

# Energy Grid Forecasting and Anomaly Detection

Cody Hill  
University of Colorado Boulder  
Boulder, USA  
cody.hill-1@colorado.edu

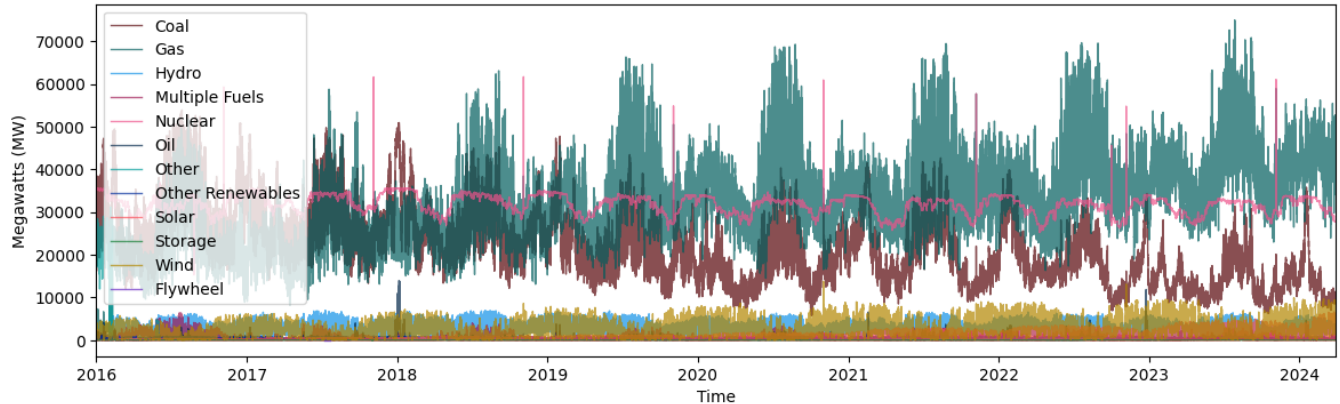


Figure 1: Energy Generated in Megawatts (MW) by Fuel Type.

## ABSTRACT

Energy forecasting is a vital part of today's energy markets and operations. It is used to set the cost of electricity, determine when power generation should be increased, and decide when power needs to be directed to or from other regions of the electrical grid. Electrical energy demand continues to rise, especially with the recent widespread adoption of electric vehicles and heat pumps. As a result, the electrical grid will need to become more robust and adaptive as this increase in demand comes in tandem with a transition away from fossil fuels and into a more diversified energy market and with it, more complex cycles. These circumstances make accurate forecasting and periodicity identification of electrical loads and generation that much more important, and it is necessary to research the differences in forecasting applications and relevance in various short-term, medium-term, and long-term settings.

Many of the statistical time series forecasting techniques of the past are still in use today, each with their own strengths and weaknesses, often trading compute efficiency with lesser accuracy in long-term forecast horizons or ability to capture multiple seasonal periods. More modern machine learning models and neural network-based architectures tend to generalize through forecasting horizons better, but often at great cost of compute resources making them inefficient for short forecast horizon forecasts. Though because of the potential for complexity of energy load patterns, both categories see benefits in task-specific tuning. By evaluating each forecasting model's accuracy across various context windows and forecasting horizons, this project presents a thorough comparison

on which techniques are most relevant in the task-specific energy grid context and each forecast horizon.

## CCS CONCEPTS

• **Mathematics of computing** → **Time series analysis**; • **Computing methodologies** → **Supervised learning by regression**; *Classification and regression trees*; *Support vector machines*; *Neural networks*; • **Applied computing** → **Forecasting**.

## KEYWORDS

time series forecasting, anomaly detection, energy load, renewables, energy grid

### ACM Reference Format:

Cody Hill. 2024. Energy Grid Forecasting and Anomaly Detection. In *Proceedings of Data Mining Project (Data Mining Project '24)*. ACM, New York, NY, USA, 12 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 AUTHOR'S NOTE

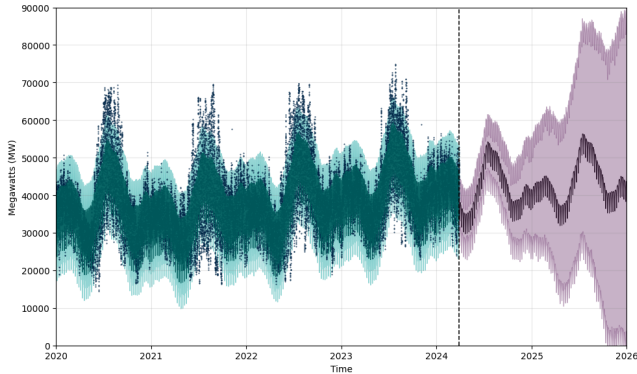
This document is to be used as a project proposal for the beginning planning stages of the project as well as a living-document as the project progresses. Up until project completion when an updated version will be created.

## 2 INTRODUCTION

Energy forecasting is an important task to not only determine the expected load on the energy grid and the requirements of energy providers to meet that demand by either ramping up production or diverting sources altogether [8]. Identifying hourly, daily, seasonal, or even yearly energy demands accurately has substantial impact on not only the logistics of the infrastructure of our energy grid, but also the complex energy marketplace. These intervals, or forecast

horizons, in the context of predictive models are defined in Table 1. Short-term forecasts can be useful for players in power markets where prices dynamically shift based on generation supply, customer demand, and financial spot and derivative contracts which are all based on energy forecasting [7]. Very short-term forecasts can be important for power distributors and operators that want to meet customer demands more efficiently. Long-term forecasts are important in infrastructure planning and energy provider and distributors long-term planning. An example of a long-term forecast can be seen in Figure 2.

In the past this predictive role was largely upheld with statistical point-estimate models, but with the growth of our predictive models along with the diversification of the energy grid, more modern probabilistic interval forecasting techniques have begun to be used [8, 13, 16]. This diversification of the energy grid not only comes from energy reforms by way of deregulating the national electric market in the 1990's, but emerging cultural desires and improved technology in renewable energies, driving a need for a better electrical grid [11].



**Figure 2: Long-Term Forecast of Natural Gas Energy Generation Using Meta's Prophet Model. Actual observations are represented by the blue points, forecasted fit and predictions are in teal and purple respectively.**

Electricity generation has always been mostly an on-demand industry and largely continues to be, where the energy we generate needs to be transmitted and used within a short period of time because we lack efficient systems for energy storage at scale [12]. This fact presents additional complications when many renewable energies only produce energy in certain conditions, making its effective use more reliant on forecasting to navigate a more diverse and complex power grid. Additionally, with the cultural and regulatory push towards end-users adopting the use of electric appliances, electric vehicles, and heat pumps, energy demands will continue to change [6]. These circumstances mean accurate forecasting is becoming even more important with improvements having the potential to better reduce reliance on fossil-fuel energies by more efficiently filling in gaps of energy demands with a more diversified and efficient energy grid.

