Review of Open Energy Efficiency Meter Methods

Task 2 Final Report

California Energy Commission

Work Authorization # 19

March 30, 2017

Reproduction or distribution of the whole, or any part of the contents of this document without written permission of kW Engineering is prohibited. Neither kW Engineering nor any of its employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any data, information, method, product or process disclosed in this document, or represents that its use will not infringe upon any privately-owned rights, including but not limited to, patents, trademarks or copyrights.

Table of Contents

I. Summary

The California Energy Commission (CEC) directed kW Engineering to review Open Energy Efficiency Meter’s savings calculation methods and to apply them to datasets from projects in the Clean Energy Jobs Act Program (CEJAP). One of the goals of this effort was to make recommendations for OEEM improvements, in particular to facilitate the use of the OEEM software in whole-building Measurement and Verification (M&V) efforts.

We used the International Performance Measurement and Verification Protocol (IPMVP, Efficiency Valuation Organization, 2012) as a guide, which established an industry-accepted framework for M&V.

The OEEM documentation materials were reviewed. We also obtained the OEEM source code in the Python programming language from the GitHub site. The OEEM programming code was reviewed by an experienced Python programmer.

Finally, we ran four test data sets through the whole-building analysis methods provided by OEEM, as well as through methods that we developed in R as part of another CEC subtask. We compared the results of these runs, and found that the monthly methods generally produced more outliers, but the daily methods were in agreement at all four sites within a few percentage points.

In general, with a few specific improvements, the OEEM code can produce IPMVP-defined estimates of savings. A number of improvements were recommended, which we will forward to the OEEM developers through the Github platform so that they can be addressed and checked in future versions.

II. Project Team

Name	Organization	Contact Information
Laith Younis Project Manager Abhilasha Wadhwa, Supervisor	California Energy Commission	1516 9th Street Sacramento, CA, 95814 laith.younis@energy.ca.gov abhilasha.wadhwa@energy.ca.gov
Richard Seppa, Senior Engineer, Sustainability Services	NORESCO	2540 Frontier Ave, #100 Boulder, CO, 80301 (303) 459-7427 rseppa@noresco.com
David Jump, Ph.D., P.E., Project Manager Devan Johnson, P.E. Principal Raphael Vitti, P.E., Senior Engineer II	kW Engineering	287 17th Street, Suite 300 Oakland, CA 94612 510-834-6420 (office) djump@kw-engineering.com johnson@kw-engineering.com rvitti@kw-engineering.com

1. Introduction

The scope of work for Subtask 2 was to review Open Energy Efficiency Meter' s (OEEM's) savings calculation methods and to apply them to datasets from projects in the Clean Energy Jobs Act Program (CEJAP). The CEJAP data sets were developed under Subtask 1 of the work authorization, and processed through whole building methods developed and implemented in R code. This task also required the results obtained from the Subtask 1 code and OEEM Python code to be compared, and make recommendations for OEEM improvements as well as define methods used in Subtask 1 that may be adopted by OEEM. This report describes the evaluation of OEEM according to these objectives. An OEEM evaluation plan was created at the outset of this project to guide the effort. It is included as Appendix A to this report.

On its website,[1] OEEM describes that OEEM was designed to provide a standardized, transparent, and repeatable methodology for estimating whole building energy savings through public domain programming code and an accessible platform. Originally targeted to residential energy efficiency projects using monthly billing data, its modeling capabilities have been expanded as data from advanced metering infrastructure (AMI data) have become available.

The goal of OEEM is to create a methodology and operating platform such that by "using the meter, private companies, utilities, and regulators will all calculate the same level of savings for a given set of building efficiency projects." Its savings estimation methods are programmed in Python and made available through GitHub's public domain software development platform. OEEM's methods are partially documented on its website. OEEM has other functionality that enables it to work with other industry tools, such as GreenButton Connect and the Department of Energy's Building Energy Data Exchange Specification (BEDES). Evaluating this functionality is not the subject of this report.

In Subtask 1, we developed whole building M&V analysis code for estimating savings using two modeling algorithms: change-point (CP) models based on ASHRAE Research Project 1050 Inverse Modeling Toolkit (Kissock, et.al., 2004), and the temperature-and-time-of-week (TTOW) model developed at Lawrence Berkeley National Laboratory (Mathieu, et.al., 2011). The CP models may be applied to monthly billing data as well as energy data in daily time intervals. The TTOW model may be applied to energy data in daily and hourly time intervals. The CP and TTOW models were developed in R programming code. We obtained energy use data for a number of schools that had completed CEJAP projects. The data was for electric and natural gas energy use. The electric data generally came in AMI format and the natural gas usage data was in monthly billing format.

As described in the OEEM evaluation plan, we reviewed the OEEM documentation materials and obtained the OEEM source code from the GitHub site. From Subtask 1, we could compile only four data sets out of the original twenty that were intended. We ran these data sets through the whole building analysis methods in R code, and ran them again using OEEM's Python code for monthly and daily analysis. This report documents our findings of this evaluation.

[1] openeemeter.org

2. Description of OEEM Methods

Energy Modeling Methods

The International Performance Measurement and Verification Protocol (IPMVP) established an industry-accepted framework for M&V, including four options for verifying savings from energy efficiency and conservation projects. IPMVP Option C Whole Building provides guidance for M&V analysis using empirical models of whole building energy use. Option C M&V analysis requires two main elements: development of empirical energy models from measured data over a baseline period, and use of the energy models to quantify savings.

