OpenEEmeter – Hourly Model

Lead Contributors

- Travis Sikes
- Armin Aligholian
- Jason Chulock
- Adam Scheer
- Brian Gerke

1 Introduction

1.1 Motivation

The OpenEEmeter 4.1 hourly model, implemented from the CalTRACK 2.0 methodology and referred to herein as the “legacy” hourly model, has led the vanguard of open-source models suitable for measuring the impacts of demand-side programs (e.g. energy efficiency, load shifting, and demand-response programs). OpenEEmeter 4.1 has gained industry-wide acceptance and is being used to measure many demand-side portfolios. Despite its success, certain limitations have emerged that have encouraged stakeholders to further advance the model. For instance:

- Solar PV Challenges: The legacy hourly model does not incorporate solar irradiance data, which leads to inaccuracies in prediction for customers with solar PV systems, particularly under varying cloud conditions.
- Flexibility limitations: For meters without solar PV, the model’s performance is sufficient, but the inability to leverage supplemental data limits its adaptability and the development of potential improvements.
- Overfit model: The legacy hourly model performs considerably better in the baseline data (training data) than in reporting period data (test data) indicating an overfitting issue.
- Efficiency: Given the improvements that were able to be made to the daily model in OpenEEmeter 4.1, it was suspected that significant efficiency gains could be achieved in the hourly model to make it faster and less costly to run.
- Ease of use: The legacy hourly model was not developed with a cohesive API in mind, resulting in an unintuitive architecture that is difficult for users and developers alike.

Addressing these shortcomings formed the rationale for developing the new hourly model.

1.2 New Model Goals

The new model was developed to address the limitations of the legacy system while maintaining or improving its strengths. The primary objectives of the new model include:

1. Improved Solar PV Prediction: Better handling of meters with solar PV by integrating solar irradiance data to capture variability caused by insolation and cloud coverage.
2. Performance Retention and Enhancement: Maintain or exceed the legacy model's accuracy for non-solar PV meters.
3. Well-fit Model: Developed model is neither underfit nor overfit, but is well fit such that baseline error metrics are reasonably predictive of reporting year error metrics.
4. Flexibility: Enable the model framework to incorporate supplemental data, such as additional time series or categorical variables, when available, to enhance predictions.
5. Faster Computation: Ensure that the model performs computations more efficiently.
6. Enhanced Usability: Simplify the API for easier and more seamless integration into workflows.

By achieving these goals, the new model is positioned as a robust, adaptable, and efficient replacement for the legacy hourly model.

1.3 Brief Model Overview

The new hourly model is a performant, flexible, data-driven framework designed to predict consumption for individual meters. Key features of the model include:

Required Inputs:

1. Temperature time series (primary input variable)
2. Energy usage time series (target variable)

Optional Inputs:

1. Solar irradiance time series (solar PV system input variable)
2. Supplemental data: The model can utilize supplemental time series or categorical variables that the user expects to be predictive of customer energy consumption.

Framework:

The model is built using the Elastic Net framework, a linear regression model with regularization. Regularization enhances predictive power by performing feature selection to limit unnecessary model complexity while maximizing model accuracy.

Prediction Mechanism:

The model operates on a 24-hour prediction framework, meaning it ingests 24-hour input data (temperature, optional solar irradiance, temporal and categorical variables) to predict consumption for a 24-hour day from the same temporal cluster (see Section 3.1.3). This allows the model to innately account for phenomena such as thermal lag and precooling if they are regularly part of a building's performance characteristics.

This architecture enables the new hourly model to outperform the legacy model in critical areas and provides an extensible solution for future energy measurement scenarios.

2 Data Input Overview

The model requires two key datasets: an hourly temperature time series and an hourly energy usage time series (electric or gas), the latter serving as the target variable. These inputs must meet specific granularity, coverage, and quality criteria to ensure robust predictions.

2.1 Data Granularity and Coverage

The model operates at an hourly resolution, meaning all input data must be recorded hourly. Data coverage is equally important, with the following criteria for both weather (temperature and solar irradiance data) and solar irradiance data:

- The model's framework requires a baseline dataset of 1 year (365 days) to capture long-term patterns such as seasonal trends.
- The baseline data must have at least 90% coverage in each month.
- Each day can have up to 50% missing data, but no more than 6 consecutive missing hours. For each meter, days that exceed these thresholds will be flagged and excluded from the model training. The rest of the meter data will be used to fit or predict with the model, provided the sufficiency criteria above are still met.
- If any data are missing for a given time stamp, all input variables are considered missing for that time stamp. This is a conservative view, but maintains the relationship between these quantities and the target variable and ensures the fractional data requirements do not drift from the prescribed quantities.

Consumption data has these additional restrictions:

- Gas consumption data must never be negative. Negative values will be viewed as erroneous and treated as missing data, including for the requirements listed previously.
- Any zeros in electricity consumption data are viewed as errors and treated as missing data.

Data Preparation and Cleaning

To manage these requirements, an **hourly data class** has been developed to validate and process the input data. Error messages are generated if any failures occur. This class performs several critical tasks, including:

1. Validation of Inputs: Ensures that hourly temperature, solar irradiance, and energy consumption time series are present and meet all sufficiency requirements.

2. Timestamp Continuity: Reindexes the data with a complete datetime series to ensure a continuous time series.
3. Handling Missing Data: Uses a correlation-based imputation method to estimate missing values, provided the missing data falls within acceptable thresholds (e.g., fewer than 6 consecutive hours missing per day). Imputed values are flagged to ensure users can identify such data points. Additional information can be found below.
4. Solar Irradiance Data Integration: When available, hourly solar irradiance data is incorporated into the feature space as an additional input. For meters with solar PV systems this enhances the model's ability to capture variability in solar energy production.

It should be noted that the hourly data class requires 100% complete supplemental data. This requirement is built on the assumption that the supplemental data could be anything. It would not make sense to try to correct data that could have unknown restrictions.

The data class also performs some data quality checks. These checks do not disqualify a meter if they are triggered but instead are intended only to inform the user of a potential issue. These checks are largely based upon the CalTRACK methodology. Additional information can be found in 8.1: Guidelines for Data Quality Issues.

2.2 Correlation-based Imputation

For each input time series feature, i.e. temperature, solar irradiance, and energy consumption, we impute all missing values. Imputation is not performed on supplemental data as it is impossible to know if imputing the quantity even makes sense. The methodology is based on autocorrelation of each feature/target. We use autocorrelation over the time series to find the *N*[1] largest peaks to get the lag/lead index for each time series (i.e., the lookahead/lookback period on which the data is the most self-similar). The lag/lead values at each missing time step for each feature are averaged to replace the missing value. This is done iteratively. On the first iteration, if there are any missing values in the lag/lead values then the imputation is skipped. Each subsequent iteration becomes more permissive allowing more missing values from the lag/lead values until only a single value is required. If there are any missing values at this point then they would be linearly interpolated, followed by forward fill and a backwards fill.

This algorithm is predicated on the premise that building behavior tends to be cyclical. Imagine a building with a consumption pattern that repeats every 24 hours. In this example, the most autocorrelated lag/lead value will be 24, which is to say Tuesday and Thursday's usage at 2 PM will be highly predictive of Wednesday's usage at 2 PM. If we were only including 2 lag/lead values, then we would average Tuesday and Thursday's usages at 2 PM to impute Wednesday's 2 PM usage. If Thursday were also missing then

[1] *N* is a function developed in a side study. For temperature and solar irradiance, it is a constant value of 6. For observed data it is $4.012 \cdot \ln\left(pct_{missing}\right) + 24.38$ where $pct_{missing}$ is the percentage of missing data. Additional information can be found in 8.2: Imputation.

we would just use Tuesday's consumption. Note that this is a simplified example. In practice, the number of lag/lead values are generally between 6 and 24.

With this methodology, the pattern of the time series is used to replace the missing value and all missing values are populated. Figure 2.1 illustrates the performance of the imputing method for 10% (upper panel) and 50% (lower panel) missing data for a single meter. This methodology has been tested up to 50% missing data to ensure that any future data sufficiency changes will not be limited by the imputation methodology. We would emphasize, however, that we will never impute such a large fraction of missing data, but instead are ensuring the robustness of the methodology by pushing this algorithm to its extreme.

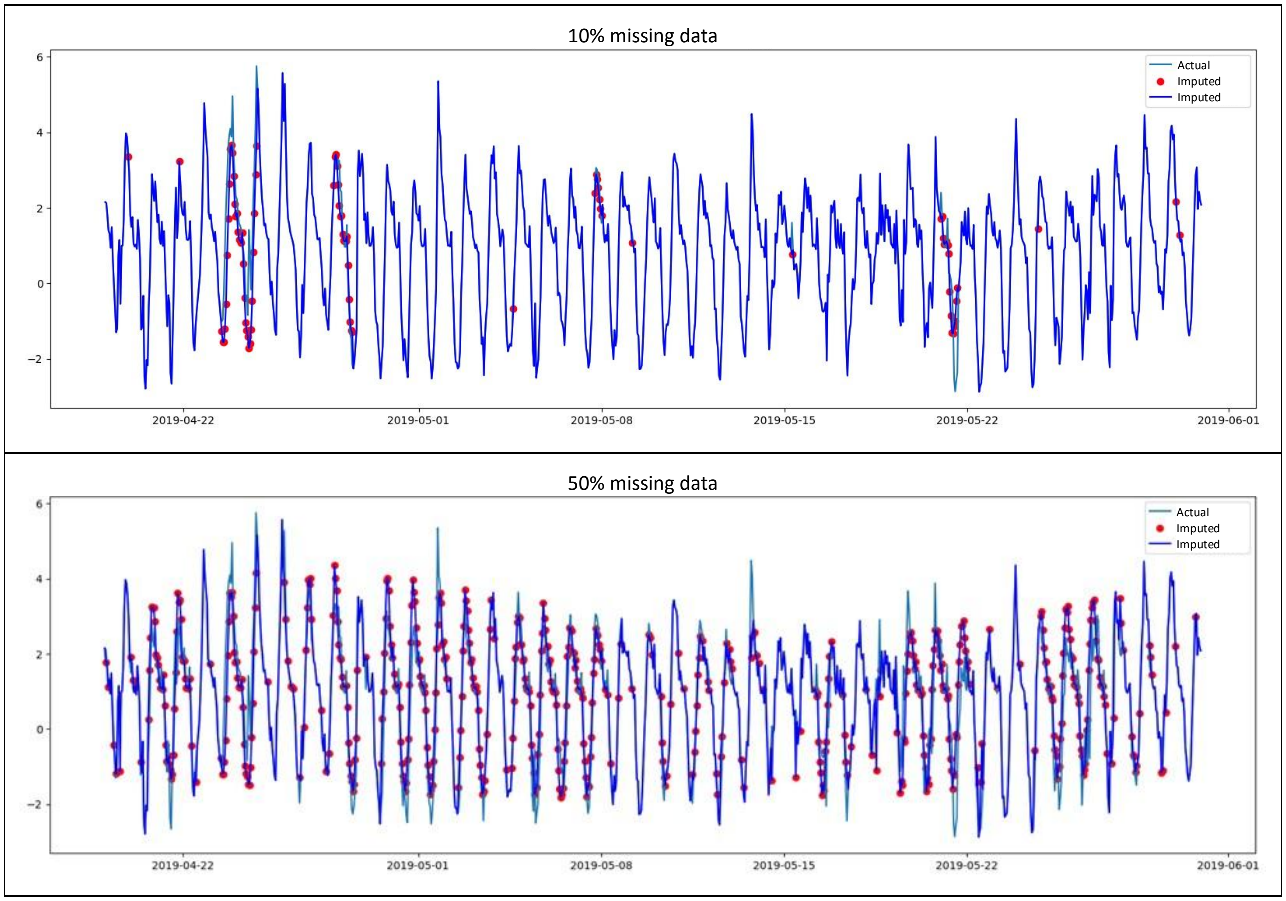

Figure 2.1. Example plots of energy usage data with 10% (upper panel) and 50% (lower panel) of data removed and used as ground truth for the correlation-based imputation methodology. The light blue lines are the actual, ground truth data and the red circles and dark blue lines show the imputed values.

By enforcing these constraints and systematically addressing data quality issues, the model can be sure to have a clean and reliable dataset. This robust preparation process enhances predictive performance and ensures flexibility for incorporating additional data in the future.

3 Model Overview

3.1 Feature Generation

Feature generation is a crucial aspect of the new hourly model and is designed to ensure that the input data is transformed into meaningful features that capture both linear and non-linear relationships among the input variables and energy consumption. The model incorporates various types of features, including time series, categorical variables, and their combinations to predict hourly consumption with high accuracy. In OpenDSM it is expected that the input dataframe has a column named “observed” to label the consumption data.

3.1.1 Temperature

Building energy consumption generally has a strong response to temperature, which varies considerably in different geographic regimes with different heating and/or cooling needs. For this reason, hourly outdoor air temperature is a required input for each day. Generally, values are taken as mean hourly outdoor dry-bulb temperature measured at the nearest weather station within the same climate zone as handled in the EEweather module of OpenDSM. In OpenDSM it is expected that the input dataframe has a column named “temperature” for this feature. To better capture the variation in building energy consumption in different temperature ranges, especially at temperature extremes, input temperature values are divided into bins, each of which is assigned an independent slope and intercept in the model. There are two categories of temperature bins in the model:

- Fixed Temperature Bins: Bins with edges of 10, 30, 50, 65, 75, 90 and 105°F are used[2]. These bin edges are fixed and each bin is assigned a unique slope and intercept representing the response of energy consumption to temperature within a given bin. Bins are required to have a minimum of 20 data points. If fewer points are present in a given bin, that bin is absorbed into the neighboring bin with the bin range expanded. For example, if there are only 10 data points of temperatures below 10°F, then instead of having a bin from $-\infty$ to 10°F, it would instead be from $-\infty$ to 30°F. The same applies from the high temperature side except in the opposite direction.
- Low and High Temperature Bins: Energy usage behavior can change dramatically in these areas. For example, an air conditioner could be turned off, continue operating as normal, struggle to keep its setpoint, or hit maximum capacity, each circumstance impacting the building’s response to temperature changes. To account for this, the lowest and highest hourly temperature bins are modeled with a non-linear term in addition to the linear slope and intercept. For these bins, exponential growth and decay rates are included to capture deviation from linear behavior, where supported by the data.

[2] An optimized choice based upon population-level optimization as discussed in Section 5.

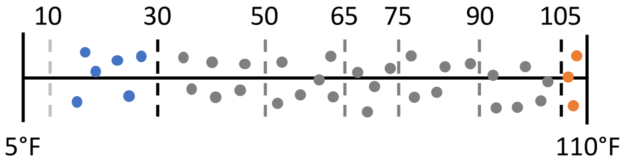

Figure 3.1. Temperature binning schematic showing how temperatures are binned where all potential bins are labeled at the top, data points are shown as circles, and dashed lines show bin edges. Circles in the extreme low temperature bin are blue, moderate bins are gray, and extreme high temperature in orange.

