2016

DR-Advisor: A Data-Driven Demand Response Recommender System

Madhur Behl
University of Pennsylvania, mbehl@seas.upenn.edu

Francesco Smarra
Universita` degli Studi dell'Aquila, francesco.smarra@univaq.it

Rahul Mangharam
University of Pennsylvania, rahulm@seas.upenn.edu

DR-Advisor: A Data-Driven Demand Response Recommender System

Abstract

Demand response (DR) is becoming increasingly important as the volatility on the grid continues to increase. Current DR ap- proaches are predominantly completely manual and rule-based or involve deriving first principles based models which are ex- tremely cost and time prohibitive to build. We consider the problem of data-driven end-user DR for large buildings which involves predicting the demand response baseline, evaluating fixed rule based DR strategies and synthesizing DR control actions. The challenge is in evaluating and taking control decisions at fast time scales in order to curtail the power consumption of the building, in return for a financial reward. We provide a model based control with regression trees algorithm (mbCRT), which allows us to perform closed-loop control for DR strategy synthesis for large commercial buildings. Our data-driven control synthesis algorithm outperforms rule-based DR by 17% for a large DoE commercial reference building and leads to a curtailment of 380kW and over $45, 000 in savings. Our methods have been integrated into an open source tool called DR-Advisor, which acts as a recommender system for the building's facilities manager and provides suitable control actions to meet the desired load curtailment while main- taining operations and maximizing the economic reward. DR-Advisor achieves 92.8% to 98.9% prediction accuracy for 8 buildings on Penn's campus. We compare DR-Advisor with other data driven methods and rank 2nd on ASHRAE's benchmarking data-set for energy prediction.

Keywords

CPS, data-driven, machine learning, demand response, energy, buildings

Disciplines

Computer Engineering | Electrical and Computer Engineering

Available online at www.sciencedirect.com

ScienceDirect

Applied Energy 00 (2016) 1–25

Applied Energy

DR-Advisor: A Data-Driven Demand Response Recommender System

Madhur Behl[a], Francesco Smarra[b], Rahul Mangharam[a]

[a]*Dept. of Electrical and Systems Engineering,*
University of Pennsylvania,
Philadelphia, USA
[b]*Department of Information Engineering, Computer Science and Mathematics,*
Università degli Studi dell'Aquila,
L'Aquila, Italy

Abstract

Demand response (DR) is becoming increasingly important as the volatility on the grid continues to increase. Current DR approaches are predominantly completely manual and rule-based or involve deriving first principles based models which are extremely cost and time prohibitive to build. We consider the problem of data-driven end-user DR for large buildings which involves predicting the demand response baseline, evaluating fixed rule based DR strategies and synthesizing DR control actions. The challenge is in evaluating and taking control decisions at fast time scales in order to curtail the power consumption of the building, in return for a financial reward. We provide a model based control with regression trees algorithm (mbCRT), which allows us to perform closed-loop control for DR strategy synthesis for large commercial buildings. Our data-driven control synthesis algorithm outperforms rule-based DR by 17% for a large DoE commercial reference building and leads to a curtailment of 380kW and over $\$45,000$ in savings. Our methods have been integrated into an open source tool called DR-Advisor, which acts as a recommender system for the building's facilities manager and provides suitable control actions to meet the desired load curtailment while maintaining operations and maximizing the economic reward. DR-Advisor achieves 92.8% to 98.9% prediction accuracy for 8 buildings on Penn's campus. We compare DR-Advisor with other data driven methods and rank 2^{nd} on ASHRAE's benchmarking data-set for energy prediction.

Keywords:
Demand Response, Regression trees, data-driven control, machine learning, Electricity curtailment, Demand side management

1. Introduction

In 2013, a report by the U.S. National Climate Assessment provided evidence that the most recent decade was the nation's warmest on record [1] and experts predict that temperatures are only going to rise. In fact, the year 2015 is very likely to become the hottest year on record since the beginning of weather recording in 1880 [2]. Heat waves in summer and polar vortexes in winter are growing longer and pose increasing challenges to an already over-stressed electric grid.

Email addresses: mbehl@seas.upenn.edu (Madhur Behl), francesco.smarra@univaq.it (Francesco Smarra), rahulm@seas.upenn.edu (Rahul Mangharam)

Furthermore, with the increasing penetration of renewable generation, the electricity grid is also experiencing a shift from predictable and dispatchable electricity generation to variable and non-dispatchable generation. This adds another level of uncertainty and volatility to the electricity grid as the relative proportion of variable generation vs. traditional dispatchable generation increases. The organized electricity markets across the world all use some variant of real-time price for wholesale electricity. The real-time electricity market at PJM, one of the world's largest independent system operator (ISO), is a spot market where electricity prices are calculated at five-minute intervals based on the grid operating conditions. The volatility due to the mismatch between electricity generation and supply further leads to volatility in the wholesale price of electricity. For e.g., the polar vortex triggered extreme weather events in the U.S. in January 2014, which caused many electricity customers to experience increased costs. Parts of the PJM electricity grid experienced a 86 fold increase in the price of electricity from $\$31/\mathrm{MW\,h}$ to $\$2,680/\mathrm{MW\,h}$in a matter of a few minutes [3]. Similarly, the summer price spiked 32 fold from an average of $\$25/\mathrm{MW\,h}$ to $\$800/\mathrm{MW\,h}$ in July of 2015. Such events show how unforeseen and uncontrollable circumstances can greatly affect electricity prices that impact ISOs, suppliers, and customers. Energy industry experts are now considering the concept that extreme weather, more renewables and resulting electricity price volatility, could become the new norm.

Across the United States, electric utilities and ISOs are devoting increasing attention and resources to demand response (DR) [4]. Demand response is considered as a reliable means of mitigating the uncertainty and volatility of renewable generation and extreme weather conditions and improving the grid's efficiency and reliability. The potential demand response resource contribution from all U.S. demand response programs is estimated to be nearly 72,000 megawatts (MW), or about 9.2 percent of U.S. peak demand [5] making DR the largest virtual generator in the U.S. national grid. The annual revenue to end-users from DR markets with PJM ISO alone is more than \$700 million [6]. Global DR revenue is expected to reach nearly \$40 billion from 2014 through 2023 [7].

