Predicting Energy Use of Institutional Buildings using nmecr

Mrinalini Sharma* David Jump† Devan Johnson‡

August 7, 2020

Normalized Metered Energy Consumption (NMEC)

Meter-based energy efficiency programs are proliferating across the globe. This proliferation is influenced by the widespread availability of high frequency energy use measurements from new metering technology as well as advancements in statistical regression and other empirical energy data modeling methodologies. Program administrators may report savings as the overall reduction in normalized metered energy consumption (NMEC). This method to determine savings is based on analysis of meter data collected prior to and after energy efficiency and conservation measures have been installed. Referred to as advanced measurement and verification (M&V) by the industry, this time-granular data and updated modeling methods provide several advantages over other methods used to quantify the benefits of energy efficiency:

- It reliably determines the actual savings achieved at the meter
- It provides fast feedback on the facility's energy performance and savings progress
- It enables identification and troubleshooting of issues that prevent savings realization

The `nmecr` package streamlines the application of the NMEC approach by enabling the:

- Management of high frequency, high volume data.
- Execution of advanced, state-of-the-art time-series data modeling algorithms.
- Comprehensive assessment of the validity of energy data models in specific applications.
- Quantification of uncertainties and risks associated with the energy savings projections in accordance with ASHRAE Guideline 14.

For an overview of nmcer, please refer nmecr_overview.pdf

Facilities outside the commercial sector have more drivers of energy use than just time and temperature. Institutional facilities, such as schools and government offices, have distinct operating profiles in different parts of the calendar year. These distinct operating profiles can be acconted for by adding variables to the dataframes that are served up to the modeling algorithms. These additional variables are indicator variables, indicating the beginning and end of an operating profile by switching between 1s and 0s.

*msharma@kw-engineering.com
†djump@kw-engineering.com
‡johnson@kw-engineering.com

Data

`nmecr` includes three datasets for this example: `school_eload`, `school_temp`, and `school_op_mode`. The data spans 1/1/2018 through 12/31/2018.

```
# Load data into an R session:

data(school_eload)
data(school_temp)
data(school_op_mode)
```

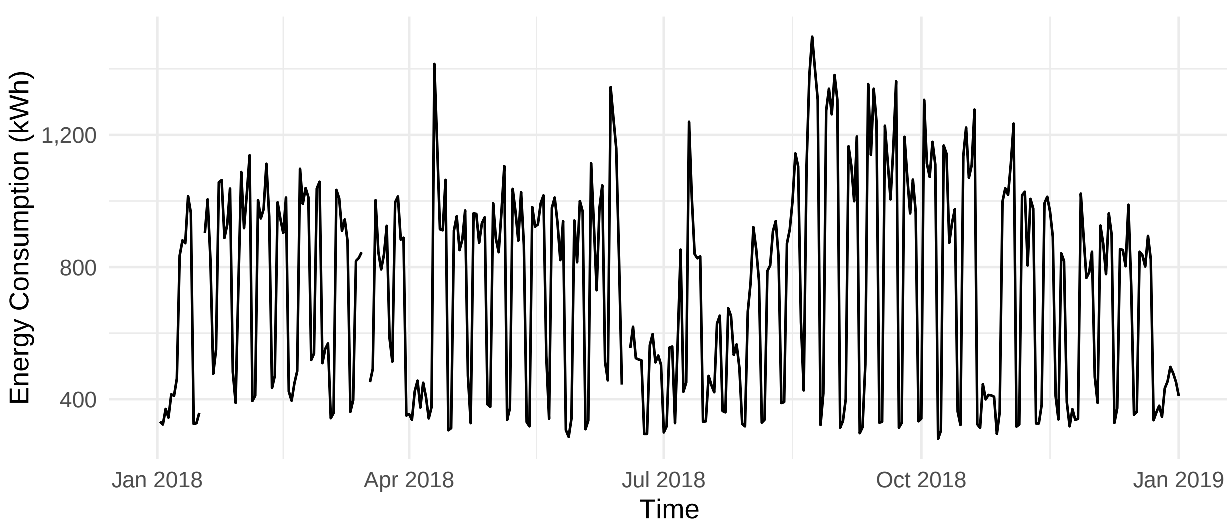
Baseline and Performance Period Dataframes for Modeling