The Figure 3.1 schematic shows how the binning algorithm works. In this example, temperature values exist between 10°F and 110°F. All of the temperature bins have a linear impact component of the model (slope and intercept). Because there are no values less than 10°F, the bin that would typically be between $-\infty$ and 10°F is removed and instead the bin is set to be from $-\infty$ to 30°F as described previously. This is depicted in the figure as a faded out dashed line for the 10°F bin edge. The points within this lowest bin are colored blue to represent that they are part of the low temperature bin and have additional non-linear terms. Likewise, values above 105°F are shown as orange to represent the high temperature bin which also has non-linear terms.

3.1.2 Non-Linear Temperature Features

The lowest and highest temperature bins have extra components in the feature space to account for nonlinearities. To model the nonlinear behavior, Eqn. 3.1 was utilized to model both a positive and negative exponential response.

$$f(T) = \frac{1}{\exp\left(\frac{1}{k}\right) - 1}\left(\exp\left(\frac{s}{k}T\right) - 1\right) \tag{3.1}$$

where s is 1 (positive exponential feature) and -1 (negative exponential feature), T is the standardized temperature time series, and k is the rate of growth/decay in response to temperature determined heuristically. The units of this equation are dimensionless, as it is a standardized quantity. This function is defined such that the maximum value is 1 in order to ensure that it is of the correct scale for the elastic net. The variable k is calculated by fitting an exponential growth/decay equation using the observed and temperature data over the entire year for each hour of the day and then using the minimum of the 24 values. An example of the non-linear features is shown below in Figure 3.2.

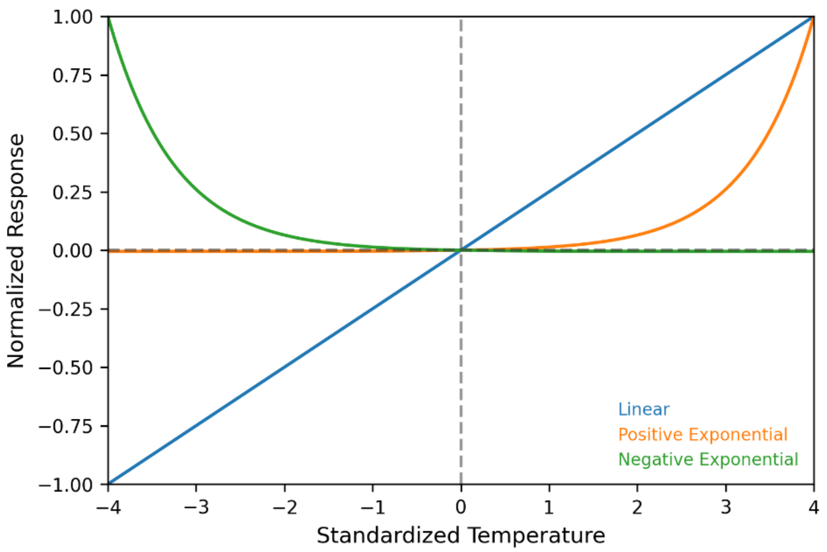

Figure 3.2. Example of response of positive exponential (green curve) and negative exponential (orange) compared to a linear response (blue).

In Figure 3.2, a linear response (blue) is shown vs standardized temperature for comparison. The two additional non-linear features are shown (orange and green). In the extreme temperature bins, all of these components are used to estimate the energy usage with respect to temperature. To do this each of the components has a coefficient in the elastic net regression. The size and sign of each coefficient controls the extent and direction of the modeled linear and non-linear response to temperature. This flexible approach allows both near-linear and more strongly non-linear behavior to be captured at extreme temperatures if the data support it.

3.1.3 Temporal Clusters

The energy consumption of an individual building depends on its occupancy and utilization, which tend to vary on a regular schedule. For instance, offices may be occupied during the week and unoccupied during weekends, while residences may have an opposite pattern. The legacy hourly model makes a rough estimate of occupancy as a pre-fitting step and includes hour-of-week as a variable to address this. However, this approach tends to be coarser than is desirable to capture the nuances of building utilization throughout the year. Another approach involves allowing different models on weekdays versus weekends, or by season, as in the OpenEEmeter 4.1 daily model. However, many buildings (e.g., theaters and restaurants) have usage variation that does not follow a strict weekday/weekend pattern. The detailed information available in hourly data allows us to discover the consumption patterns of an individual building. We can decipher these patterns and account for them by identifying clusters of similar daily load shapes throughout the year. In this way, if a particular building has distinct operational patterns on, e.g., Mondays and Tuesdays in winter compared to the rest of the year, the model can capture these differences.

To do this, we use day of the week and month groupings as the initial guess for usage pattern segmentation. For each of the 84 combinations of these patterns, (Monday, January), (Tuesday, January),

…, (Sunday, December), we take the median usage of each combination to create a 24-hour representative loadshape. Clustering is then performed on these loadshapes as described below:

- A discrete wavelet transform with 4 levels using the Haar mother wavelet (otherwise known as the Daubechies db1 wavelet) using a constant derivative to extend the boundaries decomposes the loadshape time series data into a series of wavelet coefficients and thereby out of the temporal domain.
- The coefficients are then input into a principal component analysis (PCA) to reduce the dimension of the wavelet coefficient to capture the most important parameter coefficients. The number of coefficients are decided by Minka's maximum likelihood estimation (MLE)[3].
- A spectral clustering algorithm using radial basis functions to construct the affinity matrix with a gamma of 1.05[1] is used to cluster the PCA coefficients to between 2 and 24 clusters.
- The optimal number of clusters is selected using the Variance Ratio Criterion.

Results show that the number of clusters per meter follows an exponential distribution where the majority of meters yield two or three clusters, significantly reducing complexity. Examples below show the effectiveness of the clustering algorithm. For instance, in Figure 3.3, the initial guess of 84 distinct categories is shrunk to two usage patterns, weekdays and weekends. This algorithm can also find less standard active/inactive days in a commercial building as shown in Figure 3.4. A more complicated clustering result is shown in Figure 3.5, which illustrates 3 clusters exhibiting a unique combination of seasonality and Sunday usage found specifically for this meter.

The temperature dependence of load within each temporal cluster is modeled using an interaction term between the temporal clusters, temperature bins, and temperature time series such that each cluster has its own independent linear response to temperature within each temperature bin. This is in addition to the global linear response to temperature modeled within each temperature bin irrespective of the temporal clustering. Because these time series are highly correlated, we have pushed the model towards favoring the global temperature bins by down-weighting the temporal cluster/temperature bin/temperature time series interactions by a factor of 0.524. Lastly, temporal clusters are given unique intercepts distinct from any temperature effects.

[3] Automatic choice of dimensionality for PCA, Thomas P. Minka, M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 514

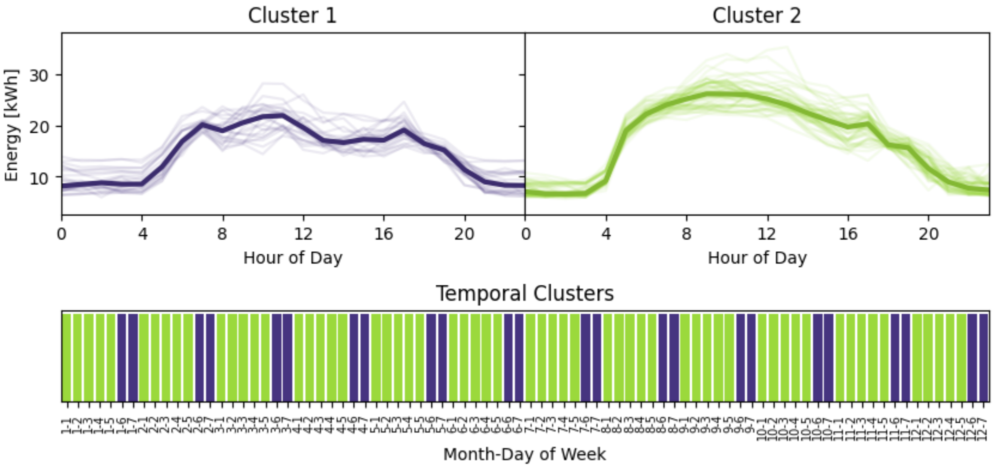

Figure 3.3. An example of temporal clustering where cluster 1 (purple) represents the weekends and cluster 2 (green) the weekdays.

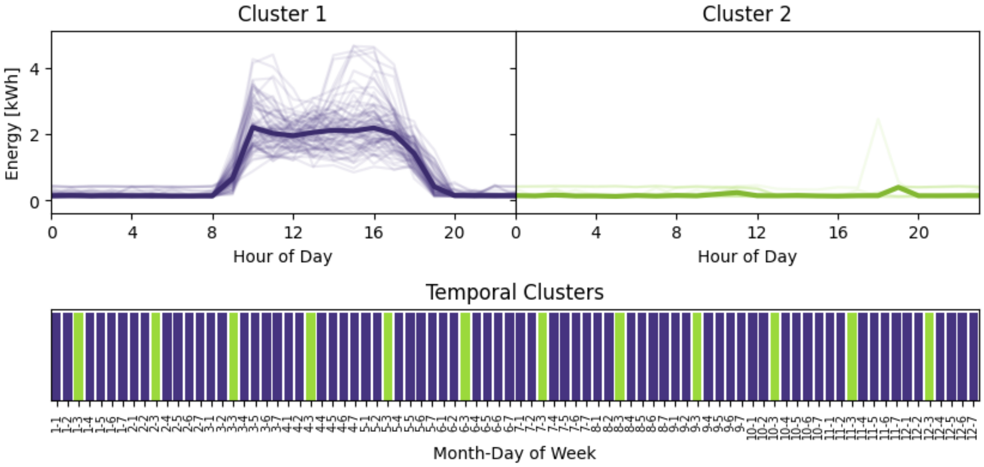

Figure 3.4. An example of temporal clustering where cluster 1 (purple) represents high usage days and cluster 2 (green) shows that this business takes off Wednesdays.

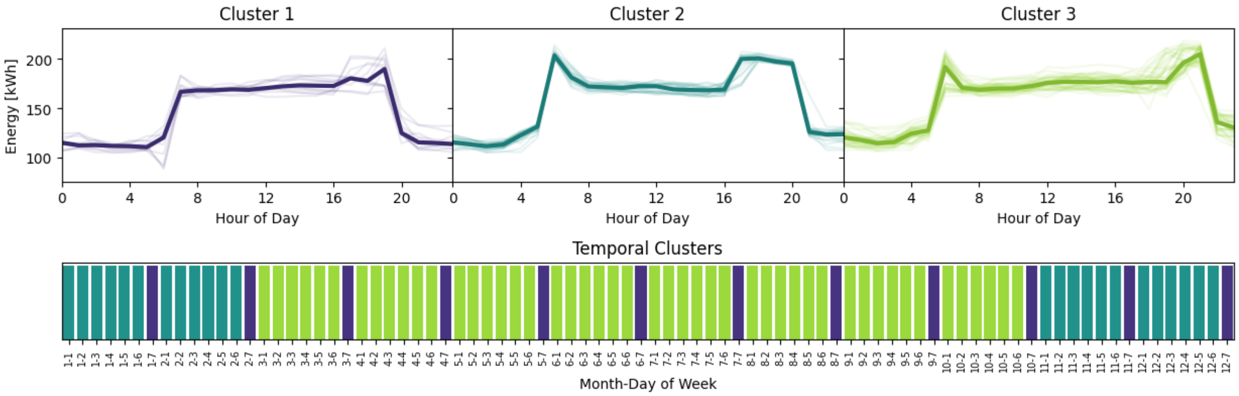

Figure 3.5. An example of temporal clustering where cluster 1 (purple) shows Sunday usage, cluster 2 (teal) shows winter seasonal usage, and cluster 3 (green) shows spring/summer/fall usage.

3.1.4 Solar Irradiance

Global Horizontal Irradiance (GHI) is used as a time series input when available to inform the new model. GHI provides a direct proxy for solar generation, being approximately linearly proportional to PV production. In OpenDSM it is expected that the input dataframe has a column named “ghi” for this feature. It is currently up to the user to obtain their own GHI data. This can be done with free, but delayed, sources such as NREL or through a commercial service. The irradiance data should be location-specific either by weather station or directly over the meter location. The model assumes a linear response to GHI that is the same across all hours of the year.

3.1.5 Supplemental Data

Optionally, supplemental datasets that may be predictive of building energy consumption, such as pump schedules or EV charging schedules, can be provided by the user. These features are treated as linear inputs when available. These datasets help refine predictions by incorporating operational patterns into the feature set. Supplemental data can be either time series or categorical variables. In either case, the data must have no missing values as stated in the sufficiency criteria above. The flexible nature of supplemental data means that it can be anything but this also means it is not safe to impute it. Therefore, it is left up to the user to correct missing data.

As an example, the hourly energy consumption of a meter during the baseline period and its corresponding model fit without supplemental data are shown in Figure 3.6, upper panel and lower panel respectively. A change point in consumption can be observed around November 11, 2019, attributed to the installation of solar PV.

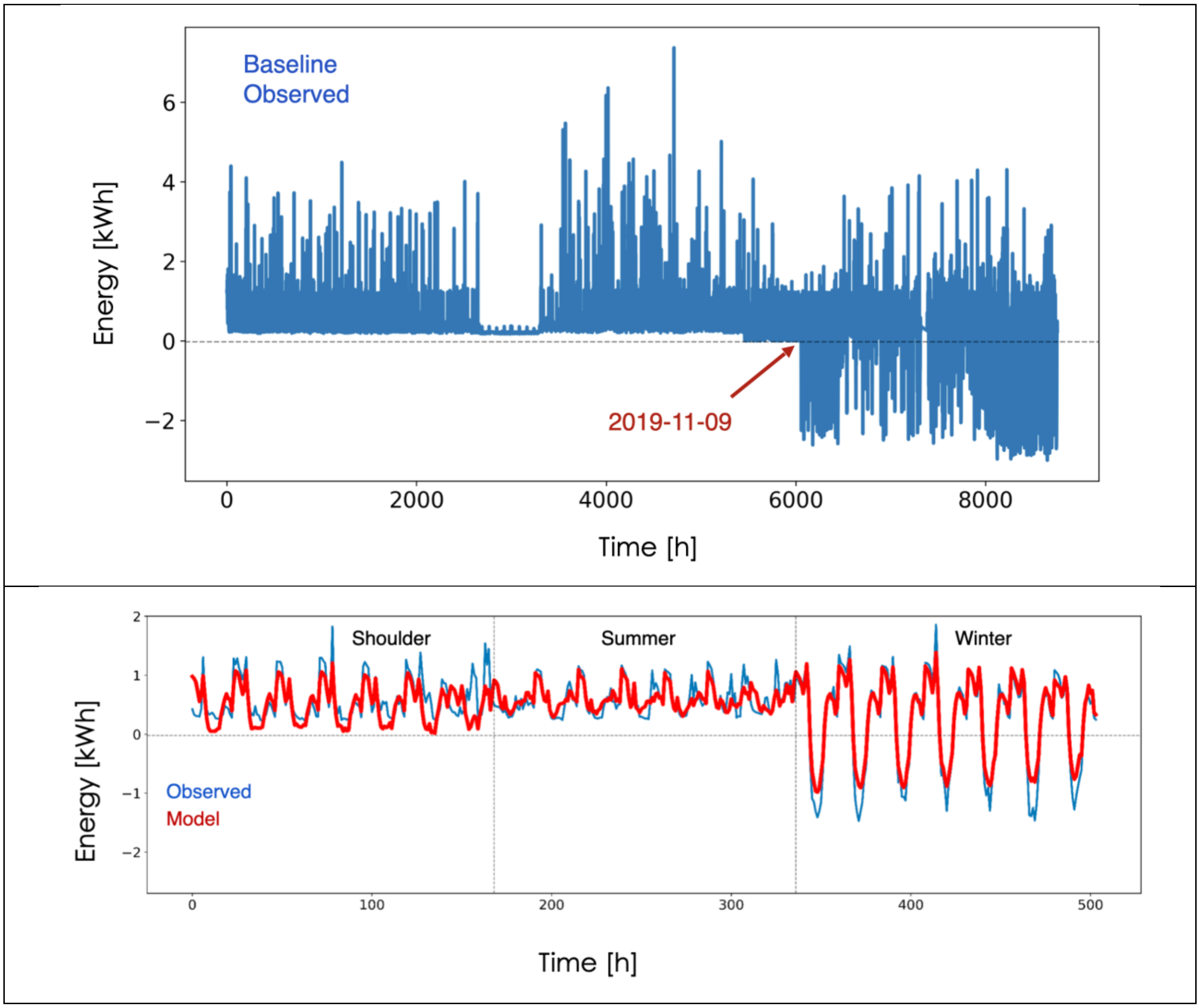

Figure 3.6. An energy consumption plot of a building likely undergoing a solar PV installation in the baseline period. The upper panel depicts the observed time series and the lower panel shows the corresponding seasonal hour-of-week loadshapes for the model fit and the observed data. Blue lines show the baseline data and the red line shows the model fit without supplemental data.