The volatility in real-time electricity prices poses the biggest operational and financial risk for large scale end-users of electricity such as large commercial buildings, industries and institutions [8]; often referred to as *C/I/I* consumers. In order to shield themselves from the volatility and risk of high prices, such consumers must be more flexible in their electricity demand. Consequently, large *C/I/I* customers are increasingly looking to demand response programs to help manage their electricity costs.

DR programs involve a voluntary response of a building to a price signal or a load curtailment request from the utility or the curtailment service provider (CSP). Upon successfully meeting the required curtailment level the end-users are financially rewarded, but may also incur penalties for under-performing and not meeting a required level of load curtailment. On the surface demand response may seem simple. Reduce your power when asked to and get paid. However, in practice, one of the biggest challenges with end-user demand response for large scale consumers of electricity is the following: *Upon receiving the notification for a DR event, what actions must the end-user take in order to achieve an adequate and a sustained DR curtailment?* This is a hard question to answer because of the following reasons:

1. **Modeling complexity and heterogeneity**: Unlike the automobile or the aircraft industry, each building is designed and used in a different way and therefore, it must be uniquely modeled. Learning predictive models of building's dynamics using first principles based approaches (e.g., with EnergyPlus [9]) is very cost and time prohibitive and requires retrofitting the building with several sensors [10]; The user expertise, time, and associated sensor costs required to develop a model of a single building is very high. This is because usually a building modeling domain expert typically uses a software tool to create the geometry of a building from the building design and equipment layout plans, add detailed information about material properties, about equipment and operational schedules. There is always a gap between the modeled and the real building and the domain expert must then manually tune the model to match the measured data from the building [11].

2. **Limitations of rule-based DR**: The building's operating conditions, internal thermal disturbances and environmental conditions must all be taken into account to make appropriate DR control decisions, which is not possible with using rule-based and pre-determined DR strategies since they do not account for the state of the building but are instead based on best practices and rules of thumb. As shown in Fig. 1(a), the performance of a rule-based DR strategy is inconsistent and can lead to reduced amount of curtailment which could result in penalties to the end-user. In our work, we show how a data-driven DR algorithm outperforms a rule-based strategy by 17% while accounting for thermal comfort. Rule based DR strategies have the advantage of being

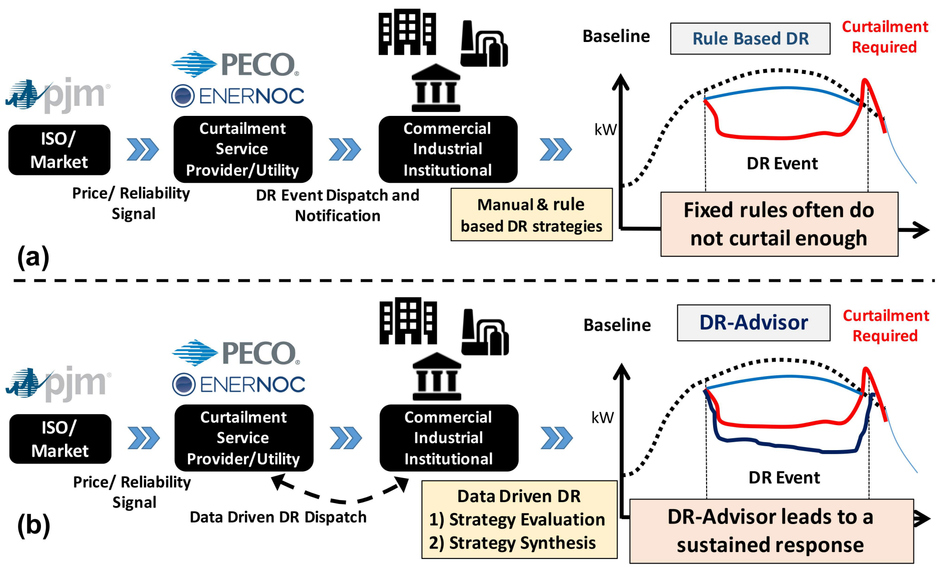

Figure 1: Majority of DR today is manual and rule-based. (a) The fixed rule based DR is inconsistent and could under-perform compared to the required curtailment, resulting in DR penalties. (b) Using data-driven models DR-Advisor uses DR strategy evaluation and DR strategy synthesis for a sustained and sufficient curtailment.

simple but they do not account for the state of the building and weather conditions during a DR event. Despite this lack of predictability, rule-based DR strategies account for the majority of DR approaches.

3. **Control complexity and scalability**: Upon receiving a notification for a DR event, the building's facilities manager must determine an appropriate DR strategy to achieve the required load curtailment. These control strategies can include adjusting zone temperature set-points, supply air temperature and chilled water temperature set-point, dimming or turning off lights, decreasing duct static pressure set-points and restricting the supply fan operation etc.. In a large building, it is difficult to asses the effect of one control action on other sub-systems and on the building's overall power consumption because the building sub-systems are tightly coupled. Consider the case of the University of Pennsylvania's campus, which has over a hundred different buildings and centralized chiller plants. In order to perform campus wide DR, the facilities manager must account for several hundred thousand set-points and their impact on the different buildings. Therefore, it is extremely difficult for a human operator to accurately gauge the building's or a campus's response.

4. **Interpretability of modeling and control**: Predictive models for buildings, regardless how sophisticated, lose their effectiveness unless they can be interpreted by human experts and facilities managers in the field. For e.g., artificial neural networks (ANN) obscure physical control knobs and interactions and hence, are difficult to interpret by building facilities managers. Therefore, the required solution must be transparent, human centric and highly interpretable.

The goal with data-driven methods for energy systems is to make the best of both worlds; i.e. simplicity of rule based approaches and the predictive capability of model based strategies, but without the expense of first principle or grey-box model development.

In this paper, we present a method called DR-Advisor (Demand Response-Advisor), which acts as a recommender system for the building's facilities manager and provides the power consumption prediction and control actions for meeting the required load curtailment and maximizing the economic reward. Using historical meter and weather data along with set-point and schedule information, DR-Advisor builds a family of interpretable regression trees to learn non-parametric data-driven models for predicting the power consumption of the building (Figure 2). DR-Advisor can be used for real-time demand response baseline prediction, strategy evaluation and control synthesis, without having to learn first principles based models of the building.

