Achieving Actionable Results from Available Inputs: Metamodels Take Building Energy Simulations One Step Further

Henry "Ry" Horsey, Katherine Fleming Ph.D., Brian Ball Ph.D., and Nicholas Long, National Renewable Energy Laboratory

ABSTRACT

Modeling commercial building energy usage can be a difficult and time-consuming task. The increasing prevalence of optimization algorithms provides one path for reducing the time and difficulty. Many use cases remain, however, where information regarding whole-building energy usage is valuable, but the time and expertise required to run and post-process a large number of building energy simulations is intractable. A relatively underutilized option to accurately estimate building energy consumption in real time is to pre-compute large datasets of potential building energy models, and use the set of results to quickly and efficiently provide highly accurate data. This process is called metamodeling.

In this paper, two case studies are presented demonstrating the successful applications of metamodeling using the open-source OpenStudio Analysis Framework. The first case study involves the U.S. Department of Energy's Asset Score Tool, specifically the Preview Asset Score Tool, which is designed to give nontechnical users a near-instantaneous estimated range of expected results based on building system–level inputs. The second case study involves estimating the potential demand response capabilities of retail buildings in Colorado. The metamodel developed in this second application not only allows for estimation of a single building's expected performance, but also can be combined with public data to estimate the aggregate DR potential across various geographic (county and state) scales. In both case studies, the unique advantages of pre-computation allow building energy models to take the place of top-down actuarial evaluations.

This paper ends by exploring the benefits of using metamodels and then examines the cost-effectiveness of this approach.

Introduction

Building energy models (BEMs) serve an important role as a key enabling technology for evaluating various building design alternatives and their impact on energy use. In a new-building context, these models allow for building experts to understand the impact of various decisions on the future energy use of the building. In retrofit contexts, BEMs allow for building experts to identify the most cost-effective energy conservation measures (ECMs) for specific buildings. Several tools exist to address these use cases. In this paper, we present two case studies that make extensive use of one such tool: OpenStudio. A software development kit funded by the U.S. Department of Energy (DOE), OpenStudio allows users to interact with the EnergyPlus whole-building energy simulation engine to determine the energy consumption of buildings using modern scripted programming languages (OpenStudio 2016 and EnergyPlus 2016).

A unique aspect of OpenStudio is a document called an OpenStudio measure. The measure is a Ruby script that accepts user inputs, loads an OpenStudio or EnergyPlus model, performs actions on the model, and then saves the changed model back to an OpenStudio or EnergyPlus model (Long, 2014). Some examples of OpenStudio measures that articulate the underlying BEM are measures such as adjusting a building's window-to-wall ratio to a user-specified input, adding stories to a building, or increasing the occupant density by a user-defined multiplier. OpenStudio measures can also be ECMs such as adding daylighting controls to a building, changing fans from constant to variable speed, or increasing envelope thermal properties. The key aspect of an OpenStudio measure is that it can programmatically apply simple user arguments to generate complex changes in the BEM. In this way, OpenStudio measures become a programmatic interface for those BEMs that use OpenStudio and EnergyPlus.

Within industry today, two well-known applications of programmatic interactions with models are calibration and optimization. The goal of calibration is to align the BEM of a specific building with a measured value—often the monthly utility bill. BEM calibration parameters, such as R-values, component efficiencies, infiltration rates, and building schedules, can be difficult to accurately identify, and as a result they introduce uncertainty into the BEM. To reduce the uncertainty and minimize the error between the BEM and the utility bill, optimization algorithms vary these parameters. The result of this process, ideally, is a calibrated BEM.

Having achieved a calibrated energy model, users can then use ECMs to maximize building energy savings while minimizing cost within the constraints of the retrofit project. The interactive effects that can occur between various combinations of ECMs often complicate this task. Optimization algorithms can be used to select the best combination of ECMs for a retrofit project given various parameters, such as the building owner's budget and required payback period. In both of these programmatic use cases (calibration of BEMs and optimization of ECMs for already-calibrated BEMs), the inputs of a programmatic BEM tool, like OpenStudio, and its energy-use outputs enable powerful workflows. In many situations, however, clearly defined inputs for BEMs are not available and, as a consequence, information about optimal sets of ECMs is significantly less clear.

Retrieving actionable insights from these problematic situations requires a retooling of existing high-level models (or the results of said models) to enable a wider array of workflows. Metamodels offer a more flexible workflow that is achieved through wrapping the imperfect/inexact results of BEM simulations inside of a metamodel created using the random forests machine-learning technique. This metamodeling process allows for less-well-defined inputs to provide valuable, even actionable, outputs by extracting the crucial intelligence contained in simulated BEMs.

In this paper, we explain the concept of a metamodel and present an intuitive example to clarify the role metamodels play in providing useful outputs from ambiguous inputs. We then present the commonly used infrastructure: first the OpenStudio Analysis Framework (OSAF), and second, random forests and their use as the metamodel regression engine. Next, we describe two case studies in which metamodels based on BEM simulations provide simple and valuable outputs to complex problems. The first examines the Asset Score Preview tool recently released by DOE, and the second demonstrates a framework for the analysis of aggregated DR capabilities. In each study, we examine the way in which metamodels enable actionable outputs with minimal inputs. Finally, we more generally discuss the useful attributes of a metamodeling workflow based off of BEMs.

