL AIRPORT, NJ
- CE: CHICAGO OHARE INTL AP, IL
- DOM: RICHMOND INTERNATIONAL AP, VA
- JC: NEW BRUNSWICK 3 SE, NJ
- PEP: WASHINGTON REAGAN NATIONAL AIRPORT, VA

Where Has the Time Gone? | Missing Datetimes

```python
# Could also use .asfreq('H') which would assign any missing rows to NaN then identify rows.
print('######### Check for Missing Weather Dates #########')
for zone in zones:
    # Find min and max dates for that zone.
    date_min = weather_df.loc[weather_df['zone'] == zone]['Date'].min()
    date_max = weather_df.loc[weather_df['zone'] == zone]['Date'].max()
    print(f'{zone} - {date_min} | {date_max}')
    # Find missing dates.
    # Weather DF - Set freq to 'D' for days.
    miss_date = pd.date_range(start = date_min, end = date_max, freq = 'D').difference(weather_df['Date'])
    print(f"{'':<5}Missing Dates: {miss_date if len(miss_date) > 0 else False}")

print('\n######### Check for Missing Energy Dates #########')
for zone in zon
    # Find min
    date_min = load_zone_df.loc[load_zone_df['zone'] == zone]['datetime_beginning_ept'].min()
    date_max = load_zone_df.loc[load_zone_df['zone'] == zone]['datetime_beginning_ept'].max()
    print(f'{zone} - {date_min} | {date_max}')
    # Find missing dates.
    # Load DF - Set freq to 'H' for hours.
    miss_date = pd.date_range(start = date_min, end = date_max, freq = 'H').difference(
        load_zone_df['datetime_beginning_ept'].loc[load_zone_df['zone'] == zone])
    print(f"{'':<5}Missing Dates: {miss_date if len(miss_date) > 0 else False}")

    # Check to see if missing dates exist.
    # Fill missing dates with average of two surrounding times.
    # TODO: There's definitely some optimizations to be had here.
    if len(miss_date) > 0:
        print(f"{'':<5}============================================================")
        print(f"{'':<5}Adding Average of Surrounding Times to {zone} Missing Dates")
        # Find index of one hour before and after missing
        miss_pre_index = (load_zone_df.loc[(load_zone_df['zone'] == zone) &
                                           (load_zone_df['datetime_beginning_ept'].isin(
                                               miss_date - pd.Timedelta(hours = 1)))]).index
        miss_post_index = (load_zone_df.loc[(load_zone_df['zone'] == zone) &
                                            (load_zone_df['datetime_beginning_ept'].isin(
                                                miss_date + pd.Timedelta(hours = 1)))]).index
        # Calculate average of two surround times.
        miss_avg = [np.round(np.mean((a,b)), 3) for a, b in
                    (zip(load_zone_df.iloc[miss_pre_index]['mw'], load_zone_df.iloc[miss_post_index]['mw']))]
        # Create DF to hold values and concat to main DF.
        temp_df = pd.DataFrame({'datetime_beginning_ept' : miss_date,
                                'zone' : zone,
                                'mw' : miss_avg})
        load_zone_df = pd.concat([load_zone_df, temp_df], join = 'outer', ignore_index = False, axis = 0)
        print(f"{'':<5}{'='*5} Done!\n")
```

(variable) load_zone_df: DataFrame

- **weather_df:** Find any missing days.
  - None found.
- **load_df:** Find any missing hours.
  - Insert missing datetime.
  - Replace **MW** with average of the next two surrounding observations.

```
PEP - 1993-01-01 00:00:00 | 2024-03-27 23:00:00
    Missing Dates: DatetimeIndex(['1993-04-04 02:00:00', '1994-04-03 02:00:00',
                   '1995-04-02 02:00:00', '1996-04-07 02:00:00',
                   '1997-04-06 02:00:00', '1998-04-05 02:00:00',
                   '1999-04-04 02:00:00', '2000-04-02 02:00:00',
                   '2001-04-01 02:00:00', '2002-04-07 02:00:00',
                   '2003-04-06 02:00:00', '2004-04-04 02:00:00',
                   '2005-04-03 02:00:00', '2006-04-02 02:00:00',
                   '2007-03-11 02:00:00', '2008-03-09 02:00:00',
                   '2009-03-08 02:00:00', '2010-03-14 02:00:00',
                   '2011-03-13 02:00:00', '2012-03-11 02:00:00',
                   '2013-03-10 02:00:00', '2014-03-09 02:00:00',
                   '2015-03-08 02:00:00', '2016-03-13 02:00:00',
                   '2017-03-12 02:00:00', '2018-03-11 02:00:00',
                   '2019-03-10 02:00:00', '2020-03-08 02:00:00',
                   '2021-03-14 02:00:00', '2022-03-13 02:00:00',
                   '2023-03-12 02:00:00', '2024-03-10 02:00:00'],
                  dtype='datetime64[ns]', freq=None)
    ============================================================
    Adding Average of Surrounding Times to PEP Missing Dates
    ===== Done!
```

Flags in the Wind | Mixed Datatypes

Though **weather_df** was not missing any days, the data logged may have string flags for certain circumstances or measurement errors.

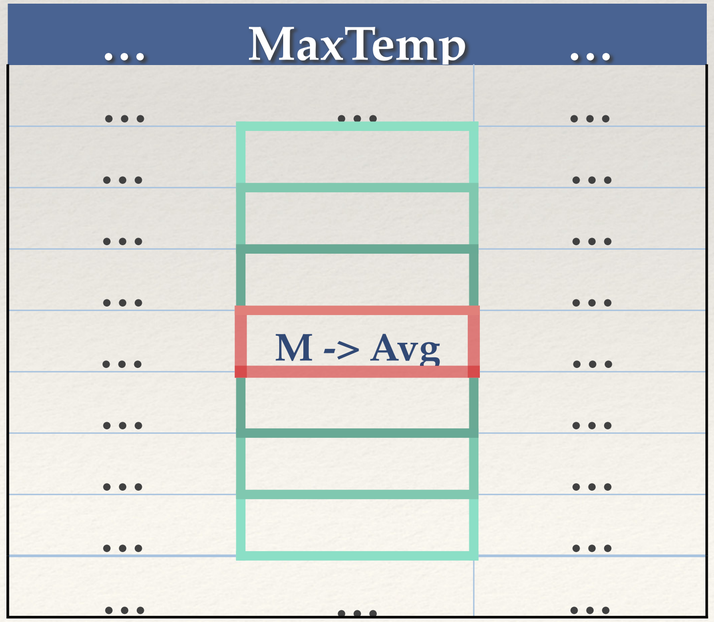
According to NOAA, weather station data may contain the following:

M = Missing (No measurement.)

T = Trace Amount (Precipitation only)

These flags need to be located and handled in different ways.

M: A window of 6 days (3 before and 3 after) will be created and the average measurement taken.

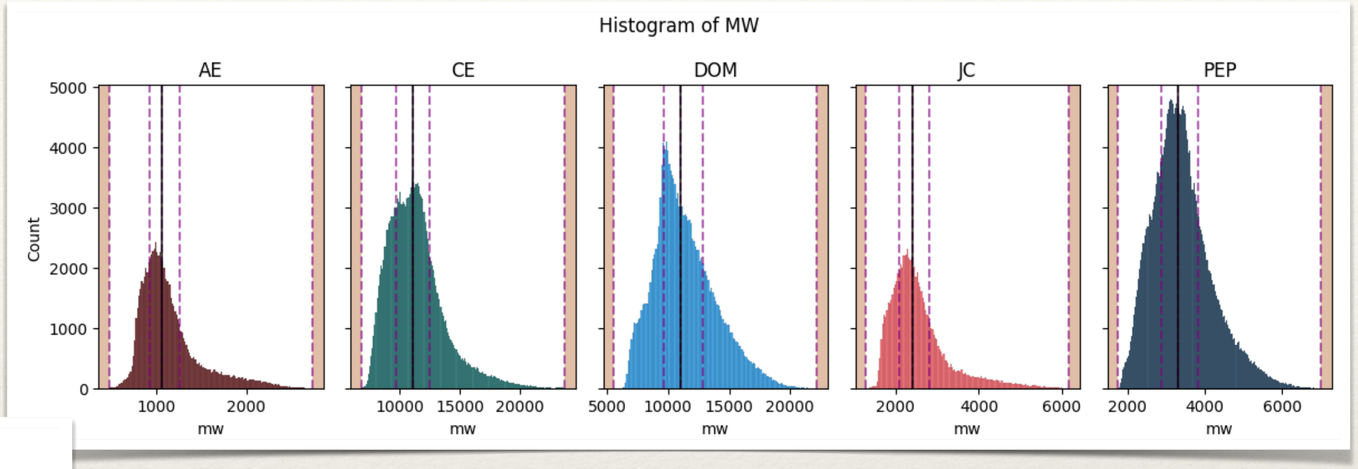


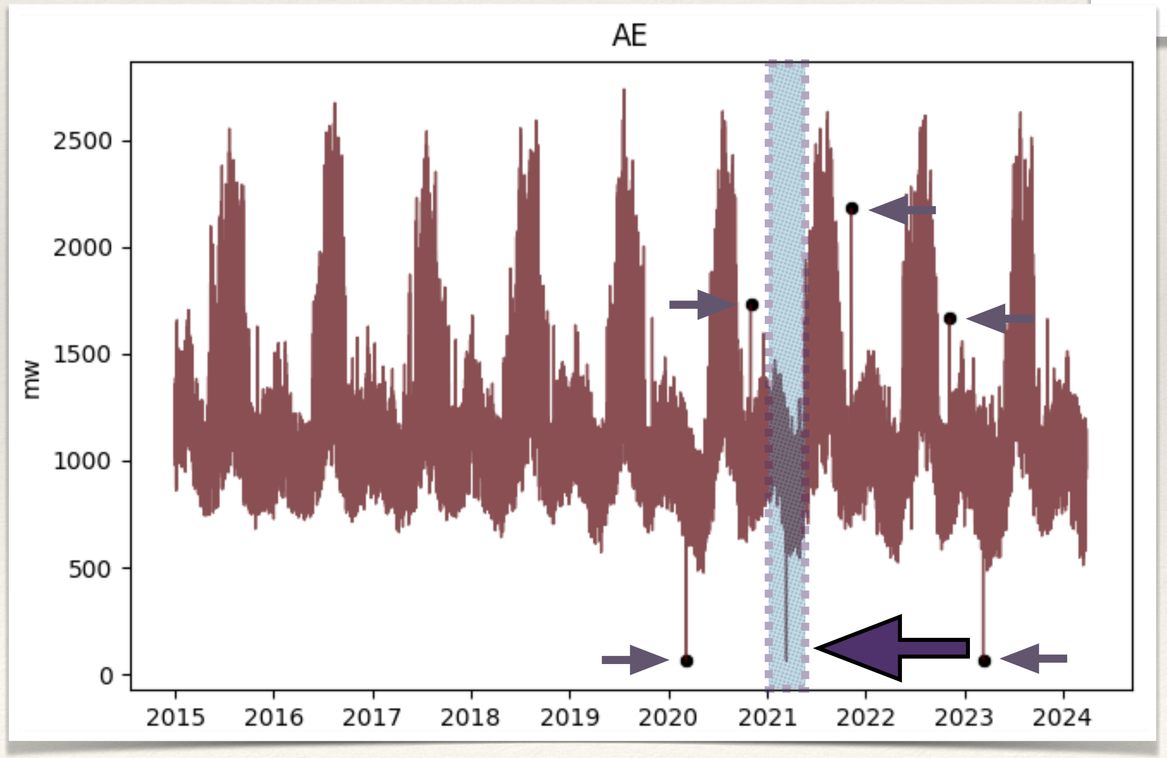
T: Replaced with a comparable measurement on this scale, 0.01.

T → **0.01**

Outlier Detection | Standard Deviation Method

Utilizing the same concept of the Interquartile Range (IQR), we can scale the median vs standard deviation to identify outliers.