At a fundamental level, IPMVP requires that both baseline and post-installation energy use be compared under the same conditions before their difference is taken to determine savings. Baseline or post-installation period energy use is 'adjusted' for routine factors such as weather and production rates that are normally expected to influence energy use, as well as for non-routine factors that are unrelated and often unexpected that influence energy use in these periods. Such non-routine 'events' can include addition or removal of electric loads, such as adding or removing office equipment, equipment shut-downs for maintenance, or occupants moving in or out of a building. In Option C, energy models are developed from data from routine factors; the models describe how they influence energy use. IPMVP recommends that the adjustments to energy use due to non-routine events be quantified separately from the Option C analysis, but provides no other guidance on how this may be done.

Under Option C, a model of whole-building energy use is developed, using baseline period energy use data, and independent variables such as weather conditions and occupancy. The baseline model is then applied to post-installation period conditions, to determine the "adjusted baseline energy use." Savings during the post-installation period, referred to as "avoided energy use" are determined by subtracting the measured post-installation period energy use from the adjusted baseline use (Efficiency Valuation Organization, 2012).

Related to avoided energy use is the concept of *normalized savings*. Normalized savings are calculated under typical operation conditions. For weather-dependent energy models, a typical year of weather data is often used to represent typical operation conditions. This is often accomplished by using typical meteorological year (TMY) weather data. To calculate normalized savings, models are developed from both baseline and post-installation energy use data, and then both models are applied to TMY weather conditions. The difference between the predicted energy uses under these normalized conditions from the two models is the normalized savings (Efficiency Valuation Organization, 2012).

IPMVP provides a framework of definitions and methods for properly assessing energy savings. It guides users to develop and carrying out good M&V Plans while adhering to its principles of accuracy, completeness, conservativeness, consistency, relevance, and transparency. No software tool alone can fully adhere to IPMVP's principles. However how a tool treats the principles of accuracy, completeness, and transparency may be assessed. The evaluation of OEEM software reviews how it conducts Option C M&V analysis and addresses these basic principles.

OEEM Methods

Open EE Meter's website provides a narrative description of the savings estimation method based on monthly billing data. This is currently run in the CalTrack Monthly Python code, which was downloaded from the GitHub public domain site. OEEM developers informed us that the savings estimation method based on daily data was in a file called: 'Seasonal ElasticNet CV Model' and that this code is in its beta phase of development. As it is in beta, the developers have not yet published the narrative documentation on its website. However, they've described the Seasonal ElasticNet CV model as being very similar to the monthly model, with linear regressions and additional features that account for days of the week and holidays. We note that we are testing a beta version of the Seasonal ElasticNet CV model, which we obtained from the Github public domain site https://github.com/openeemeter/eemeter.

The CalTrack Monthly code estimates site-level monthly savings and cumulative savings for the reporting period. The reporting period is any designated period after the measures have been installed. It can be the post-installation year, or any time period following the first post-installation year. Monthly savings estimates can be determined for natural gas or electric usage data, whether it is from monthly billing data or in a more time-granular form built up from AMI data, such as in daily time intervals. Daily temperature data is automatically collected from the weather station nearest the zip code for each site. Both energy and temperature data undergo data sufficiency checks prior to analysis.

The analysis process proceeds in two steps. In step one, the monthly billing data is converted into usage per day (UPD_m), the temperature data is used to calculate heating degree-days H_m and cooling degree-days C_m per day in the monthly billing periods. Details about the development of UPD_m, H_m, and C_m are provided on the website, and are in accordance with standard practice. The software then develops multiple baseline and post-installation period energy use models and qualifies the models if they meet the minimum significance criteria *p-value* < 0.1 and positive. The adjusted R^2 is calculated for each qualifying model and the model with the highest adjusted R^2 is ultimately selected. Four model forms are developed for each site:

1) $UPD_m = \mu + \beta_H H_m + \beta_C C_m + \epsilon_m$

2) $UPD_m = \mu + \beta_H H_m + \epsilon_m$

3) $UPD_m = \mu + \beta_C C_m + \epsilon_m$

4) $UPD_m = \mu + \epsilon_m$

With the constants β_H, β_C, and $\mu > 0$. All four model types are developed and tested with electric use data, while only the 2nd and 4th model types are used with gas usage data (cooling degree-days are not expected to influence gas use).

Once satisfactory baseline and reporting period models are developed and selected, the code estimates savings under three different conditions, two of which can be used to produce savings expressed in IPMVP terms. These are:

1) *annualized weather normal*, which is calculated by determining the heating and cooling degree-days based on the 'weather normals' from the nearest weather station to that site's zip code, then using them in the best-fit regression models to restate baseline and reporting period energy use under the same weather normal conditions. This is similar to reporting '*normalized savings*' in IPMVP parlance.
2) *gross actual savings,* which is calculated by determining the heating and cooling degree-days based on the current year weather as obtained from the nearest weather station to that site's zip code, then applying them to the selected baseline model only, and subtracting measured current year reporting period energy use. This is similar to reporting '*avoided energy use*' in IPMVP parlance.

The OEEM description of its monthly M&V analysis is based on common practice using heating and cooling degree-days as explanatory variables. Given the correct weather inputs, the algorithms calculate avoided energy use and normalized savings, as described in IPMVP. What is missing is a description of the savings uncertainty estimation process for savings stated under each of these processes.