The use of metamodeling to provide insight from unclear inputs in the context of BEMs is not new, and has been previously considered by *Eisenhower 2011* and *Romani 2015*. In addition, a critical aspect of metamodeling is the creation of regression engines. Regression engines have been widely employed in the context of building energy, notably for simplified prediction methodologies for optimal ECM recommendations and energy use forecasting. An excellent overview of this area of research and practice is presented in *Kissock et al. 2003, Harbel and Thamilseran 1998,* and *Granderson and Price 2013.*

What Is Metamodeling?

A metamodel, in the context of this paper, is a model that contains the results of previously run models—these simulation results are referred to as the underlying results set. In both case studies examined in this paper, the underlying results set is a large collection of previously computed building energy model simulations. While these results have inherent value, recasting them in a metamodel can significantly increase their value in use cases containing ambiguous inputs and outputs. The metamodel can have inputs different from the actual energy models, can base its output on the entire pre-computed results set, and is orders of magnitude faster than standard BEM simulations.

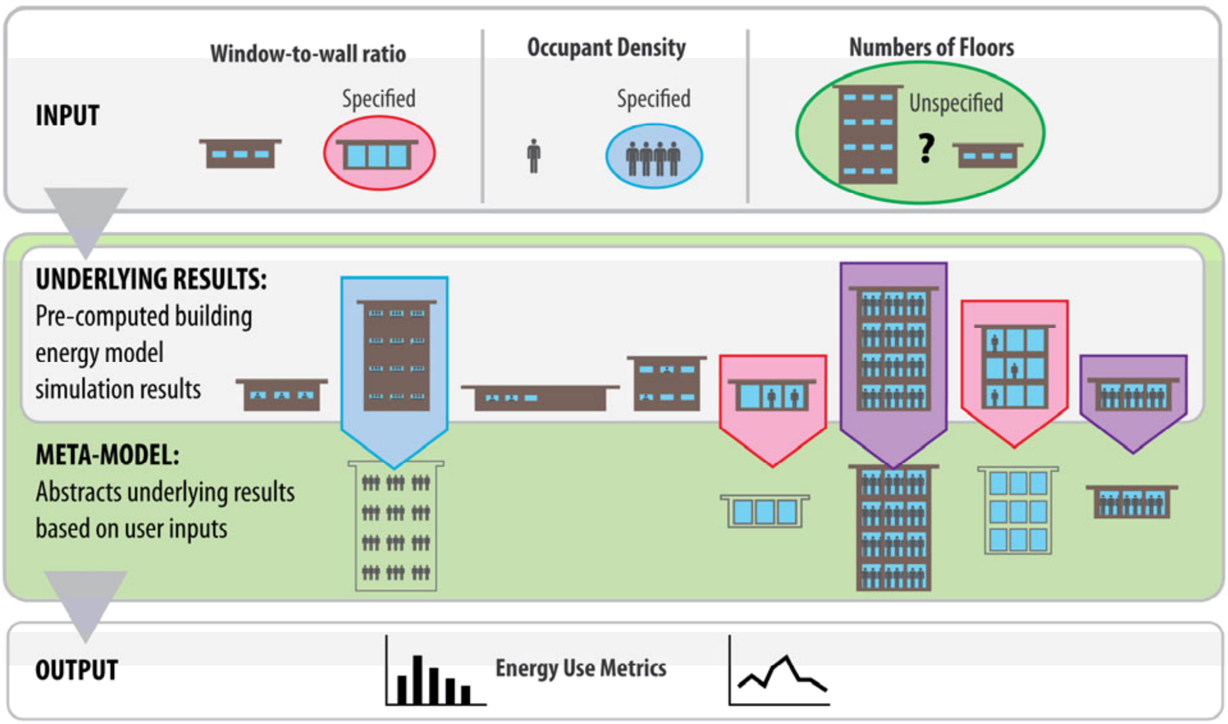

Figure 1: Diagram of a generalized metamodeling process using pre-computed building energy models.

In this paper, we define metamodels by four characteristics: the underlying model, the method used to build the metamodel regression engine, the inputs to the metamodel, and the outputs from the metamodel. Figure 1 shows a generalized example of a metamodel in the context of BEMs. Here, the underlying model is a set of results pre-computed from BEMs that have inputs of window-to-wall ratio, occupant density, and number of floors. The inputs to the metamodel are user-specified characteristics circled below—specifically, high window-to-wall ratios and high occupant density. The specification for number of floors is left undefined. Using all three input specifications, the metamodel then selects those results in the pre-computed building set that reflect the user's input specifications. In Figure 1 the buildings that only partially match the user-defined inputs have been retrieved from the underlying results set;

however, they are given a lesser weight in the consideration of results because they don't meet both the predefined criteria. The metamodel then takes the results of the buildings that it has found to be similar to the user's query and returns an amalgamation of the results back to the user.