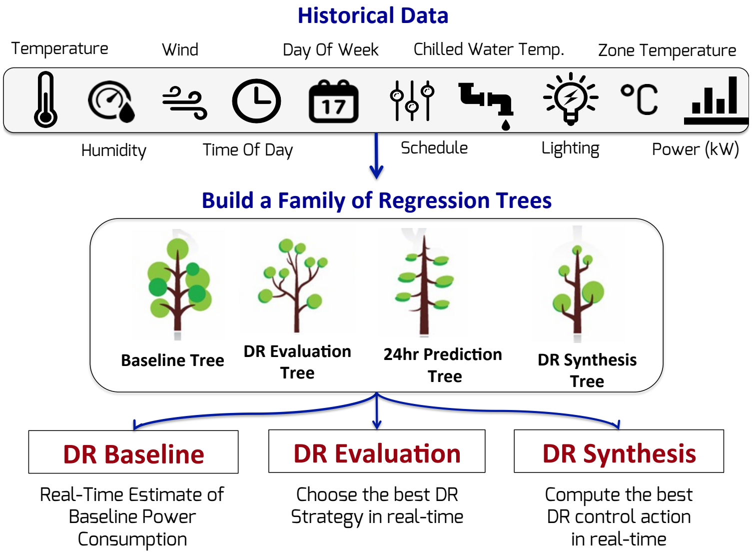

Figure 2: DR-Advisor Architecture

1.1. Contributions

This work has the following data-driven contributions:

1. **DR Baseline Prediction:** We demonstrate the benefit of using regression trees based approaches for estimating the demand response baseline power consumption. Using regression tree-based algorithms eliminates the cost of time and effort required to build and tune first principles based models of buildings for DR. DR-Advisor achieves a prediction accuracy of 92.8% to 98.9% for baseline estimates of eight buildings on the Penn campus.

2. **DR Strategy Evaluation:** We present an approach for building auto-regressive trees and apply it for demand response strategy evaluation. Our models takes into account the state of the building and weather forecasts to help choose the best DR strategy among several pre-determined strategies.

3. **DR Control Synthesis:** We introduce a novel model based control with regression trees (mbCRT) algorithm to enable control with regression trees use it for real-time DR synthesis. Using the mbCRT algorithm, we can optimally trade off thermal comfort inside the building against the amount of load curtailment. While regression trees are a popular choice for prediction based models, this is the first time regression tree based algorithms have been used for controller synthesis with applications in demand response. Our synthesis algorithm outperforms rule based DR strategy by 17% while maintaining bounds on thermal comfort inside the building.

1.2. Experimental validation and evaluation

We evaluate the performance of DR-Advisor using a mix of real data from 8 buildings on the campus of the University of Pennsylvania, in Philadelphia USA and data-sets from a virtual building test-bed for the Department of Energy's (DoE) large commercial reference building. We also compare the performance of DR-Advisor against other data-driven methods using a bench-marking data-set from AHRAE's great energy predictor shootout challenge.

This paper is organized as follows: Section 2 describes the challenges with demand response. In Section 3, we present how data-driven algorithms can be used for the problems associated with DR. Section 4, presents a new algorithm to perform control with regression trees for synthesizing demand response strategies. Section 6 describes the MATLAB based DR-Advisor toolbox. Section 7 presents a comprehensive case study with DR-Advisor using data from several real buildings. In Section 8, a detailed survey of related work has been presented. We conclude this paper in Section 9 with a summary of our results and a discussion about future directions.

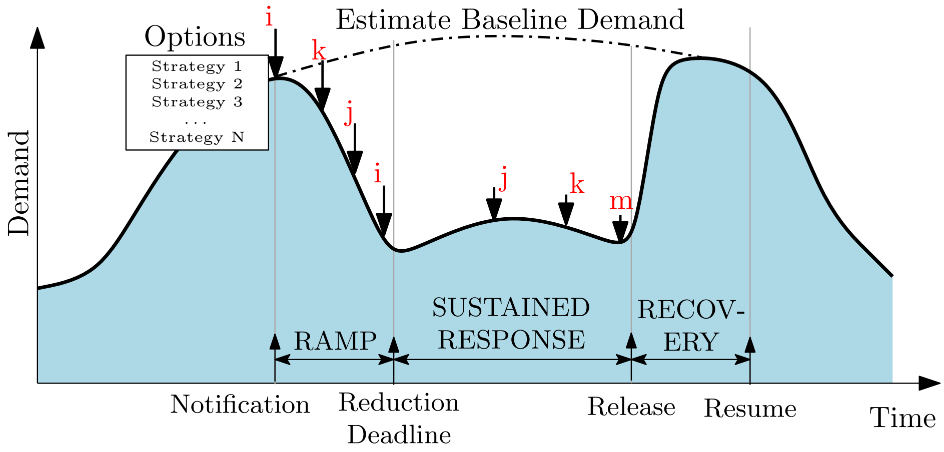

Figure 3: Example of a demand response timeline.

2. Problem definition

The timeline of a DR event is shown in Figure 3. An *event notification* is issued by the utility/CSP, at the notification time (~30mins). The time by which the reduction must be achieved, is the *reduction deadline*. The main period during which the demand needs to be curtailed is the *sustained response period* (1~6hrs). The end of the response period is when the main curtailment is released. The normal operation is gradually resumed during the *recovery period*. The DR event ends at the end of the recovery period.

The key to answering the question of what actions to take to achieve a significant DR curtailment upon receiving a notification, lies in making accurate predictions about the power consumption response of the building. Specifically, it involves solving the three challenging problems of end-user demand response, which are described next.