`create_dataframe()` combines the `eload` and `temp` dataframes into one, filters by the specified start and end dates, and aggregates to an hourly, daily, or a monthly data interval. It lines up all data such that each timestamp represents attributes up until that point, e.g. an eload value corresponding to 01/02/2018 represents the energy consumption up until then - the energy consumption of 01/01/2018. The `timestamps` parameter is optional, with permissible values: 'start' and 'end', indicating whether the timestamps mark the start or the end of the period. If operating mode data[1] is supplied, this information is added to the dataframe created by `create_dataframe()`.

```
# Baseline Dataframe

baseline_df <- create_dataframe(eload_data = school_eload,
                                temp_data = school_temp,
                                convert_to_data_interval = "Daily",
                                timestamps = "start")

baseline_df_with_op_mode <- create_dataframe(eload_data = school_eload,
                                             temp_data = school_temp,
                                             operating_mode_data = school_op_mode,
                                             convert_to_data_interval = "Daily",
                                             timestamps = "start")
```



[1] Operating mode's data interval must be same as that of the intended model. For example, for TOWT models created using daily data, upload an operating mode dataset with daily data intervals

Figure 1: Baseline Energy Use over Time

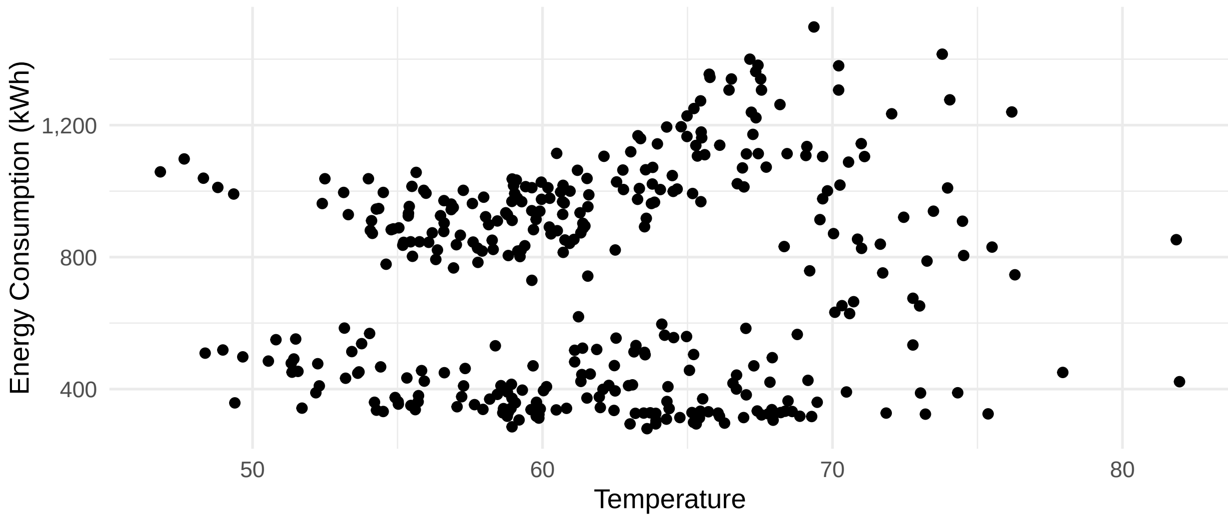


Figure 2: Baseline Energy Use over Temperature

As there isn't a high dependence on temperature, we expect that a model with Simple Linear Regression will not be able to capture all the variation in the energy use profile:

Model Creation

Simple Linear Regression:

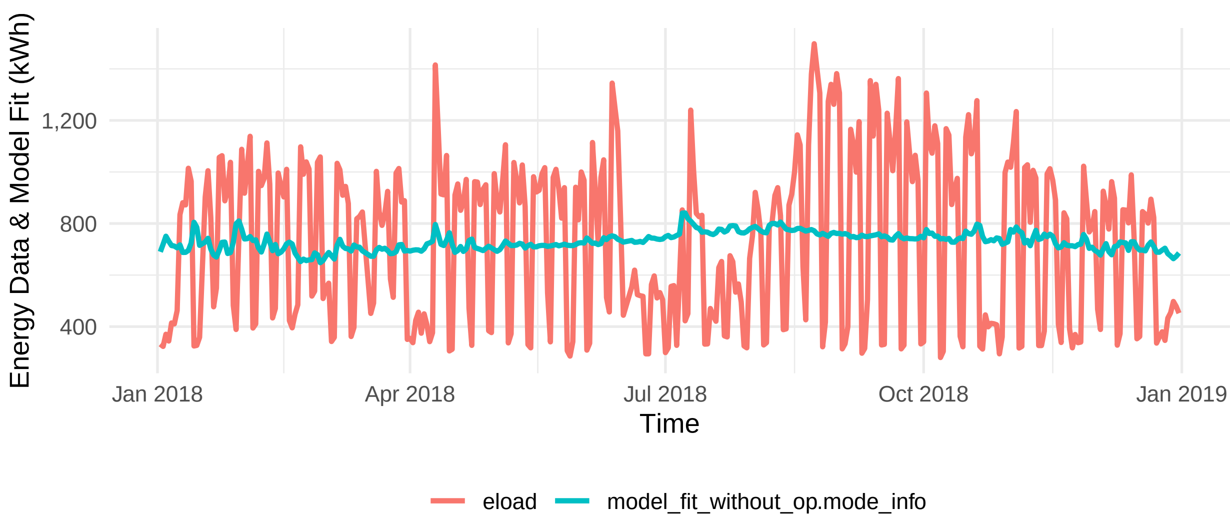


Figure 3: Baseline Energy Use modeled with Simple Linear Regression over Time

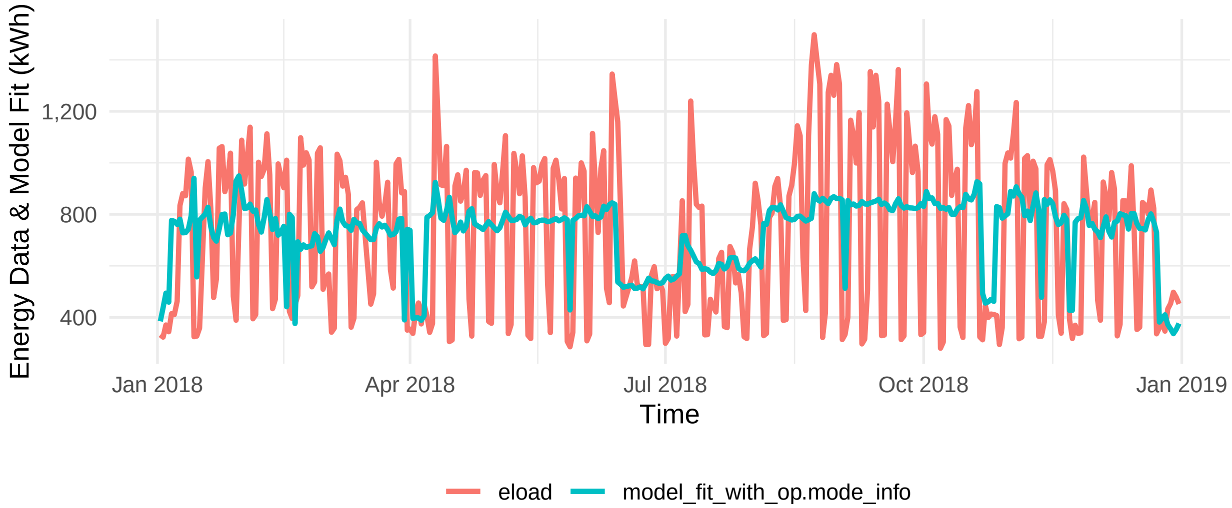


Figure 4: Baseline Energy Use modeled with Simple Linear Regression over Time with Operating Mode Information

The model with the operating model information performs better than the one without. However, it fails to capture the time-based component of the energy use variation.