Figure 1 shows that the metamodel can retrieve the pertinent BEM results given varied and potentially unspecified user inputs. The manner in which these results are processed and returned to the user, however, can be complicated but is critical for the success of the metamodel. In both presented case studies, a bootstrapping methodology is used to determine the likely output energy-use intensity (EUI) of the user-defined building as a probability distribution. In this approach, a regression model is built that considers input building characteristics (window-to-wall ratio, occupant density, and number of floors) and outputs a specified result from the simulation (EUI). Through running the regression engine on a large variety of possible input values given the user's selections, an equally large number of output estimates are calculated and used to create a probability distribution of the likely output result of the building.

This regression engine is the core of the metamodel. When a user's inputs are entered, the metamodel generates a set of potential datapoints—that is, combinations of allowable characteristics of the model. In the example above, the datapoints returned would have a fixed high window-to-wall ratio and occupant density, but no restrictions placed on the number of floors. The datapoints are then run through the regression engine, and the results form a distribution—namely, a variety of answers representative of the user's input. This distribution then serves as the output of the metamodel: a range of EUI.

Another way of considering a metamodel is as a simple wrapper model that interprets potentially ambiguous inputs and utilizes the underlying pre-computed results to produce the output. In this representation, the metamodel serves as a gray box surrounding a large number of pre-computed BEMs. The metamodel utilizes the relationships between the inputs and output of the BEMs embedded in it to take uncertain definitions from users and return results that reflect that uncertainty. In this way, the metamodel allows for a distribution of input variables to return a distribution of output metrics in close to real time.

Although metamodel implementation is often highly complex, the underlying goal is generally quite simple. In the example shown in Figure 1, the goal was to identify a distribution of potential outcomes based on the pre-computed BEMs, constrained by the user-defined inputs of buildings with high occupant density and window-to-wall ratios. In the case studies presented below, the goal is ultimately to provide actionable information to various building energy–related stakeholders, given uncertain and partial user inputs, through the use of pre-computed BEMs as a basis for metamodels.

This section presents an overview of the underlying models used for these case studies as well as the procedures used to generate the metamodels. OpenStudio measures are used to define building parameters that can be varied (for example, window-to-wall ratio, plug loads, and air-handler fan efficiency). Thousands of unique simulations are run with varying parameters, and specified results are returned. In the first case study, adjusted source EUI was the desired result, while in the second study, peak and total DR contingency event savings are of primary interest. These sets of inputs and outputs are used to build a random forest regression engine, which is a type of machine-learning algorithm. When given an input, the random forest regression engine returns an output in near-instantaneous time, and this result is postprocessed to create an actionable result for the user. This entire process is called a metamodel.

Tools Employed to Create a Metamodel

Two tools were primarily used to create the metamodels presented in both studies. The first, the OpenStudio Analysis Framework (OSAF), is a collection of open-source software tools developed by the National Renewable Energy Laboratory (NREL) to support its continued work in the field of large-scale commercial building energy modeling. The second tool is the random forest machine-learning procedure initially developed by Leo Breiman and Adele Cutler at the University of California, Berkeley.

OpenStudio Analysis Framework

The OSAF is a set of server and software tools for defining and running large-scale analyses of whole-building energy models. Presently, the most accessible user interface is a spreadsheet that enables multiple types of analyses, including parameter sampling, calibration, optimization, and sensitivity analysis. Due to the large size of the simulation set required for metamodels, simulations are run using the OSAF on Amazon's Elastic Cloud Computing (EC2) platform. A number of visualizations are generated to help users understand the results of the simulations, including parallel coordinate, radar, and Pareto plots. The OSAF, at its core, serves as an enabling technology for applying measures to whole-building energy models, both in bulk user-specified ways and in interactions with particular algorithms. The process by which these analyses are run is detailed in Figure 2 (Long et al. 2014). Enterprise and advanced users can install the OSAF platform on a workstation or, with significant expertise, a supercomputer.