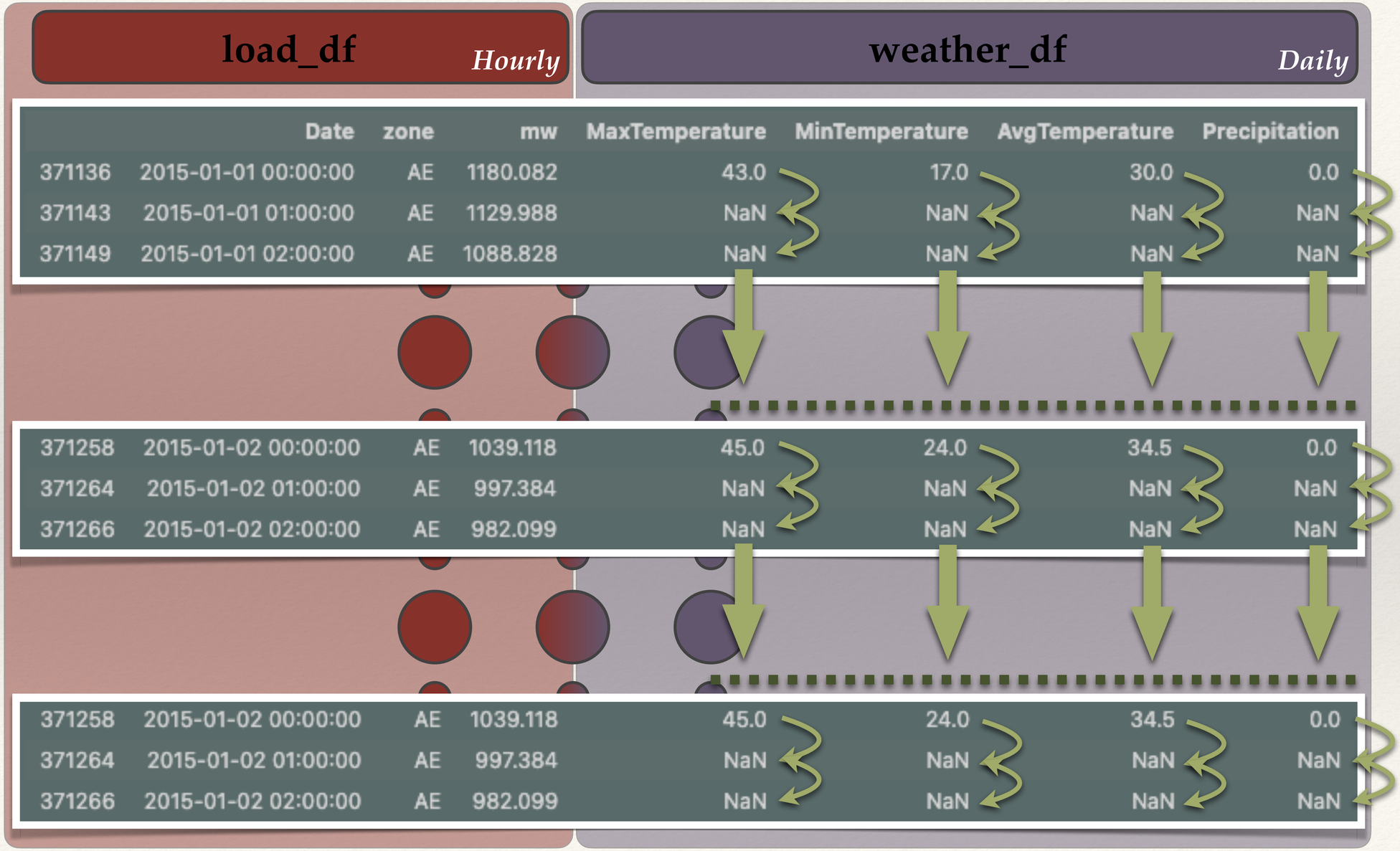




Method: $Median \pm Scalar_{Thresh} * \sigma$

1. Create tumbling window of dates.
2. Calculate median and standard deviation.
3. Calculate outlier threshold.
4. Find data outside threshold.
   - Replace with median.
5. Iterate window.

Dataset Merge



Hourly ⟷ Daily

.sort_values()

Sort by zone and Date.

.ffill()

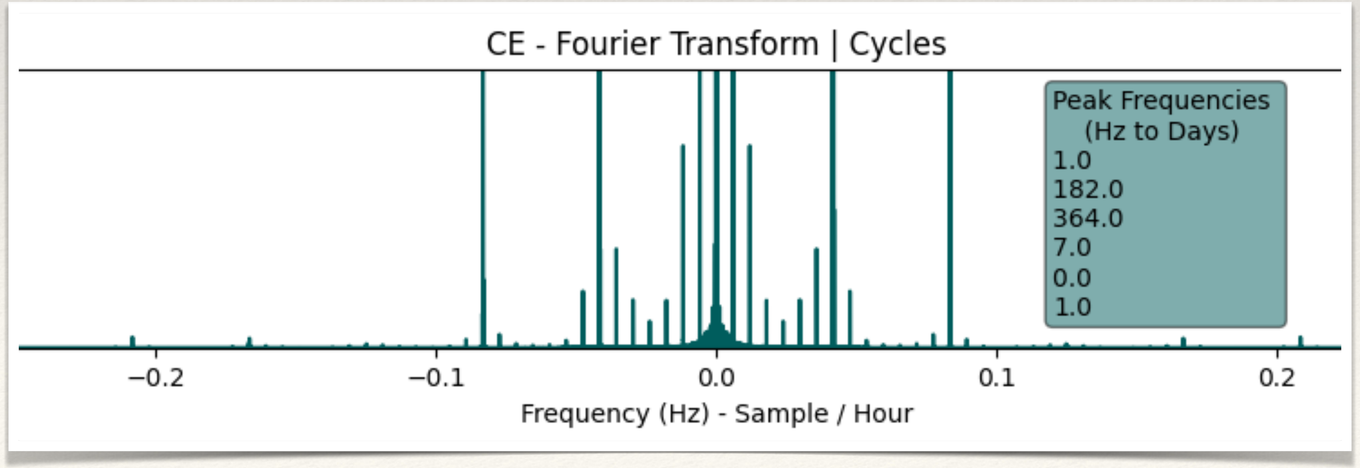
Feed forward weather data until next day.

Feature Engineering

In most time series data, past values of the target variable are highly correlated to future values.

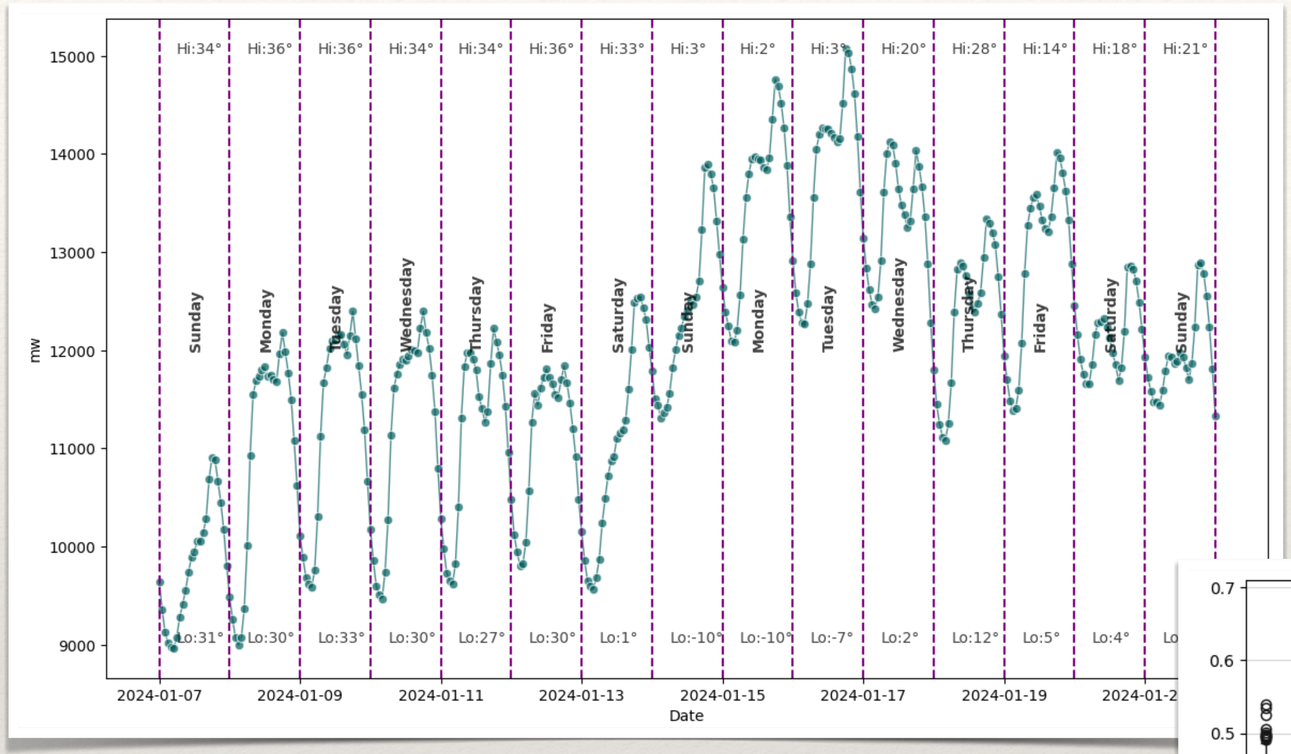
To effectively capture these relationships the following is done:

- **Date Features** - Hours, Days, Month, Years, Day of Year are encoded.
- **Rolling Mean** - Rolling window calculating mean of target variable.
- **Lagged Features** - Intervals related to seasonal periods in the data.
- **Fourier Transform** - Spectral density calculation to identify magnitudes of important seasonal periods.



Peak frequencies identified. Cast units to days with, $\frac{1}{freq * 24hours}$

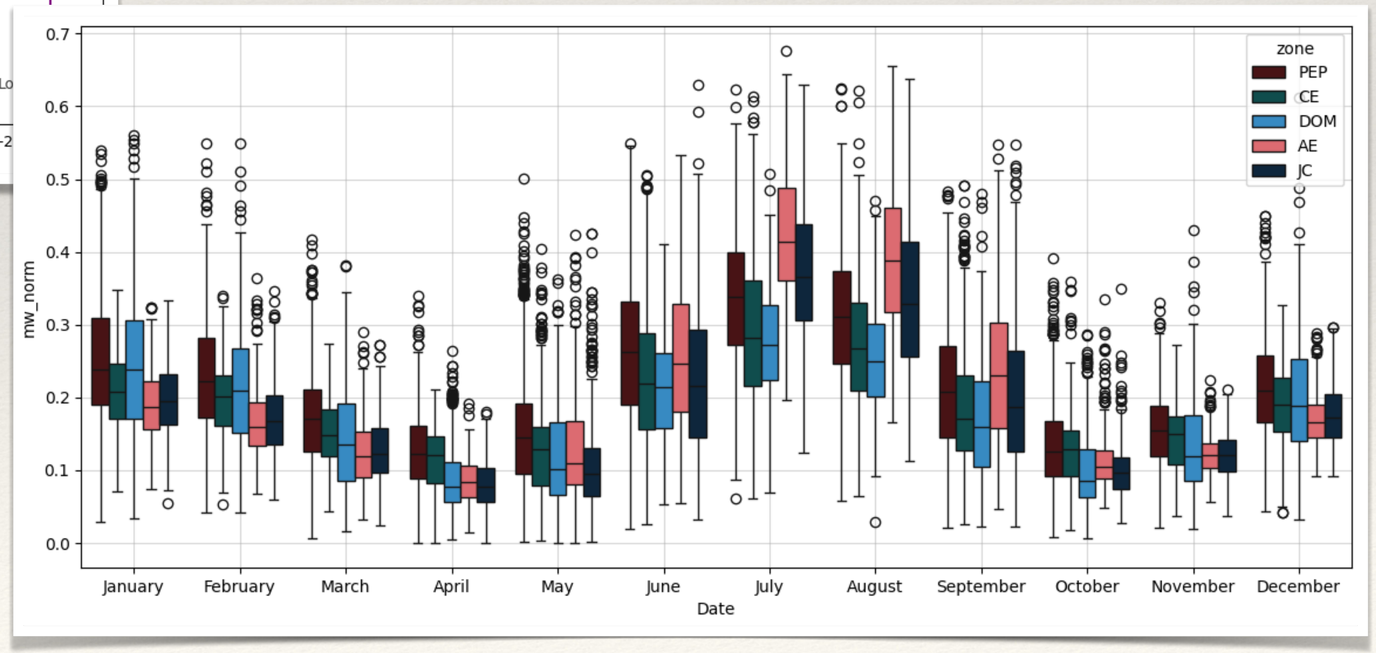
Exploratory Data Analysis (EDA)



A slight correlation between lower load (MW) and the weekend.

A strong correlation between extreme temperature and high loads.