2.1. DR baseline prediction

The DR baseline is an estimate of the electricity that would have been consumed by a customer in the absence of a demand response event (as shown in In Fig. 3) The measurement and verification of the demand response baseline is the most critical component of any DR program since the amount of DR curtailment, and any associated financial reward can only be determined with respect to the baseline estimate. The goal is to learn a predictive model $g()$ which relates the baseline power consumption estimate $\hat{Y}_{base}$ to the forecast of the weather conditions and building schedule for the duration of the DR-event i.e., $\hat{Y}_{base} = g(\text{weather, schedule})$

2.2. DR strategy evaluation

Most DR today is manual and conducted using fixed rules and pre-determined curtailment strategies based on recommended guidelines, experience and best practices. During a DR event, the building's facilities manager must choose a single strategy among several pre-determined strategies to achieve the required power curtailment. Each strategy includes adjusting several control knobs such as temperature set-points, lighting levels and temporarily switching off equipment and plug loads to different levels across different time intervals.

As only one strategy can be used at a time, the question then is, *how to choose the DR strategy from a pre-determined set of strategies which leads to the largest load curtailment?*

Instead of predicting the baseline power consumption $\hat{Y}_{base}$, in this case we want the ability to predict the actual response of the building $\hat{Y}_{kW}$ due to any given strategy. For example, in Fig. 3, there are N different strategies available to choose from. DR-Advisor predicts the power consumption of the building due to each strategy and chooses the DR strategy ($\in \{i, j, \cdots k \cdots N\}$) which leads to the largest load curtailment subject to the constraints on the thermal comfort and set-points. The resulting strategy could be a combination of switching between the available set of strategies.

2.3. DR strategy synthesis

Instead of choosing a DR strategy from a pre-determined set of strategies, a harder challenge is to synthesize new DR strategies and obtain optimal operating points for the different control variables. We can cast this problem as an optimization over the set of control variables, $\mathbb{X}_c$, such that

$$\begin{aligned} \underset{\mathbb{X}_c}{\text{minimize}} \quad & f(\hat{Y_{kW}}) \\ \text{subject to} \quad & \hat{Y_{kW}} = h(\mathbb{X}_c) \\ & \mathbb{X}_c \in \mathbb{X}_{safe} \end{aligned} \tag{1}$$

we want to minimize the predicted power response of the building $\hat{Y_{kW}}$, subject to a predictive model which relates the response to the control variables and subject to the constraints on the control variables.

Unlike rule-base DR, which does not account for building state and external factors, in DR synthesis the optimal control actions are derived based on the current state of the building, forecast of outside weather and electricity prices.

3. Data-Driven Demand Response

Our goal is to find data-driven functional models that relates the value of the response variable, say power consumption, $\hat{Y_{kW}}$ with the values of the predictor variables or features $[X_1, X_2, \cdots, X_m]$ which can include weather data, set-point information and building schedules. When the data has lots of features, as is the case in large buildings, which interact in complicated, nonlinear ways, assembling a single global model, such as linear or polynomial regression, can be difficult, and lead to poor response predictions. An approach to non-linear regression is to partition the data space into smaller regions, where the interactions are more manageable. We then partition the partitions again; this is called recursive partitioning, until finally we get to chunks of the data space which are so tame that we can fit simple models to them. Regression trees is an example of an algorithm which belongs to the class of recursive partitioning algorithms. The seminal algorithm for learning regression trees is CART as described in [12].

Regression trees based approaches are our choice of data-driven models for DR-Advisor. The primary reason for this modeling choice is that regression trees are highly interpretable, by design. Interpretability is a fundamental desirable quality in any predictive model. Complex predictive models like neural-networks , support vector regression etc. go through a long calculation routine and involve too many factors. It is not easy for a human engineer to judge if the operation/decision is correct or not or how it was generated in the first place. Building operators are used to operating a system with fixed logic and rules. They tend to prefer models that are more transparent, where it is clear exactly which factors were used to make a particular prediction. At each node in a regression tree a simple, if this then that, human readable, plain text rule is applied to generate a prediction at the leafs, which anyone can easily understand and interpret. Making machine learning algorithms more interpretable is an active area of research [13], one that is essential for incorporating human centric models in demand response for energy systems.

3.1. Data-Description

In order to build regression trees which can predict the power consumption of the building, we need to train on time-stamped historical data. As shown in Fig. 2, the data that we use can be divided into three different categories as described below:

1. **Weather Data:** It includes measurements of the outside dry-bulb and wet-bulb air temperature, relative humidity, wind characteristics and solar irradiation at the building site.

2. **Schedule data:** We create *proxy* variables which correlate with repeated patterns of electricity consumption e.g., due to occupancy or equipment schedules. *Day of Week* is a categorical predictor which takes values from $1 - 7$ depending on the day of the week. This variable can capture any power consumption patterns which occur on specific days of the week. For instance, there could a big auditorium in an office building which is only used on certain days. Likewise, *Time of Day* is quite an important predictor of powe consumption as it can adequately capture daily patterns of occupancy, lighting and appliance use without directly measuring any one of them. Besides using proxy schedule predictors, actual building equipment schedules can also be used as training data for building the trees.

3. **Building data:** The state of the building is required for DR strategy evaluation and synthesis. This includes (i) Chilled Water Supply Temperature (ii) Hot Water Supply Temperature (iii) Zone Air Temperature (iv) Supply Air Temperature (v) Lighting levels.

3.2. Data-Driven DR Baseline

DR-Advisor uses a mix of several algorithms to learn a reliable baseline prediction model. For each algorithm, we train the model on historical power consumption data and then validate the predictive capability of the model against a test data-set which the model has never seen before. In addition to building a single regression tree, we also learn cross-validated regression trees, boosted regression trees (BRT) and random forests (RF). The ensemble methods like BRT and RF help in reducing any over-fitting over the training data. They achieve this by combining the predictions of several base estimators built with a given learning algorithm in order to improve generalizability and robustness over a single estimator. For a more comprehensive review of random forests we refer the reader to [14]. A boosted regression tree (BRT) model is an additive regression model in which individual terms are simple trees, fitted in a forward, stage-wise fashion [15].