There is no direct documentation yet of the Seasonal ElasticNet CV model, which uses daily data in its M&V analysis.

3. Review of OEEM Programming Code

General Notes

The OEEM code is of good quality, it is well laid-out and properly follows general as well as python-specific coding principles, although large portions of the code currently lack proper documentation and references. It takes time to install and run as a stand-alone. Therefore, it seems more suitable for large batch analysis processes or for integration with other code.

Running the M&V portion of the code in stand-alone fashion required much time to set up and implement, as the documentation was not straightforward to understand. Feedback on this experience will be provided to the code developers through the Github platform.

CalTrack Monthly Code

The CalTrack code converts usage data to average usage per day for each calendar month of the analysis period it is applied to. Months with less than 15 days of data are not used in the analysis. The code converts daily average dry bulb temperature data to average heating degree days (HDD) and cooling degree days (CDD) for each calendar month of the analysis period, storing average HDD and CDD for a range of reference temperatures from 50°F to 85°F in 5°F increments. This results in eight reference temperatures and eight sets each of HDD and CDD independent variables.

Once preprocessing is complete, the code then tries to fit several model types as described in section 2: OEEM Methods. For electricity data, there are four model types that use a combination of HDD and CDD dependent variables. For gas data, there are two model types that only use HDD dependent variables.

As a first step, each model type that uses HDD or CDD as independent variables gets fitted several times, one for each variation and combination of reference temperatures used to calculate HDD and CDD. This results in 8 model fits for the $UPD_m = \mu + \beta_H H_m + \epsilon_m$ and $UPD_m = \mu + \beta_C C_m + \epsilon_m$ models, and in 64 model fits for the $UPD_m = \mu + \beta_H H_m + \beta_C C_m + \epsilon_m$ model. The reference temperatures that result in the highest R^2 are chosen for each model. The simple average model gets only fitted once since there are no variations for this model. All the models used in CalTrack get fitted using the ordinary least squares method.

As a second step, the code choses between the resulting two models (for gas) or four models (for electricity), by again maximizing the R^2.

Reported model fit statistics are R^2, RMSE, CV(RMSE), and number of months with more than 15 days' worth of data in the analysis period.

Predictions can then be made using the fitted model parameters. Predictions are made for the average daily usage each month of a given analysis period, then these values are multiplied by the number of days in a month to obtain monthly usage, and summed to obtain total usage.

Additionally, when making predictions using a selected model, the variance of each predicted value is calculated.[2] For example, for a two-parameter model, the following formula is used:

$$var(f) = \sigma^2 + var(\hat{\mu}) + X_p^2 var(\hat{\beta}) + 2X_p cov(\hat{\mu}, \hat{\beta})$$

where f is the predicted value,
σ^2 is estimated as the mean squared error of the model residuals,
X_p is the value of the independent parameter the prediction is based on,
$\hat{\mu}$ is the estimated intercept,
$\hat{\beta}$ is the estimated slope.

The code then multiplies the variance of each predicted value (average daily usage) for each month by the number of days in the month to obtain monthly usage variance, then these values are summed to obtain the total variance for the prediction.

When calculating avoided energy usage or normalized energy savings, the total variance of the result is calculated as the square root of the sum of the squared total variances of the baseline and reporting series. In the case of the avoided energy usage, the reporting series is metered and has a variance of zero, therefore the total variance of the avoided energy usage series is the same as the total variance of the baseline prediction.

$$var(a - b) = \sqrt{var(a)^2 + var(b)^2}$$

Based on our current understanding of the code and observed outputs, the current implementation does not report other uncertainty metrics, and therefore we could not easily and directly determine if estimated savings at our four test sites comply with ASHRAE Guideline 14 whole-building performance path requirements that maximum uncertainty be 50% of annual reported savings at 68% confidence.

Seasonal ElasticNet CV (Daily)

This model is a HDD, CDD and day-of-the-week model, designed to analyze daily data. Despite its name, the model does not seem to account for seasonal variations in usage. Based on our review of the code, the following model of daily usage data is created, which has an intercept, a HDD-based heating slope and a CDD-based cooling slope for each day of the week, as well as one global HDD×CDD interactive term, for a total of twenty-two coefficients:

$$\begin{aligned} U_d = {} & \mu_{Mon} + \mu_{Tue} + \mu_{Wed} + \mu_{Thu} + \mu_{Fri} + \mu_{Sat} + \mu_{Sun} + \\ & \beta_{H_{Mo}} H_{d_{Mo}} + \beta_{H_{Tu}} H_{d_{Tu}} + \beta_{H_{We}} H_{d_{We}} + \beta_{H_{Thu}} H_{d_{Thu}} + \beta_{H_{Fri}} H_{d_{Fri}} + \beta_{H_{Sa}} H_{d_{Sa}} + \beta_{H_{Su}} H_{d_{Su}} + \\ & \beta_{C_{Mo}} C_{d_{Mo}} + \beta_{C_{Tu}} C_{d_{Tu}} + \beta_{C_{We}} C_{d_{We}} + \beta_{C_{Thu}} C_{d_{Thu}} + \beta_{C_{Fri}} C_{d_{Fri}} + \beta_{C_{Sa}} C_{d_{Sa}} + \beta_{C_{Su}} C_{d_{Su}} + \\ & \gamma_{HC} C_d H_d + \\ & \epsilon_d \end{aligned}$$

[2] A reference for calculating this variance can be found at: http://www.uio.no/studier/emner/sv/oekonomi/ECON4150/v04/seminar/Var_f.pdf

The code uses an R-style formula for daily data that has twenty-five terms:

```
energy ~ 1 + CDD + HDD + CDD:HDD + (CDD) * C(tempF.index.weekday) + (HDD) * C(tempF.index.weekday) + C(tempF.index.weekday)
```

The first three terms (global intercept, cooling slope and heating slope) are redundant and ignored by the python patsy library when creating the design matrix and performing the regression, therefore the output has twenty-two coefficients as expected.