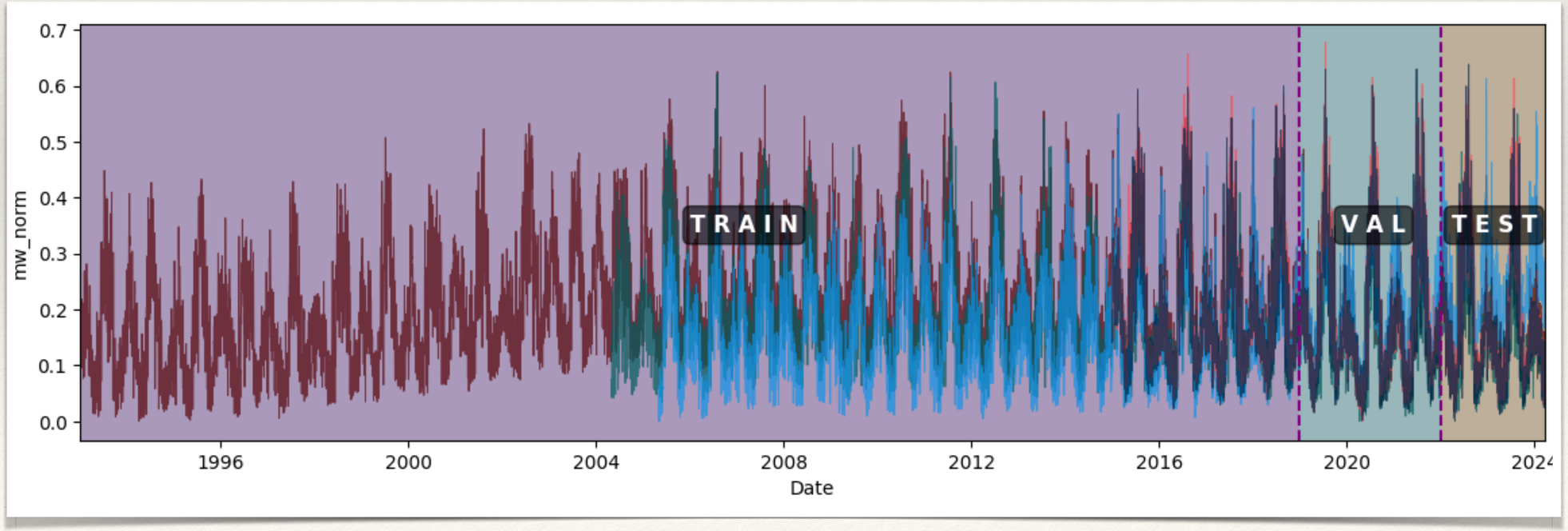
Summer is correlated with the highest loads.



Data Split | Train, Val, and Test Sets

1. To maintaining temporal relationships and prevent data leakage, the test set will always be assigned the most recent dates.

2. Each set contains a reasonable amount of each periodicity that extends beyond planned forecast horizons.



Model Val & Testing | Forecast Horizons

Table 1: Defining prediction forecast horizon intervals. The forecast horizon determines how far into the future a forecast model predicts.

No.	Forecast Horizon Intervals	Duration
1	Very Short-Term	$X < 1$ Hour
2	Short-Term	1 Hour $\leq X <$ 1 Week
3	Medium-Term	1 Week $< X \leq 1$ Year
4	Long-Term	$X > 1$ Year

- 6H - 6 Hours
- 3D - 3 Days
- 1W - 1 Week
- 1M - 1 Month
- 6M - 6 Months
- 2Y - 2 Years

Short ⟷ **Long**

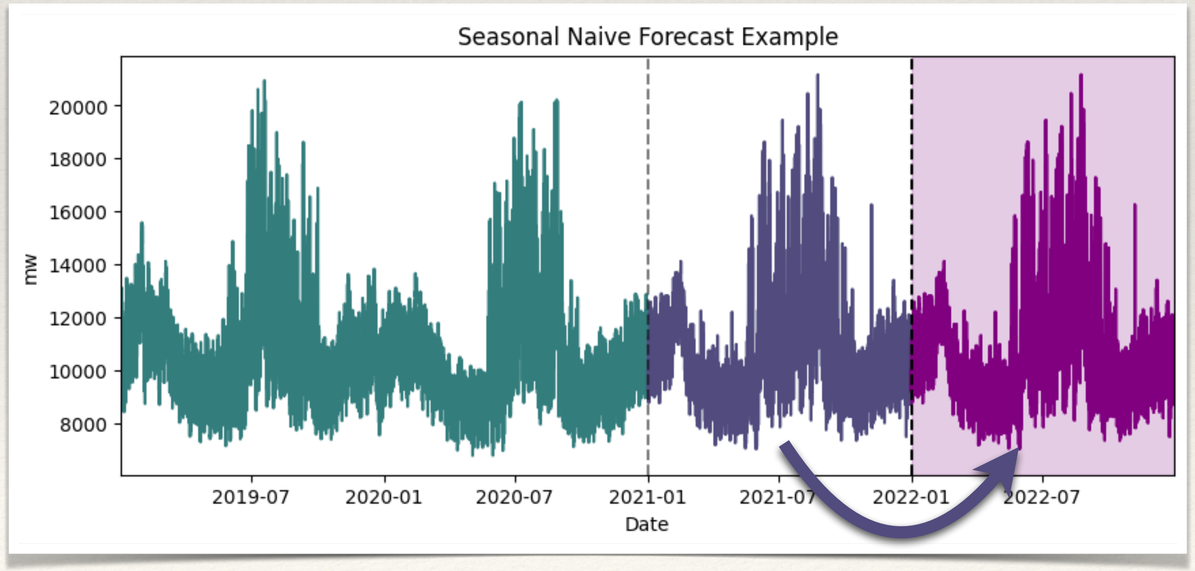
Baseline Models

Seasonal Naive Forecast Model

Forecasts last observed value of previous seasonal period.

Forecast: *2022-01-01 to Horizon (1 Yr)*

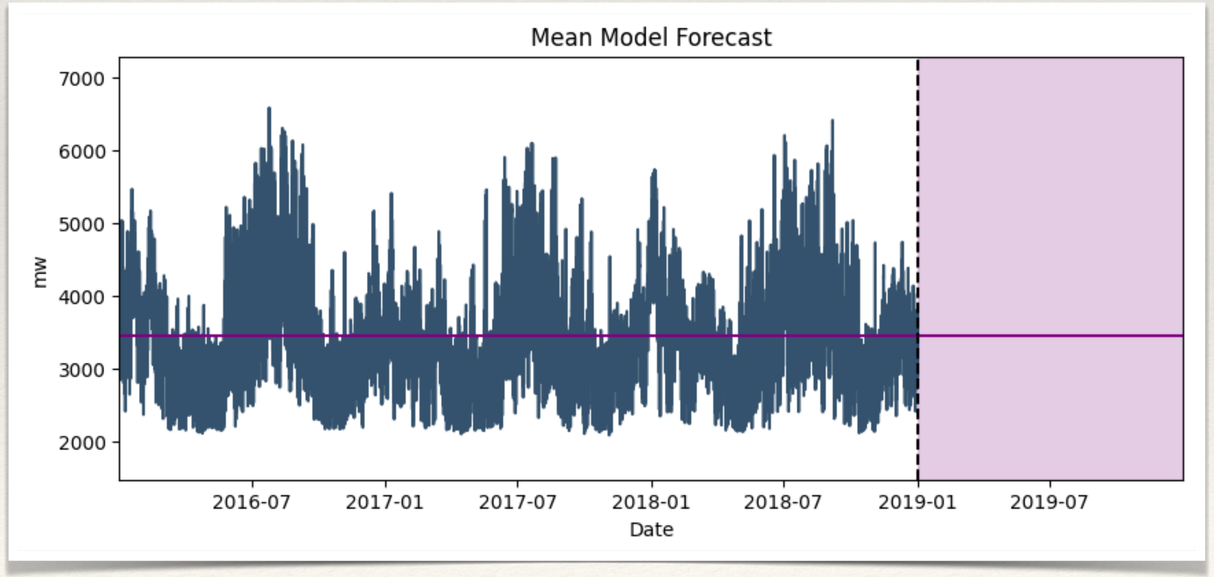
Last Observed: *2021-01-01 up to 2021-12-31*



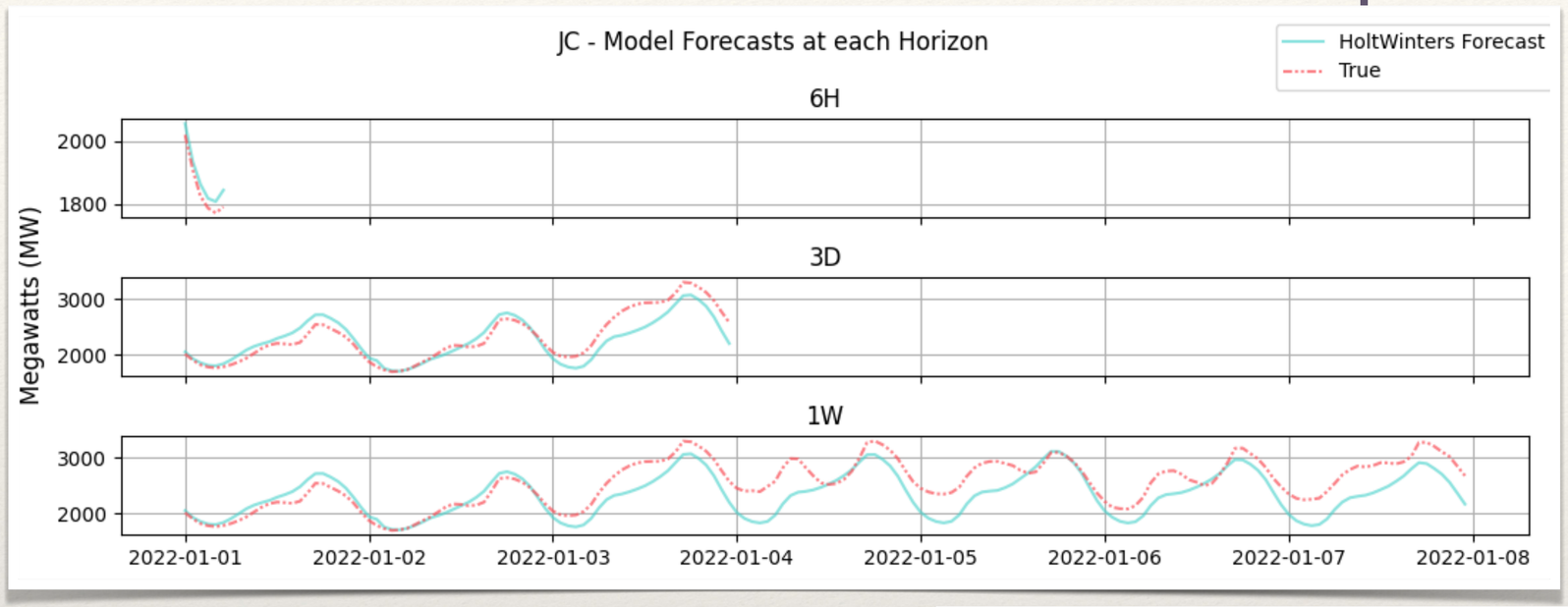
Mean Model

Calculates mean of training set and uses that as all future forecasts.