In the upper panel of Figure 3.6, we see a time series of the observed consumption. It appears that at hour ~5500, the solar PV system becomes operational but is not exporting, hence the minimum consumption drops to zero. At hour ~6000, 11-09-2019, the solar PV system begins to export, which appears as significant negative consumption. In the blue trace of the lower panel, this observed trace is aggregated to a seasonal hour-of-week loadshape. Because the solar PV system only begins to export in November, the loadshape has a completely different structure in the shoulder and summer seasons as compared to the winter. Without information on solar installation date the model would expect a similar loadshape will be seen in the reporting year; it only knows that there is a significant change in behavior between the seasons and tries to best match this behavior.

Indeed, this is what we see in the upper panel of Figure 3.7; the shoulder and summer months are treated as if there is no solar PV because that is what the model learned from the baseline year during those time periods. However, we can improve predictive performance by using the solar PV installation date as a categorical variable when training the model on the baseline data. The categorical value is assigned as 0 for days before the installation date and 1 for days after it. The resulting predictive improvement is dramatic. Using the solar PV installation date, as shown in the lower panel of Figure 3.7, yields a 30% improvement in error. Because the model could only learn from winter months, where solar production is at a minimum, the negative consumption throughout the year is somewhat underpredicted. Nevertheless, it is clear that supplementing the model with an indicator of the solar installation date has allowed the model to better utilize the available data and drastically improved its predictive power.

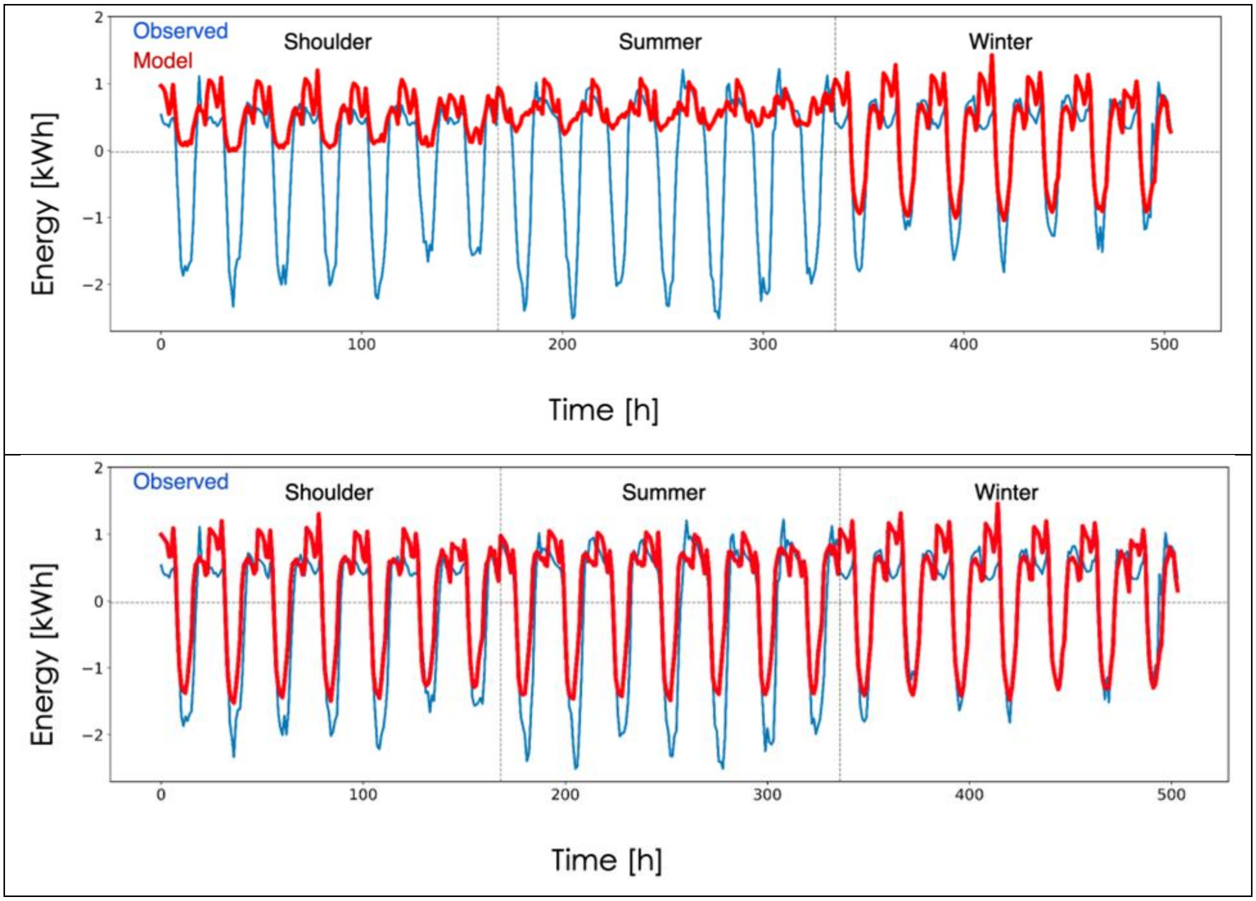

Figure 3.7. An example of how model predictive performance can be significantly enhanced using supplemental data with seasonal hour-of-week loadshapes. The upper panel shows the reporting year prediction from a model trained without the solar PV installation date, and the lower panel shows the result of a model trained with the solar PV installation date. The blue lines represent the observed data in the reporting period and the red lines show the model prediction during the same time period.

Use of supplemental data is an experimental feature of OpenDSM. It cannot be used in OpenDSM-compliant measurements because it's impossible to know what a potential user might use as an input. If certain supplemental time series/categorical features are negotiated and agreed upon by all parties to perform a measurement, then it is reasonable to be included, but such measurements should not be referred to as OpenDSM compliant.

3.1.6 Total Number of Features

To aid in understanding how the model ingests data, Eqn. 3.2 reflects the number of features considered for each hour for a given fit using the energy efficiency model:

$$N_{features} = 2\, N_{T_{bins}} \left(N_{t_{clusters}} + 1\right) + N_{t_{clusters}} + 4 + GHI + N_{supp} \tag{3.2}$$

where $N_{T_{bins}}$ is the number of temperature bins as defined in Section 3.1.1, $N_{t_{clusters}}$ is the number of temporal clusters as defined in Section 3.1.3, 4 is the number of nonlinear temperature features for the extreme temperature bins also defined in Section 3.1.1, GHI is either 0 or 1 if GHI is included or not, and N_{supp} is the number of supplemental columns provided. The number of features in the model varies depending on the specific characteristics of each meter. Since temperature binning and clustering are optimized individually for each location, the number of temperature bins and clusters are unique to the meter being analyzed. Furthermore, if solar irradiance (GHI) or other supplemental data is available, the feature count changes accordingly. This adaptive approach ensures that the model remains tailored to the specific data and characteristics of each meter.

To demonstrate how many features are considered, assume that we are using the solar model on a meter which has 6 populated temperature bins and 7 temporal clusters. Due to the multiple interactions This results in a staggering 108 features to be considered. Even if the number of temporal clusters is reduced to 2, this would still be 43 features. Clearly, we are going to need feature selection to not create an overfit model.

3.2 Model Framework

3.2.1 Building the 24-Hour Input-Output Framework

To enhance the model's ability to capture intricate lead-lag relationships among temperature, solar irradiance, and consumption, we implemented a 24-hour input-output segmentation framework. Historically, M&V models have predicted consumption per hour without considering correlations among different hours. The new model processes a full day's data as input and output, enabling it to capture correlations across different time leads/lags for each meter. Generally, model inputs will be shifted such that hour 0 is midnight meaning that for the vast majority of the day, where weather patterns are changing most significantly, leads and lags will be properly fit. If the leads and lags are greater than 24 hours or during nighttime, this framework will still broadly capture the correlated response as weather of a given day is highly correlated with the days surrounding it.

The 24-hour framework is created by taking a feature vector (1×N) as an input and converting it to a matrix (N×24), rather than a vector, to map the input feature to 24 hours of output (consumption prediction). This matrix of coefficients allows the model to learn the lead/lag correlation between the input data and the output data. For instance, consumption in hour 11 can be impacted by the temperature in the hours preceding or following it. With this framework we can account for effects such as thermal inertia or operational schedule variation that may be missed by a purely hourly model.

As a result of the 24-hour framework, we have introduced a significant number of coefficients for the model to fit. Let us revisit the example in Section 3.1.6 where 2 temporal clusters and 6 populated temperature bins resulted in 43 features. These 43 features then create a matrix of 572×24 coefficients + 24 intercepts or in other words 13752 total coefficients. We have more coefficients than we do hours in the year! As previously stated, it is critical that we have a model capable of some level of feature selection so that we do not overfit to the baseline.

3.2.2 Elastic Net

We chose Elastic Net regression as the core modeling framework after an extensive literature review of energy consumption M&V techniques. During the review, we evaluated a range of models, including simple linear regression, decision tree-based models, and advanced approaches like neural networks/transformers. Simple models, such as linear regression, were deemed inadequate due to their inability to perform feature selection and tendency to overfit when all desirable features are included. While decision tree-based models could better handle nonlinearities, they often struggled with overfitting in large datasets and performed poorly when extrapolating. Advanced models like neural networks and transformers were highly capable but came with a significant computational burden, making them impractical for use across large datasets.

Elastic Net offered the proper balance between feature selection, interpretability, and computational efficiency. Coercing the framework to handle both linear and nonlinear relationships made it even more effective for modeling energy consumption. Its computational efficiency ensured that it could scale to large datasets without becoming prohibitively resource-intensive.

Elastic Net is a regularized regression method that combines the strengths of L1 (Lasso) and L2 (Ridge) regularization. This dual regularization approach offers several advantages:

1. Feature Selection and Sparsity: The L1 penalty encourages sparsity in the model by shrinking some coefficients to zero, effectively performing feature selection. This is especially useful when handling a large number of features and coefficients, as it helps focus on the most relevant predictors.
2. Handles Multicollinearity: The L2 penalty helps stabilize coefficient estimates in the presence of multicollinearity, where predictors are highly correlated. By blending L1 and L2, Elastic Net balances sparsity with robustness.
3. Flexible: Elastic Net allows for a mix of linear and nonlinear modeling by adjusting its regularization parameters, making it adaptable to various types of data.
4. Computationally Efficient: Elastic Net is computationally efficient, particularly when applied to large datasets. Its matrix-based optimization methods are well-suited for high-dimensional data, making it scalable for applications involving millions of meters.
5. Interpretable: Despite its ability to model complex relationships, Elastic Net remains interpretable because each feature for each hour has an assigned coefficient. This allows users to interrogate coefficients to gain understanding of the model.

These advantages make Elastic Net an ideal framework for the new hourly model.

3.2.3 Adaptive, Robust, Hour-of-Day Weighting

Hourly data shows high variance over a year where hours range from very negative due to solar PV generation, to zero because of power outages or maintenance, all the way to extremely high usages of EV charging + electric dryers + hot tubs. Managing these outliers is critical to fitting a model capable of predicting the normal usage of a building. We use a robust, adaptive loss function from Barron[4] to accomplish this task.

The adaptive loss function is a continuous function capable of replicating many other loss functions. In addition to this it also has a statistically justified penalization so that when the function is optimized for a given dataset, it does not excessively down-weight outliers.

In Barron's work data is not assumed to be normalized or standardized, but for simplicity, in our implementation of these equations, we always center and rescale the data, such that $\mu = 0$ and $c = 1$, prior to going into these functions. The key parameter in this function is α, also called the shape parameter as it dictates the shape of the loss function as shown in the left panel of Figure 3.8. The right panel of Figure 3.8 shows the loss responses with the penalization factor included. Use of this function ensures that we are optimally down-weighting outliers. If there is little justification to reject outliers, then sum of squared errors will be used, and if there are enough outliers to justify a more extreme loss such as Welsch, it will be selected.

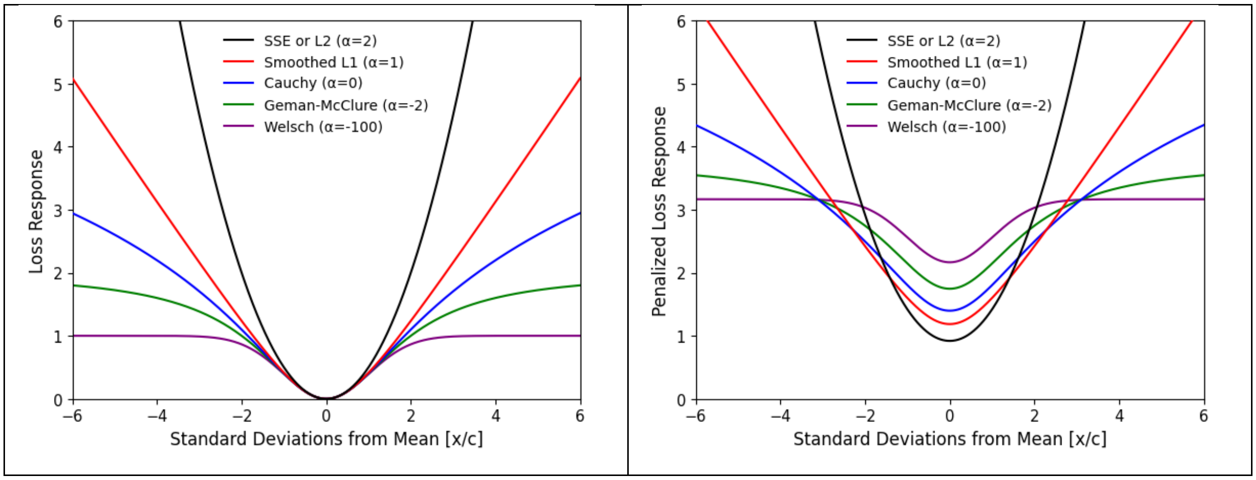

Figure 3.8. How the robust, adaptive loss function responds to values. In the left panel, the values are shown without penalization so that their response curves can be more easily compared. In the right panel are the loss responses with the appropriate penalization. Here $\alpha = -100$ is used as a proxy for $\alpha = -\infty$.