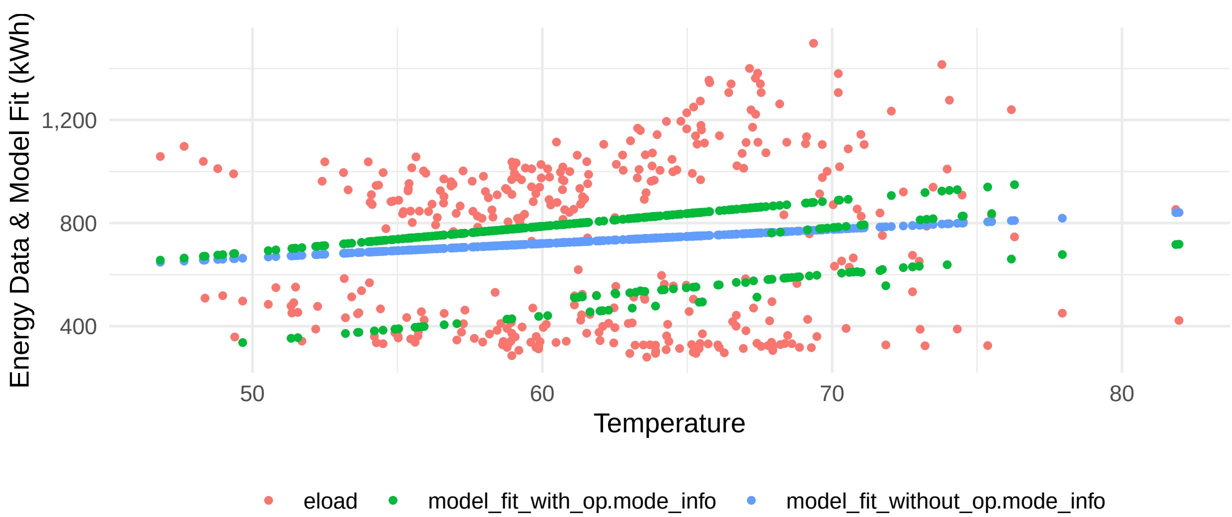


Figure 5: Baseline Energy Use modeled with Simple Linear Regression over Temperature

Figure 5 shows that the model without the operating mode information tends to average between the high and the low usage. The model with the operating mode information captures this variation better.

Time of Week and Temperature:

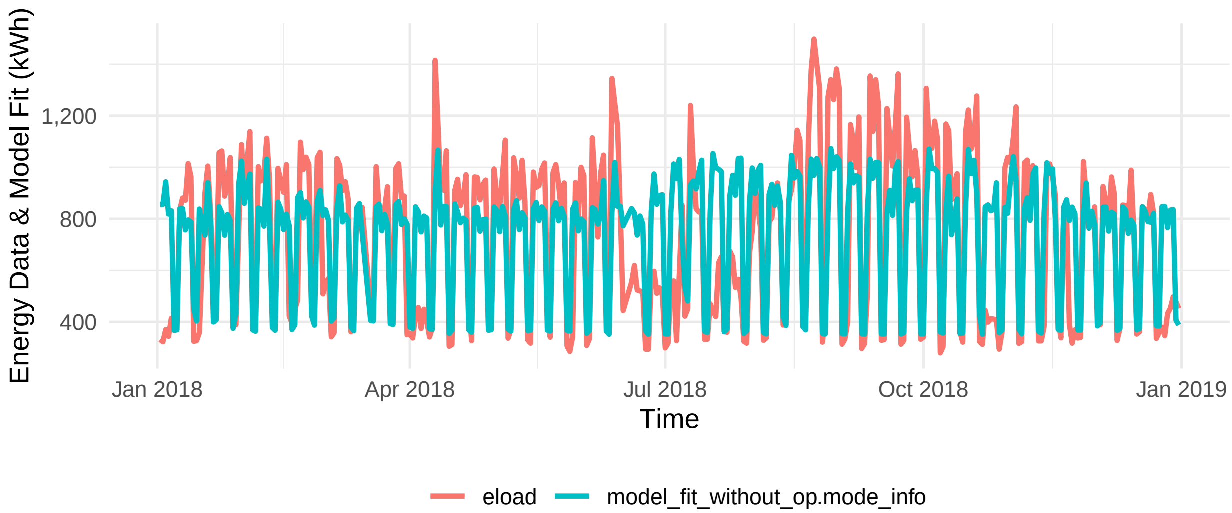


Figure 6: Baseline Energy Use modeled with Time-of-week and Temperature over Time