Mean: 3427.68



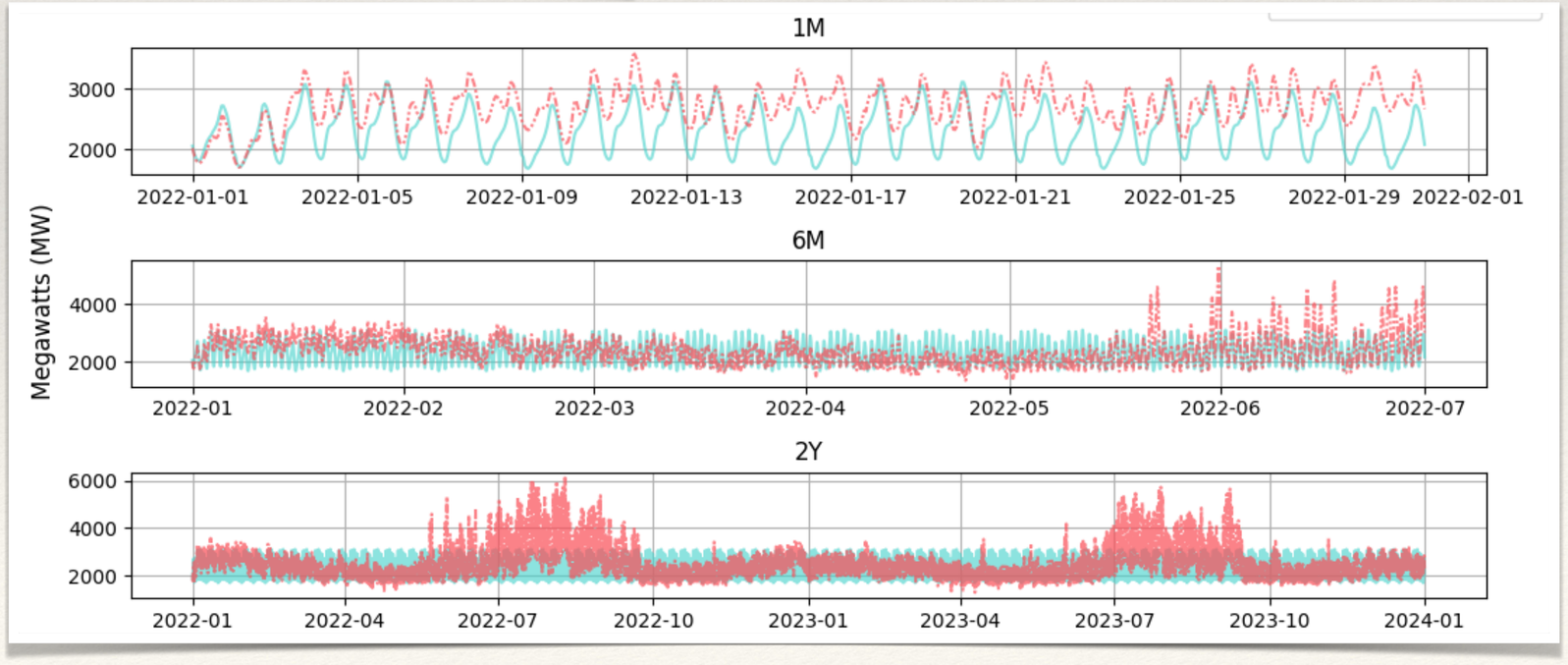
Statistical Models | Holt-Winters



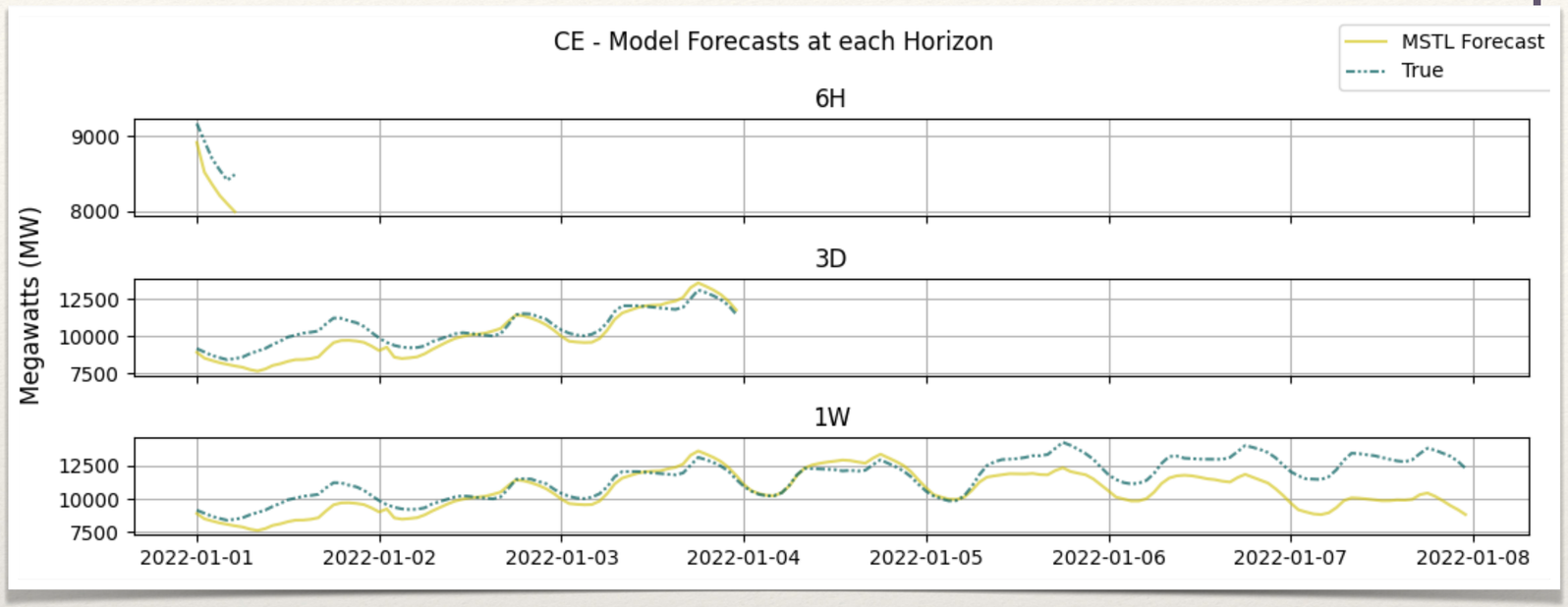
An extension of exponential smoothing for data that contains both trend and seasonality.

Only one seasonal period can be used.

1 Week



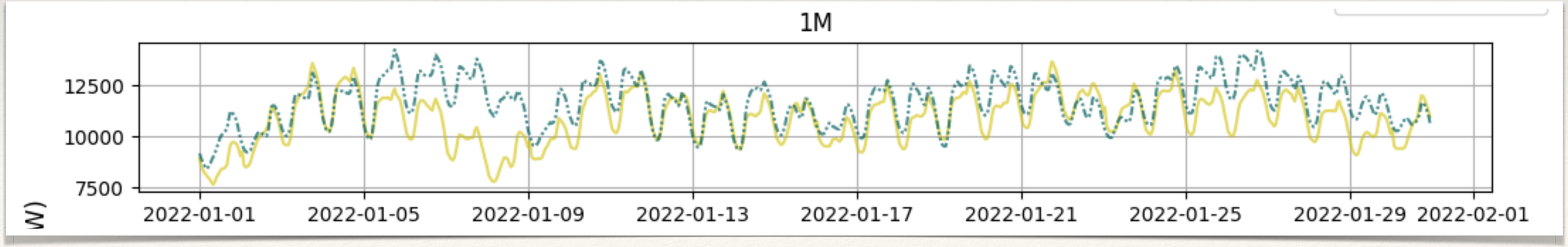
Statistical Models | MSTL



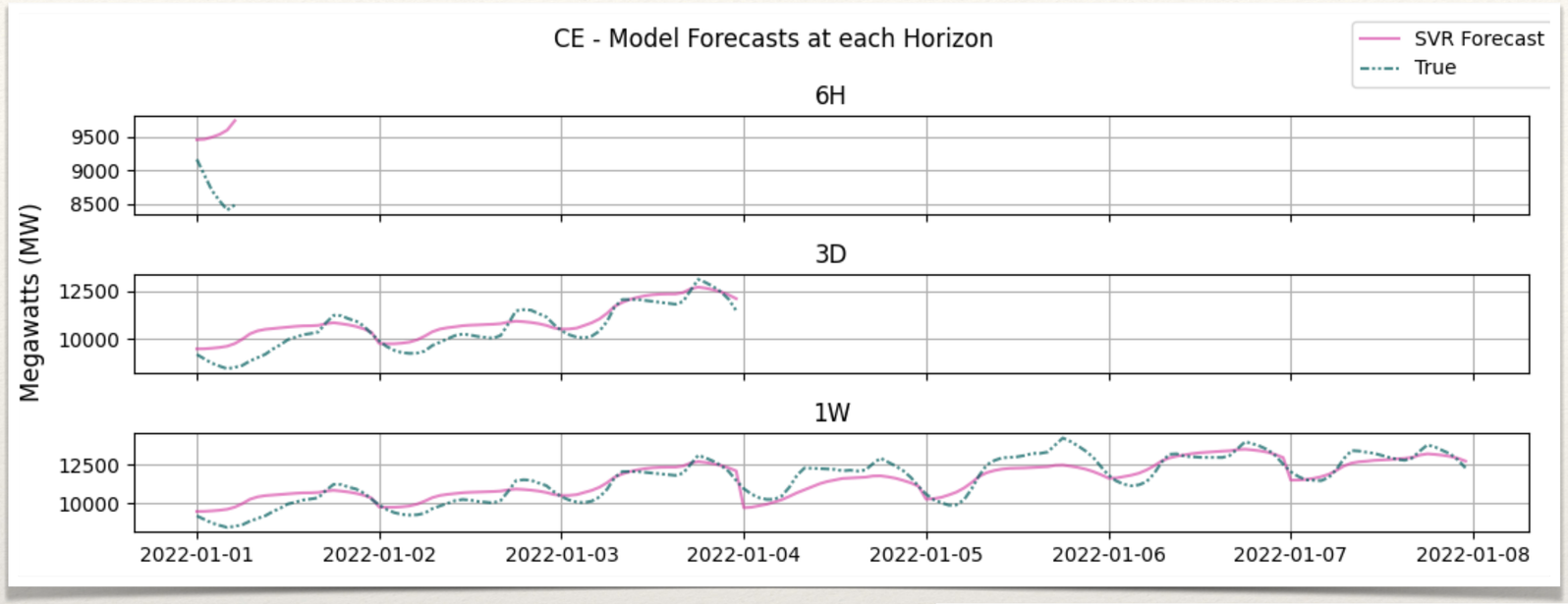
Multiple Seasonal-Trend decomposition using LOESS (MSTL)

Decomposes time series into trend, seasonalities, and residual components.

Can handle multiple Seasonal Periods. *1 Day, 1 Week, 6 Months, 1 Year*



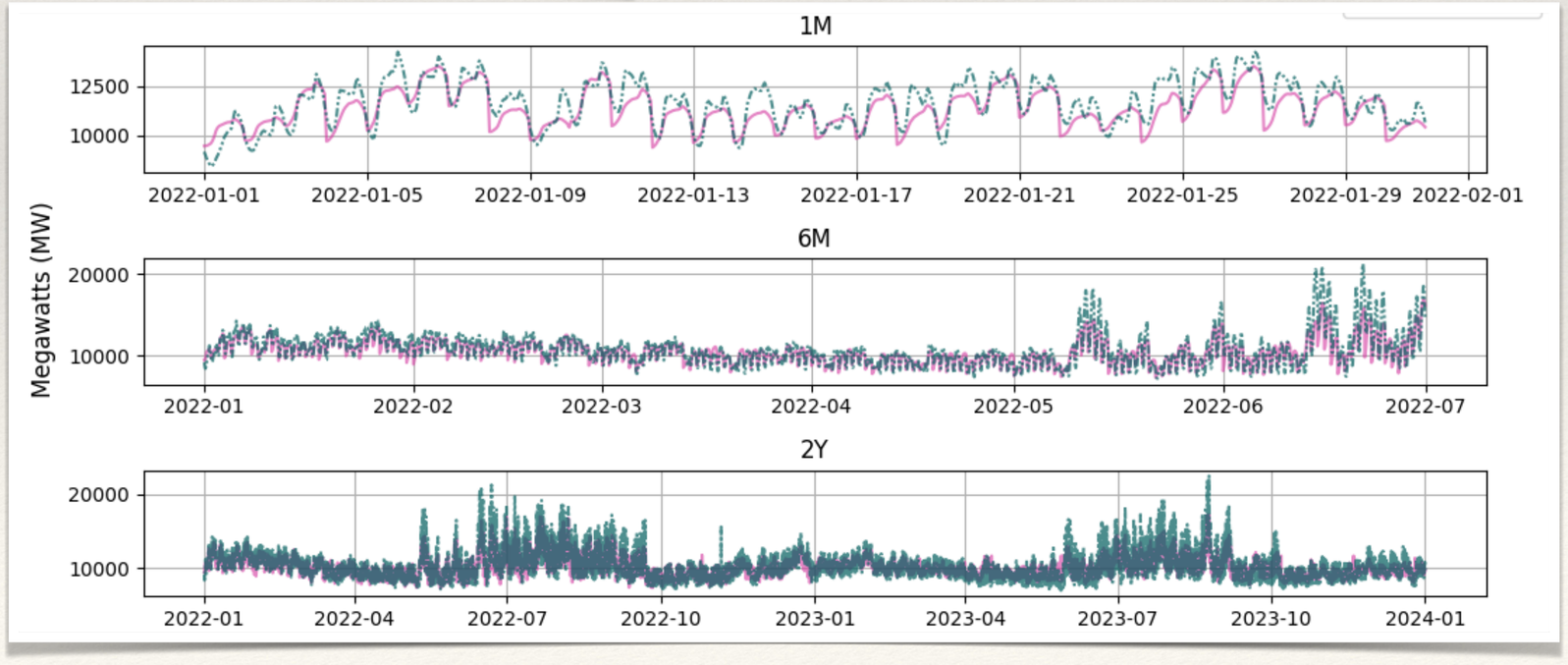
Machine Learning Models | Support Vector Regression (SVR)