In general, α is selected by optimizing the penalized adaptive, robust loss function shown in the right panel of Figure 3.8. With the optimal α, the last step is to determine the weights. Conveniently, Barron

[4] J.T. Barron, A general and adaptive, robust loss function; Computer Vision Foundation, 2019.

developed a function for this based upon the adaptive, robust loss function. The resulting weight response curves are shown below in Figure 3.9.

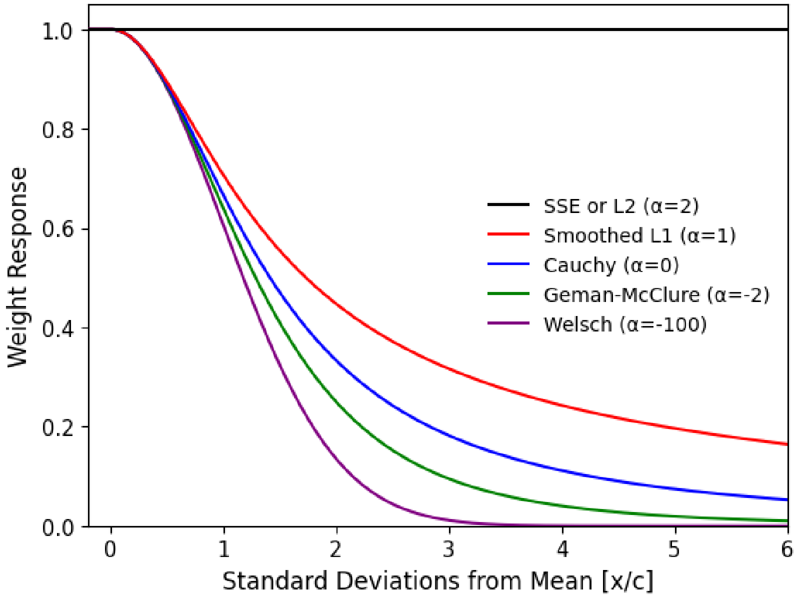

Figure 3.9. Weight response of the adaptive robust loss function based upon α.

In practice for the OpenDSM hourly model, α is determined through an optimization step on the recentered and rescaled residuals for each hour of the day for the entire year. That is to say that there are 24 unique α values corresponding to each hour of the day. Once α has been optimized, each data point is given an appropriate weight as depicted for a few specified α in Figure 3.9. The process of fitting the model to the data and optimizing the adaptive weights is repeated until convergence to account for the model changing as the weights change. The most frequent α used is 2, representing SSE, but other values can be found as well in data with particularly large outliers.

Additional information including the equations for this procedure are given in the appendix (Section 8.3).

4 Error Metrics

Before model performance can be assessed, it is critical that we define what model performance is. Root mean squared error (RMSE) or mean absolute error (MAE) are standard statistical metrics to determine model performance. In building energy modeling, mean bias error (MBE)[5] is another common metric which will be used in this report. RMSE, MAE, and MBE are defined in Eqns. 4.1 – 4.3 as follows:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n-1}} \tag{4.1}$$

[5] ASHRAE. Guideline 14-2002: Measurement of Energy and Demand Savings; ASHRAE: Atlanta, GA, USA, 2002.

$$MAE = \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{n-1} \tag{4.2}$$

$$MBE = \sum_{i=1}^{n} \frac{\hat{y}_i - y_i}{n-1} \tag{4.3}$$

where $\hat{y}$ are the predicted values, y are the observed values, the subscript i represents a given data point, n is the total number of data points. Note that RMSE and MAE will always be positive due to the square term in RMSE and the absolute value in MAE. MBE is a signed metric which, if positive, indicates that the predicted value is larger than the observed value and vice versa. A potential issue with MBE is that it does allow errors to cancel which can result in a situation where one large outlier has an outsized effect on MBE or where a few large errors of one sign cancel out many small errors of the opposite sign.

These metrics would be fine to assess the performance of a model on an individual meter, but we need to determine performance on thousands of meters which have their own unique behavior and magnitudes. Magnitude is particularly important because it means that one meter having a model RMSE of 29 kWh would be acceptable if the meter is regularly using 1000 kWh but would be terrible if the meter is only using 5 kWh. In order to be able to assess model performance on thousands of meters, we must normalize these error metrics by a value so that how well fit a model is can be objectively determined even for meters with largely varying magnitudes.

4.1 Normalized Error Metric Definitions

Historically CVRMSE and NMBE[5] have been the normalized error metrics by which model quality is judged for weather-normalized utility meter modeling. For reference these equations are presented below as Eqns. 4.4 and 4.5.

$$CVRMSE = \frac{RMSE}{\overline{obs}} \tag{4.4}$$

$$NMBE = \frac{MBE}{\overline{obs}} \tag{4.5}$$

where $\overline{obs}$ is the mean of the observed values. These metrics provide a convenient assessment of the size of the model error relative to the meter's overall energy consumption. If they are smaller than 1, the error is smaller than a typical consumption value, which is commonly used as an indicator of an acceptable fit for measurement purposes.

These normalized metrics are generally adequate so long as meters do not have mean observed values which are close to zero or negative. In the case of meters with rooftop PV, however, it is common to see annual average values that are near or below zero because PV systems are often designed to match the annual consumption of the building. When the average value is near zero, the denominators of Eqns. 4.4 and 4.5 become very small resulting in the normalized metrics rapidly approaching vertical asymptotes, even if the model error magnitude is small. Worse yet, when the annual average is below zero, these metrics take on negative values, which are difficult to interpret. In these scenarios, the normalized error metrics are no longer serving their intended purpose, since models will appear to have unacceptable (or unclear) values. With rooftop PV becoming increasingly common in the building stock and one of the primary goals of this model being to improve performance on solar PV meters, it is clear that we need new normalized metrics for assessing model fit.

To address both issues of negative and near-zero denominators, we developed an alternative that will be generally applicable to meters with behind-the-meter generation as well as those without. Several options were considered such as only averaging positive signals, averaging nighttime signals, or using peak load. We ultimately chose to instead utilize the interquartile range (IQR) of the observed load as a means to normalize the error, in place of the average. The IQR is useful in this context because it will always be positive. It is still possible for the IQR to be near zero, but in that case the meter would have very little variability in its energy consumption, so the model error would also be expected to be small. The resulting metrics are referred to as percentile-normalized RMSE (PNRMSE) and percentile-normalized MBE (PNMBE) and are defined in Eqns. 4.6 and 4.7.

$$PNRMSE = \frac{RMSE}{obs_{Q_3} - obs_{Q_1}} \tag{4.6}$$

$$PNMBE = \frac{MBE}{obs_{Q_3} - obs_{Q_1}} \tag{4.7}$$

where $obs_{Q_1}$and obs_{Q_3} are the first (25%) and third (75%) quartiles of the observed consumption values, respectively.

4.2 Disqualification Criteria

Models within OpenDSM are frequently used to measure the impact of demand response events and/or energy efficiency interventions. In this context, some meters are more regular and able to be modeled than others. This implies that there are also meters which are random and unable to be modeled well which means that their predictions would be of poor quality and not reflective of their true usage. It is therefore necessary to have metrics by which a meter can be "disqualified" in the event that the model is unable to predict future usage with adequate accuracy. As discussed above, CVRMSE and NMBE are inadequate to disqualify meters with behind-the-meter generation due to their tendency to take on inflated (or negative) values when annual metered consumption is small. PNRMSE and PNMBE have the potential to act as replacements or supplement the existing normalized metrics, but given that CVRMSE has been the standard, we will need to see how PNRMSE compares to it to determine if it is a suitable replacement.

To see how well PNRMSE correlates with CVRMSE, 1000 non-solar residential customers' error values are compared in Figure 4.1. By comparing these two metrics on meters where CVRMSE is valid (i.e., for non-solar customers), we can assess the relationship between the two. It is common to think about CVRMSE as a measure of error relative to how much energy a meter uses. PNRMSE can be thought of in a similar manner except that it is normalized by the variation in a meter's usage, which provides a more stable and robust normalization when consumption values can be negative.

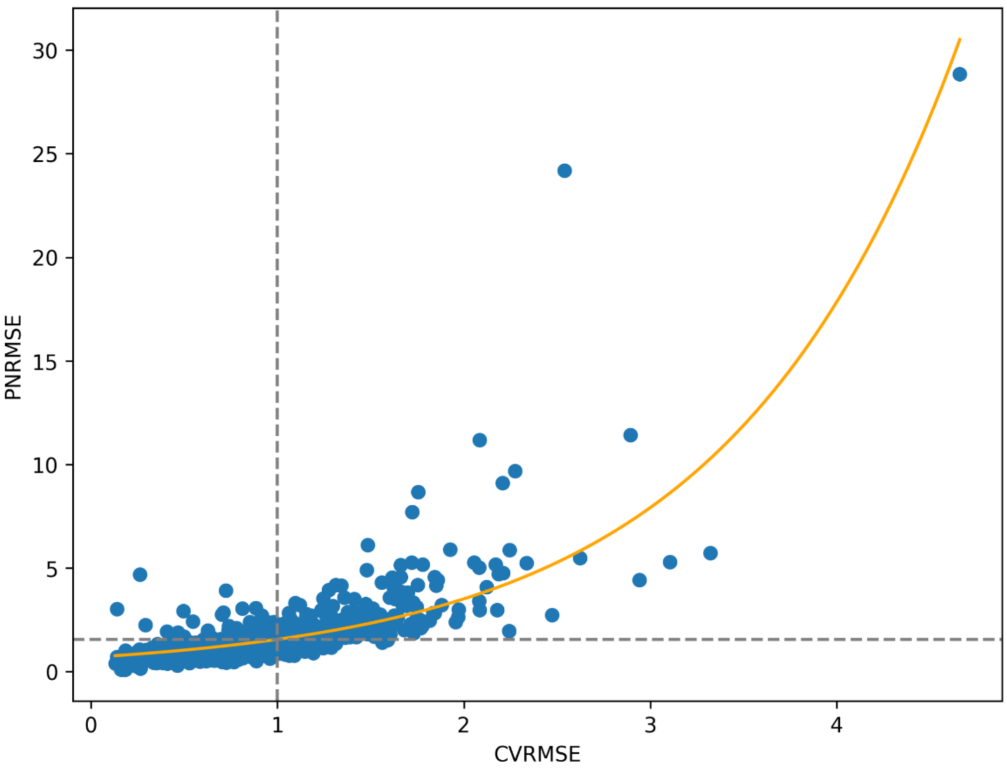

Figure 4.1. Correlation between PNRMSE and CVRMSE for 1000 non-solar meters. Each meter's metric is represented in blue and a best fit in orange. Dashed lines depict the intersection of CVRMSE at 1 with PNRMSE at 1.6 utilizing the fit curve.

Since PNRMSE and CVRMSE are fairly well correlated, it might be reasonable to consider replacing CVRMSE with PNRMSE for the purposes of judging goodness of fit. However, we are not proposing such a drastic change, because CVRMSE remains a useful metric for non-solar meters, which are still the majority of meters in nearly all contexts, and because PNRMSE also can become artificially large in some cases. In many cases, it will be equally reasonable to judge goodness of fit using either CVRMSE or PNRMSE. However, both metrics are inadequate in certain scenarios: CVRMSE will tend to be artificially high when the average consumption is small (e.g., in when rooftop PV is present) and PNRMSE will be artificially high when the IQR is small (e.g., for buildings with very flat load profiles). We therefore propose that a meter's model should be considered sufficient if it would meet goodness-of-fit criteria for either metric. This ensures that, for example, a meter with rooftop PV that receives an inflated CVRMSE value will have a second chance to be qualified for measurement via its PNRMSE statistic.

What remains, then, is to determine appropriate disqualification thresholds in both statistics. The OpenEEmeter 4.1 daily model recommends a CVRMSE threshold of 1.0 to disqualify meters from measurement. Such a threshold is likely to be too conservative for use with hourly data which shows significantly more variability than daily data. To determine an appropriate CVRMSE threshold for hourly data, we analyzed thousands of meters fitted using both the daily model and the new hourly model; non-solar meters with a daily CVRMSE of 1.0 were typically found to have an hourly CVRMSE of 1.4. We therefore recommend that the CVRMSE threshold for hourly models be 1.4 and that the PNRMSE threshold be set at a corresponding 2.2, based on the trend line shown in Figure 4.1. Moreover, models that yield negative CVRMSE values should be considered to have failed the CVRMSE disqualification criterion, since their CVRMSE values are hard to interpret, and the disqualification decision should be made based on PNRMSE. In summary, we recommend that a model fit should be considered sufficient for measurement purposes if $0 \leq CVRMSE \leq 1.4$ *or* $PNRMSE < 2.2$.

5 Hyper-Parameter Optimization

5.1 Test Population

To fine-tune the new hourly model, we conducted extensive hyperparameter optimization using data from over 33,000 residential and commercial non-participant meters. More specifically, data was used from MCE's service territory, primarily in the California Bay Area and North Coast regions as previously utilized in developing comparison groups and demand response methodologies.[6] The regions that the data are from tend to be temperate to hot with few cold weather locations present. In this particular case, three years of data were utilized ending on March 1, 2020 in order to avoid the impacts of the COVID-19 pandemic and to ensure a steady time period with as few changes as possible. Meters were selected that had not enrolled in EE or DR programs, to ensure that no known interventions have been performed. Using non-participant meters means that the reporting year savings is expected to be zero at a population level (within uncertainty bands and given that there is likely some small population drift).

To maximize the amount of data available for optimization and comparison purposes, we used a rolling test/train period, as shown in Figure 5.1. Each rolling period of each meter is treated as a unique meter. This offers flexibility in which we can either continue to treat them as independent meters or average the rolling periods for each meter to get an expected result.

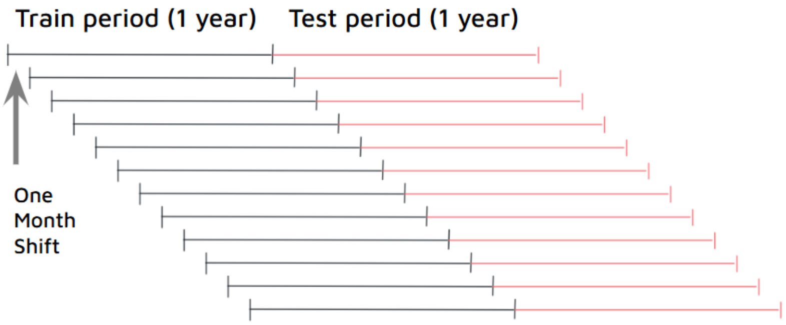

Figure 5.1. Rolling test/train periods utilized in the test population to reduce outlier behavior unduly affecting modeling results

Despite limiting the data to only non-participant meters, we found that there were still significant differences between the training period and test periods for some meters, with certain meters showing

[6] Demand Response Advanced Measurement Methodology; CAISO, 2021

drastic and lasting increases or decreases in consumption (i.e., non-routine events). In some cases, such meters were particularly influential (e.g., large commercial buildings with evident operational changes). This was causing some configurations of the new model to appear "better" according to error metrics, but it was simply due to random chance on some of these influential meters that exhibited differences in the training and test data.