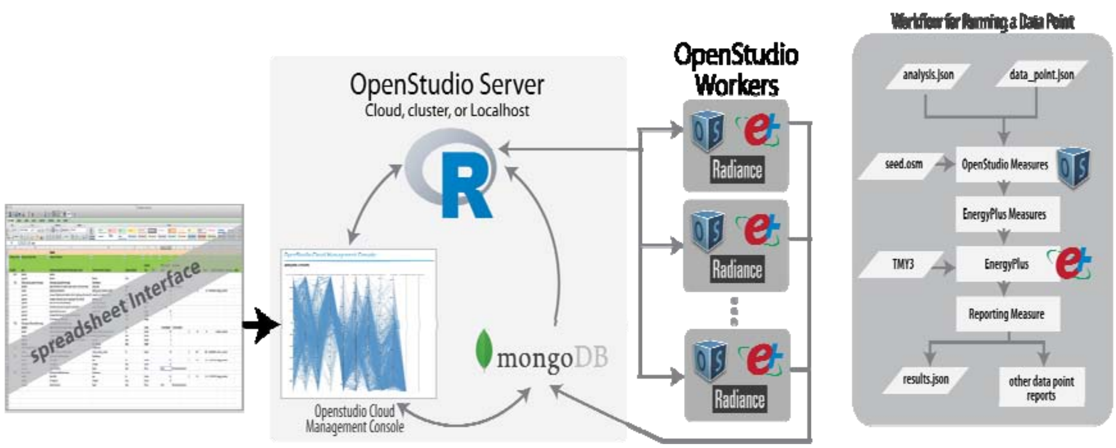

Figure 2: OpenStudio Analysis Framework system diagram.

Each simulation is executed by the worker node managed by the OSAF server. In each simulation, a seed OpenStudio Model (OSM) file is loaded, and then each OpenStudio measure is run with server-defined variable and argument values until all OpenStudio measures have been run. At this point, nearly all changes to the OSM have been made. The OSM is then translated into an EnergyPlus Input Definition File (IDF), and any additional measures, such as advanced controls, are then executed. Once these measures have run, EnergyPlus is called to simulate the final IDF. The results are passed through a standard-output reporting measure, and then through any additional reporting measures defined in the user-defined workflow. The set of variable measure inputs for the specific simulation as well as the high-level results of the simulation are then passed back to the server and registered in the results data table. Additional file outputs of reporting measures can be flagged to be available to the user as a download.

In the first case study, the results data table was used to populate the metamodel. The variable measure inputs were defined as the input of the random forest regression engine, and the output of the random forest regression engine was defined as the adjusted source EUI of the building. In the second case study, the parameters of the random forest regression engine were slightly more complicated. The DR contingency potential of each simulated building was evaluated through a simulated DR event for each day of a yearly simulation. The input associated with each event was defined as the set of environmental variables at the beginning of a day's DR event—such as dry-bulb temperature and direct isolation—as well as the measure inputs associated with that simulation. The output of each event was defined as the maximum kilowatts shed during the event period and the total kilowatt-hour savings during the event.

Random Forests

A random forest is a machine-learning model called an ensemble learner. In general terms a random forest is a large collection of independent decision trees that acts as a regression engine. The random forest as a whole achieves a high degree of reliability and accuracy from the diversity in the set of trees; each tree is trained on a random subset of the data initially provided to the random forest as a training set. Each tree attempts to minimize the variance in the training dataset at each split, thereby increasing the certainty of the result at each node of the tree. The details of the random forest machine-learning procedure are described in mathematical detail in Breiman 2001, and an excellent introduction to the topic can be found in Breiman and Cutler 2016.

A variety of machine-learning techniques could have been used to achieve similar outcomes to those described in the two case studies; however, the use of random forests was primarily driven by its ease of development and speed of results. While many machine-learning techniques are highly sensitive to various tuning parameters, random forests have only two main tuning parameters, and both have been shown by Breiman to be relatively insensitive. Additionally, it is worth noting that random forests are nonlinear and can model categorical variables (names rather than numbers)—such as heating fuel type, where the options are either electricity or gas. These two properties of random forests enable them to respond well to wide input ranges and alleviate the need for complex preprocessing or postprocessing of the data.

Case Study: Asset Score Preview

Motivation

The U.S. DOE's Asset Scoring Tool is a tool aimed at helping building owners, operators, and tenants to evaluate the energy-efficiency potential of a building's assets (Wang 2015, and 2016). To achieve this, the building assets are defined as the set of building systems that function independently of the occupant and are assumed to experience no degradation in performance due to lack of maintenance. The output of the Asset Score Tool is a rating on a 10-point scale, where a score of 10 represents buildings with high potential efficiency, and a score of 1 represents buildings with a low potential for efficiency. To allow for a high level of accuracy when rating the Asset Score of commercial building stock, a relatively exhaustive dataset of building characteristics is required. To avoid the Asset Score Tool's use being limited to building owner/operators who have access to this level of data, it was decided to develop a "lite" version of the tool for owners/operators with less access to technical specifications. The lite version requires fewer inputs with reduced complexity from the user (the minimum being building vintage, location, floor-area, orientation, number of stories, and building-use type) and returns a range of possible scores as the output. A metamodel provides an excellent framework to solve the unusual requirements of this problem, and was developed and deployed as Asset Score Preview (BTO 2016).

The challenges involved in developing the functionality for Asset Score Preview stem from the need to translate a nonspecific user-supplied variable, like building vintage, into a useful modeling characteristic of building performance, like R-value. The full Asset Scoring Tool uses OpenStudio measures to incorporate 38 user inputs—many of which require a building engineer to obtain—into an energy model, and then runs a full simulation of the building. For Asset Score Preview, however, the input-gathering burden placed on the user is limited to general information about the building and an indication of agreement or disagreement with default values of building systems such as envelope construction type, lighting technology, HVAC system type, and service water-heating fuel type. The properties of a building's subsystems can be set to nondefault values when the user has particular knowledge of the values. The high-level inputs of the Preview allow nonbuilding experts to interact with and gain confidence in the tool, and may encourage them to perform a full Asset Score audit in the future.