3.3. Data-Driven DR Evaluation

The regression tree models for DR evaluation are similar to the models used for DR baseline estimation except for two key differences: First, instead of only using weather and proxy variables as the training features, in DR evaluation, we also train on set-point schedules and data from the building itself to capture the influence of the state of the building on its power consumption; and Second, in order to predict the power consumption of the building for the entire length of the DR event, we use the notion of auto-regressive trees. An auto-regressive tree model is a regular regression tree except that the lagged values of the response variable are also predictor variables for the regression tree i.e., the tree structure is learned to approximate the following function:

$$\hat{Y_{kW}}(t) = f([X_1, X_2, \cdots, X_m, Y_{kW}(t-1), \cdots, Y_{kW}(t-\delta)]) \tag{2}$$

where the predicted power consumption response $\hat{Y_{kW}}$ at time t, depends on previous values of the response itself $[Y_{kW}(t-1), \cdots, Y_{kW}(t-\delta)]$ and δ is the order of the auto-regression. This allows us to make finite horizon predictions of power consumption for the building. At the beginning of the DR event we use the auto-regressive tree for predicting the response of the building due to each rule-based strategy and choose the one which performs the best over the predicted horizon. The prediction and strategy evaluation is re-computed periodically throughout the event.

4. DR synthesis with regression trees

The data-driven methods described so far use the forecast of features to obtain building power consumption predictions for DR baseline and DR strategy evaluation. In this section, we extend the theory of regression trees to solve the demand response synthesis problem described earlier in Section 2.3. This is our primary contribution.

Recall that the objective of learning a regression tree is to learn a model f for predicting the response Y with the values of the predictor variables or features $X_1, X_2, \cdots, X_m$; i.e., $Y = f([X_1, X_2, \cdots, X_m])$ Given a forecast of the features $\hat{X_1}, \hat{X_2}, \cdots, \hat{X_m}$ we can predict the response $\hat{Y}$. Now consider the case where a subset, $\mathbb{X}_c \subset \mathbb{X}$ of the set of features/variables $\mathbb{X}$'s are manipulated variables i.e., we can change their values in order to drive the response $(\hat{Y})$ towards a certain value. In the case of buildings, the set of variables can be separated into disturbances (or non-manipulated) variables like outside air temperature, humidity, wind etc. while the controllable (or manipulated) variables would be the temperature and lighting set-points within the building. Our goal is to modify the regression trees and make them suitable for synthesizing the optimal values of the control variables in real-time.

4.1. Model-based control with regression trees

The key idea in enabling control synthesis for regression trees is in the separation of features/variables into manipulated and non-manipulated features. Let $\mathbb{X}_c \subset \mathbb{X}$ denote the set of manipulated variables and $\mathbb{X}_d \subset \mathbb{X}$ denote the set of disturbances/ non-manipulated variables such that $\mathbb{X}_c \cup \mathbb{X}_d \equiv \mathbb{X}$. Using this separation of variables we build upon the idea of simple model based regression trees [16, 17] to *model based control with regression trees (mbCRT).*

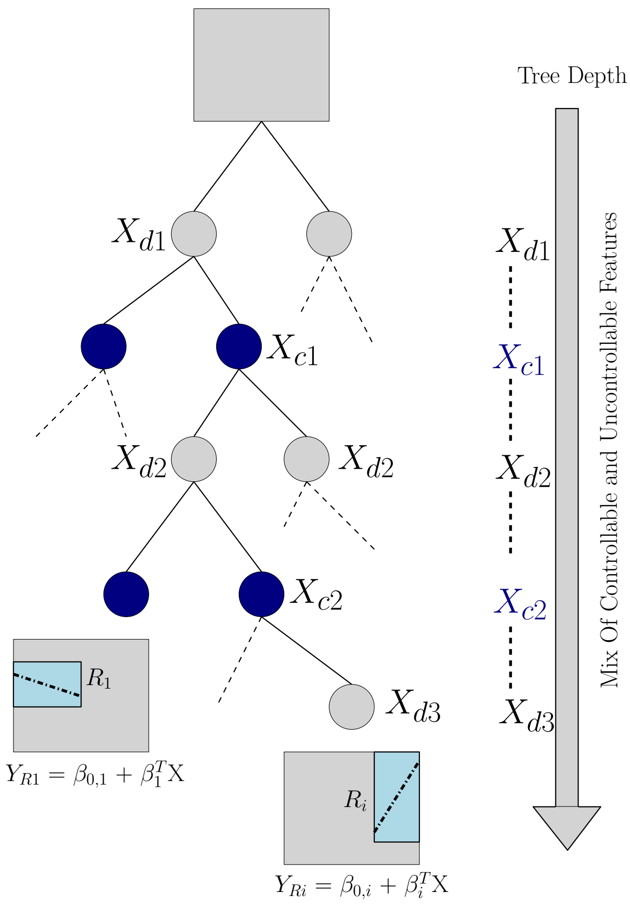

Figure 4: Example of a regression tree with linear regression model in leaves. Not suitable for control due to the mixed order of the controllable X_c (solid blue) and uncontrollable X_d features.

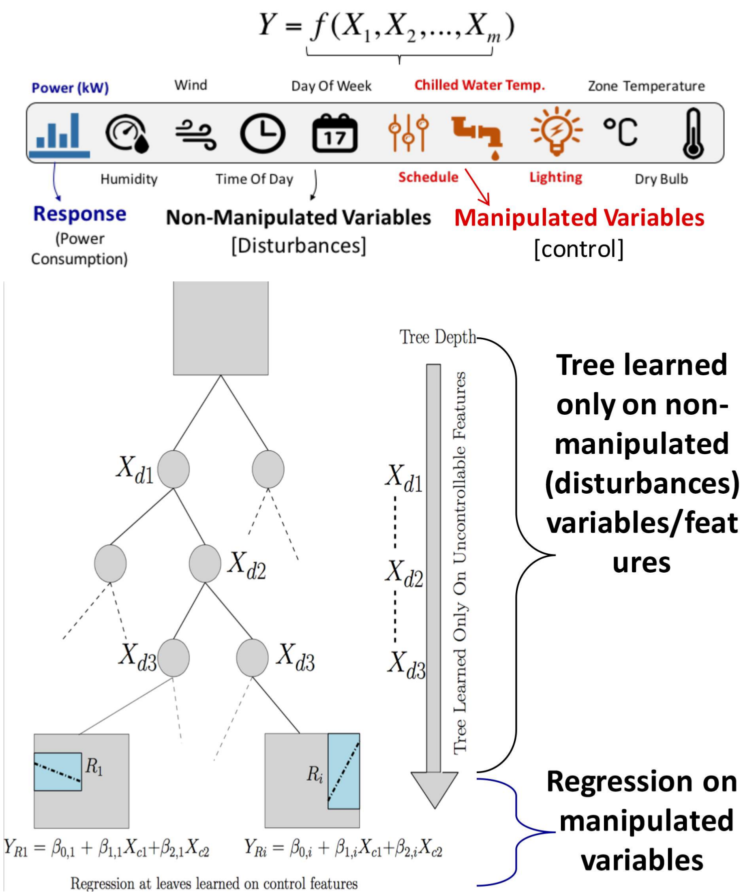

Figure 5: Example of a tree structure obtained using the mbCRT algorithm. The separation of variables allows using the linear model in the leaf to use only control variables.