There is little consensus among researchers and professionals about what exemplifies state-of-the-art in time series forecasting.

**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	$X < 1 \text{ Hour}$
2	Short-Term	$1 \text{ Hour} \leq X < 1 \text{ Week}$
3	Medium-Term	$1 \text{ Week} < X \leq 1 \text{ Year}$
4	Long-Term	$X > 1 \text{ Year}$

This seems to be due to many things including that time series forecasting is heavily task-specific, meaning one model may perform well with one type of data and forecast horizon, but might perform poorly when compared to other techniques with a different horizon or data attributes such as seasonality [17]. This is why it is still important to experiment and compare forecasting techniques on domain-specific time series data.

### 3 RELATED WORK

Seasonal Autoregressive Integrated Moving Average (SARIMA) and Exponential Smoothing models tend to be useful in short and medium-term forecasts but can experience limitations modeling complex periodicity and interactions [9, 14]. Newer, but certainly not state-of-the-art, machine learning techniques such as Support Vector Machines, XGBoost, and a variety of Long Short-Term Memory (LSTM) network architectures tend to generalize better and excel across multiple forecast horizons [14, 15].

Hybrid models and ensemble methods, utilizing a variety of preprocessing techniques and combining models into a unified framework, are becoming more popular as of late. The process typically involves splitting the job of extracting different types of features from the data, say the temporal and spatial relationships of the time series, and feeding those outputs into yet another separate model to perform the forecasting [10]. This technique combines the strengths of multiple models resulting in improved results across all forecast horizons [9, 10]. Many of these still utilize LSTM layers as their attention mechanism [15].

Transformer-based models and other frameworks developed and popularized in the Large Language Model (LLM) space have found a foothold in time series forecasting in the last few years as well, but their use in time series forecasting is still being researched [18]. Similar to the LSTM, transformer-based models capture complex context and relationships in data, but across greater distances of the training corpus with increased efficiency because of their ability to be parallelized effectively [19]. This type of model benefits from pre-training on a large corpus of diverse data which simply does not exist in the public domain for time series data [1]. With more time and data, researchers believe transformer-based models can be effective at time series forecasting [1, 19].

### 4 METHODS AND PROPOSED WORK

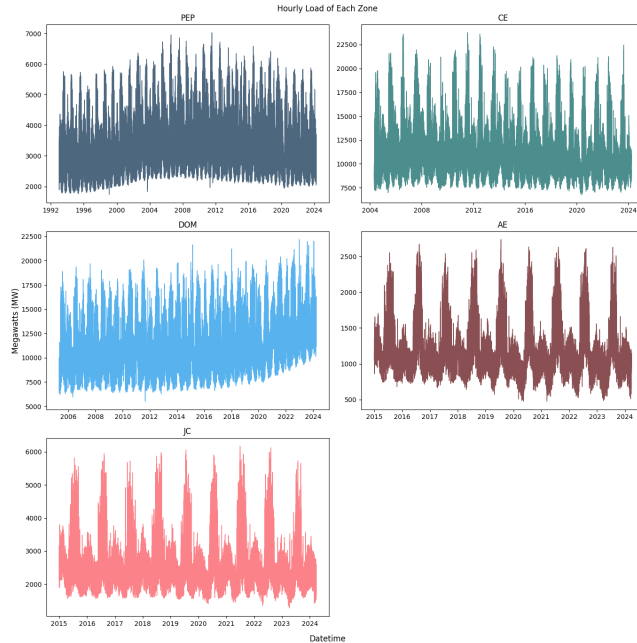
Several models stemming from different statistical, machine-learning, and deep learning families will be trained to perform electricity load forecasting across the different forecast horizon intervals listed in Table 1. Each model's amount of compute required for training

and evaluation scores will be compared across six different forecast horizons and 5 different time series. Due to the sample rate of the load data being hourly, the shortest forecast horizon tested will be 6 hours. See evaluation section for more details on the specific metrics used.

#### 4.1 Data

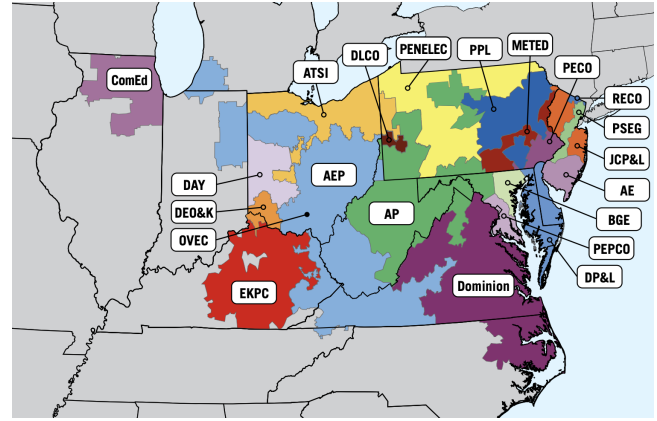
Electric load data was collected by the Pennsylvania-New Jersey-Maryland Interconnection (PJM). PJM is a regional electrical transmission organization which coordinates electricity transmission in all or parts of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia, and the District of Columbia [2]. A map of this can be seen in Figure 4.

Weather data was retrieved from the National Oceanic and Atmospheric Administration's (NOAA) Regional Climate Center (RCC) Applied Climate Information System (ACIS) [3]. The data was manually downloaded by selecting a weather station and choosing relevant dates and features with the daily data listings form.



**Figure 3: Megawatt vs Time plot of the selected zones.**

The electric load data sourced from PJM is a set of hourly load in megawatt-hours verified by the individual electric distribution companies [4]. The hourly load data is flagged with geographical location data which will be leveraged to include regional weather data as a covariate in the models. After data discovery, 5 zones were chosen to represent the electrical load data for their sample size, variety in signal shape, and data continuity. The following zones were chosen: 1.) "AE" - Atlantic City Electric Co.. 2.) "CE" - ComEd. 3.) "DOM" - Dominion. 4.) "JC" - Jersey Central Power & Light 5.) "PEP" - Potomac Electric Power Co.. A plot of each zone's time series can be seen at Figure 3.



**Figure 4: Regional Transmission Zones Labeled by Distributor IDs [5].**

The weather data has a daily sample frequency which was the only data available at the time. Hourly data would have been preferable to match the frequency of the load data, but even at the mismatched sample rate, will be a valuable exogenous covariate. Features included in this dataset are the maximum and minimum recorded temperature, average temperature, precipitation in inches. After the 5 zones were selected, weather data was retrieved to match. Along with the zone map in Figure 4, weather stations were chosen to represent each zone under the following criteria: 1.) Weather station is geographically central to the zone. 2.) Weather station uptime and reliability (airport weather stations often chosen for this quality).

An overview of the full datasets can be seen in Table 2.