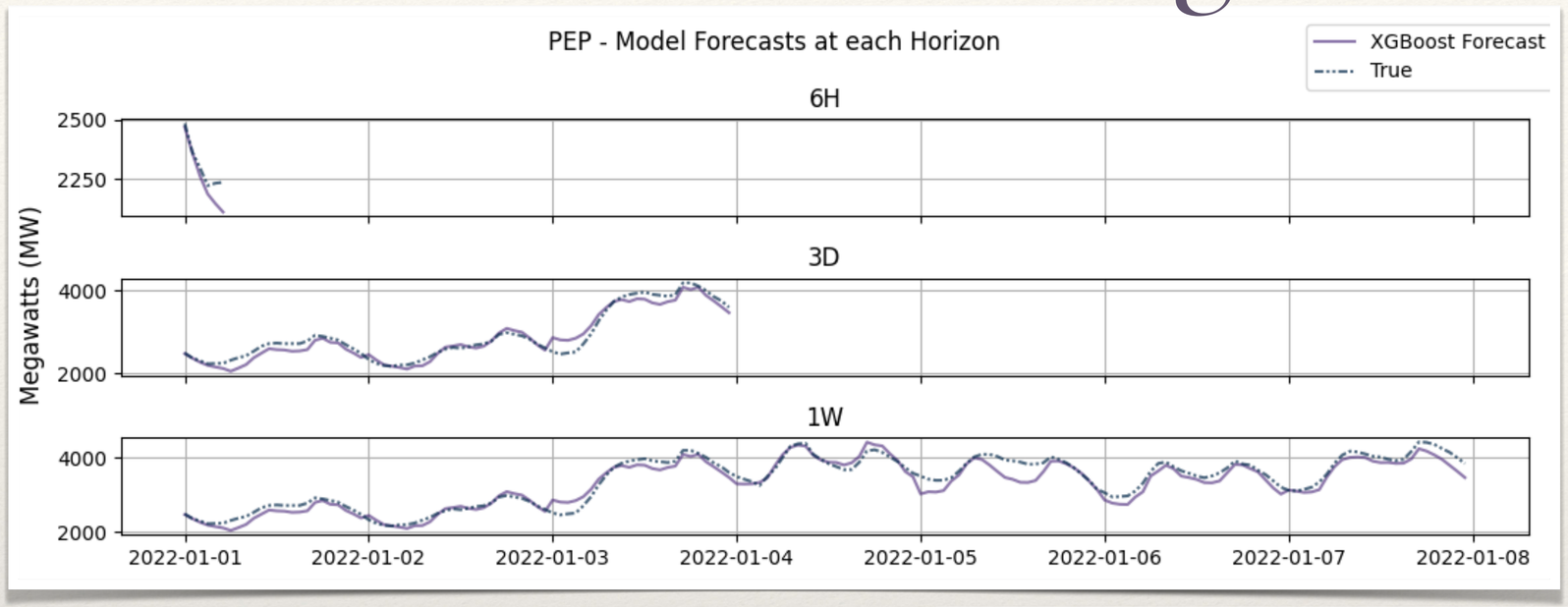
Calculates a hyperplane that best fits the data points in continuous space.

Hyper-parameter optimization is mandatory.

Used Radial Basis Function (RBF) kernel…



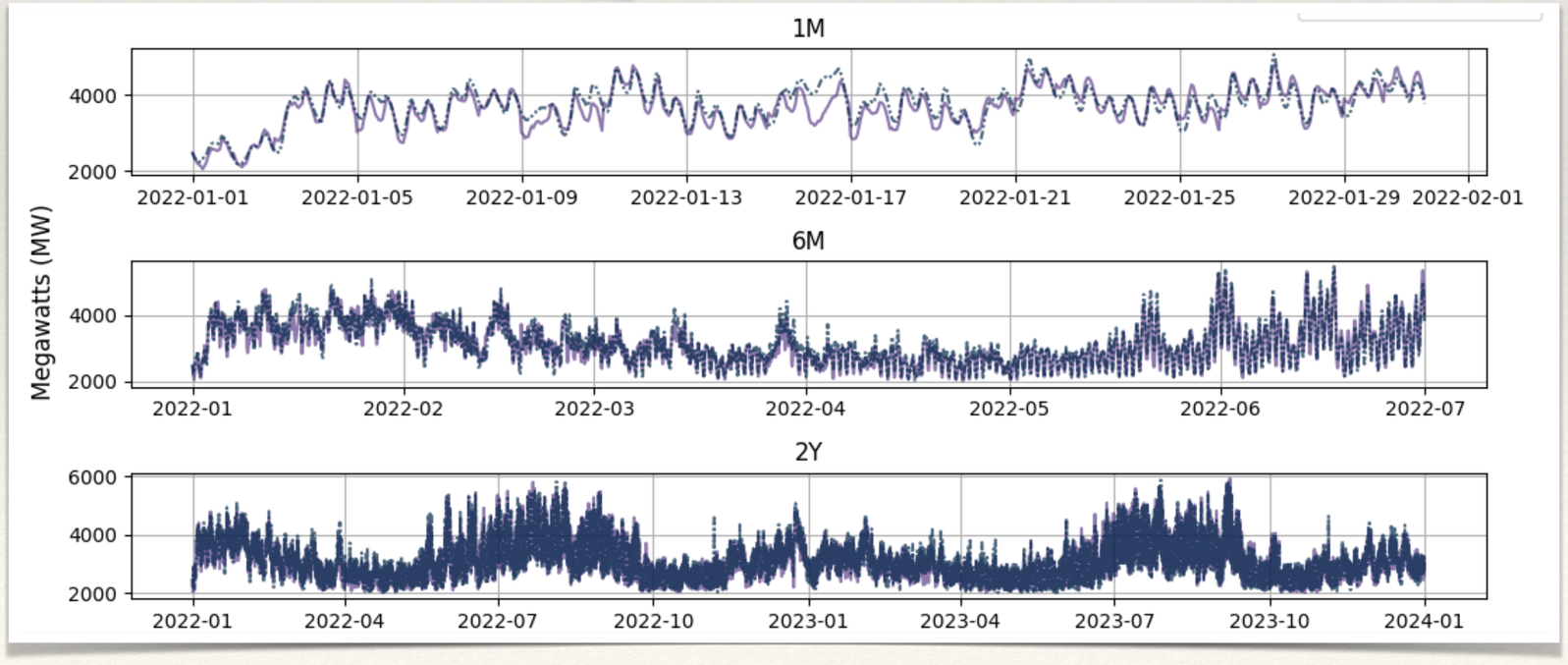
Machine Learning Models | XGBoost



Ensemble gradient boosted tree method which sequentially builds trees.

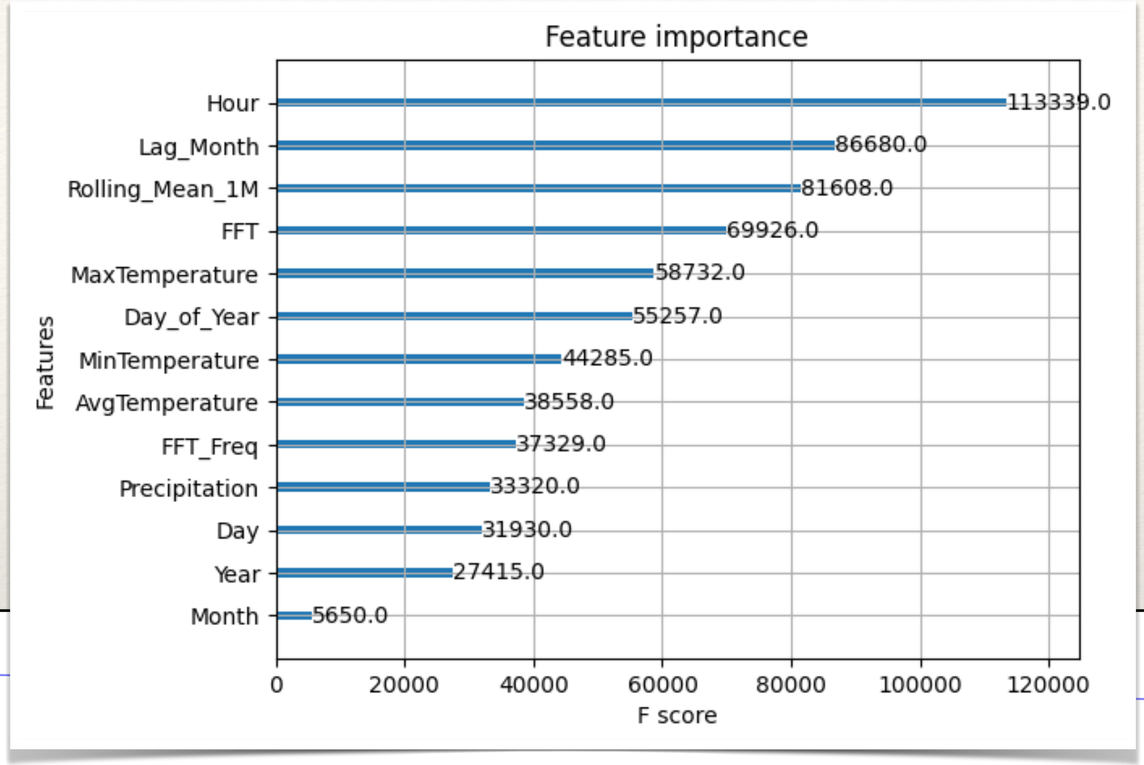
Hyper-parameter optimization is mandatory.

Max depth 7, Max leaf None, n_estimators 6000…

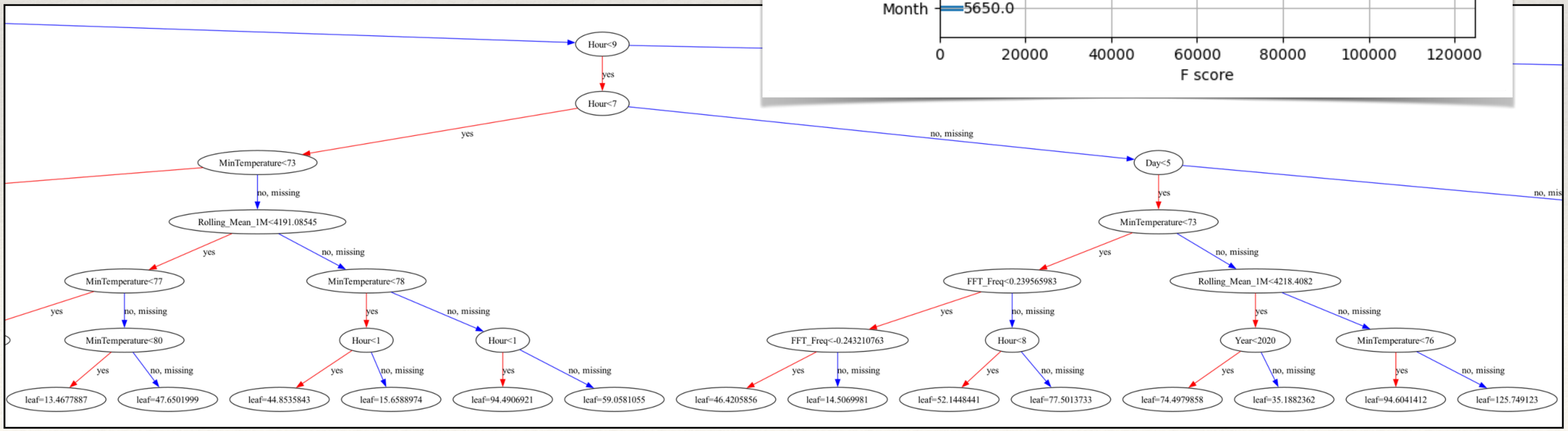


Machine Learning Models | XGBoost Cont...

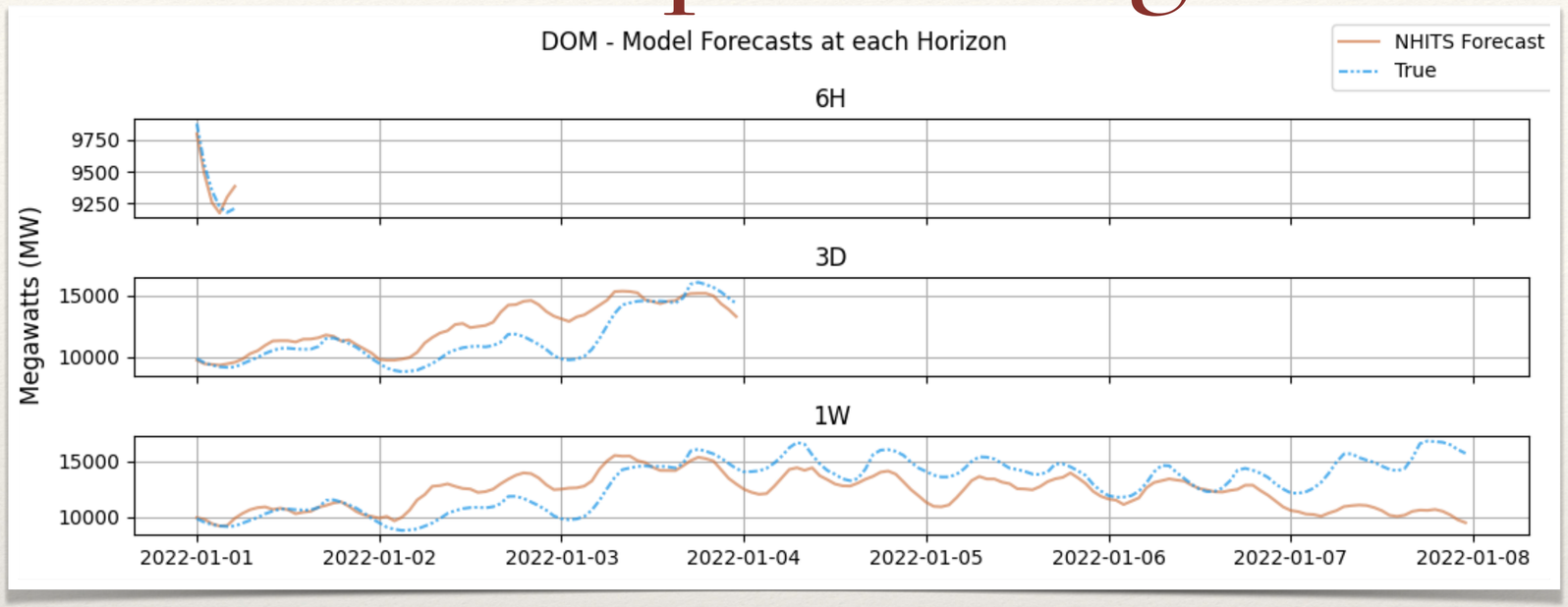
Feature importance according to XGBoost.