Additionally, the model can account for holidays in certain conditions. Holidays are determined using a standard python library and we could not find a built-in functionality to customize holidays. When holidays are accounted for, one additional intercept term is used. The heating and cooling slopes do not appear to account for holidays, and are only based on the actual day of the week.

The current iteration of the model uses fixed CDD and HDD reference temperatures, provided by the user.

The regression to determine model coefficients is performed using an elastic net regression with cross-validation, which is a regularized, iterative regression method that combines the LASSO and RIDGE regression methods. Regularized methods are typically used to attenuate overfitting issues by penalizing complexity and extreme coefficients values.

For our four test sites, we tested replacing this regression method with the ordinary least squares method for fitting, and found that some of the coefficients occasionally had significantly different values. As expected due to the regularization, the R^2 and CV(RMSE) of the models were always slightly better for the ordinary least squares method. The variance of predictions was always better using the elastic net regression method. Further investigation should be performed to fully document the usefulness of the elastic net regression method in determining coefficients for this model form.

Table 3.1: Comparison of OLS and ElasticNet Fitting Methods using Selected Metrics

Site Data	ElasticNet R^2	OLS R^2	ElasticNet CV(RMSE)	OLS CV(RMSE)	ElasticNet Variance*	OLS Variance*
7998 Base	0.7718	0.7865	12.67%	12.25%	12,150,370	16,116,666
7998 Post	0.7090	0.7218	14.01%	13.70%	26,818,823	28,799,289
7931 Base	0.4602	0.4628	38.10%	38.00%	230,365,378	238,915,465
7931 Post	0.6200	0.6270	22.59%	22.38%	10,985,262	11,083,203
1322 Base	0.5368	0.5392	39.70%	39.60%	187,452,285	204,127,351
1322 Post	0.4399	0.4865	41.00%	39.26%	151,985,430	151,075,274
5778 Base	0.5421	0.5443	39.38%	39.28%	140,835,736	198,121,004
5778 Post	0.3090	0.4736	47.43%	41.40%	100,042,366	101,069,397

* Variance is the total variance of predictions, reported for base and post models applied to annualized weather data. Variance is based on prediction error calculated using a bootstrapped empirical method.

Error is calculated using a type of block bootstrap method with fixed block length. We did not find any references in the code to document the methodology. Default values are 200 minimum number of points in series, 99 points used to draw bootstrap blocks, for a total of 50 such overlapping blocks of length 50. It does not appear that random sampling of the blocks is used, instead all blocks are used. The blocks are overlapping and all the blocks are formed using the end of the data time series, by default starting at index -100. For each block, a model is fitted using all the preceding data in the series, then a prediction is made over the block and residual errors are calculated.

The resulting matrix of residuals (default dimensions 50 by 50) is used to calculate the coefficients of an equation that describes the standard deviation of a series of predictions based on the length n of the series: $std(n) = \beta.n^{\alpha}$. We could not find a reference for this methodology. The standard error function for a model is then used to calculate the variance of predictions made using that model.

This empirical error calculation methodology allows users to take advantage of the regularized regression method, which yields regression coefficients that can give more reliable predictions despite having typically worse fitting statistics as calculated over the entire training period (such as R^2 and CV(RMSE) as presented in Table 3.1).

We recommend that this methodology be fully documented and that the standard deviation be used to calculate uncertainty and relative uncertainty values at various confidence levels.

4. Analysis Results

Comparison of Runs for Four Sites with R Code

We used electricity data from four school sites that were previously analyzed using kW Engineering R code in a separate CEC Prop 39 task. The models include temperature-time-of-week (TTOW) hourly and daily models, as well as change point models for daily and monthly data. We loaded the same data within OEEM, ran the CalTrack and Seasonal ElasticNet models, then compared the results, in particular avoided energy use estimates, normalized savings estimates, uncertainty and fit. Uncertainty for the TTOW and CP models from the R code was determined using the savings uncertainty formulation in ASHRAE Guideline 14 (ASHRAE, 2014). For the OEEM models, we used the reported variance of the prediction errors to calculate the uncertainty of the savings estimates.

Only 1 of the 4 sites showed large energy savings of more than 10% for more than two models. The 3 other sites have low savings or no savings.