Figure 4 shows an example of how manipulated and non-manipulated features can get distributed at different depths of model based regression tree which uses a linear regression function in the leaves of the tree:

$$\hat{Y_{Ri}} = \beta_{0,i} + \beta_i^T \mathbb{X} \tag{3}$$

Where $\hat{Y_{Ri}}$ is the predicted response in region R_i of the tree using all the features $\mathbb{X}$. In such a tree the prediction can only be obtained if the values of all the features X's is known, including the values of the control variables X_{ci}'s. Since the manipulated and non-manipulated variables appear in a mixed order in the tree depth, we cannot use this tree for control synthesis. This is because the value of the control variables X_{ci}'s is unknown, one cannot navigate to any single region using the forecasts of disturbances alone.

The mbCRT algorithm avoids this problem using a simple but clever idea. We still partition the entire data space into regions using CART algorithm, but the top part of the regression tree is learned only on the non-manipulated features $\mathbb{X}_d$ or disturbances as opposed to all the features $\mathbb{X}$ (Figure 5) In every region at the leaves of the "disturbance" tree a linear model is fit but only on the control variables $\mathbb{X}_c$:

$$Y_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X}_c \tag{4}$$

Separation of variables allows us to use the forecast of the disturbances $\hat{\mathbb{X}_d}$ to navigate to the appropriate region R_i and use the linear regression model ($Y_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X}_c$) with only the control/manipulated features in it as the valid prediction model for that time-step.

Algorithm 1 mbCRT: Model Based Control With Regression Trees

```
DESIGN TIME
procedure MODEL TRAINING
    Separation of Variables
    Set X_c ← non-manipulated features
    Set X_d ← manipulated features
    Build the power prediction tree T_kW with X_d
    for all Regions R_i at the leaves of T_kW do
        Fit linear model kŴ_Ri = β_0,i + β_i^T X_c
        Build q temperature trees T1, T2 ··· Tq with X_d
    end for
    for all Regions R_i at the leaves of Ti do
        Fit linear model T̂i = β_0,i + β_i^T X_c
    end for
end procedure
RUN TIME
procedure CONTROL SYNTHESIS
    At time t obtain forecast X̂_d(t + 1) of disturbances X̂_d1(t + 1), X̂_d2(t + 1), ···
    Using X̂_d(t + 1) determine the leaf and region R_rt for each tree.
    Obtain the linear model at the leaf of each tree.
    Solve optimization in Eq5 for optimal control action X_c^*(t)
end procedure
```

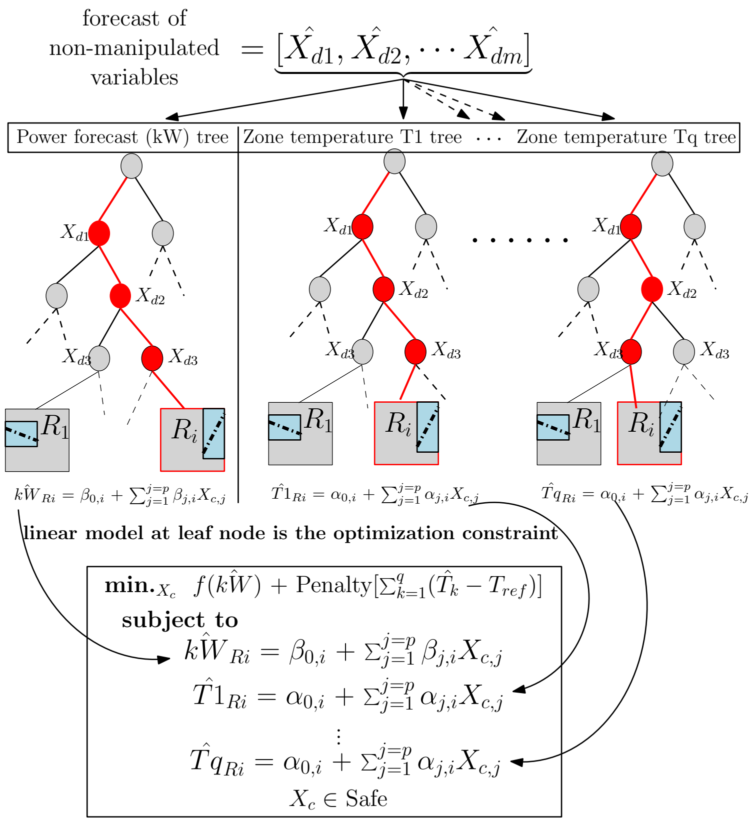

Figure 6: DR synthesis with thermal comfort constraints. Each tree is responsible for contributing one constraint to the demand response optimization.

4.2. DR synthesis optimization

In the case of DR synthesis for buildings, the response variable is power consumption, the objective function can denote the financial reward of minimizing the power consumption during the DR event. However, the curtailment must not result in high levels of discomfort for the building occupants. In order to account for thermal comfort, in addition to learning the tree for power consumption forecast, we can also learn different trees to predict the temperature of different zones in the building. As shown in Figure 6 and Algorithm 1, at each time-step during the DR event, a forecast of the non manipulated variables is used by each tree, to navigate to the appropriate leaf node. For the power forecast tree, the linear model at the leaf node relates the predicted power consumption of the building to the manipulated/control variables i.e., $k\hat{W} = \beta_{0,i} + \beta_i^T \mathbb{X}_c$.