To address the period variance issue, we developed a methodology in which we compared the difference between the test period and training period on a seasonal hour-of-week loadshape basis. The difference between these two was normalized by both the average of the observed (effectively the loadshape CVRMSE) and the IQR of the observed (the loadshape PNRMSE) to normalize each meter. For a given subsample, the distribution of both the loadshape CVRMSE and PNRMSE could then be analyzed. Meters were rejected and removed from the training set if they fell outside either the CVRMSE or PNRMSE threshold as determined by using the 1.5 IQR outlier rule. This helped to significantly clean up the datasets and eliminate meters with large changes in year-on-year consumption.

The combination of treating rolling periods as unique meters and then cleaning the datasets with the aforementioned algorithm leaves 21336 meters for the hyperparameter optimization and a further 12907 meters for use in evaluation as an out-of-sample population. A more detailed distribution of these meters is given in Table 5.1.

Table 5.1. Breakdown of development/evaluation populations

Sector	Solar PV Status	Development Meter Count	Evaluation Meter Count
Residential	Non-Solar	7845	7372
	Solar	2664	2882
Commercial	Non-Solar	5269	1234
	Solar	5558	1419
Total	All	21336	12907

While this dataset lacks significant cold weather data (due to its location in coastal California), in every other way it gives us a robust representation of diverse real-world buildings suitable for performing hyperparameter optimization.

5.2 Objective Function

The hyperparameter optimization was conducted at a population level, ensuring that the values selected were optimal for the entire population level across both residential and commercial sectors. While important, the objective function's primary purpose was to guide guesses. During each evaluation, 130 various aggregate metrics were recorded. The objective function that guided these guesses is the summation of the metrics and weights shown in Table 5.2.

Table 5.2. Objective function metrics and weights.

Metric	Period	Weight
Annual PNRMSE	Baseline	0.0252
	Reporting	0.2268
Seasonal hour-of-week loadshape PNRMSE	Baseline	0.0252
	Reporting	0.2268
Extreme temperature PNMAE	Baseline	0.0180
	Reporting	0.1620
GHI loss bins PNRMSE	Baseline	0.0216
	Reporting	0.1944
% Absolute loadshape overfit	Baseline/Reporting	0.1000

The annual PNRMSE is the PNRMSE over the entire baseline or reporting year, as appropriate. This metric will give an assessment of overall performance. Given that this is an hourly model, we also need a metric which ensures that time-based behaviors are not lost, so we accomplish this by aggregating the data to seasonal hour-of-week loadshapes and comparing the predicted vs observed using the residuals' PNRMSE. The extreme temperature PNMAE is calculated by looking at error within the lowest 5% and highest 5% of temperatures. The mean PNMAE of these metrics is called the extreme temperature PNMAE as it represents how the model performs in these regions using a metric which does not cancel but also has a linear response to error similar to MBE. Similarly, he GHI loss bins PNRMSE is calculated by binning the data based upon GHI loss $\left(GHI - GHI_{clearsky}\right)$. The RMSE is calculated within each bin and then the RMSE of all of the bins is calculated and normalized by the IQR. This helps to push the model to minimize the error based upon cloudiness. Finally, the % absolute loadshape overfit is the absolute value of the reporting year, seasonal hour-of-week loadshape divided by the same metric derived from the baseline year minus 1. The absolute value is taken so that underfitting is not preferred to overfitting and it is shifted to 0 so that in the optimal case where the baseline error = reporting error would be 0.

This objective function served to guide the optimization, but did not dictate the final hyperparameters; rather it showed promising areas of hyperparameter space that were further tuned through detailed investigation.

While performing the optimization numerous other error metrics were stored for many permutations of PNRMSE, PNMAE, and PNMBE for years, months, extreme temperature bins, and various GHI bins. These metrics were compared against one another and after a final full factorial comparison between all of the top contenders, the final hyperparameters were selected.

5.3 Optimized Parameters

The key hyperparameters optimized during this process included parameters of the Elastic Net, the temperature binning definition, the adaptive weighting parameters, and parameters of the temporal clustering algorithm. The specific hyperparameters optimized in this process are as follows:

Elastic Net Parameters (Section 3.2.2)

1. Alpha: This parameter controls the overall strength of the regularization applied in the model. Higher values of alpha increase the regularization effect, shrinking coefficients more aggressively, which can help prevent overfitting but may reduce the model's ability to capture complex patterns.
2. L1 Ratio: This parameter defines the balance between L1 regularization (Lasso) and L2 regularization (Ridge).

Temperature Bin Parameters (Section 3.1.1)

3. Temperature Bin Option: Select among the following fixed temperature binning options:

Bin Type	**Bin Edges [°F]**
CalTRACK bins + extreme bins	[10, 30, 45, 55, 65, 75, 90, 100]
Modified CalTRACK bins	[10, 30, 50, 65, 75, 90, 105]
Centered at 65°F with 15°F bins	[5, 20, 35, 50, 65, 80, 95, 110]
Centered at 65°F with 20°F bins	[5, 25, 45, 65, 85, 105]
Centered at 70°F with 15°F bins	[10, 25, 40, 55, 70, 85, 100]

Adaptive Weighting Parameters (Section 3.2.3)

4. Adaptive Weights Sigma: At how many equivalent standard deviations does the down-weighting begin to take effect.
5. Adaptive Weights Threshold Algorithm: There are several different ways to calculate the weighting threshold. Options included standard deviation, median absolute deviation, and using the interquartile range rule. All options have been scaled to be equivalent when used on a normal distribution.
6. Adaptive Weights Window: Each hour could look at surrounding hours to ensure that a single hour's weighting threshold is informed by surrounding hours in case multi hour events would overly penalize a single hour due to random chance.

Temporal Clustering Parameters (Section 3.1.3)

7. Spectral Clustering Gamma: This parameter affects the number of clusters created when spectral clustering is used.
8. Clustering Scoring Metric: Which scoring metric is used to select the optimal number of clusters.

The final parameter values are summarized in Table 5.3.

Table 5.3. Final optimized hyperparameters for the new hourly model.

Optimized Parameters	Value
Alpha	0.0139
L1 Ratio	0.871
Temperature Bin Option	Modified CalTRACK Bins
Adaptive Weights Sigma	4.55
Adaptive Weights Threshold Algo	IQR
Adaptive Weights Window	3 (2 hours surrounding)
Spectral Clustering Gamma	1.05
Clustering Scoring Metric	Variance Ratio

These hyperparameters serve as the foundational parameters for the model and remain constant across all meters. While the hyperparameters are consistent across the population, the primary model coefficients and parameters, such as the weights assigned to individual features, are unique to each meter. These coefficients are determined based on the baseline data and input features for each meter, allowing the model to adapt to specific usage patterns and environmental factors.

These parameters are designed to work for all versions of the new hourly model including both the solar model and non-solar model. It is recommended to only use the non-solar model, which does not utilize solar irradiance data, on non-solar customers. Use of the non-solar model on solar customers will have reduced predictive capabilities on par with the OpenEEmeter 4.1 model we are seeking to improve upon. The solar model incorporates solar irradiance data, can be used on both solar and non-solar customers, and is encouraged for universal application if possible.

6 Population Results

These results show data from all meters described in Table 5.1 from Section 5.1 including both in-sample and out-of-sample meters. None of these meters received a known intervention–i.e., they are all "non-treatment" or "non-participant" meters in the context of measuring the impacts of demand-side interventions. Hence, the expectation is that the overall change from baseline to reporting year will be small on a population level. Before we present the results, it should be highlighted that the new model is between 4 – 5x faster than the prior hourly model. This should result in significant savings when run on thousands to millions of meters when used in practice. In the interest of a fair comparison between the legacy and the new model, only reporting year (testing period) results will be shown, rather than baseline period (training period) results, unless otherwise specified. This is because the OpenEEmeter 4.1 hourly model is known to be significantly overfit and hence will appear more accurate when applied to training data than it will when being used for prediction. Comparing the models' performance on reporting period data thus allows for a fair comparison.

The OpenEEmeter 4.1 hourly model will be referred to as the legacy model herein. The non-solar model refers to the newly developed hourly model without solar irradiance used as an input and the solar model refers to the new model using solar irradiance.

6.1 Computational Efficiency

Before we present the results, it should be highlighted that the new model is between 4 – 5x faster than the prior hourly model. This will result in significant cost savings when run on thousands to millions of meters in practice. For example, assume that new model takes 2 seconds on average to fit a meter and the legacy model 10 seconds. Fitting and predicting 100,000 meters would take $\sim$2.5 cpu-days instead of $\sim$11.5 cpu-days.

6.2 Non-Solar Meter Results

Two of the goals we laid out for the new model were to meet or exceed the accuracy of the legacy model and reduce overfitting. This section examines the new model's performance against those goals for meters without behind-the-meter solar.

Figure 3.8 demonstrates that the new model has a significant reduction in overfitting. The figures show distributions of an overfitting score assigned to individual model runs for a large number of residential (upper panel) and commercial (lower panel) meters. The overfitting score compares the RMSE of the model in the baseline and reporting periods: a model that is not overfit at all would be expected to have equal RMSE values in both cases. Here, we see that the distribution of the overfitting score is shifted significantly toward zero for the new hourly model, regardless of whether it is run with or without solar irradiance data.

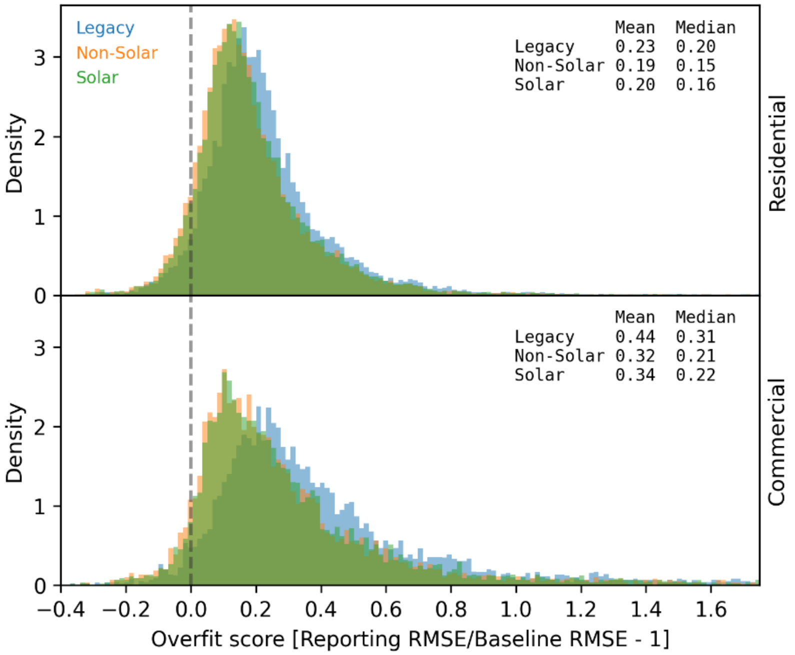

Figure 6.1. Overfitting using annual RMSE compared between the legacy, non-solar, and solar models on non-solar data for both residential and commercial meters. The overfit score is defined as the reporting year RMSE/baseline year RMSE – 1. A perfectly fit model would have the same RMSE for reporting and baseline therefore it would have an overfit score of zero, represented here as the grey dashed line. Values below zero mean that the model exhibits better accuracy in the reporting period than in the baseline period, while values greater than zero indicate lower accuracy in the reporting period.

On residential meters, the non-solar model sees a 17% average improvement compared to the legacy model; whereas, the solar model is slightly more overfit but still shows a 13% average improvement. We see even more improvement for commercial customers with 27% and 23% improvements over the legacy model for the non-solar and solar models, respectively. Median improvements are slightly higher as one might expect from the skewed distribution of the histograms.

Because we disqualify meters based upon normalized RMSE, it is important that our models be as well fit as possible. In the case of the legacy model, these results would indicate that either the thresholds at which we disqualify meters are either too large to account for the amount of overfitting that we see, or we are not disqualifying meters because they show low error when in fact, they should be disqualified given their poor reporting-period performance. By reducing the degree of overfitting, we are reducing the impact of such concerns. In fact, it may be valuable in future work to revisit the data sufficiency and disqualification criteria to better reflect the accuracy of the new model.

Over the course of a year the new models have approximately equivalent accuracy to the legacy model when applied to meters without PV. Figure 6.2 shows this through histograms of percent improvement of the new model's error metrics over the legacy model. It shows CVRMSE and NMBE improvement distributions for meters without behind-the-meter solar for both the solar and non-solar models.

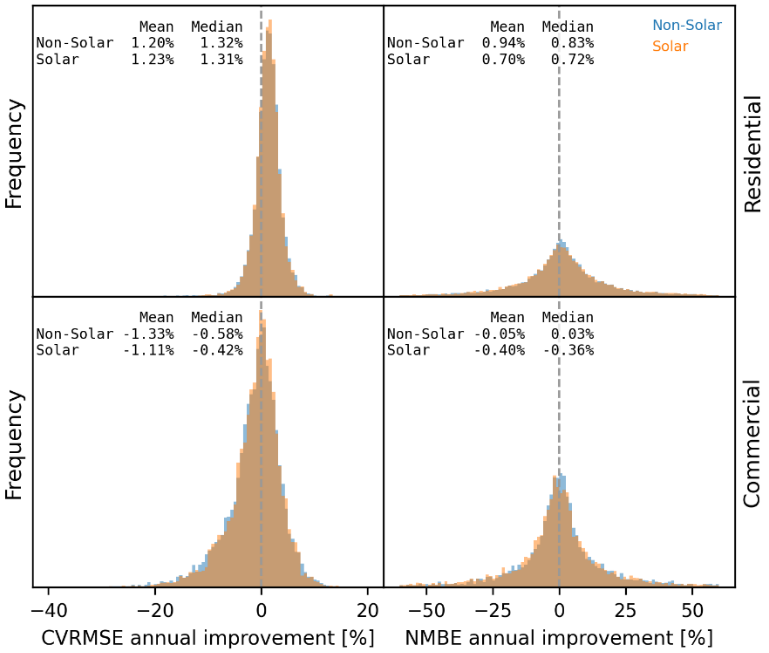

Figure 6.2. Histogram of the new model's improvement over the legacy model on non-solar residential (upper panels) and commercial meters (lower panels) for both CVRMSE and NMBE (left and right panels). A dashed vertical line represents matching the accuracy of the legacy model.

The CVRMSE distribution for residential customers is approximately normally distributed with a mean value of 1.2%, indicating a small overall improvement in model accuracy compared to the legacy model. For commercial customers, the distribution is more skewed with a mode that is near zero, but the negative tail pulls the mean to a small reduction in accuracy of ∼1.2% (averaging the two models). The NMBE distributions for all models in all sectors exhibit little skewness. The residential sector shows an insignificant improvement of 0.8% and the commercial sector shows an equally insignificant reduction of ∼0.2%. Overall, the models are nearly equivalent with small differences depending upon sector. Individual meters may show more variation, but this is inevitable when a significant modeling reformulation is performed.