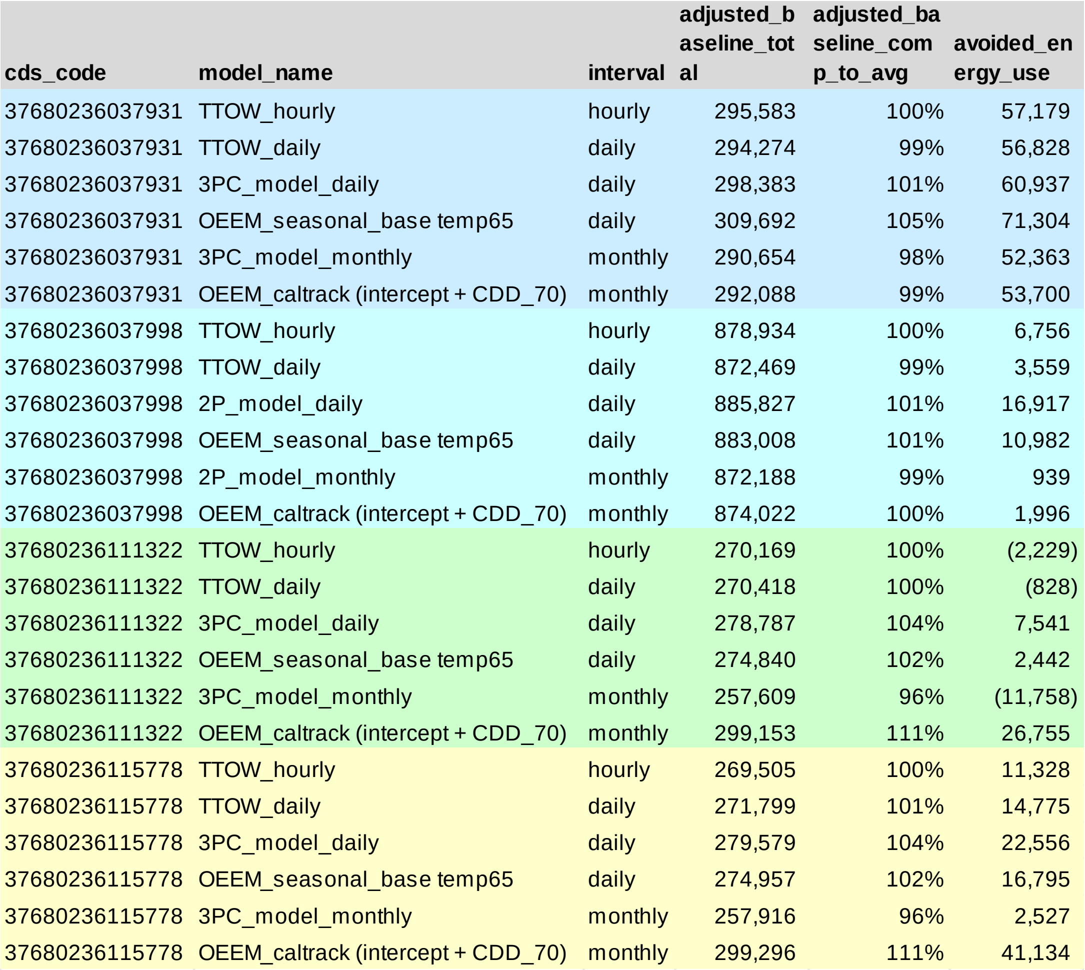
Overall we found significant differences in the calculated savings between the different models, up to 13% of the calculated baseline total. Daily models tended to have more consistent results than the monthly models: the TTOW, the CP and the ElasticNet models yielded results within 4% of the calculated baseline total for each site. The R code CP and CalTrack monthly models seemed to produce more anomalies, in particular in their avoided energy usage results. This is likely due to weaknesses of these two monthly models when applying them to less than one year of data.

Discussion of Results

The R change point monthly model appears to underestimate by 2% or more the adjusted baseline usage during the post-implementation period at 3 of the 4 test sites when comparing to the average result from the six models included in this test. We further looked into this slight discrepancy and believe it may be due to the fact that we are missing July and August post-implementation data at all of our test sites. The baseline model is developed using 12 months of data, including June and July data when the schools included in our data set are occasionally closed, thereby pulling the average electricity usage down. When applied to 10 months of data that exclude July, the models therefore underestimate the actual usage. This is in contrast with the R change point daily and R TTOW daily and hourly models, which all include treatment of holiday data based on the actual school schedule, and therefore can more reliably calculate total adjusted baseline usage on a post-implementation period of less than 12 months.

Comparatively, the OEEM Caltrack monthly model overestimates by 10% or more the adjusted baseline usage during the post-implementation period at 2 of the 4 sites. This model also does not include treatment of holiday data based on the school schedule, however this does not explain an overestimation of the adjusted baseline usage.

The OEEM Seasonal ElasticNet model is generally in line with other models but overestimates by 2% or more the adjusted baseline usage during the post-implementation period at 3 of the 4 sites.