**Table 2: Dataset size and date ranges.**

Dataset	Size (Rows)	Date Range
Hourly Load	776,016	1993/01/01 - 2024/03/27
Daily Weather	32,400	1993/01/01 - 2024/04/09

#### 4.2 Preprocessing

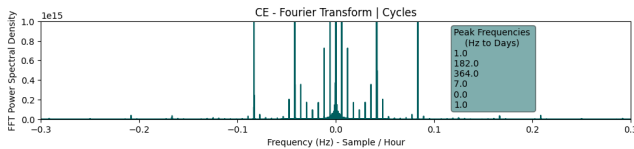
The electrical load data is retrieved from PJM's Data Miner 2 and each year is downloaded into a .csv file. These .csv files are then zipped so they can be pushed to Github. All data is imported into Pandas dataframes as it will allow all the following data transformations to be frictionless. Also, since several different models will be used, Pandas allows users to easily reformat the data between float arrays, tensors, and otherwise.

The first step in preprocessing the datasets is checking for any missing dates. Any missing rows are added by inserting the average value of the two surrounding observed data. Then the data is analyzed for outliers, using a tumbling window method, replacing any identified outliers with the median of the window size. The two datasets is then be date aligned and merged.

### 4.3 Feature Engineering

Feature engineering is a large part of time series forecasting, typically past values of the target variable are highly correlated to future values. Several methods were used to generate covariates that either have a signal smoothing effect, identifies seasonal periods, or encodes exogenous information.

- (1) **Date features** are encoded in categorical features representing Hours, Days, Months, Years, and Day of Year - this will help the models identify recurring temporal relationships, or seasonality, in the target variable.
- (2) **Rolling mean** of the target variable of the previous 1 month of data points, adding the rolling mean helps smooth out the signal, increasing model performance.
- (3) **Lagged features** at specific intervals are useful if an effort is made to match known seasonal periods within the data. Lagged features of the target variable at 1 hour, 6 hours, 24 hours, 1 week, and 1 month will be used. *Note, a 1 year lagged feature would also be effective, but would have null values for the first year which can be problematic for some models and those rows would need to be removed.*
- (4) **Fourier transform spectral density**, similar to lagged features, helps identify correlated seasonal periods. By performing the Fast Fourier Transform on the target variable and taking the absolute value of it, the most relevant seasonal frequencies will have the greatest magnitudes. An example of the resulting spectral density plot can be seen in Figure 5.



**Figure 5: Fourier transform spectral density plot of the ComEd (CE) zone. To cast frequency units to days:  $\frac{1}{freq \cdot 24}$ .**

- (5) **First and Second-Order Derivatives/Differencing** finds the instantaneous rate of change between two observed data points and the acceleration of the rate of change respectively. This encodes direction, magnitude, and change of direction in time series data. An example plot of both orders of derivatives can be seen in Figure 12 in Appendix A.

Specific intervals of features such as lag, rolling mean, derivatives need to be considered carefully when used as exogenous features calculated from the target variable. This can be a great source of data leakage into the testing set if these are not data you would have in a prediction setting. One method to overcome this is to take a naive approach and assign last season's values. Another is to create a hybrid model which forecasts the exogenous features to include in the forecast. Finally, a rolling forecast can be used predict new datapoints and retrain the model with those predictions up until the intended forecast horizon is achieved. Here, the shorter length interval features have been excluded to prevent data leakage, but will remain implemented but not included as an exercise in time series feature engineering.

### 4.4 Tools and Techniques

The load data is then analyzed and plotted for historical trends and to identify periodicity (cycles) at different context window intervals which helps set model expectations for the various forecast horizons.

Models of varying complexity and methodology were implemented to better represent the broad spectrum of techniques used today, in an attempt to capture a good representation of which contemporary models work best for different forecast horizons.

#### Proposed models:

##### Baseline

- Seasonal Naive Forecast Model
- Mean

##### Statistical

- Seasonal Autoregressive Integrated Moving Average (SARIMA)\*
- Holt-Winters' Method
- Multiple Seasonal-Trend decomposition using LOESS (MSTL)

##### Machine Learning

- Support Vector Regression (SVR)
- XGBoost

##### Deep Learning

- Neural Hierarchical Interpolation (N-HITS)
- Temporal Fusion Transformers (TFT)\*

\* SARIMA and TFT implementations were attempted but removed from results section as they both required more computation than the hardware could reasonably dedicate to their training. More details in the discussion section.

Some models require data transformation or normalization to perform effectively. For example, the SVM and TFT increase performance with scaled feature matrices.

$$Z = \frac{(X - \bar{X})}{S}$$

A standard Z standardization is used which centers inputs around 0 with the sample mean, and scales them with the sample standard deviation.

## 5 EVALUATION

### 5.1 Training, Validation, and Testing Data Split

The data will be split into a training, validation, testing split. A validation set is necessary here because several of the proposed forecasting models will require hyperparameter tuning. Once the data has been aggregated, cleaned, and finalized the exact ratio may be adjusted to ensure the test set is large enough to span the largest forecast horizon at least one or several times, depending on what is feasible with the data on hand.

**Table 3: Dataset size and date ranges.**

Split Set	Date Range
Training	1993/01/01 - 2018/12/31
Validation	2019/01/01 - 2021/12/31
Testing	2022/01/01 - 2024/03/27



To avoid data leakage and maintaining temporal relationships, the test set will always be assigned the most recent dates with zero overlap in the training or validation sets. Also, it is ensured that each set contains a reasonable amount of each periodicity. Details about the resulting split can be seen in Table 3 and Figure 6.

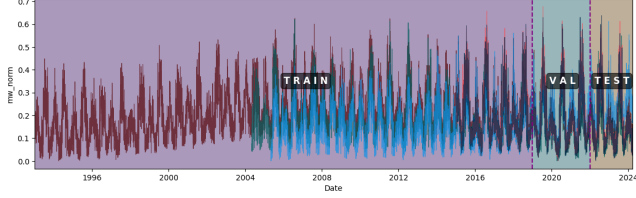


Figure 6: Plot of the training, validation, and testing split.

An important question arises when using a validation set to tune models in a time series domain because of the temporal nature of the data. Oftentimes, but not always, the most recent observations have the biggest impact on future forecasted predictions. This can be an issue in the process of hyperparameter optimization because the fit model now contains a time-gap between the training set and the testing set.

A choice remains:

- (1) Use the validation set only in model selection and hyperparameter tuning, knowing it may not generalize well to new data.
- (2) After using the validation set to specify the model and hyperparameters, refit including the validation set, hoping this will not overfit the now combined training and validation sets.

Both have their merits and are highly dependent on the dataset and target value. In this project, choice 2 has been the chosen after model optimizations are complete, the validation set is combined with the training set and used on the model's final fit.