While the annual results are promising, since this is an hourly model, it is also of interest to examine the model's accuracy at an hourly resolution. An 8760-hour plot has far too much variation and detail to glean information from, so we typically will aggregate the information to seasonal averages by hour of week for the sake of presentation. The following 4 plots, Figure 6.3 – 6.6 show seasonal hour-of-week comparisons between the observed, legacy, and new models for both residential and commercial non-solar meters with the plots starting at midnight on Monday. In each of these plots the first panel shows the average baseline loadshape normalized by the average observed consumption over the course of the baseline year; all panels are normalized by the average observed usage. The second row shows the normalized average reporting year loadshape, and the following two rows show the normalized baseline year error and normalized reporting year error.

Figure 6.3 and Figure 6.4 are very similar and are only different in that Figure 6.3 is the non-solar model and Figure 6.4 is the solar model used on residential meters without behind the meter solar. The seasons in these figures are defined using months as summer: 6, 7, 8, 9; shoulder: 4, 5, 10, 11; and winter: 1, 2, 3, 12. The hour of the week starts on Monday at midnight. The most notable difference between the legacy model and the new models is that the new models appear to have somewhat reduced accuracy on the weekends while exhibiting slightly better accuracy during the summer and winter weekdays, as evidenced by the reporting-year error's being more consistently near zero. The solar model seems to do slightly better in that its errors are slightly more centered around zero compared to the non-solar model.

A more subtle but critical difference between the models can be seen by comparing the error in the baseline against the reporting-year error for each of the models. Here it becomes clear that the baseline error profile for the legacy model is not representative of the reporting year error in either magnitude or behavior. This indicates the significant overfitting in the legacy model that we have noted elsewhere. Importantly, if the baseline error is not representative of the reporting year error, then it is not a good indicator of the expected quality of the model's predictions, nor of the expected error in measurements performed using those predictions.

Conversely, the new models' baseline error profiles are quite similar to one another, meaning that they can be used reliably to understand the error in model predictions. Moreover, if one were to create comparison groups based upon model error, such as is done in the comparison group clustering method in OpenDSM's GRIDmeter module, this would improve the accuracy of corrections resulting from this methodology using the new models.

Similar plots are shown for commercial meters in Figure 6.5 and Figure 6.6. Again, accuracy is modestly improved on weekdays and reduced on weekends. There is also a striking signature of overfitting in the legacy model, with very small errors in the baseline period but much larger errors in the reporting period. By contrast, the error profiles of the new hourly model are consistent across both periods. This is yet more evidence that the overfitting in the legacy hourly model leads to error metrics that are not reflective of predictive accuracy. The accumulating evidence leads us to the conclusion that even if the new models had mildly reduced accuracy, it would still be beneficial to adopt the new models so that error the metrics could be trusted. After all, measurements are only as good as their uncertainty.

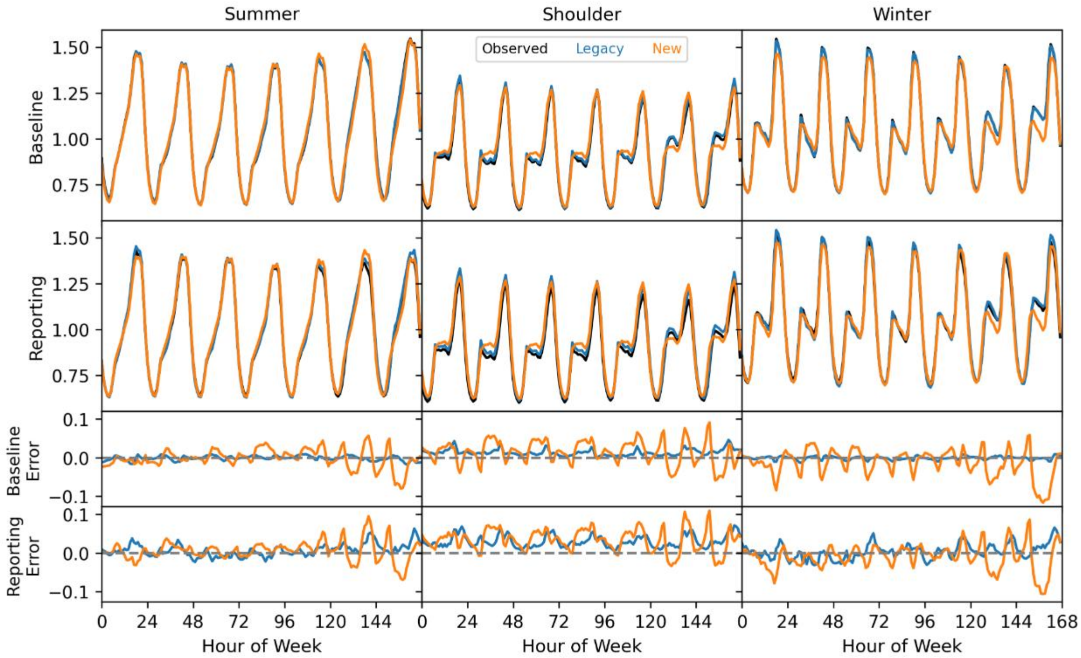

Figure 6.3. Mean seasonal hour-of-week loadshapes normalized by the baseline annual average for non-solar residential customers using the legacy and non-solar model. The top two panels show the baseline and reporting year loadshapes. The bottom two panels show the baseline error and reporting error loadshapes.

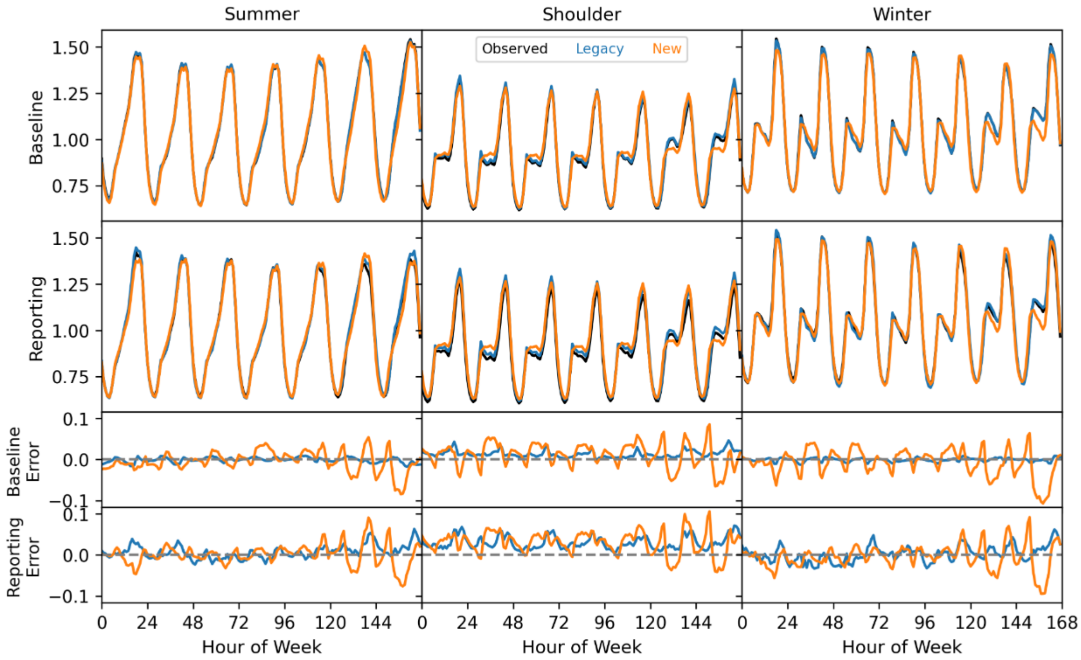

Figure 6.4. Mean seasonal hour-of-week loadshapes normalized by the baseline annual average for non-solar residential customers using the legacy and solar model. The top two panels show the baseline and reporting year loadshapes. The bottom two panels show the baseline error and reporting error loadshapes.

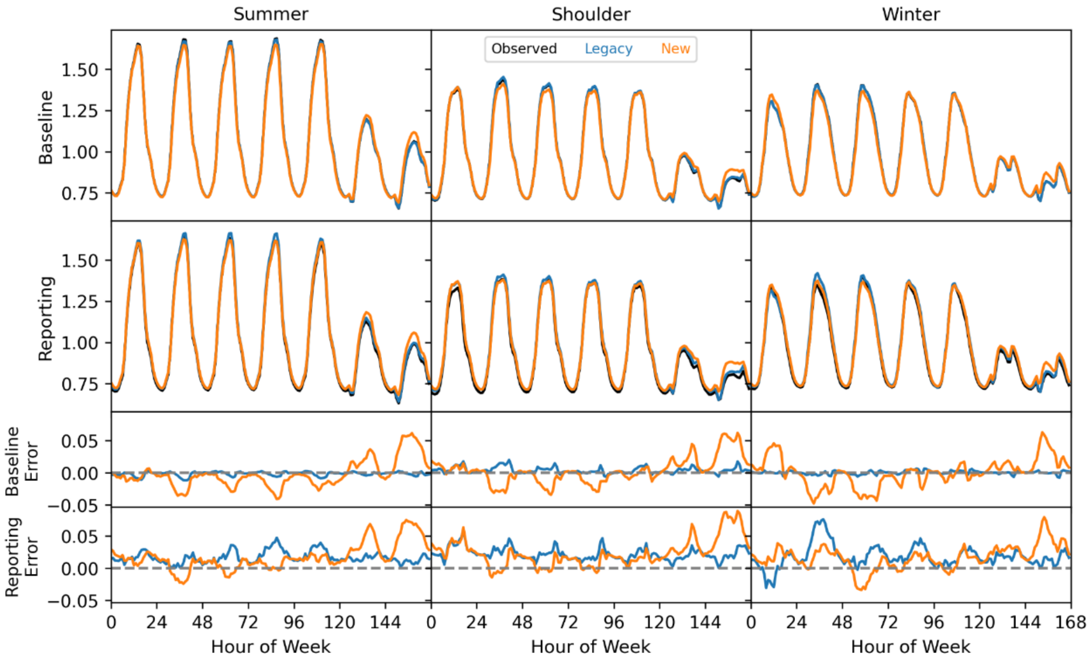

Figure 6.5. Mean seasonal hour-of-week loadshapes normalized by the baseline annual average for non-solar commercial customers using the legacy and non-solar model. The top two panels show the baseline and reporting year loadshapes. The bottom two panels show the baseline error and reporting error loadshapes.

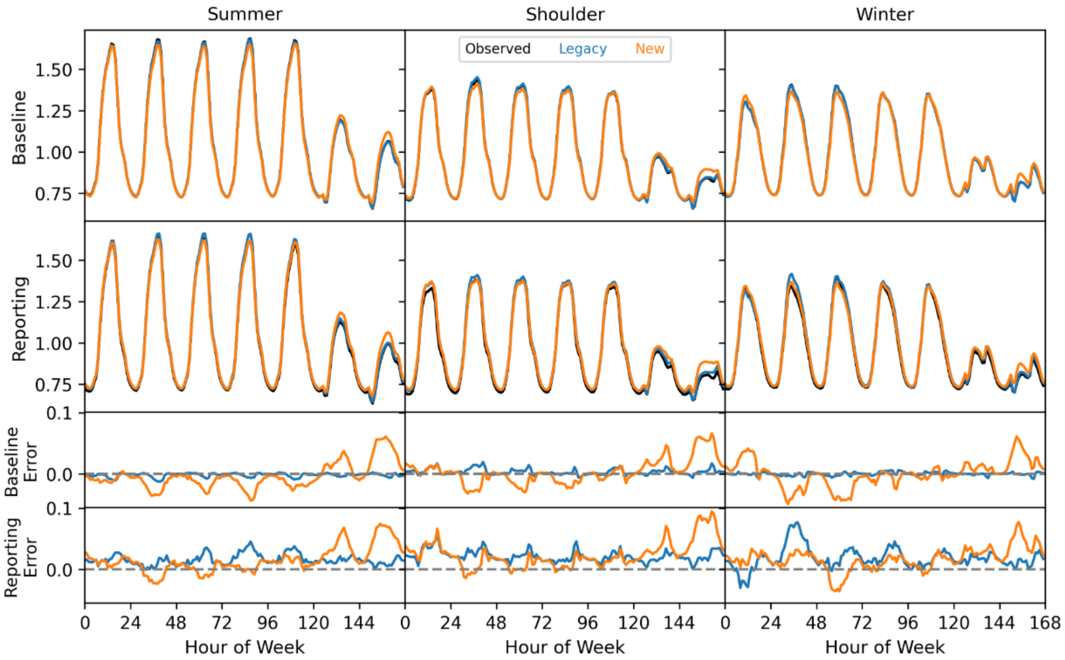

Figure 6.6. Mean seasonal hour-of-week loadshapes normalized by the baseline annual average for non-solar commercial customers using the legacy and solar model. The top two panels show the baseline and reporting year loadshapes. The bottom two panels show the baseline error and reporting error loadshapes.

Another view that one could take of the models' performance is to ask how they perform as a function of temperature, since it may be of particular interest to measure savings on peak days, for example. We address this question in Figure 6.7 and Figure 6.8. These figures show the average NMBE and CVRMSE for temperature bins defined for each meter using the non-solar model on an hourly basis. The temperature bins are 5% quantiles, so that bin 1 can be viewed as the lowest 5% of temperatures that each meter sees and bin 20 being the highest 5% of temperatures. Figure 6.7 shows this for residential customers and Figure 6.8 for commercial customers.

These plots show mixed results, with broadly similar model performance between the legacy and new models. In terms of NMBE, the absolute magnitudes of the binned results are roughly the same between the legacy model and the new models. For residential customers, we see slightly more bias in NMBE at high temperatures, but less at low temperatures. Commercial customers show an improvement at both low and high temperatures. Mid temperatures of both residential and commercial customers are roughly the same between both models. CVRMSE is more consistent in showing improvement (albeit small) for the new model in most bins except at high temperatures. Our takeaway is that from this view, the models have nearly equivalent performance with small trade-offs depending upon which temperature bin is compared.

With all of this information, we argue that we have met our goals of meeting or exceeding the legacy model's accuracy while reducing overfitting. The overfitting results show a clear and stark improvement. Annual fit-quality metrics are nearly identical with differences at the 1% level. When looking at loadshapes, we largely see minor improvements in weekday predictions trading off against minor reductions in weekend prediction quality. Finally, when looked at from the viewpoint of temperature, the new models perform just as well as the legacy model.