Method

The first step in defining the metamodel for Asset Score Preview is specifying the inputs and outputs. The required inputs are use type, location, vintage, orientation, conditioned floor area, and number of floors. Asset Score Preview provides the remaining subsystem inputs as defaults. The user then has the option to either confirm or change the default. If the user does neither, the input is flagged as uncertain for the subsequent analysis. The underlying result set is a large collection of simulations run on various models across all climate zones. The metamodel holds the required and confirmed subsystem inputs constant and considers the range of all uncertain subsystem inputs. Details on this process can be found in Goel et al. 2016b. As an example, if a user confirms that a retail building in Denver has T8 lighting, but is uncertain of all other parameters, the metamodel considers the full range of subsystem inputs except for lighting power density and the required inputs. The metamodel returns a distribution of adjusted source EUIs that are translated into a potential Asset Score range. For a detailed description of the process by which this was achieved, please refer to Goel et al. 2016a and 2016b and Long et al. 2015.

The level of detail incorporated in the underlying models is an important characteristic of a robust metamodel. While some details are easy to account for, such as translating user-defined T8 lighting into lighting power density, other details, such as inferring a building's geometry from unspecified volumetric conditioned space and envelope surface area, are complex and intractable. The key to problems such as unspecified geometry is to make sure that the range of variables considered in the underlying result set is broad and dense enough to capture all potential user inputs. For a more lengthy discussion on how this problem was approached in Asset Score Preview, please refer to Goel et al. 2016a.

Outcomes

Although the results of a metamodeling approach can be difficult to demonstrate, in this case, an excellent example is provided by one of the pilot buildings used to test the Asset Score Preview tool. The distribution of adjusted source EUIs returned by the Asset Score Preview metamodel under various user-confirmed input conditions is shown in Figure 3. In each test case, the default subsystem definitions were set to the correct values while the user's confirmation was varied between the samples. As can be seen in the purple density plot, when no inputs were certain, the range of possible values was quite wide—from 100 to 260 kBTU/sqft/yr. When the HVAC inputs were certain, the upper range of possible EUIs was scaled back by approximately 30 kBTU/sqft/yr—from 100 to 220 kBTU/sqft/yr. When the lighting technology was certain, the range of potential EUIs was drastically reduced—from 155 to 260 kBTU/sqft/yr—and when all inputs were fixed, the range was reduced to 170 to 200 kBTU/sqft/yr. In addition, the most likely outcome was closely aligned with the EUI returned by a full Asset Score simulation: 185 kBTU/sqft/yr. What is even more critical than the accuracy of the metamodel is its ability to respond dynamically to user input. In this way, the metamodel can serve its primary purpose: to encourage users to perform a full Asset Score evaluation and to identify which buildings might benefit most from additional examination.