## 5.2 Metrics

**Symmetric Mean Absolute Percentage Error (sMAPE)** will be the evaluation metric used to compare all models. This ensures comparisons of error across different models and forecasting horizons are normalized.

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

where  $y_i$  and  $\hat{y}_i$  are the actual values and forecasted values respectively, and  $n$  is the number of points in the set. This implementation outputs a percentage ranging from 0 to 200 percent.

One possible limitation of sMAPE is if a predicted value or actual value equals zero, the metric balloons to the maximum error value, and if both equals zero, the metric is undefined.

**Root Mean Squared Error (RMSE)** will also be used in model evaluation to complement sMAPE. The value of including RMSE is that it is an easily interpretable quantitative metric resulting in a value that is in the same unit as the response. It also heavily penalizes predictions the further they are from the actual value.

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

where  $y_i$  and  $\hat{y}_i$  are the actual values and forecasted values respectively, and  $n$  is the number of points in the set.

In conjunction with the RMSE, a **Normalized RMSE (NRMSE)** is also used to compare models between zones, since each zone have varying scales of the target variable.

$$\text{NRMSE} = \frac{\text{RMSE}}{\bar{Y}} \quad (3)$$

where  $\bar{Y}$  is the mean of the actual values.

## 6 DISCUSSION

After data discovery and feature engineering, during the preliminary model setup it was determined a few of the proposed models would not be moving forward to testing due to the size of the dataset and feature matrix.

The ARIMA-based models such as SARIMA, even when specified as univariate models, require substantial setup and tuning which on the machine it was being worked on became untenable. Train times for this and a few neural-network models including TFT, even with reduced training set sizes, were not able to be completed due to hardware limitations. Unfortunately these two models will not be completed to a scope able to compare results. However, this does add justification to the thesis that forecast horizon lengths and models need to be chosen at a task-specific level because lengthy training durations on short forecast horizon lengths heavily outweigh the balance of tradeoffs.

Furthermore, two models have reduced scope due to the same reasons above and the following concessions were made due to hardware limitations.

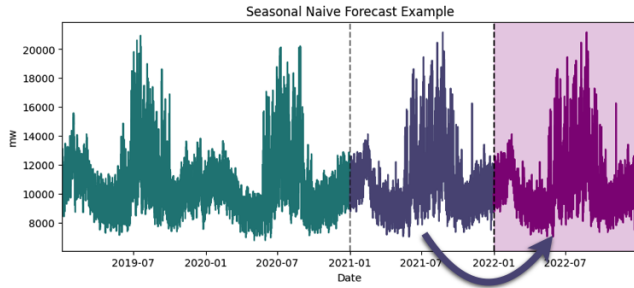
- **MSTL:** After model tuning on a reduced training set, the MSTL model was unable to train on the entirety of the combined training and validation set so the final fit was trained solely on the validation set. Also, forecast horizon was limited up to 1 month and failed to train to the 6 month and 2 year horizons, again due to hardware limitations.
- **N-HITS:** After model tuning on a reduced training set, the N-HITS model was also unable to train on the entirety of the combined training and validation sets. The final fit was performed on just the last 30 days of the validation set. This also limited its forecast horizon to up to 1 month.

During Exploratory Data Analysis (EDA), Fourier Transform analysis, and Autocorrelation (ACF) and Partial Autocorrelation (PACF) plot analysis it was noted that each zone's time series data contained multiple seasonalities. These seasonality periods included, 24 hour, 1 week, 6 month, 1 year cycles, and are important to understand when specifying time series models.

The following section includes a brief model summary and details of notable parameters.

- (1) **Seasonal Naive Forecast Model:** The Seasonal Naive Forecasting Model is a simple but often effective baseline model that can handle any length forecast horizon as long as there is at least the same amount of training data available. Forecasts of future values are equal to past values of the target

variable ('MW') shifted backwards one period of the input season. As we predict different forecast horizons, the length of the horizon can be used as a seasonal-lag to adjust the naive forecast lengths. For example, a forecast one week into the future is equal to the previously observed values from the same week last year. An example is shown in Figure 7.



**Figure 7: Example of a 1 year seasonal naive forecast. The prediction forecast is in the shaded area.**

- (2) **Mean:** A simple baseline model that calculates the mean of the historical time series data (training set) and uses that mean value to predict all future values.
- (3) **Holt-Winters' Method:** Holt-Winters' method (sometimes referred to as triple exponential smoothing) is an extension of exponential smoothing for data that contains both trend and seasonality. Only one seasonal period can be selected to capture.
  - During tuning, a seasonal period of 1 week performed the best. A period length of one year would also likely be effective, but training times became a constraint.
- (4) **Multiple Seasonal-Trend decomposition using LOESS (MSTL):** MSTL decomposes time series data into a trend component, multiple seasonal components, and a residual component. MSTL uses Seasonal-Trend decomposition using LOES (STL) to extract seasonal components from a time series. Unlike the Holt-Winters method, and as the name suggests, MSTL can handle multiple seasonalities.
  - Season periods included in the model: 24 hour, 1 week, 6 month, 1 year cycles.
- (5) **Support Vector Regression (SVR):** The SVR calculates a hyperplane that best fits the data points in a hyper-dimensional continuous space.
  - kernel: Radial Basis Function (RBF)
- (6) **XGBoost:** Ensemble gradient boosted tree method which sequentially builds trees.
  - max depth: 7
  - max leaf: None
  - estimators: 6000
- (7) **Neural Hierarchical Interpolation (N-HiTS):** Developed as an improvement on NBEATS, it uses several multilayer perceptrons (MLPs) with ReLU non-linearities to interpolate across blocks each specializing in separate signal frequencies.
  - epochs: 1000
  - learning rate: 0.001

- pool kernel size: [2, 2, 1]
- freq downsample: [4, 2, 1]
- mlp units: 3 \* [[512, 512]]

*Note: This is not an exhaustive list of parameters used.*

## 6.1 Project Timeline

Table 4 outlines the specific tasks and their projected schedule.

**Table 4: A general framework onto which the project will progress. This timeline is subject to change.**

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
Days 6-8	Exploratory Data Analysis	–	X
	Train/Val/Test Split	–	X
	Anomaly Detection		
	Forecast Model Setup	–	X
	Training	–	X
Days 9-10	Val. Optimizations	–	X
	Build Eval. Functions	–	X
	Test Results	–	X
	Discussion	–	X
	Conclusion	–	X

## 6.2 Challenges & Alternative Approaches

A few challenges arose throughout, impeding progress in several areas of the project. As mentioned in the discussion section, the SARIMA model was removed due to hardware limitations. MSTL and N-HiTS both had their scope reduced in limiting the length of forecast horizon and, in the case of N-HiTS, reducing training set size as well. Also, the scope of the project overall was reduced to only focus on the electrical load data, not also generation data, and anomaly detection was removed from the project as well.