cds_code	model_name	interval	adjusted_baseline_total	adjusted_baseline_comp_to_avg	avoided_energy_use
37680236037931	TTOW_hourly	hourly	295,583	100%	57,179
37680236037931	TTOW_daily	daily	294,274	99%	56,828
37680236037931	3PC_model_daily	daily	298,383	101%	60,937
37680236037931	OEEM_seasonal_base temp65	daily	309,692	105%	71,304
37680236037931	3PC_model_monthly	monthly	290,654	98%	52,363
37680236037931	OEEM_caltrack (intercept + CDD_70)	monthly	292,088	99%	53,700
37680236037998	TTOW_hourly	hourly	878,934	100%	6,756
37680236037998	TTOW_daily	daily	872,469	99%	3,559
37680236037998	2P_model_daily	daily	885,827	101%	16,917
37680236037998	OEEM_seasonal_base temp65	daily	883,008	101%	10,982
37680236037998	2P_model_monthly	monthly	872,188	99%	939
37680236037998	OEEM_caltrack (intercept + CDD_70)	monthly	874,022	100%	1,996
37680236111322	TTOW_hourly	hourly	270,169	100%	(2,229)
37680236111322	TTOW_daily	daily	270,418	100%	(828)
37680236111322	3PC_model_daily	daily	278,787	104%	7,541
37680236111322	OEEM_seasonal_base temp65	daily	274,840	102%	2,442
37680236111322	3PC_model_monthly	monthly	257,609	96%	(11,758)
37680236111322	OEEM_caltrack (intercept + CDD_70)	monthly	299,153	111%	26,755
37680236115778	TTOW_hourly	hourly	269,505	100%	11,328
37680236115778	TTOW_daily	daily	271,799	101%	14,775
37680236115778	3PC_model_daily	daily	279,579	104%	22,556
37680236115778	OEEM_seasonal_base temp65	daily	274,957	102%	16,795
37680236115778	3PC_model_monthly	monthly	257,916	96%	2,527
37680236115778	OEEM_caltrack (intercept + CDD_70)	monthly	299,296	111%	41,134

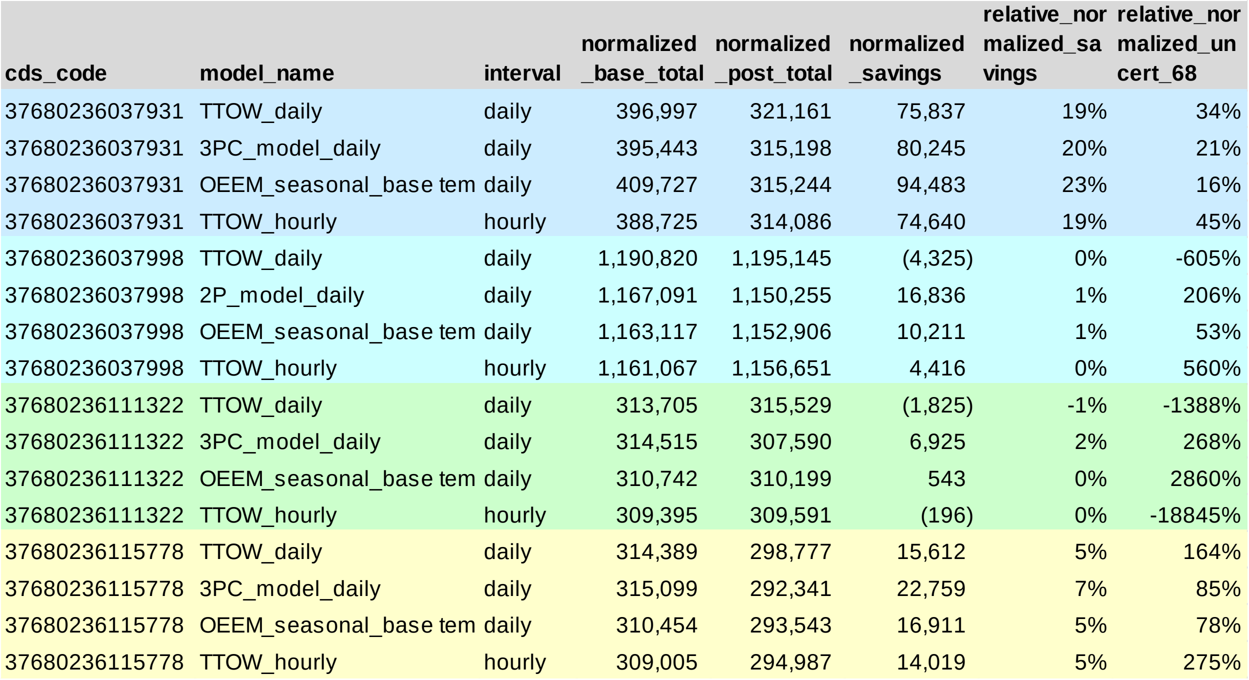
Figure 4.1: Comparison of adjusted baseline total usage (kWh) and avoided energy use (kWh) calculated with six different models at four sites

The hourly and daily model results were fairly consistent for all 4 sites and all 4 models. Figure 4.2 shows the normalized savings results, including relative uncertainty at 68% confidence.

The relative normalized savings values are consistent within 4% for all four sites. Relative uncertainty values at 68% confidence for daily models are also comparable for the first and fourth test sites. The uncertainty for the hourly models is typically greater because we used the autocorrelation penalty method suggested by Guideline 14, which is a simplified method.

For sites 2 and 3 the relative uncertainty is different between the models, however the savings are low enough (2% or less) that relative uncertainty values are not particularly significant.