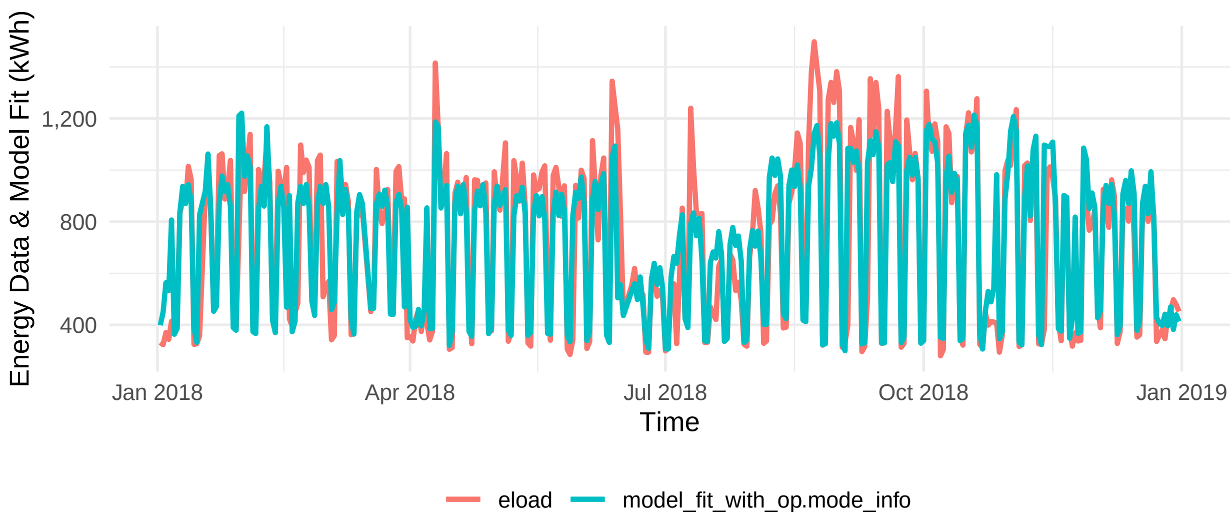


Figure 7: Baseline Energy Use modeled with Time-of-week and Temperature over Time with Operating Mode Information