The granularity of the data also posed issues in accurately forecasting the shorter forecast horizons. Since all collected data has a time-step (sample rate) granularity of one hour, predictions less than one hour have little basis in reality. Data with smaller a time-step would be required for accurate shorter forecast horizons.

## 7 RESULTS

The testing set results, seen in Appendix A - Figure 8, show a full picture of balancing the tradeoff model complexity and forecast horizon length. A plot of each model's forecast prediction on one of the zones across all forecast horizons can also be seen in Appendix A - Figure 9 and 10.

### Analysis by Forecast Horizon:

- **6 Hours:** Statistical models Holt-Winters and MSTL show good performance on the shortest forecast horizon. Also, N-HITS performed well on all zones, performing the best out of all models on DOM and JC time series.
- **3 Day:** XGBoost performed the best in almost all zones except AE where the SVR received the best score. We also begin to see more models performing worse than the baseline models in a few zones.
- **1 Week:** XGBoost continues to evaluate the best with SVR close behind. However, now the statistical models are beginning to perform worse than the baseline models. This is to be expected for the Holt-Winters model as it only captures one seasonal period in a dataset that contains several, but was initialized for 1 week which demonstrates how effecting a seasonal naive forecast can be. Also, as expected N-HITS falls off from here on because of the constraints put on the model in this implementation described in the discussion session.
- **1 Month:** Models scored similarly to the 1 week forecasts, though with MSTL performing worse.
- **6 Month:** XGBoost continues to evaluate and to a lesser extend, SVR. Holt-Winters out-performed the baseline in a few zones but referencing the plots confirms this is due to random chance.
- **2 Year:** XGBoost again outperforms all other tested models. The SVR results fluctuate between zones.

Though these calculations would be performed at an enterprise level, compute times are very relevant to the thesis of this project. Please find the single data point on compute times on my local machine in Table 5.

**Table 5: Aggregated training and prediction times of each model on a Macbook Pro (2021) M1 - 16GB RAM.** <sup>1</sup> *MSTL only forecasted up to 1 month.* <sup>2</sup> *N-HITS only forecasted up to 1 month, but also only trained on 30 days of data. 570 minutes is an estimate by linear extrapolation.*

Model	Time
Seasonal Naive	Instant
Mean	Instant
Holt-Winters	9 Min
MSTL	95 Min <sup>1</sup>
SVR	59 Min
XGBoost	2 Min
N-HITS	9 Min → 570 Min <sup>2</sup>

## 8 CONCLUSION

By analyzing and modeling energy time series data we can efficiently and accurately forecast into the future different interval forecast horizons, which correspond to real-world decisions. Depending on the time-scale in which those decisions rely, analysts must appropriately identify what side of the accuracy and efficiency tradeoff is more important in their use case. Modern techniques in time series forecasting have reduced the errors in all forecast horizon intervals and continue to generalize those prediction abilities across many domains, but many of the most accurate models take a considerable amount of time or hardware to train. This type of research and technology is necessary for uninterrupted energy generation and distribution due to the ever-increasing demands and complexity of our energy grids.

### 8.1 Key Findings

XGBoost performs surprisingly well in this domain at most forecast horizons, greatly improving over the baseline methods. On top of its performance the computational demands are very low and training takes negligible time compared to other machine learning and neural network-based methods. Predictions are also performed very quickly once the tree has been fitted. However, it may be susceptible to data drift as new data is introduced, requiring it to be refitted.

Statistical methods such as MSTL and Holt-Winters models perform best at short horizon forecasts. Holt-Winters should only be used in forecasts horizon lengths in which there is only one predominant seasonality otherwise it needs to be augmented to handle multiples.

N-HITS even though was computationally the most expensive, performed well on shorter forecast horizons. As it was built for long-term horizons, no doubt it would also excel at them when fitted with the full training set.

### 8.2 Future Work

Training and testing should be rerun on more robust hardware so every model can be tested on each forecast horizon. Also, more attention should be given to computational efficiency and statistical measures of training and prediction times across each model and forecast horizon. This would help give a clearer picture of which model to choose in what situations.

## 9 APPENDICES

Appendix A is on the page after references.

## ACKNOWLEDGMENTS

Sqyd, for always brewing the morning coffee – I know it's 8 cups of water, now.

Luna, for breaking up the hyper-focused monotony with mind-clearing afternoon walks.

## REFERENCES

- [1] 2024. <https://huggingface.co/blog/autoformer>
- [2] 2024. <https://www.pjm.com/about-pjm/who-we-are/territory-served>
- [3] 2024. <https://scacis.rcc-acis.org/#>
- [4] 2024. [https://dataminer2.pjm.com/feed/hrl\\_load\\_metered/definition](https://dataminer2.pjm.com/feed/hrl_load_metered/definition)
- [5] 2024. <https://www.pjm.com/library/maps.aspx>

- [6] James Archsmith, Erich Muehlegger, and David S. Rapson. 2022. Future Paths of Electric Vehicle Adoption in the United States: Predictable Determinants, Obstacles, and Opportunities. *Environmental and Energy Policy and the Economy* 3 (2022), 71–110. <https://doi.org/10.1086/717219> arXiv:<https://doi.org/10.1086/717219>
- [7] Derek W. Bunn. 2004. *Modelling Prices in Competitive Electricity Markets*. Wiley.
- [8] Tao Hong and Shu Fan. 2016. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting* 32, 3 (2016), 914–938. <https://doi.org/10.1016/j.ijforecast.2015.11.011>
- [9] Konstantin Hopf, Hannah Hartstang, and Thorsten Staake. 2023. Meta-Regression Analysis of Errors in Short-Term Electricity Load Forecasting. In *Companion Proceedings of the 14th ACM International Conference on Future Energy Systems (Orlando, FL, USA) (e-Energy '23 Companion)*. Association for Computing Machinery, New York, NY, USA, 32–39. <https://doi.org/10.1145/3599733.3600248>
- [10] Heng Hua, Mingping Liu, Yuqin Li, Suhui Deng, and Qingnian Wang. 2023. An ensemble framework for short-term load forecasting based on parallel CNN and GRU with improved ResNet. *Electric Power Systems Research* 216 (2023), 109057. <https://doi.org/10.1016/j.epsr.2022.109057>
- [11] Paul Joskow. 2001. U.S. Energy Policy During the 1990s. *National Bureau of Economic Research, Inc, NBER Working Papers* (01 2001).
- [12] Chun Sing Lai, Giorgio Locatelli, Andrew Pimm, Xiaomei Wu, and Loi Lei Lai. 2021. A review on long-term electrical power system modeling with energy storage. *Journal of Cleaner Production* 280 (2021), 124298. <https://doi.org/10.1016/j.jclepro.2020.124298>
- [13] Eunjung Lee and Wonjong Rhee. 2021. Individualized Short-Term Electric Load Forecasting With Deep Neural Network Based Transfer Learning and Meta Learning. *IEEE Access PP* (01 2021), 1–1. <https://doi.org/10.1109/ACCESS.2021.3053317>
- [14] Shahzad Muzaffar and Afshin Afshari. 2019. Short-Term Load Forecasts Using LSTM Networks. *Energy Procedia* 158 (2019), 2922–2927. <https://doi.org/10.1016/j.egypro.2019.01.952> Innovative Solutions for Energy Transitions.
- [15] Lichao Sun, Hang Qin, Krzysztof Przystupa, Michał Majka, and Orest Kochan. 2022. Individualized Short-Term Electric Load Forecasting Using Data-Driven Meta-Heuristic Method Based on LSTM Network. *Sensors* 22 (10 2022), 7900. <https://doi.org/10.3390/s22207900>
- [16] Rafał Weron. 2014. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting* 30, 4 (2014), 1030–1081. <https://doi.org/10.1016/j.ijforecast.2014.08.008>
- [17] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yu Tong, and Bixiong Xu. 2021. TS2Vec: Towards Universal Representation of Time Series. In *AAAI Conference on Artificial Intelligence*. <https://api.semanticscholar.org/CorpusID:237497421>
- [18] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2022. Are Transformers Effective for Time Series Forecasting? arXiv:2205.13504 [cs.AI]
- [19] Vincent Zhihao Zheng and Lijun Sun. 2024. Multivariate Probabilistic Time Series Forecasting with Correlated Errors. arXiv:2402.01000 [stat.ML]