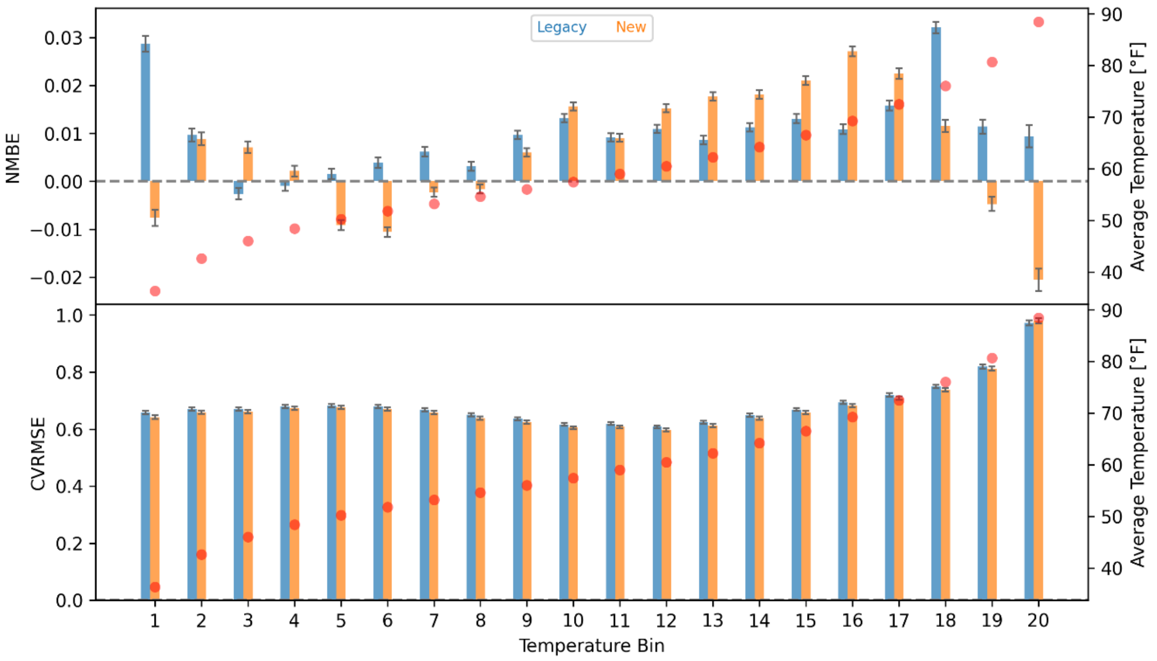

Figure 6.7. Average error metrics for residential customers binned by temperature. Temperature bins are defined as 5% quantiles for each meter so bin 1 will always show the lowest temperatures experienced by each meter and bin 20 the highest. The average temperature for each bin is given as red circles and correspond to the secondary y-axis on the right.

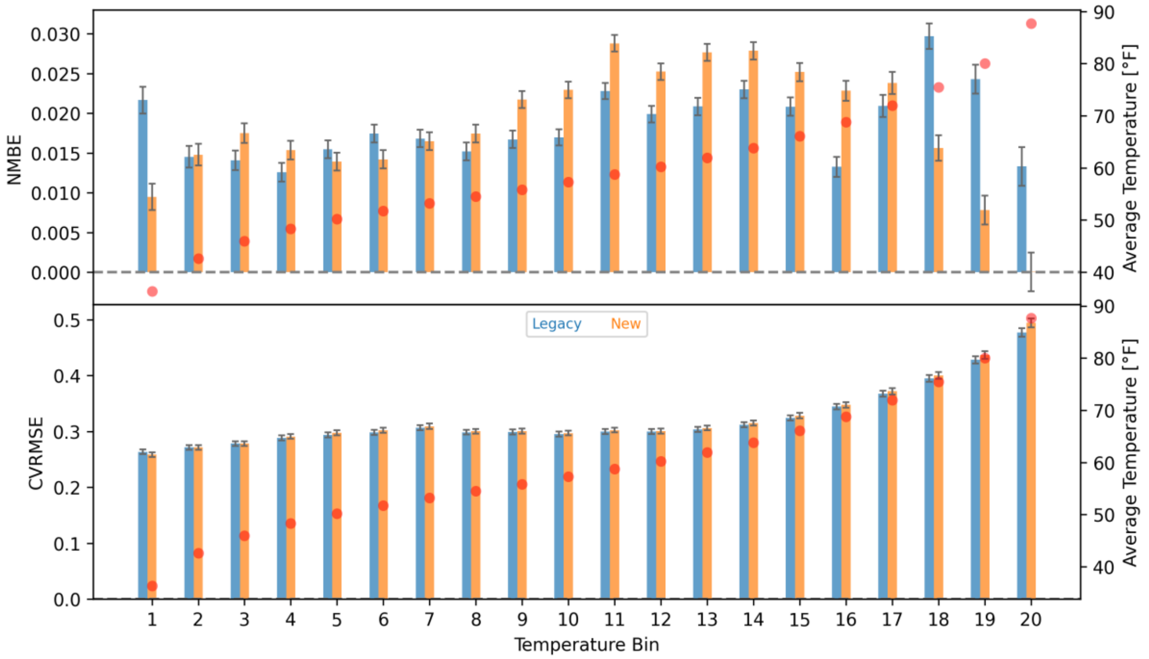

Figure 6.8. Average error metrics for commercial customers binned by temperature. Temperature bins are defined as 5% quantiles for each meter so bin 1 will always show the lowest temperatures experienced by each meter and bin 20 the highest. The average temperature for each bin is given as red circles and correspond to the secondary y-axis on the right.

6.3 Solar Meter Results

In this section we will compare the new solar model vs. the legacy model on solar meters. It is important to highlight that our expectation is that we will minimize the risk of making poor predictions in the context of variable cloud cover, not necessarily that solar predictions will improve on an annual basis. The legacy model is fit over the course of a year, so if the reporting year has a similar proportion and seasonal distribution of sunny days as the baseline year (a likely scenario in many regions), the model's prediction will be reasonably accurate on an annual basis. But relying on this to be true creates an uncontrolled potential for error in the measurement of demand-side interventions. This translates into significant financial risk when real-world projects are compensated based on measured performance. Where we can make significant improvements with the new model is on cloudy days or days which differ from the average solar irradiance that the model was trained on in the baseline period. This will significantly de-risk measurement for solar meters.

Figure 6.9 and Figure 6.10 show the seasonal hour-of-week loadshape comparison for solar residential and commercial meters, respectively, as described in Table 5.1. Focusing on the reporting-year error profiles, residential meters have similar error characteristics in the legacy and new models, except in the winter season, where the new model exhibits notably better performance. Given that winter is more likely to have cloudy days in northern California, this outcome is in line with expectations. The overfitting issue with the legacy model is also even more noteworthy for these meters, especially in winter, where the legacy model's error profile indicates a very accurate fit, but the prediction error is extremely large by comparison. Inspection of the plots for commercial meters yields similar conclusions.

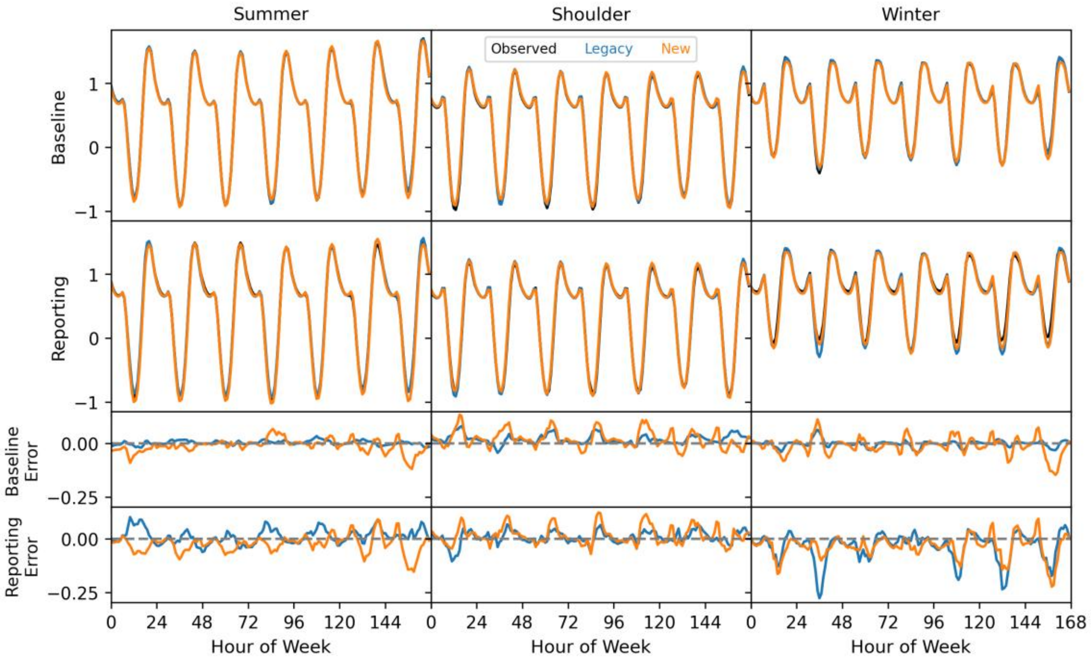

Figure 6.9. Mean seasonal hour-of-week loadshapes normalized by the baseline annual interquartile range for solar residential customers using the legacy and solar model. The top two panels show the baseline and reporting year loadshapes. The bottom two panels show the baseline error and reporting error loadshapes.

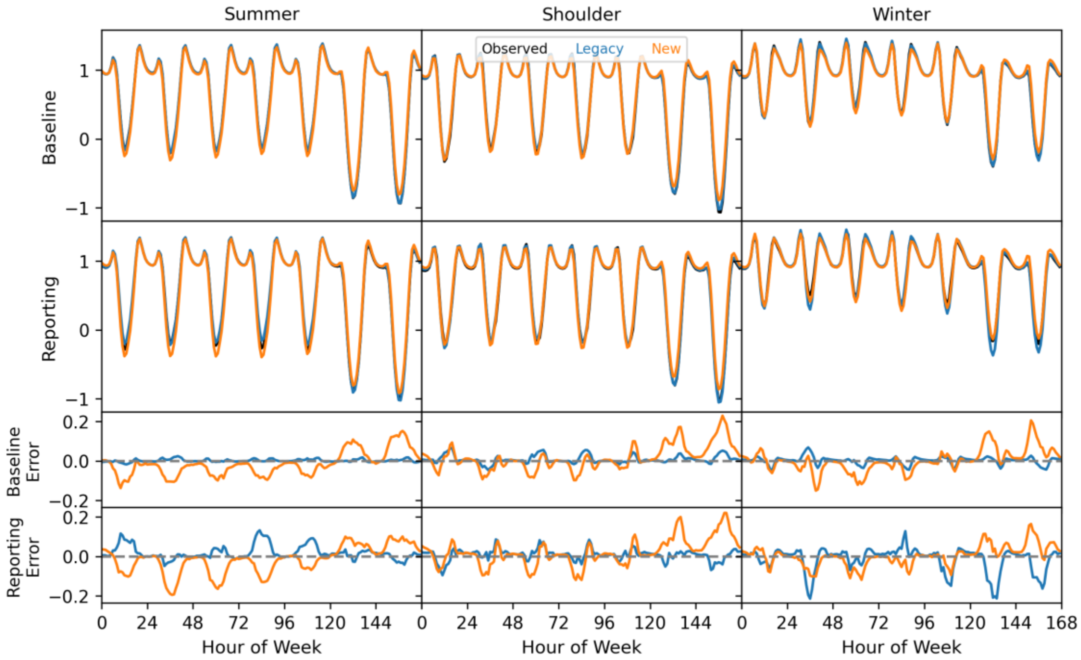

Figure 6.10. Mean seasonal hour-of-week loadshapes normalized by the baseline annual interquartile range for solar commercial customers using the legacy and solar model. The top two panels show the baseline and reporting year loadshapes. The bottom two panels show the baseline error and reporting error loadshapes.

Figure 6.9 and Figure 6.10 strongly suggest that the new model is yielding substantially more accurate predictions on cloudy days. To address this topic more directly, Figure 6.11 and Figure 6.12 show comparisons between the new solar model with GHI included and the legacy model for residential and commercial meters, respectively.

These figures show average model error in bins of the difference between the expected clear-sky GHI and the actual GHI, which represents a good indicator of cloudiness. The bins in both figures are constructed to have equal sample sizes in each bin. This yields more data points to the left side of each plot with increasingly separated data points towards the right, because there are fewer cloudy days than sunny days. The upper panels show PNMBE; the better model will be closer to the dashed line at 0. The lower panels show PNRMSE; the ideal result would be for the new model to be lower than the prior model.

Both residential and commercial meters show a minor PNMBE improvement for very sunny days and a significant improvement for moderately to heavily cloudy days. Residential PNRMSE is better at all levels of cloudiness. Commercial PNRMSE shows equivalent performance on very sunny days with significant improvement as cloudiness increases. The story here is clear: the solar model derisks predictions for solar PV customers with no caveats.

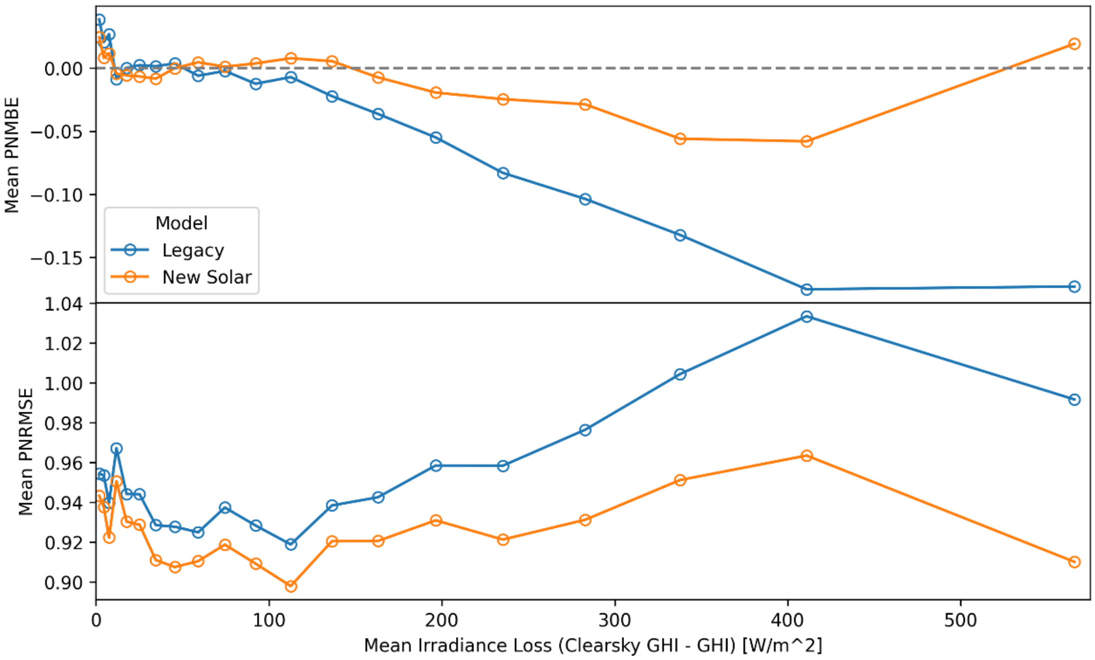

Figure 6.11. A comparison of model performance on residential meters with solar PV showing mean PNMBE (upper panel) and PNRMSE (lower panel) vs. the average hourly irradiance loss for bins with equal sample sizes. The horizontal axis represents an indicator of cloudiness, with 0 being a sunny day with no clouds and 500 being a very overcast day.