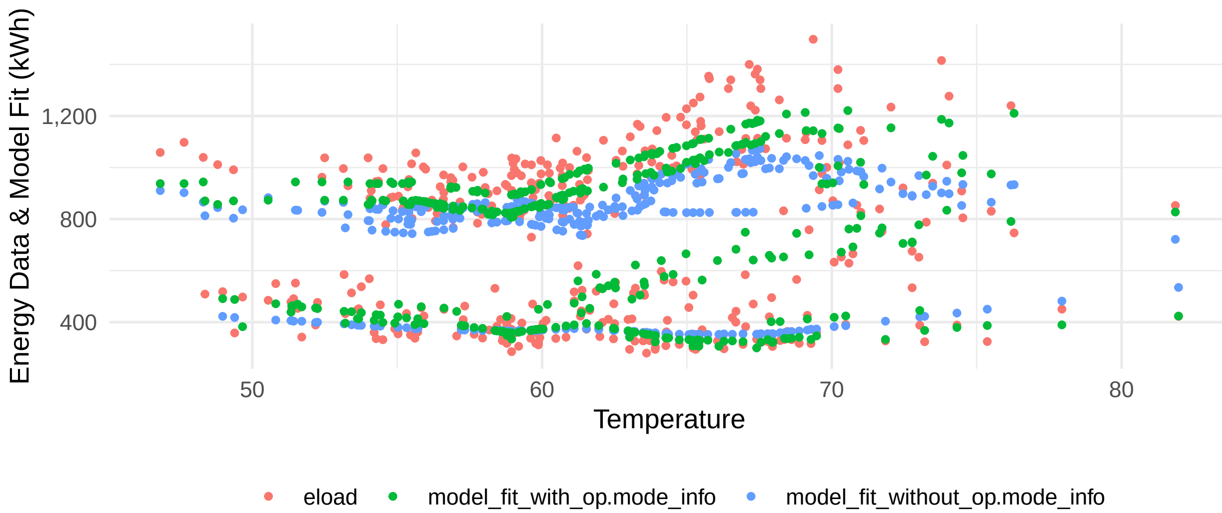


Figure 8: Baseline Energy Use modeled with Time-of-week and Temperature over Temperature

The plots show that the Time-of-Week and Temperature algorithm models the energy use profile better than SLR.

This insight can be confirmed through model statistics by using the `calculate_summary_statistics()` function. The following table summarizes the results from this function for each of the models assessed above:

```
SLR_stats <- calculate_summary_statistics(SLR_model)

SLR_with_op_mode_stats <- calculate_summary_statistics(SLR_model_with_op_mode)

TOWT_stats <- calculate_summary_statistics(TOWT_model)

TOWT_with_op_mode_stats <- calculate_summary_statistics(TOWT_model_with_op_mode)


all_stats <- bind_rows(SLR_stats, SLR_with_op_mode_stats, TOWT_stats, TOWT_with_op_mode_stats)

model_names <- c("SLR", "SLR with op_mode", "TOWT", "TOWT with op_mode")

all_stats <- bind_cols("Model Name" = model_names, all_stats)

all_stats
#>          Model Name R_squared CVRMSE %          NDBE %          NMBE % #Parameters
#> 1               SLR      0.01    43.60 -8.420678e-16 -8.467721e-16           2
#> 2  SLR with op_mode      0.17    40.22 -8.650051e-16 -8.797077e-16           6
#> 3              TOWT      0.54    30.43  1.879754e-13  1.984491e-13          19
#> 4 TOWT with op_mode      0.77    21.98 -7.938584e-14 -8.584192e-14          27
```

- CVRMSE: Coefficient of Variation of Root Mean Squared Error
- NDBE: Net Determination Bias Error
- MBE: Mean Bias Error
- R_squared: Coefficient of Determination

Assessing project fit for NMEC

Using the modeling statistics and the savings uncertainity for 10%, we can determine the validity of the NMEC approach for a certain project. For the building decribed here, following are the key values to be used for this assessment

1. CVRMSE: 22% (should be < 25%)
2. NDBE: 0.0000 (should be < 0.005%)

The California Public Utilities Commission requires savings to be detectable above model variations. `nmecr` interprets this using ASHRAE Guideline 14 - 2014's formulation for savings uncertainty, which relates the savings uncertainty to the model goodness of fit metric CV(RMSE), the confidence level, the amount of savings, the amount of data used to develop the model, and the amount of data required to report savings. It includes a correction when autocorrelation is present (which occurs mainly in models developed from daily and hourly data). LBNL has shown this uncertainty formulation with correction for autocorrelation underestimates the savings uncertainty. More work on this issue is needed. Until a better formulation is available, `nmecr` uses ASHRAE's method only as an estimation.

```
TOWT_savings <- calculate_savings_and_uncertainty(prediction_df = NULL,
                                                  savings_fraction = 0.1,
                                                  modeled_object = TOWT_model_with_op_mode,
                                                  model_summary_statistics = TOWT_with_op_mode_stats,
                                                  confidence_level = 68)

TOWT_savings$savings_summary_df
#>   savings_fraction savings_uncertainty savings_frac_for_50pct_uncertainty
#> 1              0.1           0.2035816                         0.04071631
#>   confidence_level
#> 1               68
```

3. Savings Uncertainty for 10% savings (at 68% confidence level): 20.4% (shoule be < 50%).

The savings percentage required to meet the threshold of 50% uncertainty, at the 68% confidence level, is 3.9% (as shown by the output above, see: savings_pct_for_50pct_uncertainty).

In addition to evaluating the model metrics, it is essential to ensure that the modeled profile follows the actual energy use closely (see Figure 8 above).