## Appendix A

NRMSE								sMAPE							
Metric	Model	Horizon	AE	CE	DOM	JC	PEP	Metric	Model	Horizon	AE	CE	DOM	JC	PEP
NormRMSE	Naive	6H	0.136277	0.087492	0.150645	0.184597	0.235583	sMAPE	Naive	6H	12.580831	8.368642	13.676927	16.850913	20.837337
	Mean	6H	0.340522	0.314349	0.186621	0.397084	0.488823		Mean	6H	28.959313	27.093643	16.941791	33.020723	39.234910
	HoltWinters	6H	0.014868	0.046767	0.011711	0.020384	0.013049		HoltWinters	6H	1.281159	4.399481	1.074516	1.976006	1.146197
	MSTL	6H	0.010456	0.042240	0.020137	0.026714	0.025732		MSTL	6H	0.924006	4.236191	1.540820	2.242525	1.879716
	SVR	6H	0.054108	0.104567	0.076867	0.085008	0.190672		SVR	6H	4.607263	9.269047	6.653134	7.561048	18.970828
	XGBoost	6H	0.045109	0.066064	0.041571	0.035517	0.028640		XGBoost	6H	3.375682	6.152227	4.115225	3.328029	2.278010
	NHITS	6H	0.028110	0.045254	0.011302	0.017790	0.016293		NHITS	6H	2.248364	4.536223	1.046940	1.470468	1.328477
NormRMSE	Naive	3D	0.119357	0.109039	0.171870	0.113405	0.200999	sMAPE	Naive	3D	10.564294	8.141624	14.682704	10.053777	17.745348
	Mean	3D	0.208615	0.141610	0.183489	0.219810	0.280960		Mean	3D	17.931932	11.818520	13.749959	18.790199	24.905756
	HoltWinters	3D	0.096301	0.104986	0.128598	0.084460	0.129481		HoltWinters	3D	6.795567	9.117568	7.209524	6.459062	7.638461
	MSTL	3D	0.132737	0.077898	0.111262	0.133788	0.131693		MSTL	3D	10.137742	6.672670	7.716595	11.919681	9.536786
	SVR	3D	0.088246	0.058785	0.094406	0.131786	0.118316		SVR	3D	6.649310	5.024464	7.196546	9.249314	11.662376
	XGBoost	3D	0.098207	0.032681	0.082099	0.064133	0.050121		XGBoost	3D	7.755542	2.698701	5.322590	4.371418	4.240230
	NHITS	3D	0.120104	0.076831	0.141640	0.110050	0.147622		NHITS	3D	9.395042	7.082582	10.380827	9.056158	12.270452
NormRMSE	Naive	1W	0.114612	0.117962	0.137086	0.092925	0.143650	sMAPE	Naive	1W	9.833909	10.046109	11.853713	7.963604	12.959489
	Mean	1W	0.152446	0.126032	0.221700	0.163112	0.190239		Mean	1W	13.027409	11.028017	19.665086	13.869585	16.964675
	HoltWinters	1W	0.129742	0.162546	0.208207	0.120589	0.212457		HoltWinters	1W	10.738640	15.046106	17.100893	10.238545	18.103634
	MSTL	1W	0.124733	0.131192	0.119044	0.126353	0.148563		MSTL	1W	10.071614	10.309518	9.181221	10.920468	12.810960
	SVR	1W	0.070804	0.055620	0.089589	0.095575	0.089091		SVR	1W	5.717646	4.692355	6.849986	7.107230	8.467719
	XGBoost	1W	0.072556	0.045279	0.069561	0.053372	0.052927		XGBoost	1W	5.374305	3.589565	5.300548	3.915818	4.300709
	NHITS	1W	0.114543	0.159284	0.171549	0.104196	0.172830		NHITS	1W	9.455587	15.526011	13.352938	9.039466	14.380739
NormRMSE	Naive	1M	0.121239	0.100637	0.161737	0.100371	0.148098	sMAPE	Naive	1M	10.163846	7.986211	13.466885	8.498115	12.421239
	Mean	1M	0.126805	0.101653	0.283969	0.133212	0.162745		Mean	1M	10.515516	8.513869	28.365910	10.961452	14.056335
	HoltWinters	1M	0.167221	0.167445	0.304187	0.176096	0.286739		HoltWinters	1M	15.198061	16.639998	30.038587	16.547223	28.951588
	MSTL	1M	0.149286	0.101509	0.187906	0.226123	0.189383		MSTL	1M	13.202894	8.138927	16.189155	22.238330	17.300937
	SVR	1M	0.084019	0.057702	0.087019	0.084813	0.090183		SVR	1M	6.603718	4.732827	7.115081	6.492116	7.793409
	XGBoost	1M	0.067560	0.040742	0.089747	0.068048	0.066061		XGBoost	1M	5.180421	3.061406	7.495047	5.063363	5.188263
	NHITS	1M	0.159065	0.212156	0.331945	0.136681	0.299784		NHITS	1M	13.960246	21.936399	31.881035	12.069495	29.509419
NormRMSE	Naive	6M	0.195920	0.194636	0.173055	0.207272	0.188107	sMAPE	Naive	6M	13.572251	12.176701	13.512928	12.933855	14.080367
	Mean	6M	0.240628	0.199582	0.218384	0.231947	0.218194		Mean	6M	18.941483	15.168616	16.169793	18.350397	18.375323
	HoltWinters	6M	0.205409	0.176356	0.221137	0.195561	0.216097		HoltWinters	6M	15.425375	12.174435	15.976172	14.355150	15.959039
	MSTL	6M	nan	nan	nan	nan	nan		MSTL	6M	nan	nan	nan	nan	nan
	SVR	6M	0.130074	0.093505	0.094867	0.137879	0.129636		SVR	6M	9.653313	6.016505	7.315464	9.054637	11.440345
	XGBoost	6M	0.081284	0.054166	0.071832	0.114838	0.060866		XGBoost	6M	6.139266	3.743250	5.560432	6.804394	4.516422
	NHITS	6M	nan	nan	nan	nan	nan		NHITS	6M	nan	nan	nan	nan	nan
NormRMSE	Naive	2Y	0.193376	0.183827	0.169782	0.209542	0.174865	sMAPE	Naive	2Y	13.257603	12.475873	14.042065	13.456625	12.813002
	Mean	2Y	0.296964	0.207625	0.233131	0.282705	0.225961		Mean	2Y	21.589680	16.402006	17.586856	20.421968	18.642190
	HoltWinters	2Y	0.283207	0.171086	0.231524	0.254468	0.214478		HoltWinters	2Y	18.524866	11.674018	18.222272	16.527110	15.562225
	MSTL	2Y	nan	nan	nan	nan	nan		MSTL	2Y	nan	nan	nan	nan	nan
	SVR	2Y	0.167635	0.097154	0.102399	0.157525	0.278998		SVR	2Y	12.905069	6.720007	8.251229	10.808048	28.784374
	XGBoost	2Y	0.072943	0.058035	0.078654	0.105739	0.054516		XGBoost	2Y	5.588003	4.122110	6.700303	6.550198	4.082743
	NHITS	2Y	nan	nan	nan	nan	nan		NHITS	2Y	nan	nan	nan	nan	nan