Overall, the results on these four test sites show agreement between the OEEM and R daily models as well as the R hourly model. We recommend that additional comparisons be performed on more test sites, and that diverging results be investigated.



cds_code	model_name	interval	normalized_base_total	normalized_post_total	normalized_savings	relative_normalized_savings	relative_normalized_uncert_68
37680236037931	TTOW_daily	daily	396,997	321,161	75,837	19%	34%
37680236037931	3PC_model_daily	daily	395,443	315,198	80,245	20%	21%
37680236037931	OEEM_seasonal_base tem	daily	409,727	315,244	94,483	23%	16%
37680236037931	TTOW_hourly	hourly	388,725	314,086	74,640	19%	45%
37680236037998	TTOW_daily	daily	1,190,820	1,195,145	(4,325)	0%	-605%
37680236037998	2P_model_daily	daily	1,167,091	1,150,255	16,836	1%	206%
37680236037998	OEEM_seasonal_base tem	daily	1,163,117	1,152,906	10,211	1%	53%
37680236037998	TTOW_hourly	hourly	1,161,067	1,156,651	4,416	0%	560%
37680236111322	TTOW_daily	daily	313,705	315,529	(1,825)	-1%	-1388%
37680236111322	3PC_model_daily	daily	314,515	307,590	6,925	2%	268%
37680236111322	OEEM_seasonal_base tem	daily	310,742	310,199	543	0%	2860%
37680236111322	TTOW_hourly	hourly	309,395	309,591	(196)	0%	-18845%
37680236115778	TTOW_daily	daily	314,389	298,777	15,612	5%	164%
37680236115778	3PC_model_daily	daily	315,099	292,341	22,759	7%	85%
37680236115778	OEEM_seasonal_base tem	daily	310,454	293,543	16,911	5%	78%
37680236115778	TTOW_hourly	hourly	309,005	294,987	14,019	5%	275%

Figure 4.2: Comparison of Normalized Savings Results for 4 Sites, Daily Models Only

Recommended Improvements

We recommend adding custom holiday input capability within the OEEM framework, because schools typically have long period of low-occupancy in the summer and for a few weeks during the rest of the year. Based on the results discussed above, this did not seem to be a significant issue at our four test sites, possibly because energy usage at these sites remains high during breaks. Additionally, for the Seasonal ElasticNet CV model, we recommend considering the addition of a holiday slope factor, rather than having holidays only modify the intercept.

Similarly, we recommend improvements to both the monthly R change point model and the OEEM CalTrack model in order to account for holidays and breaks. For example, the number of holidays within each period could be added as a variable, or the number of days within each period could otherwise be decreased by a factor based on the number of holidays in the period.

Finally, for improved confidence in the model results, we recommend completing the documentation of the Seasonal ElasticNet CV model, in particular by adding references for the error calculation methodology.

Summary of Methodologies

Table 3 below summarizes the findings of our evaluation of OEEM's two M&V analysis tools.

IPMVP Option C Requirement	CalTrack Monthly	Seasonal ElasticNet CV
1. Development of empirical energy models for baseline and reporting periods Model type Model accuracy checks	Linear models based on heating and cooling degree days with varying reference temperatures Ordinary least squares regressions Model selection process based on *p-value* threshold and R^2 maximization. No input for holidays, vacations, or other building-specific expected events	Linear model based on heating and cooling degree days with fixed reference temperatures Elastic net regularized regression Reference temperatures may be input by user Standard holidays can be used, but there is no straightforward input to add holidays, vacations or other building-specific events
2. Calculation of Avoided Energy Use	Baseline model applied to reporting period temperature yields adjusted baseline. Measured energy values during reporting period are subtracted from baseline model predictions to obtain avoided energy use.	Same as for CalTrack Monthly Same as for CalTrack Monthly
3. Calculation of Normalized Savings	Baseline model and reporting period models are developed. The models are applied to temperature variables calculated from a standard year of data from a chosen weather station to create normalized baseline and post-installation usage predictions, the difference is normalized savings.	Same as for CalTrack Monthly
4. Accuracy and Uncertainty	Variance of the prediction errors is calculated based on the variance of the estimated model coefficients. For normalized savings, the total variances of the prediction errors for the normalized baseline and the normalized post-installation estimates are combined. The actual savings uncertainty is not an output of the code.	Variance of the prediction errors is calculated based on an empirical bootstrap methodology. Same as for CalTrack Monthly Same as for CalTrack Monthly

5. Conclusions

Open Energy Efficiency Meter's two energy modeling methods of interest are contained in the eemeter software repository, which were downloaded from the Github public domain software development platform. These software programs used the Python programming language – a public domain software free to the public and in wide use in academic and professional environments. The two Python methods of interest estimate whole building energy savings based on data in monthly (CalTrack Monthly) and daily (Seasonal ElasticNet CV) time intervals. While the CalTrack Monthly code has completed its peer review process, the Seasonal ElasticNet CV code is still in its beta stage of development. Consequently, narrative documentation of the algorithms to develop models and estimate savings are available on the OEEM website for the CalTrack Monthly code, but not yet available for the Seasonal ElasticNet CV code. For the latter, an experienced Python programmer was able to read through the code and understand its algorithms.

In each case, the OEEM code follows IPMVP Option C procedures. Both OEEM methods perform data quality and sufficiency checks. They develop regression-based energy models from both baseline and reporting (post-installation) period data, and use these models to adjust energy use to a common set of conditions before estimating savings. Both *avoided energy use* and *normalized savings* may be estimated using this software.

The CalTrack Monthly modeling method is in accordance with common industry practices. It uses heating and cooling degree days as independent variables in model development and savings estimation. It develops models using the ordinary least squares method. It fits up to four different forms of linear models, then disqualifies models if the significance of their coefficients (*p-value*) does not meet a fixed threshold, then selects the best fitting model based on their coefficient of determination (R^2 value). This is unique in that common modeling methods, such as change-point models, select models based on the lowest coefficient of variation of the root-mean-squared-error (CV(RMSE)) and highest R^2 values.