Similarly, for each zone $1, 2, \cdots q$, a tree is built whose response variable is the zone temperature Ti. The linear model at the leaf node of each of the zone temperature tree relates the predicted zone temperature to the manipulated variables $\hat{T}i = \alpha_{0,j} + \beta_j^T \mathbb{X}_c$. Therefore, at every time-step, based on the forecast of the non-manipulated variables, we obtain $q + 1$ linear models between the power consumption and q zone temperatures and the manipulated variables. We can then solve the following DR synthesis optimization problem to obtain the values of the manipulated variables $\mathbb{X}_c$:

$$
\begin{aligned}
& \underset{\mathbb{X}_c}{\text{minimize}} f(k\hat{W}) + \text{Penalty}[\sum_{k=1}^{q} (\hat{T}_k - T_{ref})] \\
& \text{subject to} \\
& \qquad k\hat{W} = \beta_{0,i} + \beta_i^T \mathbb{X}_c \\
& \qquad \hat{T}1 = \alpha_{0,1} + \beta_1^T \mathbb{X}_c \\
& \qquad \cdots \\
& \qquad \hat{T}d = \alpha_{0,q} + \beta_q^T \mathbb{X}_c \\
& \qquad \mathbb{X}_c \in \mathbb{X}_{safe}
\end{aligned}
\tag{5}
$$

The linear model between the response variable Y_{Ri} and the control features $\mathbb{X}_c$ is assumed for computational simplicity. Other models could also be used at the leaves as long as they adhere to the separation of variables principle. Figure 7 shows that the linear model assumption in the leaves of the tree is a valid assumption.

The intuition behind the mbCRT Algorithm 1 is that at run time t, we use the forecast $\hat{\mathbb{X}}_d(t + 1)$ of the disturbance features to determine the region of the *uncontrollable* tree and hence, the linear model to be used for the control. We then solve the simple linear program corresponding to that region to obtain the optimal values of the control variables.

The mbCRT algorithm is the first ever algorithm which allows the use of regression trees for control synthesis.

5. The case for using regression trees for demand response

Trees share the advantage of being a simple approach, much like other data-driven approaches. However, they offer several other advantages in addition to being interpretable, which make them suitable for solving the challenges of demand response discussed in Section 2. We list some of these advantages here:

1. **Fast computation times:** Trees require very low computation power, both running time and storage requirements. With N observations and p predictors trees require $pNlogN$ operations for an initial sort for each predictor, and typically another $pNlogN$ operations for the split computations. If the splits occurred near the edges of the predictor ranges, this number can increase to N^2p. Once the tree is built, the time to make predictions is extremely fast since obtaining a response prediction is simply a matter of traversing the tree with fixed rules at every node. For fast demand response, where the price of electricity could change several times within a few minutes, trees can provide very fast predictions.

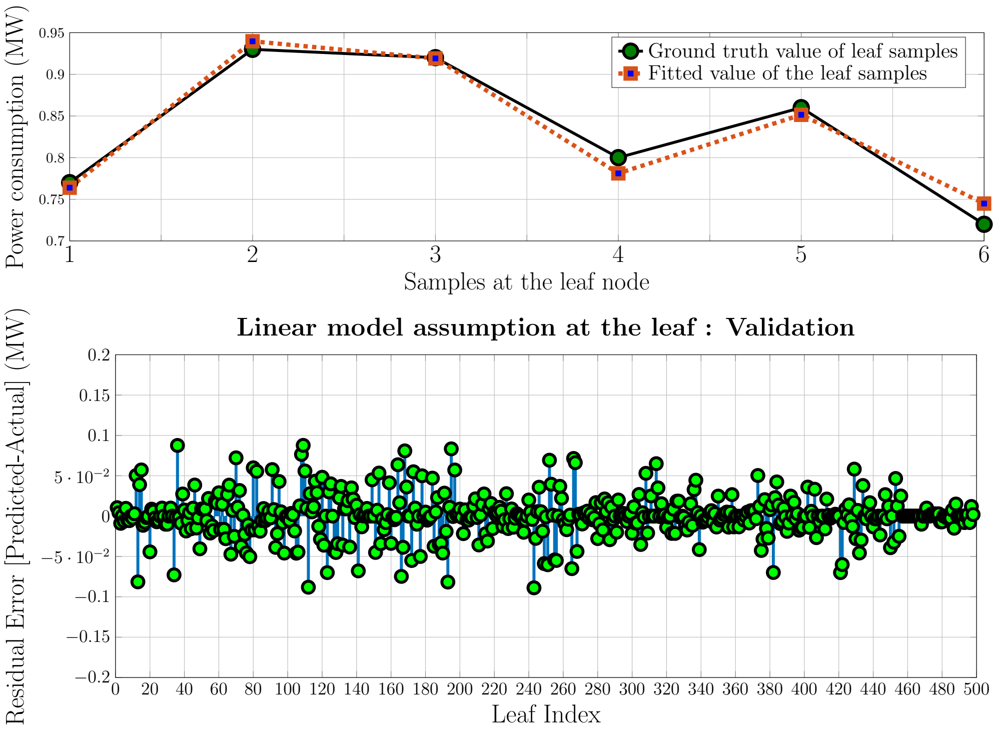

Figure 7: Linear model assumption at the leaves. The top figure shows the comparison between fitted values and ground truth values of power consumption for one of the leafs in the power consumption prediction tree. The bottom figure shows the residual error between fitted and actual power consumption values for all the leaf nodes of the tree.

2. **Handle a lot of data and variables:** Trees can easily handle the case where the data has lots of features which interact in complicated and nonlinear ways. In the context of buildings, a mix of weather data, schedule information, set-points, power consumption data is used and the number of predictor variables can increase very quickly. A large number of features and a large volume of data can become too overwhelming for global models, like regression, to adequately explain. For trees, the predictor variables themselves can be of any combination of continuous, discrete and categorical variables.