Figure 8: NRMSE and sMAPE results of every model at each forecast horizon and zone. Scores highlighted in green indicate the best score in that forecast horizon and zone. Scores highlighted in red indicate the model scored worse than either of the baseline models. Values of nan indicate models that were not fit to that forecast horizon due to hardware limitations.

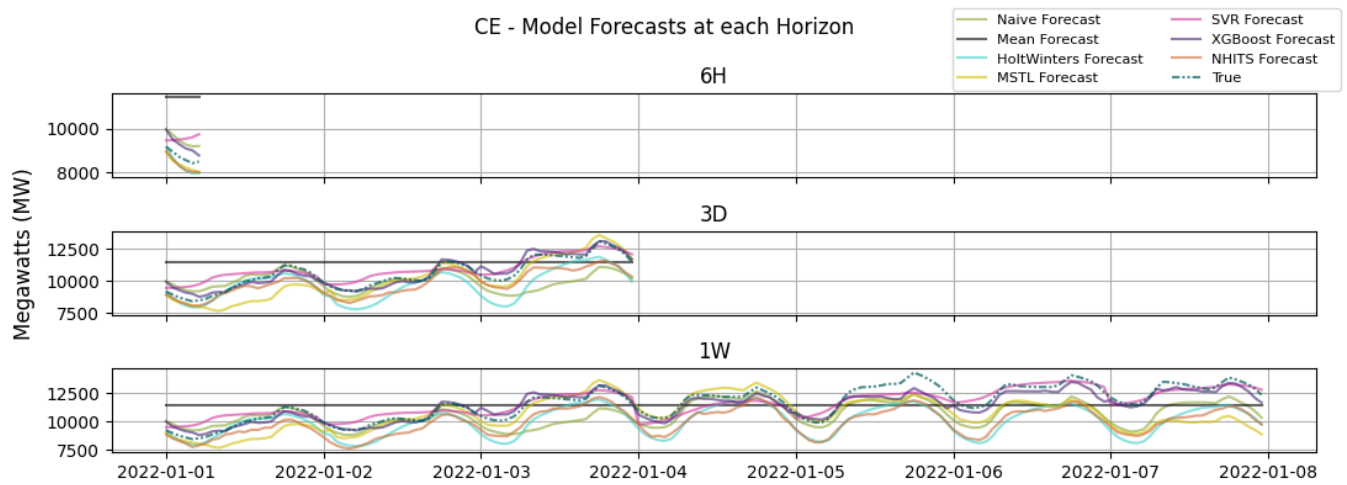


Figure 9: Plot of each model's forecast prediction on the CE zone, across 6 hours, 3 days, and 1 week forecast horizons.

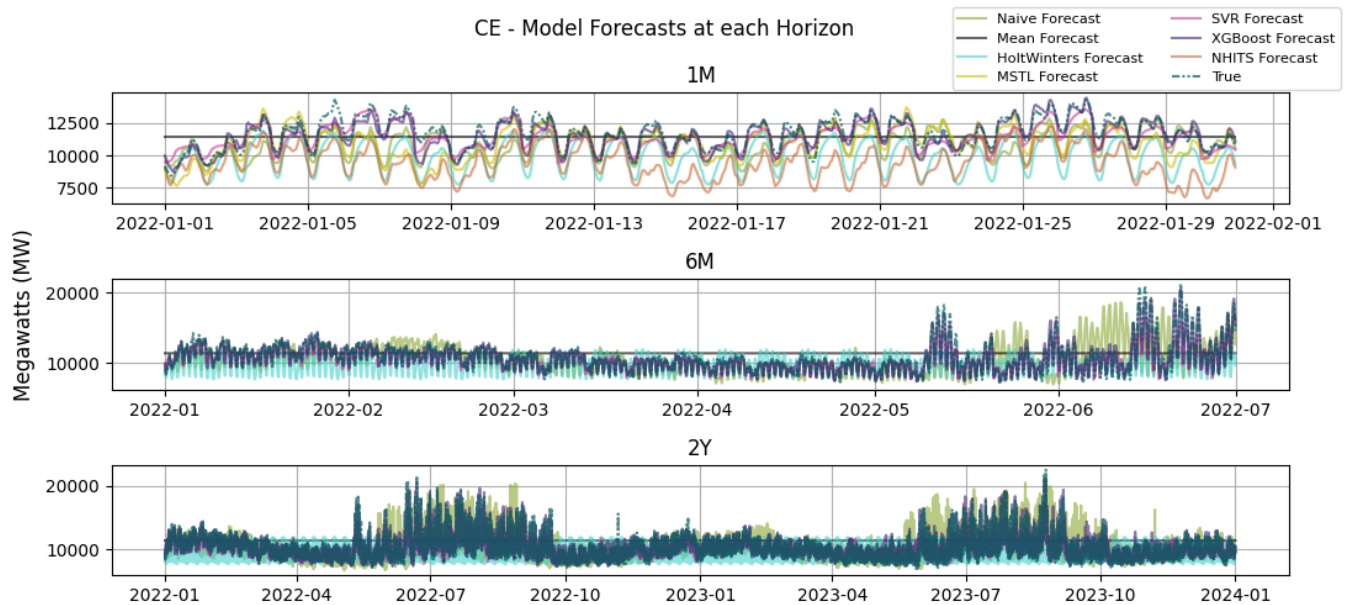


Figure 10: Plot of each model's forecast prediction on the CE zone, across 1 month, 6 months, and 2 years forecast horizons.