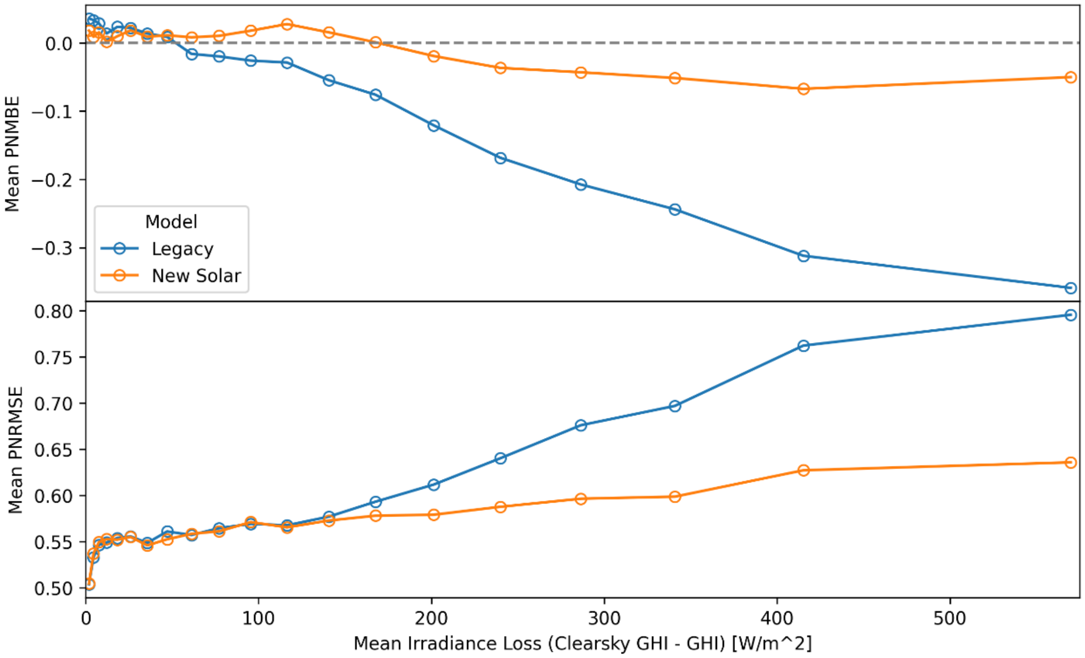

Figure 6.12. A comparison of model performance on commercial meters with solar PV showing mean PNMBE (upper panel) and PNRMSE (lower panel) vs. the average hourly irradiance loss for bins with equal sample sizes. The horizontal x-axis represents an indicator of cloudiness, with 0 being a sunny day with no clouds and 500 being a very overcast day.

7 Conclusion

In this work, we have developed a new hourly model for OpenDSM with several goals in mind.

1. Improved Solar PV Prediction: Better handling of meters with solar PV by integrating solar irradiance data to capture variability caused by insolation and cloud coverage.
2. Performance Retention and Enhancement: Maintain or exceed the legacy model's accuracy for non-solar PV meters.
3. Well-fit Model: Developed model is neither underfit nor overfit, but is well fit such that baseline error metrics are reasonably predictive of reporting year error metrics.
4. Flexibility: Incorporate supplemental data, such as additional time series or categorical variables, when available, to enhance predictions.
5. Faster Computation: Ensure that the model performs computations more efficiently.

The details of the model framework have been described, as has the optimization framework, the test population, and the final optimized model. The question is: did we accomplish our goals? The answer can be summarized by a table of improvements and losses.

Losses	Improvements
-	Derisked solar PV meter predictions
Modestly reduced weekend accuracy	Comparable or better weekday accuracy
-	Considerably less overfit
-	Allows R&D supplemental data
-	Simplified code structure and API
-	4 to 5 times faster

Given these results, we recommend that the new model be accepted as a replacement of the prior OpenEEmeter 4.1 hourly model in all aspects, both non-solar and solar applications.

We encourage the community to try the new hourly model. A working model is available on PyPI as the newest OpenDSM release. Tools for obtaining solar irradiance data will unfortunately not be provided given the lack of standard public sources for near-real-time solar data that would be comparable to the NOAA weather station data handled by EEweather. For analysts hoping to use the solar model, we suggest using either historical data from NREL's National Solar Radiation Database or a paid commercial service.

We will convene a final OpenDSM hourly model Working Group meeting on July 15, 2025 to allow a venue for final comments. In the interim, if you do have any questions, concerns, or comments, please feel free to reach out to us.

8 Appendix

8.1 Guidelines for Data Quality Issues

In many cases, data quality issues can be resolved by going back to the source to resolve issues in export or transfer. This guidance is a second line of defense for handling or correcting for common data issues, and are provided in the hope of mitigating the myriad issues and discrepancies which arise using different methods for data cleaning. These recommendations largely come from the CalTRACK methodology.

8.1.1 Impossible dates

1. If conducting billing analysis, and if day of month is impossible (e.g., 32nd of Jan), use first of month.
2. If month (e.g., 13) or year (e.g. 2051) is impossible, flag the date and remove it from the dataset. Check for mis-coding, such as 2015 -> 2051.

8.1.2 Timezone/Daylight Savings Time

1. Ensure that meter and temperature data is using matching and correct timezone and daylight-savings handling across all data sources.

8.1.3 Weather Resolution

1. NOAA weather is sampled roughly hourly with minute-level timestamps. This should be converted to hourly by first computing a minute-resolution time series using near interpolation of data points with a limit of 60 minutes, then downsampling to hourly temperature by taking mean of linearly-interpolated minute-level readings.

8.1.4 Duplicated meter or temperature records

1. Combine available versions into a single time series by dropping duplicate records, using the most complete version possible. If a record for a particular timestamp conflicts with another version, flag the project for possible existence of multiple meters or submeters. If this is confirmed, the usage from multiple meters may be aggregated.

8.1.5 Unreported Net Metering

1. Negative meter data values should be flagged for review as they indicate the possible unreported presence of net metering.

8.1.6 Extreme Values

1. Usage values that are more than three interquartile ranges less than the first quartile or greater than the third quartile should be flagged as outliers and manually reviewed.

8.1.7 Frequency

1. Roll up data if not given with expected frequency.

8.2 Imputation

8253 solar and non-solar PV meters were used from residential and commercial sectors. We removed data from temperature and observed time series by randomizing the day and start hour followed by the pct missing. This process was performed 5 times per meter. PNRMSE was calculated for bins of percentage missing and then normalized for each pct_missing (or missing fraction) bin. The percent missing was tested all the way up to 50% for robustness.

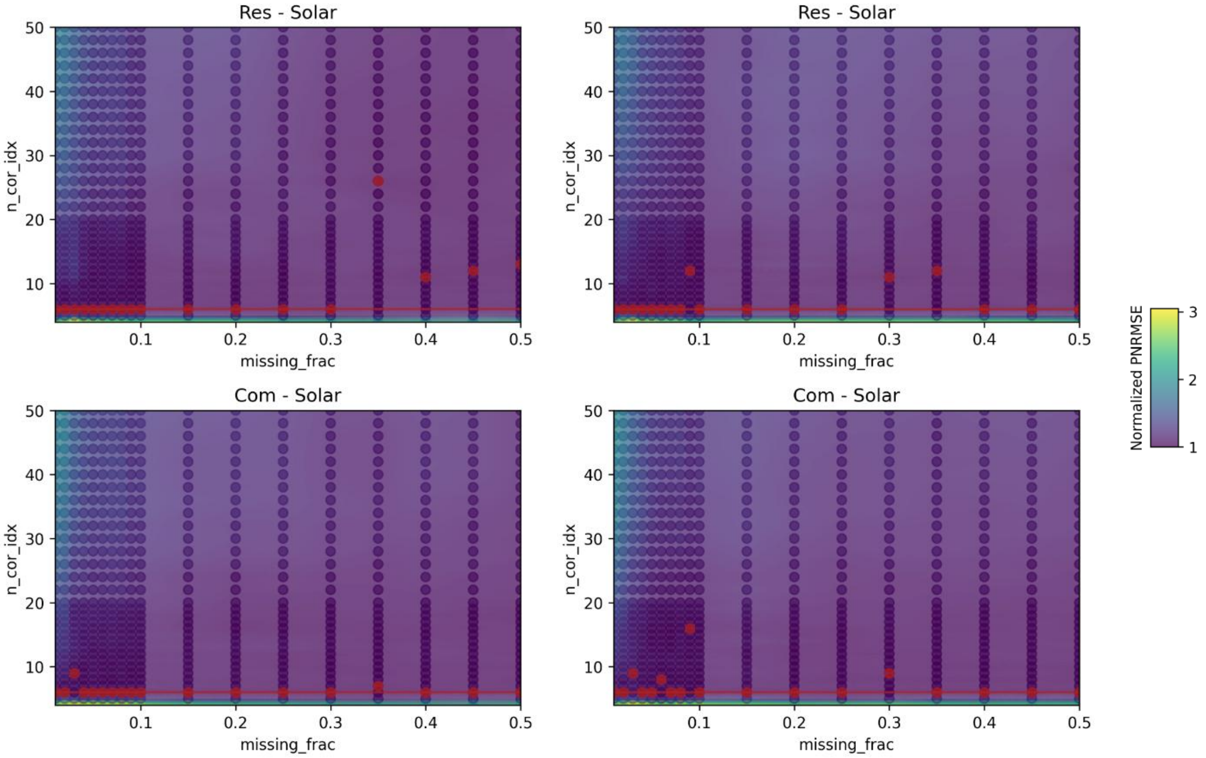

Figure 8.1. Surface plot where the color indicates the normalized PNRMSE for each column of missing temperature data percentage. The number of correlated indices is the y-axis. The red circles indicate the minimum normalized PNRMSE for each missing fraction percentage and the solid line is the best fit. The solid line is a constant of 6.

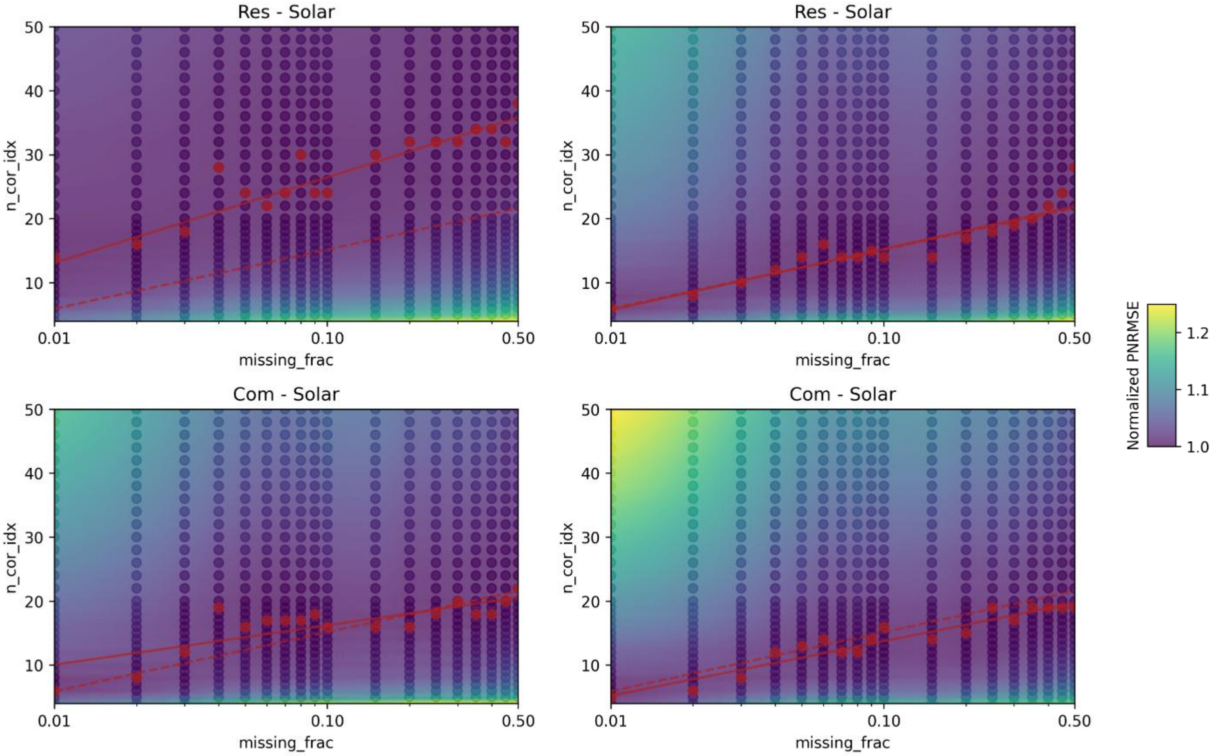

Figure 8.2. Surface plot where the color indicates the normalized PNRMSE for each column of missing observed data percentage. The number of correlated indices is the y-axis. The red circles indicate the minimum normalized PNRMSE for each missing fraction percentage and the solid line is the best fit. The solid line is the function $4.012 \cdot \ln(missing\_frac) + 24.38$.

8.3 Adaptive, Robust Loss Function Math

To do this, we use a generalized loss function, Eqn. 8.1, developed by Barron[7].

$$\rho(x,\mu,\alpha,c) = \left(\frac{|\alpha-2|}{\alpha}\right)\left(\left(\frac{\left(\frac{x-\mu}{c}\right)^2}{|\alpha-2|}+1\right)^{\alpha/2}-1\right) \tag{8.1}$$

where x is a data value, α a shape parameter, μ centers x, and c scales x. In this work, μ is the median of usage for a given hour with outliers removed using the 1.5 interquartile range rule, and c is the upper IQR outlier threshold on the absolute value of $|x-\mu|$ normalized to a standard deviation if the data are normally distributed. Another way to think about c is that this gives a range at which the down-weighting will start to take effect.

If α were to be optimized in Eqn. 8.1, it would always select $\alpha = -\infty$ because this would cause the function to be at a minimum, this is visualized in the left panel of Figure 3.8. To overcome this, Barron constructed a probability distribution, Eqn. 8.2 and 8.3, to penalize α.

[7] J.T. Barron, A general and adaptive, robust loss function; Computer Vision Foundation, 2019.

$$Pr(x \mid \mu, \alpha, c) = \frac{1}{cZ(\alpha)} \exp\big(-\rho(x, \mu, \alpha, c)\big) \tag{8.2}$$

$$Z(\alpha) = \int_a^b \exp\big(-\rho(x, 0, \alpha, 1)\big)\, dx \tag{8.3}$$

It should be noted that Eqn. 8.3 only converges between its proper bounds of $-\infty$ and ∞ when $\alpha \geq 0$; however, Chebrolu et al.[8] suggest using a truncated integration between $-10c$ and $10c$ to deal with most outlier distributions. In our library, $\ln\big(Z(\alpha)\big)$ has been numerically calculated and then fit using a series of B-splines which result in a maximum error of $< 5 \times 10^{-7}$. Finally, the original loss function Eqn. 8.1, can be penalized using Eqn. 8.3 as shown in Eqn. 8.4.

$$\rho_{penalized}(x, \mu, \alpha, c) = \rho(x, \mu, \alpha, c) + \ln\big(c\, Z(\alpha)\big) \tag{8.4}$$

[8] N. Chebrolu et al., Adaptive Robust Kernels for Non-Linear Least Squares Problems; IEEE Robotics and Automation Letters, 2021.