A small section of the fitted tree.



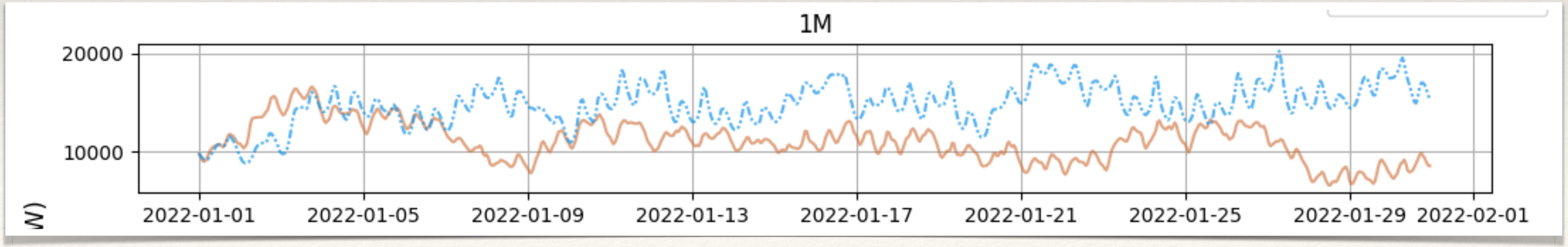
Deep Learning Models | N-HiTS



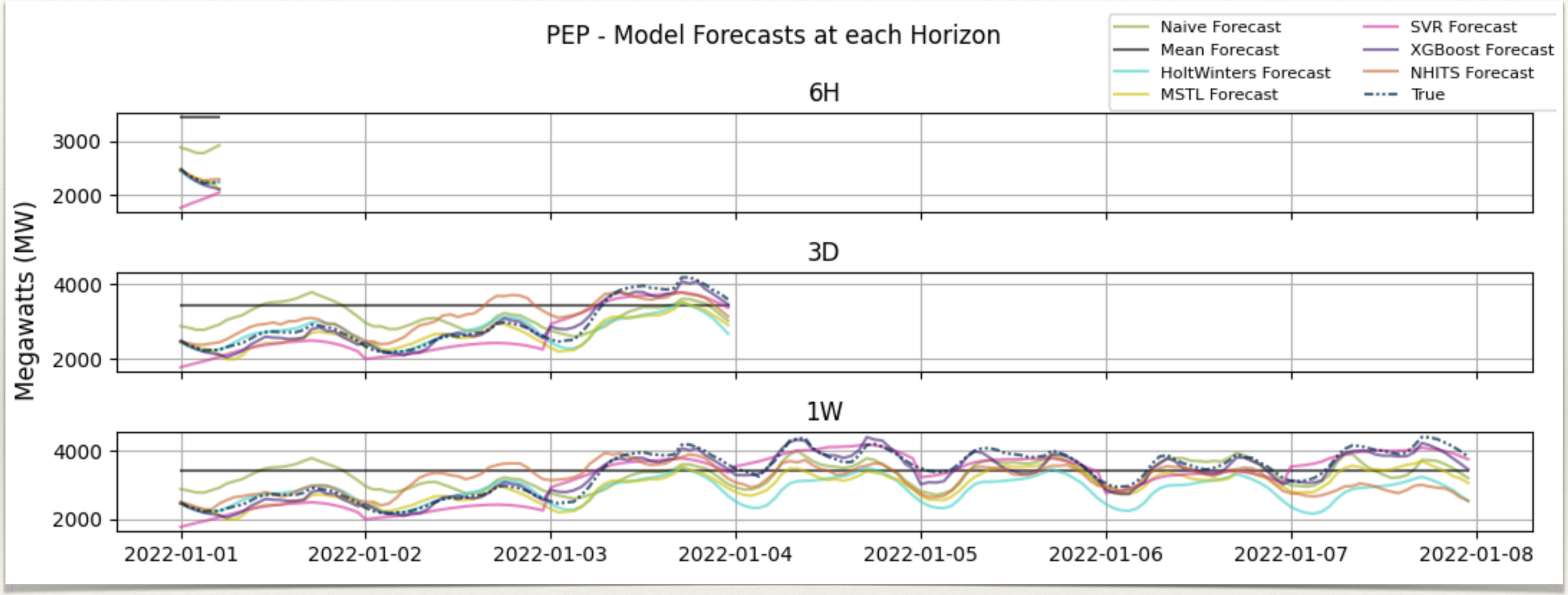
Neural Hierarchical Interpolation (N-HiTS)

An improvement on NBEATS, it is several multilayer perceptrons (MLPs) with ReLU…

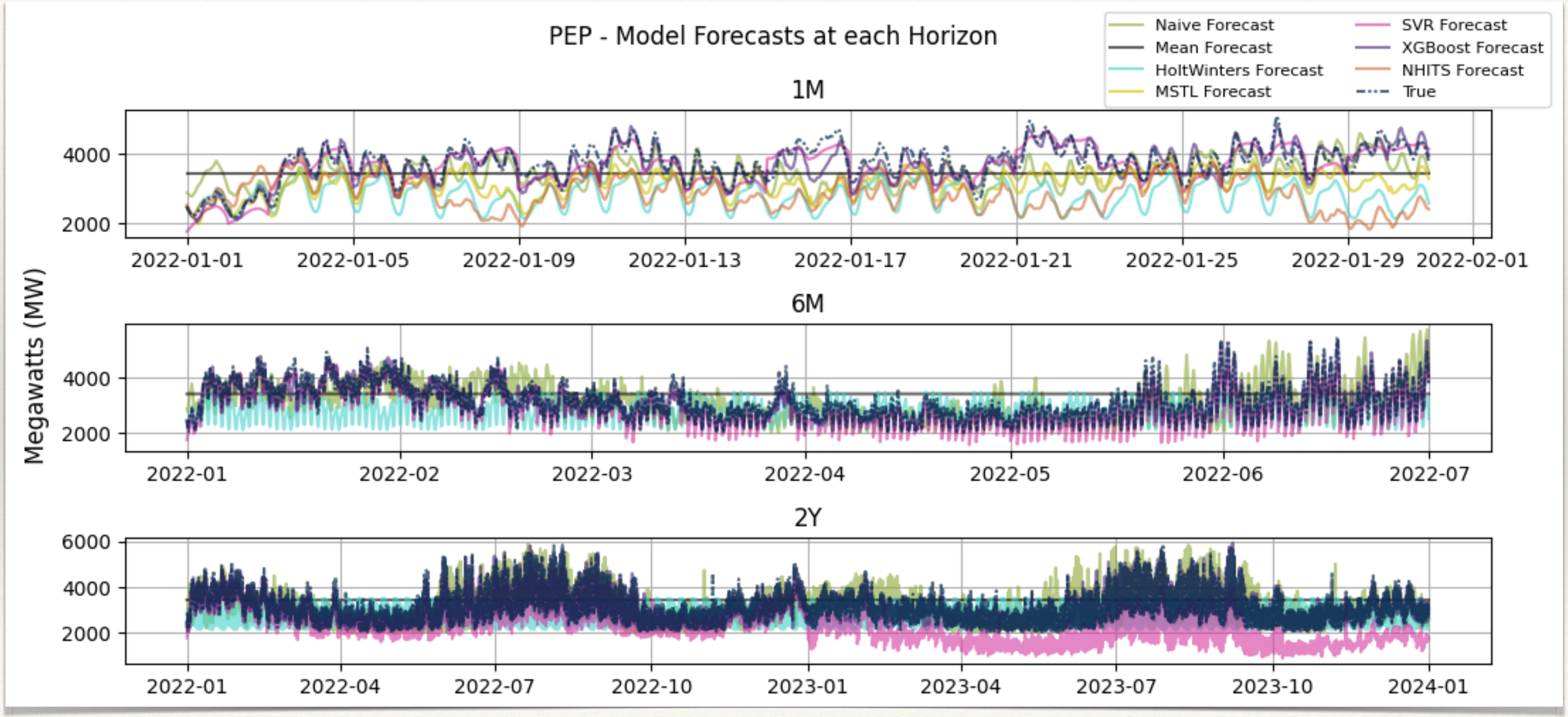
…N-HiTS interpolates across blocks which each specializing in separate signal frequencies.



Results | Model & Horizon Plots | Short



Results | Model & Horizon Plots | Long



Model Evaluation & Metrics

Symmetric Mean Absolute Percentage Error (sMAPE):

Ensures comparisons of error across different models and forecasting horizons are normalized.

$$\text{sMAPE} = \frac{200}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}$$

Root Mean Squared Error (RMSE):

Easily interpretable quantitative metric resulting in a value that is in the same unit as the response. Also, heavily penalizes predictions the further they are from the actual value. Normalized variation also included to compare zones with different scales.

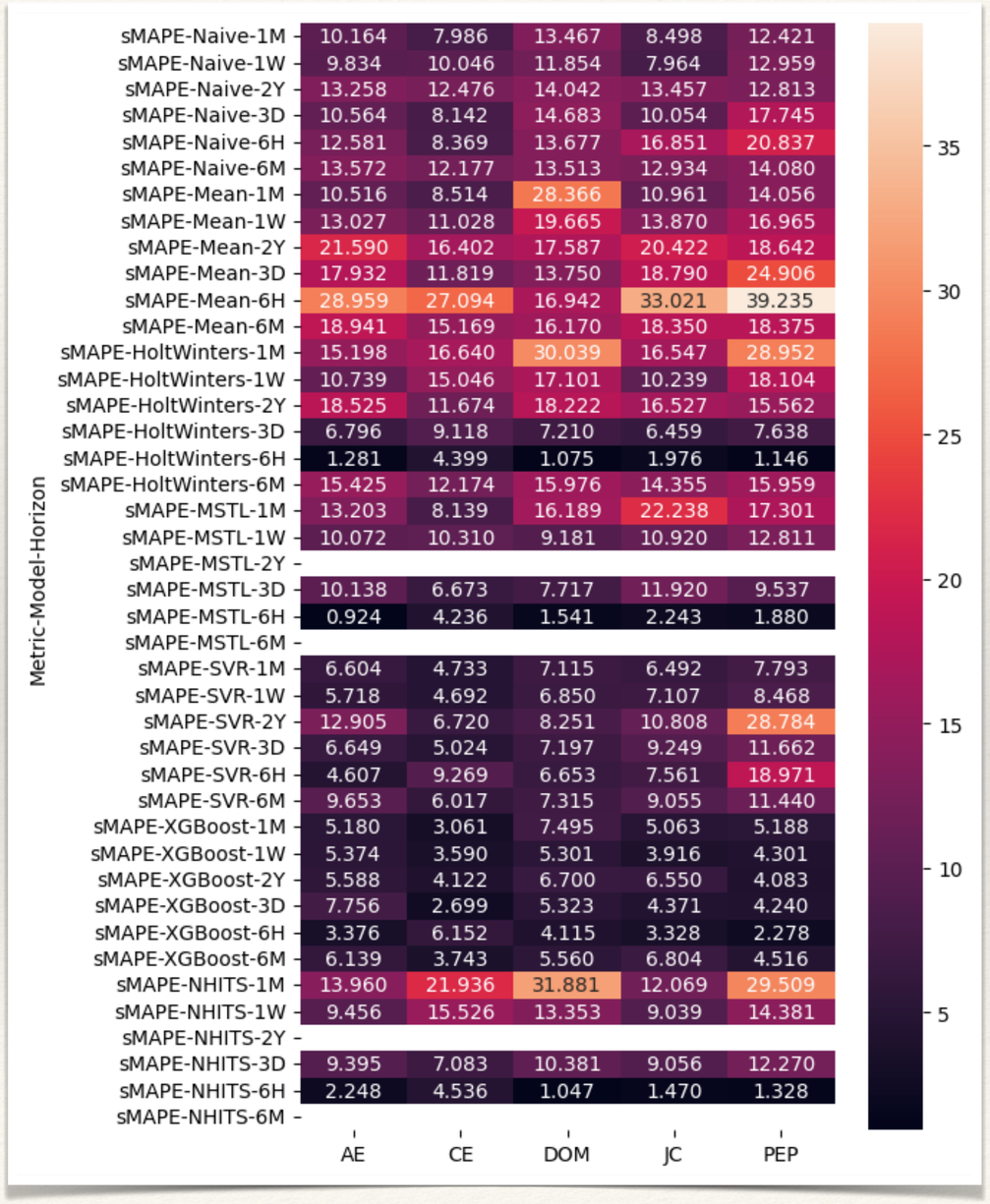
Horizon	Model	sMAPE	NRMSE
Short-Term	Naive	—	—
Long-Term	Naive	—	—
Short-Term	XGBoost	—	—
…	…	…	…