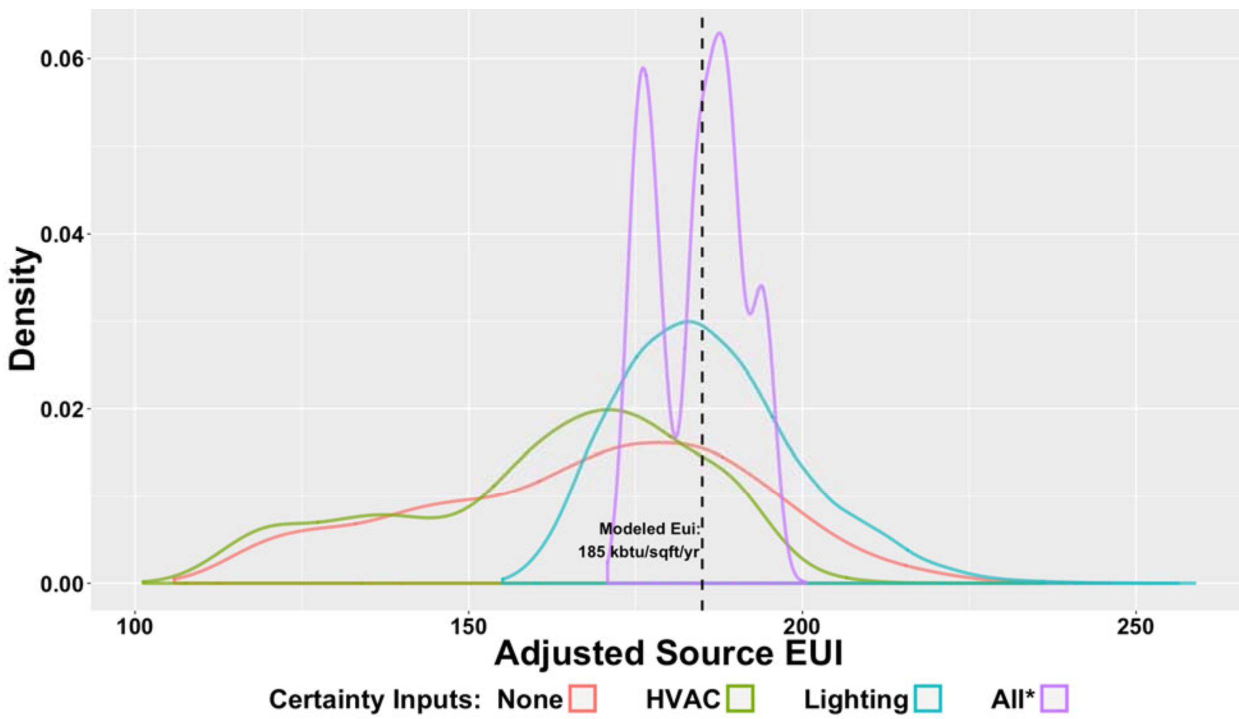

Figure 3: Metamodel results for varying user-certainty inputs when previewing the Asset Score of a medium office building with a packaged VAV system.

Case Study: Demand-Side Grid Integration and Development (DSGRID)

Motivation

The commercial building stock has the potential to provide several grid-related services, including contingency, flexibility, regulation, and capacity, through DR events. The maximum potential capacity for each grid service needs to be quantified to understand the overall potential for building-grid interactions. NREL has funded an internal project to begin to model the aggregate potential capacity reserves, in megawatts and megawatt-hours, provided by the commercial building stock. The project first uses the OSAF to model large sets of buildings and then uses a metamodel to aggregate the results of the individual models by geographic region. The goal of this project is to enable analysis of building-stock DR potential for various use cases such as integrated resource plans, national projections, and policy decisions using physics-based BEMs. In addition, the physics-based models can provide inputs to various grid-modeling software tools to enhance their fidelity.

Method

The process of analyzing the DR potential of a specific building type across a geographic region can be broken down into four separate steps. First, OpenStudio measures are applied to the baseline models to vary building parameters such as lighting power density, HVAC system efficiencies, and building envelope parameters. Second, a single DR event is modeled each day of the annual simulation for each model in the set of articulated BEMs. Third, the results of each simulation have to be processed to determine both the maximum reduction in load during the event and the total energy not used during the event (maximum shed and total savings, respectively). These results, merged with information about the articulated baseline model, form the pre-computed results used as the underlying results set. Fourth, the properties of the commercial building stock of interest are estimated using available datasets. The results of this step represent the inputs to the metamodel. As in the previous case study, the metamodel uses a random forest regression engine.

The case study estimates the contingency reserves potential of a thermostat setpoint setback reduction in the stand-alone retail building-type segment in Colorado. The building stock examined was represented by various vintages of the stand-alone retail reference model with articulated envelope, lighting, and HVAC properties. To represent the highly variable weather conditions of Colorado, 26 weather files from the state were sampled to create the simulations. The metamodel inputs were derived from a combination of the U.S. Energy Information Administration’s Commercial Building Energy Consumption Survey (CBECS) 2003 microdata files and the U.S. Census’ County Business Patterns (CBP) data. This work is currently in publication, and is expected to appear in 2017.

Outcomes

The metamodeling approach used in the DSGRID case study allows for a minute-by-minute analysis of the HVAC DR potential of retail buildings in Colorado on a county-by-county basis. The unique aspects are the granularity of the results on both a temporal and geographic scale. A combination of CBECS and CBP data provides estimates of the number of buildings in each county in Colorado, as well as estimates of building vintage. A variety of building characteristics and environmental factors remain undefined; however, they were predicted across their entire range.

In this case study, the metamodeling approach is also used to provide input files to the PLEXOS modeling software, which is the primary grid-modeling tool used by NREL researchers for high-renewable penetration studies. PLEXOS requires a CSV input file specifying the expected available contingency reserve at each hour of the year. The metamodel generates the input file by weighting the weather files to represent the relative geographic distribution of retail buildings in Colorado that are modeled at each time step. Figure 4 shows the total contingency potential of these buildings in full geographic granularity during a week in July, with each stacked color representing the potential contingency provided by a specific county. The total potential at each time step can be used to inform high-level analyses, while the county-by-county results can be used in finer-granularity grid studies.

The unique aspect of the metamodel in this use case is its ability to provide an aggregated representation of the results of BEMs to a grid-modeling tool through an easily defined interface. To generate resources for other grid-modeling tools, the metamodel results simply need to be processed into a form acceptable for that specific tool. Once again, the need to develop additional whole-building energy simulations is avoided. The metamodel, in short, becomes a currency of exchange between building-focused and grid-focused analysis and modeling tools.