The Seasonal ElasticNet CV modeling method is more advanced. While it uses heating and cooling degrees days as independent variables, its model development follows a regularized regression approach not commonly used in the industry, but commonly available in public domain statistical software packages, such as Python. This is another way to determine the coefficients of the linear equation form. Seasonal ElasticNet CV evaluates only one linear model form with fixed reference temperatures for the heating and cooling degree days.

The OEEM software do not output a value of savings uncertainty. Conducting error and uncertainty analysis is a requirement of both IPMVP and ASHRAE Guideline 14's whole-building performance path. In addition to the statistical analysis the OEEM code uses to select the model that best fits the data, it quantifies the variance of the model prediction errors, including the combined variance of prediction errors for calculated energy savings series. This variance can easily be used to calculate savings uncertainty for both avoided energy use and normalized savings, however it requires a sufficient understanding of statistical analysis and should not be left to the user.

One advantage the Seasonal ElasticNet CV method has is that its prediction variances are smaller than those developed from ordinary least squares methods, as estimated using empirical error estimation methods. This can result in lower overall savings uncertainty estimates.

The testing of the OEEM code with CEJAP project data and comparison with results from our R code showed agreement in predicted annual usages used in avoided energy use and normalized savings calculations. We noted that the monthly models produced more outliers when calculating avoided energy usage or normalized savings. However, the relative avoided energy usage and normalized savings estimated from OEEM and R daily models were within 5% of each other. Only one site had significant savings of more than 10% of baseline usage, while the other site's savings estimates were very low. No conclusion could be drawn based on a comparison of estimated uncertainties.

In general, with a few specific improvements, the OEEM code can produce IPMVP-defined estimates of savings. Despite the absence of documentation for the Seasonal ElasticNet CV model, its algorithms were transparent and could be understood by skilled programmers. While the code assesses modeling accuracy in its unique fashion, with a few modifications it can easily produce estimates of savings uncertainty. Its linear heating and cooling degree day models are consistent with common practice. A number of improvements were recommended, which we will forward to the OEEM developers through the Github platform so that they can be addressed and checked in future versions.

6. References

Association of Heating, Refrigeration, and Air-conditioning Engineers (ASHRAE), “Guideline 14 – 2014: Measurement of Energy, Demand and Water Savings,” (2014), www.ashrae.org.

Efficiency Valuation Organization, (2012). International Performance Measurement and Verification Protocol—Concepts and Options for Determining Energy and Water Savings, Volume 1. Efficiency Valuation Organization, 2012. www.evo-world.org.

Kissock, K. K., J. S. Haberl, D. E. Claridge, “ASHRAE Research Project 1050: Development of a Toolkit for Calculating Linear, Change-Point Linear, and Multiple-Linear Inverse Building Energy Analysis Models,” (2004), available at: www.ashrae.org.

Mathieu, J. L., P. N. Price, S. Kiliccote, and M. A. Piette, “Quantifying Changes in Building Electricity Use, with Applications to Demand Response,” IEEE Transactions on Smart Grid, 1949-3053, (2011).

Appendix A: OEEM Evaluation Plan

The scope of work for Subtask 2 required a review of Open EE Meter's (OEEM) energy savings calculation methods. OEEM was designed to provide a transparent, repeatable, and standardized methodology for estimating energy savings through public domain programming code and an accessible platform. OEEM addresses one of the most confounding barriers to achieving energy efficiency and conservation at scale: the high cost and controversy of implementing, reviewing, and evaluating savings estimations by involved parties, which serve ultimately to lower an investor's confidence in achieving expected returns on energy efficiency investments.

Based on a review of available website and related documentation, the OEEM method for estimating savings follows an IPMVP Option C Whole Building approach. Initially designed to address the residential and small/medium building (SMB) market sectors, its analysis methods use monthly billing data and follow well-known energy modeling methods. OEEM developers will soon complete development of methods that are based on daily data, which will increase its potential application to other market sectors, including schools. Methods based on hourly data will soon follow.

"Instances" of OEEM are available through a Github website. Github is an on-line platform where stakeholders (developers, reviewers, and users) may collaborate on the development and review of software code, and track code versions as they are completed and tested. It provides forums for discussion of related topics among its user community, among many other features.

OEEM developers reported that development of the final monthly analysis code was scheduled for completion at the end of February 2017, and the proposed final version of the daily analysis code would be made available for review in February. This schedule allows us to complete our evaluation of OEEM's energy savings estimation methods and process.

Under our Subtask 1.3, we will determine savings from twenty schools that have completed CEJAP projects and that have twelve months of data prior to and twelve months of data after CEJAP measures have been installed. We will use the R code we have developed for this analysis under Subtasks 1.1. and 1.2, and report several aspects of the analysis, including site characteristics, modeling type and methodology, goodness of fit and accuracy metrics, savings (avoided energy use and normalized savings) and savings uncertainty. The R code we have developed employs algorithms that may be applied to monthly, daily, and hourly data.

To complete our evaluation of OEEM's methods, we are directed to run the same data for the Subtask 1.3 projects through OEEM code. The data generally includes electric meter data in AMI form, and natural gas data in monthly billing form. We will make direct comparisons of the outputs of OEEM monthly and daily algorithms with the outputs of the R code for each project, and for the population of twenty projects. We will obtain OEEM source code and review its analysis procedures, identifying adherence with industry standard procedures, and making recommendations for improvements as necessary. We will utilize Github as a means to track comments on methods that can be easily translated into improvements in the OEEM code base.

We will document our Subtask 2 deliverable in a report.