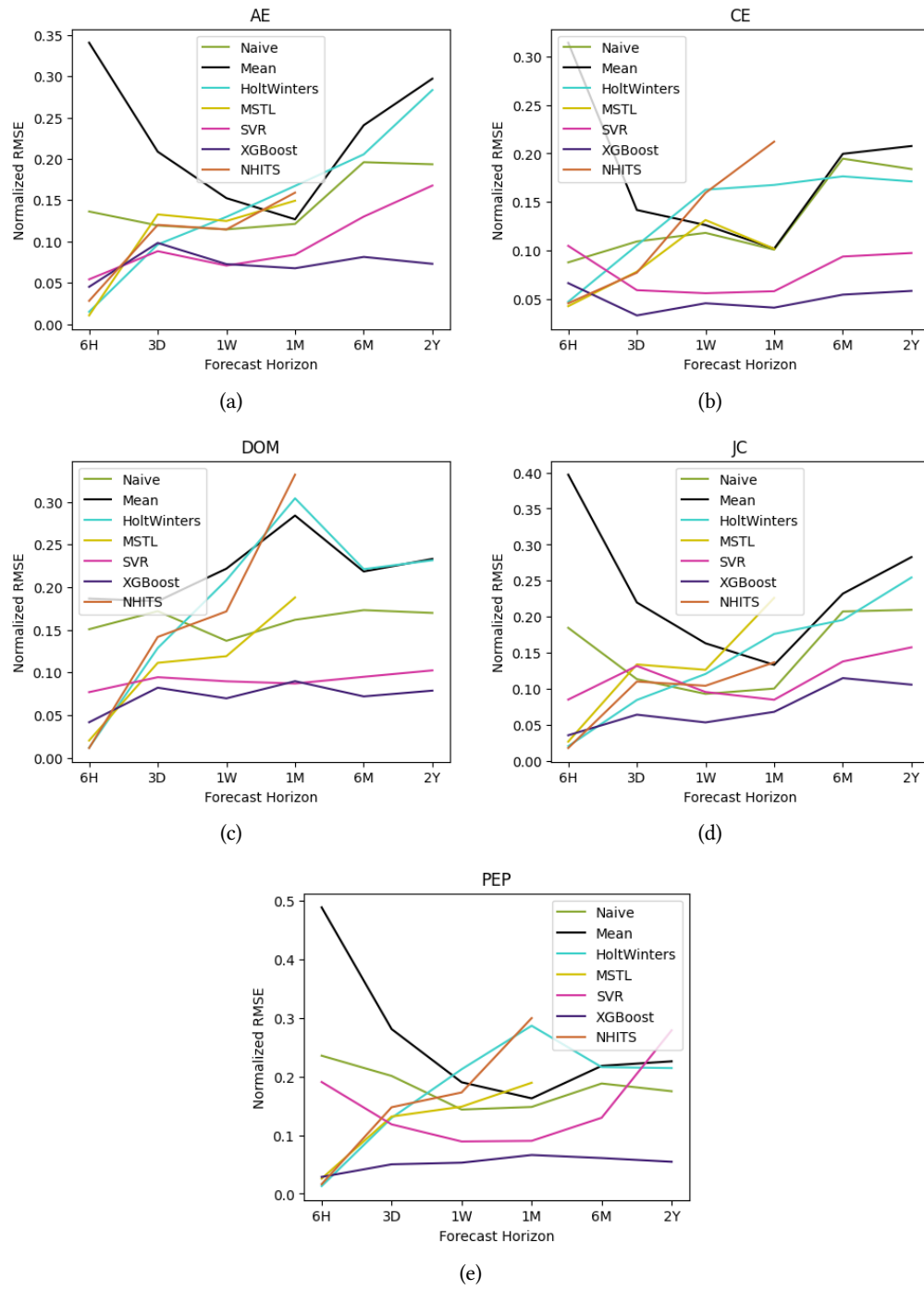


Figure 11: NRMSE of each model and forecast horizon organized by zone. (a) Atlantic City Electric Co. (b) ComEd. (c) Dominion. (d) Jersey Central Power & Light (e) Potomac Electric Power Co.

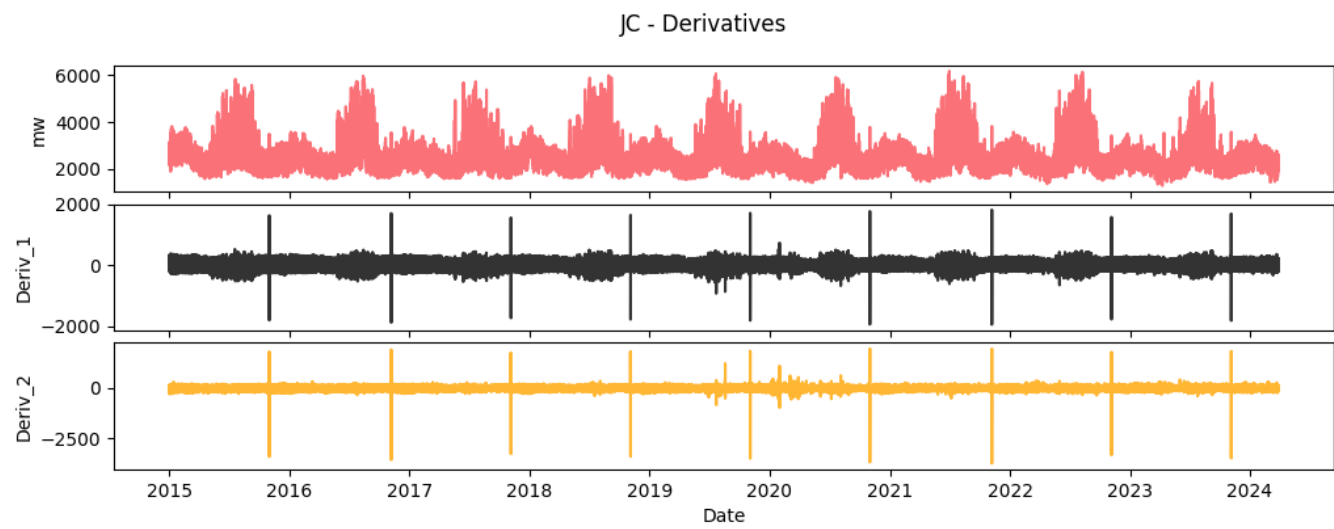


Figure 12: First and second-order derivatives/differencing of Megawatts in black and yellow respectively.