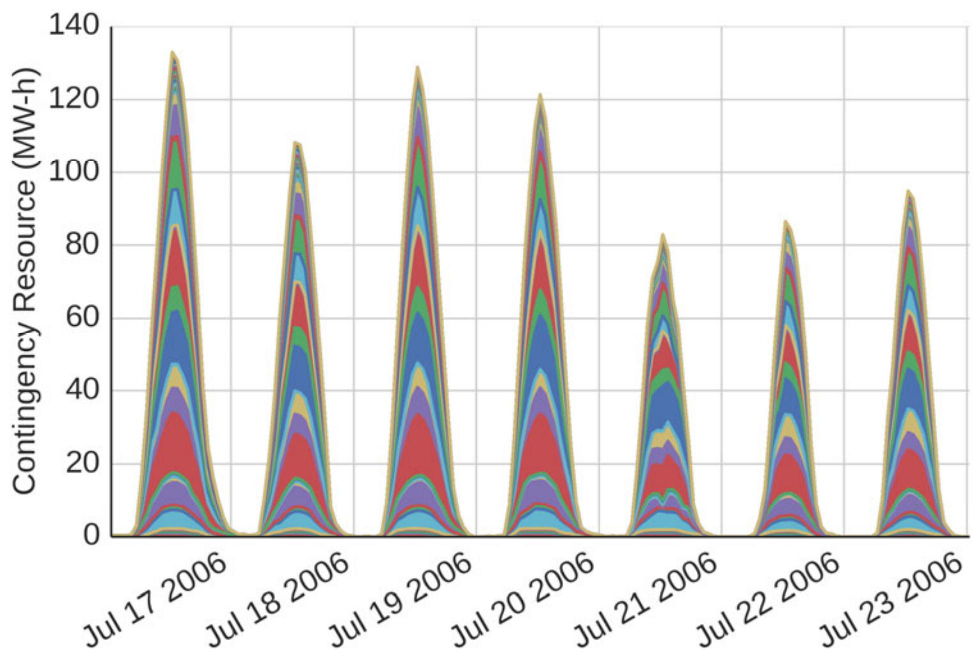

Figure 4: Example of the time series output for PLEXOS based on retail HVAC DR resources in Colorado.

The Benefits of Metamodeling

The Asset Score Preview and DSGRID case studies present a general framework in which metamodels take the results of detailed BEMs and use them to derive actionable information for wider applications. There are several aspects of metamodels that make them attractive for use in extending the applicability of BEMs. First, given current cloud computing infrastructure, the actual costs associated with the pre-computed underlying results set are relatively minimal. Second, the flexibility of inputs provided through the use of sampling in the metamodel allows for a single metamodel to be used in a variety of frameworks. Finally, the modularity provided by the use of automated BEM creation and simulation using the OSAF facilitates high iteration speed and version control of the results.

Cost

One of the most critical enabling technologies for the approach described in this paper is the low cost of cloud computing. The OSAF is currently deployable on the Amazon Web Services' EC2 platform, and the results presented for both case studies relied on this computational resource (Amazon 2016). At the time of writing, a 32-core server (on-demand compute-optimized c3.8xlarge instances) costs $1.68 per hour of usage. The average computation time of an Asset Score Preview building model was approximately five minutes. Assuming an additional 50 percent overhead rate for the creation, provisioning, and destruction of servers, the total cost of the analysis is approximately $1,800, which is negligible in comparison to staff costs associated with the analysis.

Flexibility

Developing highly complex models can be difficult to justify, especially if the models are only applicable in one use case. Metamodels, however, are only truly limited by the accuracy and parameter space of the underlying results set, as well as the accuracy of the regression engine built using this results set. All regression engines used in the case studies presented had R^2 values above 0.90 out of the box, with an average R^2 of 0.95. EnergyPlus, the simulation engine that OSAF relies on, is generally accepted to be highly accurate, although absolute measures of this are difficult (Judukoff et al. 2011). The input criteria to the metamodels are incredibly flexible due to the ability to sample the regression engine across the range of each unspecified variable. Additionally, the outputs can be postprocessed to provide various desired outcomes, as presented in the DSGRID case study.

Modularity

The complexity of whole-building energy modeling ensures that base models and the measures interacting with them can always be improved. Quite often this means that labor-intensive tasks must be performed throughout the entire workflow chain every time revisions are made. In the workflow presented in both case studies, the programmatic nature of OSAF, along with the separation of interfaces provided by the metamodel regression engine, makes whole-building energy-modeling iteration a relatively painless process. As an example, when the control strategies associated with the thermostat setpoint adjustment measure in the DSGRID case study needed to be changed, it was a simple task to insert the updated measure, rerun the simulations on EC2, and rebuild the metamodel regression engine on the new underlying dataset. No changes were required to the interfaces with the metamodel. This property of the described workflow has an important secondary effect: it is easy to version the results produced by the metamodel by incrementing the version whenever the underlying model or the interface to the metamodel is updated. This enables reproducible results, which is a critical requirement in a variety of use cases.

Conclusion

The two use cases described in this paper demonstrate how metamodeling can use data provided by BEMs to produce actionable results for nonbuilding scientists. Due to the fast run times of the random forest regression engine and the detailed nature of the underlying results set, users are able to access insights using metamodels that they may not have had the expertise or wherewithal to otherwise discover. This process can also serve as an excellent way to extract new value from existing BEM result sets.