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$$

&

$$\text{NRMSE} = \frac{RMSE}{\bar{Y}}$$

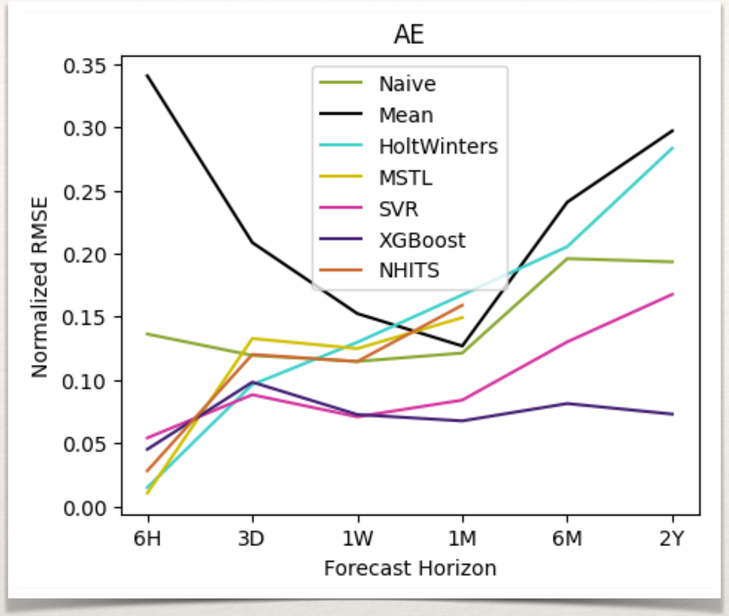
Results | sMAPE - Heatmap

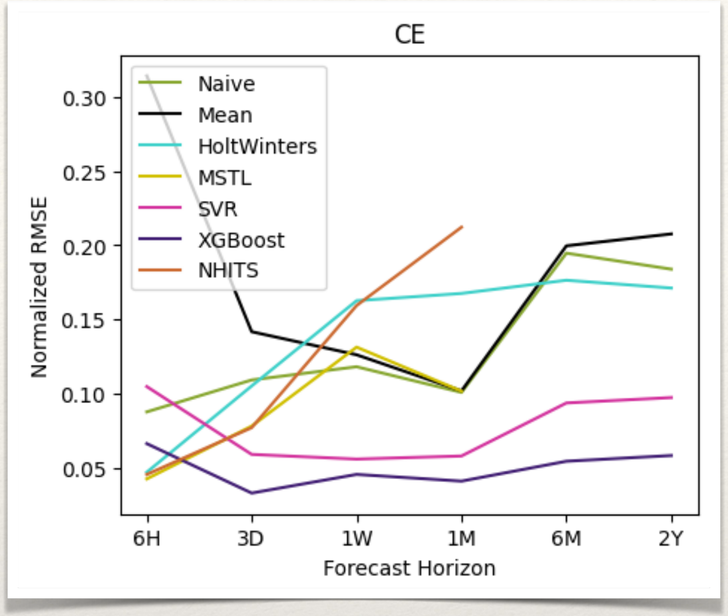


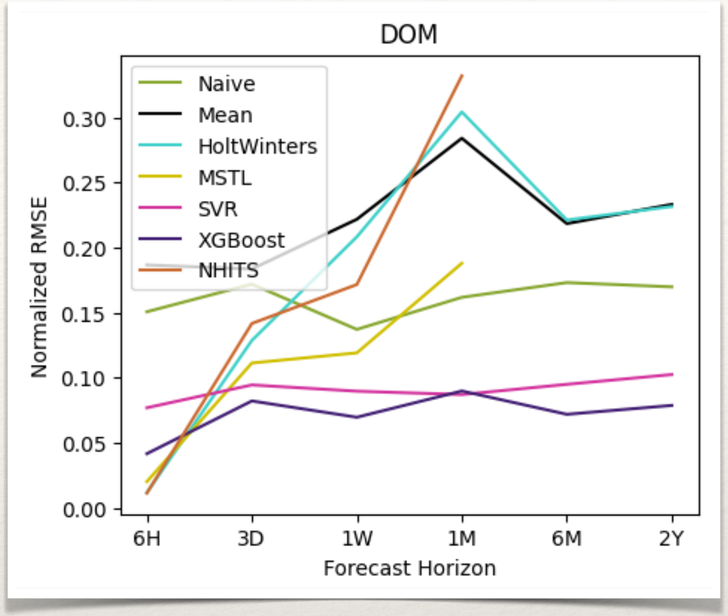
Heatmap of the results for every model and horizon.

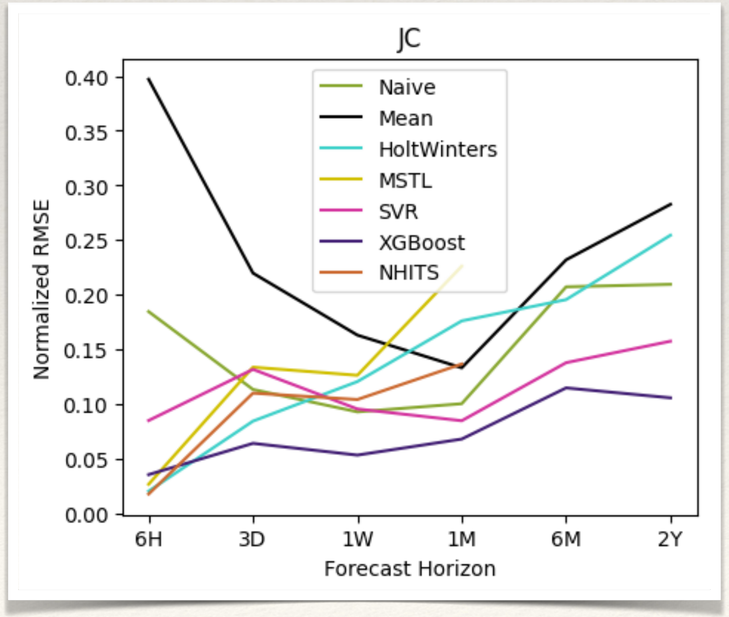
- PEP scored worse than other zones.
- XGBoost generalized across horizons the best.
- HoltWinters, MSTL, N-HiTS excelled on the 6 hour horizon.

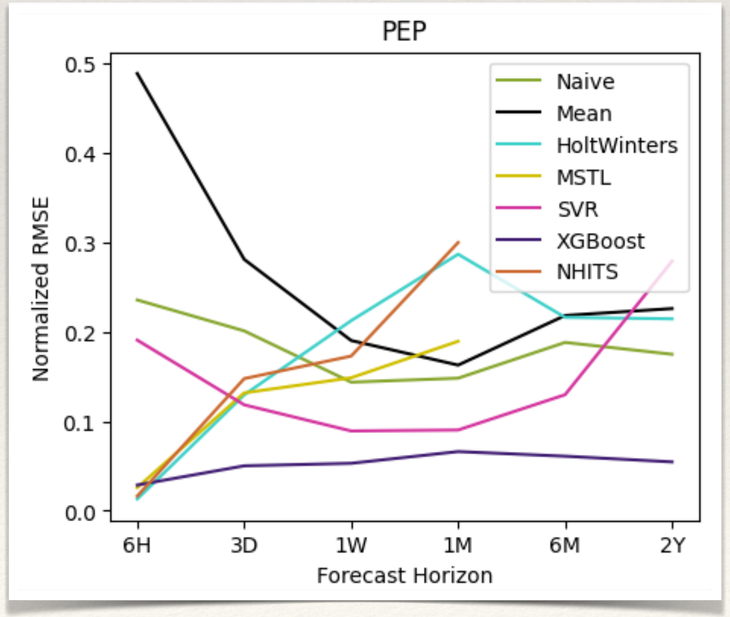
Results | NRMSE by Zone & Horizon











Results | Metrics

NRMSE

Metric	Model	Horizon	AE	CE	DOM	JC	PEP
NormRMSE	Naive	6H	0.136277	0.087492	0.150645	0.184597	0.235583
	Mean	6H	0.340522	0.314349	0.186621	0.397084	0.488823
	HoltWinters	6H	0.014868	0.046767	0.011711	0.020384	0.013049
	MSTL	6H	0.010456	0.042240	0.020137	0.026714	0.025732
	SVR	6H	0.054108	0.104567	0.076867	0.085008	0.190672
	XGBoost	6H	0.045109	0.066064	0.041571	0.035517	0.028640
	NHITS	6H	0.028110	0.045254	0.011302	0.017790	0.016293
NormRMSE	Naive	3D	0.119357	0.109039	0.171870	0.113405	0.200999
	Mean	3D	0.208615	0.141610	0.183489	0.219810	0.280960
	HoltWinters	3D	0.096301	0.104986	0.128598	0.084460	0.129481
	MSTL	3D	0.132737	0.077898	0.111262	0.133788	0.131693
	SVR	3D	0.088246	0.058785	0.094406	0.131786	0.118316
	XGBoost	3D	0.098207	0.032681	0.082099	0.064133	0.050121
	NHITS	3D	0.120104	0.076831	0.141640	0.110050	0.147622
NormRMSE	Naive	1W	0.114612	0.117962	0.137086	0.092925	0.143650
	Mean	1W	0.152446	0.126032	0.221700	0.163112	0.190239
	HoltWinters	1W	0.129742	0.162546	0.208207	0.120589	0.212457
	MSTL	1W	0.124733	0.131192	0.119044	0.126353	0.148563
	SVR	1W	0.070804	0.055620	0.089589	0.095575	0.089091
	XGBoost	1W	0.072556	0.045279	0.069561	0.053372	0.052927
	NHITS	1W	0.114543	0.159284	0.171549	0.104196	0.172830
NormRMSE	Naive	1M	0.121239	0.100637	0.161737	0.100371	0.148098
	Mean	1M	0.126805	0.101653	0.283969	0.133212	0.162745
	HoltWinters	1M	0.167221	0.167445	0.304187	0.176096	0.286739
	MSTL	1M	0.149286	0.101509	0.187906	0.226123	0.189383
	SVR	1M	0.084019	0.057702	0.087019	0.084813	0.090183
	XGBoost	1M	0.067560	0.040742	0.089747	0.068048	0.066061
	NHITS	1M	0.159065	0.212156	0.331945	0.136681	0.299784
NormRMSE	Naive	6M	0.195920	0.194636	0.173055	0.207272	0.188107
	Mean	6M	0.240628	0.199582	0.218384	0.231947	0.218194
	HoltWinters	6M	0.205409	0.176356	0.221137	0.195561	0.216097
	MSTL	6M	nan	nan	nan	nan	nan
	SVR	6M	0.130074	0.093505	0.094867	0.137879	0.129636
	XGBoost	6M	0.081284	0.054166	0.071832	0.114838	0.060866
	NHITS	6M	nan	nan	nan	nan	nan
NormRMSE	Naive	2Y	0.193376	0.183827	0.169782	0.209542	0.174865
	Mean	2Y	0.296964	0.207625	0.233131	0.282705	0.225961
	HoltWinters	2Y	0.283207	0.171086	0.231524	0.254468	0.214478
	MSTL	2Y	nan	nan	nan	nan	nan
	SVR	2Y	0.167635	0.097154	0.102399	0.157525	0.278998
	XGBoost	2Y	0.072943	0.058035	0.078654	0.105739	0.054516
	NHITS	2Y	nan	nan	nan	nan	nan