Using cloud computing resources, the process used to generate the underlying results sets is cost-effective and removes the need for extensive local computing resources. In addition, modular workflows allow users to quickly change measures and rerun the analysis in a versionable and traceable manner, increasing robustness and reducing user errors. In each case, the ability for metamodels to work with ambiguous inputs and provide valuable insights through realistic uncertainty proves to be reliable and consistent. The usefulness of metamodels may have a larger impact in other areas; however, each use case needs to be scrutinized with care to ensure that the accuracy and vastness of the parameter space are reasonably approximated. This required scrutiny highlights the critical role that building scientists and energy engineers will continue to play in the evolving building energy analytics market, regardless of the ongoing proliferation of low-cost computing and highly automated learning algorithms.

Acknowledgements

The authors wish to thank Supriya Goel, Nora Wang, Juan Gonzales, Noel Merket, Elaine Hale, Brady Stoll, Ian Dobber, and Ambarish Nag for their critical work with regards to the presented case studies, as well as the development teams for OpenStudio and EnergyPlus.

Citations and References

Amazon. 2016. https://aws.amazon.com/ec2/pricing/. (Accessed on 8 March, 2016).

ASHRAE (American Society of Heating, Refrigeration, Air Conditioning Engineers). 2004. ASHRAE Standard 90.1. Atlanta, GA.

BTO (Buildings Technology Office, U.S. Department of Energy). 2016. Asset Score Preview. https://buildingenergyscore.energy.gov/buildings. (Accessed on 6 March, 2016).

Deru, M., K Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, M. Rosenberg, M. Yazdanian, J. Huang, and D. Crawley. 2011. *U.S. Department of Energy Commercial Reference Building Model of the National Building Stock.* National Renewable Energy Laboratory.

Eisenhower, B., Z. O'Neill, S. Narayana, V. Fonoberov, and I. Mezic. 2012. "A Methodology for Meta-Model Based Optimization in Building Energy Models." *Energy and Buildings* 47: 292–301.

EnergyPlus. 2016. http://energyplus.net (Accessed on 8 March, 2016).

Granderson, J. and P. Price. 2013. *Development and Application of a Statistical Methodology to Evaluate the Predictive Accuracy of Building Energy Baseline Models*. Lawrence Berkley National Laboratory.

Goel, S., H. Horsey, and N. Long. 2016a. "Building Efficiency and Uncertainty Analysis with DOE's Asset Score Preview*.*" *2016 ASHRAE/IBPSA-USA Building Simulation Conference*, Salt Lake City, UT.

Goel, S., N. Wang, H. Horsey, and N. Long. 2016b. "Streamlining Building Efficiency Evaluation with DOE's Asset Score Preview." *Proceedings of the ACEEE Summer Study*. Asilomar, CA.

Haberl, J. and S. Thamilseran. 1996. "Predicting Hourly Building Energy Use: The Great Energy Predictor Shootout II: Measuring Retrofit Savings—Overview and Discussion of Results." *ASHRAE Transactions*, 102 (Pt. 2).

Haberl, J., A. Sreshthaputra, D. Claridge, and J. Kissock. 2003. "Inverse Modeling Toolkit: Applications and Testing." *ASHRAE Transactions*, 109 (Pt. 2).

Judkoff, R., B. Polly, and M. Bianchi. 2011. *The Building Energy Simulation Test for Existing Homes (BESTEST-EX) Methodology*. NREL-51655, National Renewable Energy Laboratory

Long, N., B. Ball, K. Fleming, and D. Macumber. 2014. Scaling Building Energy Modeling Horizontally in the Cloud with OpenStudio. *2014 ASHRAE/IBPSA-USA Building Simulation Conference*, Atlanta, GA.

Long, N., S. Goel, and H. Horsey. 2015. U.S. Department of Energy's Asset Score Sensitivity and Scale Implementation*. Building Simulation 2015*, Hyderabad, India.

OpenStudio. 2016. www.openstudio.net. (Accessed on March 8, 2016).

Romani, Z., A. Draoui, and F. Allard. 2015. "Metamodeling the Heating and Cooling Needs and Simultanious Building Envelope Optimization for Low Energy Building Design in Morroco." *Energy and Buildings* 102: 139–148.

USDOE. 2015b. Commercial Reference Buildings. http://energy.gov/eere/buildings/commercial-reference-buildings. (Accessed 10 January, 2016).

Wang, N., S. Goel, V. Srivastava, and A. Makhmalbaf. 2015. *Commercial Building Energy Asset Score System: Program Overview and Technical Protocol (Version 1.2)*. PNNL-22045 Rev 1.2, Pacific Northwest National Laboratory.

Wang, N., S. Goel, A. Makhmalbaf, and N. Long. 2016. "Development of Building Energy Asset Rating Using Stock Modeling in the USA." *Journal of Building Performance Simulation*.