sMAPE

Metric	Model	Horizon	AE	CE	DOM	JC	PEP
sMAPE	Naive	6H	12.580831	8.368642	13.676927	16.850913	20.837337
	Mean	6H	28.959313	27.093643	16.941791	33.020723	39.234910
	HoltWinters	6H	1.281159	4.399481	1.074516	1.976006	1.146197
	MSTL	6H	0.924006	4.236191	1.540820	2.242525	1.879716
	SVR	6H	4.607263	9.269047	6.653134	7.561048	18.970828
	XGBoost	6H	3.375682	6.152227	4.115225	3.328029	2.278010
	NHITS	6H	2.248364	4.536223	1.046940	1.470468	1.328477
sMAPE	Naive	3D	10.564294	8.141624	14.682704	10.053777	17.745348
	Mean	3D	17.931932	11.818520	13.749959	18.790199	24.905756
	HoltWinters	3D	6.795567	9.117568	7.209524	6.459062	7.638461
	MSTL	3D	10.137742	6.672670	7.716595	11.919681	9.536786
	SVR	3D	6.649310	5.024464	7.196546	9.249314	11.662376
	XGBoost	3D	7.755542	2.698701	5.322590	4.371418	4.240230
	NHITS	3D	9.395042	7.082582	10.380827	9.056158	12.270452
sMAPE	Naive	1W	9.833909	10.046109	11.853713	7.963604	12.959489
	Mean	1W	13.027409	11.028017	19.665086	13.869585	16.964675
	HoltWinters	1W	10.738640	15.046106	17.100893	10.238545	18.103634
	MSTL	1W	10.071614	10.309518	9.181221	10.920468	12.810960
	SVR	1W	5.717646	4.692355	6.849986	7.107230	8.467719
	XGBoost	1W	5.374305	3.589565	5.300548	3.915818	4.300709
	NHITS	1W	9.455587	15.526011	13.352938	9.039466	14.380739
sMAPE	Naive	1M	10.163846	7.986211	13.466885	8.498115	12.421239
	Mean	1M	10.515516	8.513869	28.365910	10.961452	14.056335
	HoltWinters	1M	15.198061	16.639998	30.038587	16.547223	28.951588
	MSTL	1M	13.202894	8.138927	16.189155	22.238330	17.300937
	SVR	1M	6.603718	4.732827	7.115081	6.492116	7.793409
	XGBoost	1M	5.180421	3.061406	7.495047	5.063363	5.188263
	NHITS	1M	13.960246	21.936399	31.881035	12.069495	29.509419
sMAPE	Naive	6M	13.572251	12.176701	13.512928	12.933855	14.080367
	Mean	6M	18.941483	15.168616	16.169793	18.350397	18.375323
	HoltWinters	6M	15.425375	12.174435	15.976172	14.355150	15.959039
	MSTL	6M	nan	nan	nan	nan	nan
	SVR	6M	9.653313	6.016505	7.315464	9.054637	11.440345
	XGBoost	6M	6.139266	3.743250	5.560432	6.804394	4.516422
	NHITS	6M	nan	nan	nan	nan	nan
sMAPE	Naive	2Y	13.257603	12.475873	14.042065	13.456625	12.813002
	Mean	2Y	21.589680	16.402006	17.586856	20.421968	18.642190
	HoltWinters	2Y	18.524866	11.674018	18.222272	16.527110	15.562225
	MSTL	2Y	nan	nan	nan	nan	nan
	SVR	2Y	12.905069	6.720007	8.251229	10.808048	28.784374
	XGBoost	2Y	5.588003	4.122110	6.700303	6.550198	4.082743
	NHITS	2Y	nan	nan	nan	nan	nan

6 Hours

3 Days

1 Week

1 Month

6 Months

2 Years

Green: Best score in forecast horizon.

Red: Worse score than either baseline model.

nan: Model not fitted for that horizon due to hardware limitations.

Results | Compute

NRMSE

Metric	Model	Horizon	AE	CE	DOM	JC	PEP
NormRMSE	Naive	6H	0.136277	0.087492	0.150645	0.184597	0.235583
	Mean	6H	0.340522	0.314349	0.186621	0.397084	0.488823
	HoltWinters	6H	0.014868	0.046767	0.011711	0.020384	0.013049
	MSTL	6H	0.010456	0.042240	0.020137	0.026714	0.025732
	SVR	6H	0.054108	0.104567	0.076867	0.085008	0.190672
	XGBoost	6H	0.045109	0.066064	0.041571	0.035517	0.028640
	NHITS	6H	0.028110	0.045254	0.011302	0.017790	0.016293
NormRMSE	Naive	3D	0.119357	0.109039	0.171870	0.113405	0.200999
	Mean	3D	0.208615	0.141610	0.183489	0.219810	0.280960
	HoltWinters	3D	0.096301	0.104986	0.128598	0.084460	0.129481
	MSTL	3D	0.132737	0.077898	0.111262	0.133788	0.131693
	SVR	3D	0.088246	0.058785	0.094406	0.131786	0.118316
	XGBoost	3D	0.098207	0.032681	0.082099	0.064133	0.050121
	NHITS	3D	0.120104	0.076831	0.141640	0.110050	0.147622
NormRMSE	Naive	1W	0.114612	0.117962	0.137086	0.092925	0.143650
	Mean	1W	0.152446	0.126032	0.221700	0.163112	0.190239
	HoltWinters	1W	0.129742	0.162546	0.208207	0.120589	0.212457
	MSTL	1W	0.124733	0.131192	0.119044	0.126353	0.148563
	SVR	1W	0.070804	0.055620	0.089589	0.095575	0.089091
	XGBoost	1W	0.072556	0.045279	0.069561	0.053372	0.052927
	NHITS	1W	0.114543	0.159284	0.171549	0.104196	0.172830
NormRMSE	Naive	1M	0.121239	0.100637	0.161737	0.100371	0.148098
	Mean	1M	0.126805	0.101653	0.283969	0.133212	0.162745
	HoltWinters	1M	0.167221	0.167445	0.304187	0.176096	0.286739
	MSTL	1M	0.149286	0.101509	0.187906	0.226123	0.189383
	SVR	1M	0.084019	0.057702	0.087019	0.084813	0.090183
	XGBoost	1M	0.067560	0.040742	0.089747	0.068048	0.066061
	NHITS	1M	0.159065	0.212156	0.331945	0.136681	0.299784
NormRMSE	Naive	6M	0.195920	0.194636	0.173055	0.207272	0.188107
	Mean	6M	0.240628	0.199582	0.218384	0.231947	0.218194
	HoltWinters	6M	0.205409	0.176356	0.221137	0.195561	0.216097
	MSTL	6M	nan	nan	nan	nan	nan
	SVR	6M	0.130074	0.093505	0.094867	0.137879	0.129636
	XGBoost	6M	0.081284	0.054166	0.071832	0.114838	0.060866
	NHITS	6M	nan	nan	nan	nan	nan
NormRMSE	Naive	2Y	0.193376	0.183827	0.169782	0.209542	0.174865
	Mean	2Y	0.296964	0.207625	0.233131	0.282705	0.225961
	HoltWinters	2Y	0.283207	0.171086	0.231524	0.254468	0.214478
	MSTL	2Y	nan	nan	nan	nan	nan
	SVR	2Y	0.167635	0.097154	0.102399	0.157525	0.278998
	XGBoost	2Y	0.072943	0.058035	0.078654	0.105739	0.054516
	NHITS	2Y	nan	nan	nan	nan	nan

sMAPE

Metric	Model	Horizon	AE	CE	DOM	JC	PEP
sMAPE	Naive	6H	12.580831	8.368642	13.676927	16.850913	20.837337
	Mean	6H	28.959313	27.093643	16.941791	33.020723	39.234910
	HoltWinters	6H	1.281159	4.399481	1.074516	1.976006	1.146197
	MSTL	6H	0.924006	4.236191	1.540820	2.242525	1.879716
	SVR	6H	4.607263	9.269047	6.653134	7.561048	18.970828
	XGBoost	6H	3.375682	6.152227	4.115225	3.328029	2.278010
	NHITS	6H	2.248364	4.536223	1.046940	1.470468	1.328477
sMAPE	Naive	3D	10.564294	8.141624	14.682704	10.053777	17.745348
	Mean	3D	17.931932	11.818520	13.749959	18.790199	24.905756
	HoltWinters	3D	6.795567	9.117568	7.209524	6.459062	7.638461
	MSTL	3D	10.137742	6.672670	7.716595	11.919681	9.536786
	SVR	3D	6.649310	5.024464	7.196546	9.249314	11.662376
	XGBoost	3D	7.755542	2.698701	5.322590	4.371418	4.240230
	NHITS	3D	9.395042	7.082582	10.380827	9.056158	12.270452
sMAPE	Naive	1W	9.833909	10.046109	11.853713	7.963604	12.959489
	Mean	1W	13.027409	11.028017	19.665086	13.869585	16.964675
	HoltWinters	1W	10.738640	15.046106	17.100893	10.238545	18.103634
	MSTL	1W	10.071614	10.309518	9.181221	10.920468	12.810960
	SVR	1W	5.717646	4.692355	6.849986	7.107230	8.467719
	XGBoost	1W	5.374305	3.589565	5.300548	3.915818	4.300709
	NHITS	1W	9.455587	15.526011	13.352938	9.039466	14.380739
sMAPE	Naive	1M	10.163846	7.986211	13.466885	8.498115	12.421239
	Mean	1M	10.515516	8.513869	28.365910	10.961452	14.056335
	HoltWinters	1M	15.198061	16.639998	30.038587	16.547223	28.951588
	MSTL	1M	13.202894	8.138927	16.189155	22.238330	17.300937
	SVR	1M	6.603718	4.732827	7.115081	6.492116	7.793409
	XGBoost	1M	5.180421	3.061406	7.495047	5.063363	5.188263
	NHITS	1M	13.960246	21.936399	31.881035	12.069495	29.509419
sMAPE	Naive	6M	13.572251	12.176701	13.512928	12.933855	14.080367
	Mean	6M	18.941483	15.168616	16.169793	18.350397	18.375323
	HoltWinters	6M	15.425375	12.174435	15.976172	14.355150	15.959039
	MSTL	6M	nan	nan	nan	nan	nan
	SVR	6M	9.653313	6.016505	7.315464	9.054637	11.440345
	XGBoost	6M	6.139266	3.743250	5.560432	6.804394	4.516422
	NHITS	6M	nan	nan	nan	nan	nan
sMAPE	Naive	2Y	13.257603	12.475873	14.042065	13.456625	12.813002
	Mean	2Y	21.589680	16.402006	17.586856	20.421968	18.642190
	HoltWinters	2Y	18.524866	11.674018	18.222272	16.527110	15.562225
	MSTL	2Y	nan	nan	nan	nan	nan
	SVR	2Y	12.905069	6.720007	8.251229	10.808048	28.784374
	XGBoost	2Y	5.588003	4.122110	6.700303	6.550198	4.082743
	NHITS	2Y	nan	nan	nan	nan	nan

6 Hours

3 Days

1 Week

1 Month

6 Months

2 Years

Times include training and prediction computes.
Some model predictions are near instant.

Seasonal Naive: Instant

Mean: Instant

Holt-Winters: 9 min

MSTL: 95 min*

SVR: 59 min

XGBoost: 2 min

N-HiTS: 9 min => 570 min†

* MSTL only forecasted up to 1 Month

† N-HiTS only forecasted up to 1 Month, but also only trained on 30 days of data. 570 min is an estimate by linear extrapolation.

Key Findings & Future Work

- XGBoost shows strong performance in all forecast horizons
  - Low computational demands
  - Quick training and prediction times.
- Statistical methods, MSTL and Holt-Winters, excel in short-term forecasts.
- Deep learning methods like N-HiTS, show great efficacy even with small training set.

- Test models on more robust hardware.
- Consider statistical measures analyzing efficiency accuracy tradeoffs.
- Further data processing and feature engineering
  - Adding Fourier series or signal smoothing, could enhance model performance, especially for capturing multiple seasonal periods.

Project Timeline

Timeline	Agenda Item	Progress	Done
Days 1 - 5	Data Mining	–	X
	Hourly Load Data	–	X
	Hourly Generation Data	–	X
	Weather Data	–	X
	Cleaning/Preprocessing	–	X
	Datetime Alignment	–	X
	Aggregate by Zones	–	X
	Outlier Inspection	–	X
	Corrupted Data Removal	–	X
	Feature Engineering	–	X
	Exploratory Data Analysis	–	X
	Train/Val/Test Split	–	X
Days 6-8	Anomaly Detection		
	Forecast Model Setup	–	X
	Training	–	X
	Val. Optimizations	–	X
	Build Eval. Functions	–	X
Days 9-10	Test Results	–	X
	Discussion	–	X
	Conclusion	–	X

Alternate Approaches

If the target deadlines are not met the following can be changed:

1. The number of models can be reduced, leaving at least one remaining from each model type.
2. Model complexity can be reduced by utilizing more out-of-the-box approaches.
3. Further reduce the scope to solely focus on load forecasting.