3. **Handle Missing Data:** Sometimes, data has missing predictor values in some or all of the predictor variables. This is especially true for buildings, where sensor data streams fail frequently due to faulty sensors or faulty communication links. One approach is to discard any observation with some missing values, but this could lead to serious depletion of the training set. Alternatively, the missing values could be imputed (filled in), with say the mean of that predictor over the non-missing observations. For tree-based models, there are two better approaches. The first is applicable to categorical predictors: we simply make a new category for "missing". From this we might discover that observations with missing values for some measurement behave differently than those with non-missing values. The second more general approach is the construction of surrogate variables. When considering a predictor for a split, we use only the observations for which that predictor is not missing. Having chosen the best (primary) predictor and split point, we build a list of surrogate predictors and split points. The first surrogate is the predictor and corresponding split point that best mimics the split of the training data achieved by the primary split. The second surrogate is the predictor and corresponding split point that does second best, and so on. When sending observations down the tree either in the training phase or during prediction, we use the surrogate splits in order, if the primary splitting predictor is missing.

4. **Robust to outliers:** Tree based models are generally not affected by outliers but regression based models are. The intuitive reasoning behind this is that during the construction of the tree the region of the data with outliers is likely to be partitioned in a separate region.

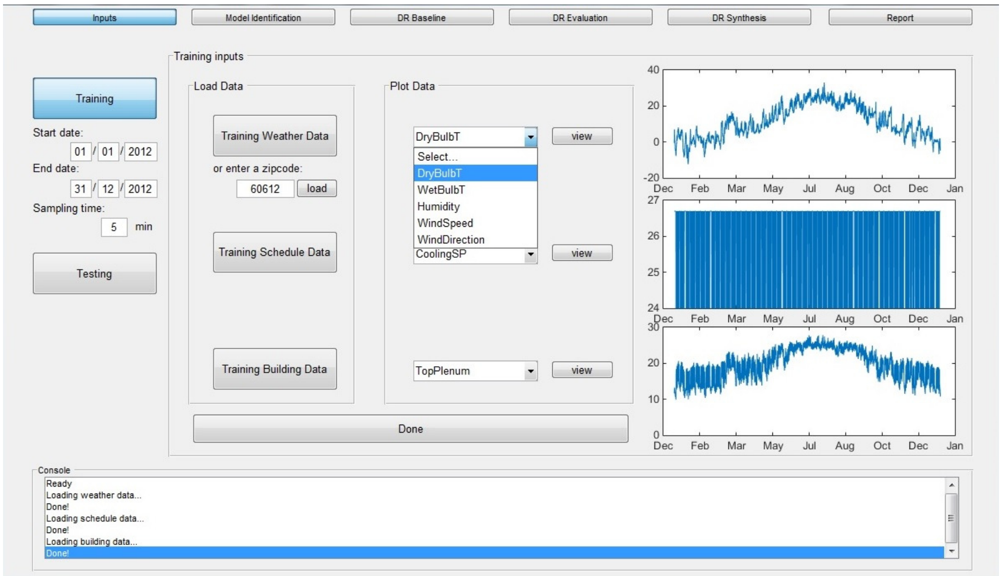

Figure 8: Screenshot of the DR-Advisor MATLAB based GUI.

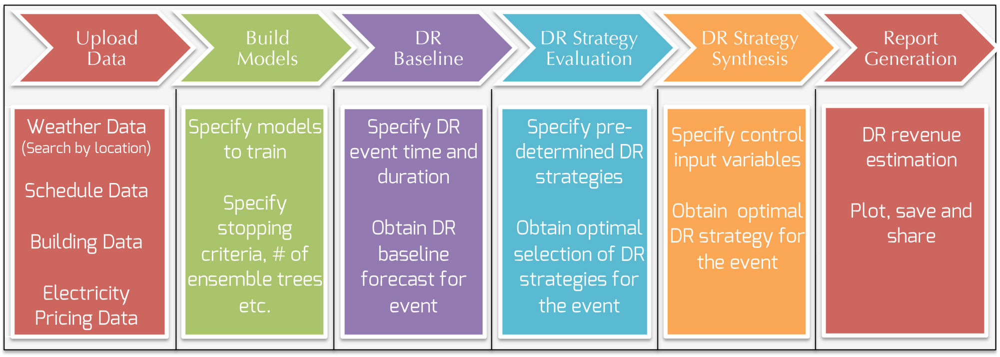

Figure 9: DR-Advisor Workflow

6. DR-Advisor:Toolbox design

The algorithms described thus far, have been implemented into a MATLAB based tool called DR-Advisor. We have also developed a graphical user interface (GUI) for the tool (Figure 8) to make it user-friendly.

Starting from just building power consumption and temperature data, the user can leverage all the features of DR-Advisor and use it to solve the different DR challenges. The toolbox design follows a simple and efficient workflow as shown in Figure 9. Each step in the workflow is associated with a specific tab in the GUI. The workflow is divided into the following steps:

1. **Upload Data:** When the toolbox loads, the Input tab of the GUI (Figure 8) is displayed. Here the user can upload and specify any sensor data from the building which could be correlated to the power consumption. This includes historical power consumption data, any known building operation schedules and zone temperature data. The tool is also equipped with the capability to pull historical weather data for a building location from the web. The user can also specify or upload electricity pricing or utility tariff data. Once the upload process is complete the data structure for learning the different tree based models is created internally. The GUI also has a small console which is used to display progress, completion and alert messages for each action in the upload process.

2. **Build Models:** In the next step of the workflow, the user can specify which tree-based models should be learned as shown in Figure 10. These include, a single regression tree (SRT), cross-validated regression tree (CV-RT), random forest (RF), boosted regression tree (BRT) and M5 model based regression tree (M5). For each method the user may change the parameters of the training process from the default values. These parameters include the stopping criteria in terms of `MinLeaf` or the number of trees in the ensemble and the value for the number of folds in cross validation. After the models have been trained, the normalized root mean square value for each method on the test data is displayed. The user can also visualize and compare the predicted output vs the ground truth data for the different methods. For the ensemble methods, the convergence of the resubstitution error and the feature importance plots can also be viewed.

3. **DR Baseline:** In the DR baseline tab, the user can specify the start and end times for a DR event and DR-Advisor generates the baseline prediction for that duration using the methods selected during the model identification. The user can also specify if the baseline uses only weather data or it uses weather plus building schedule data.

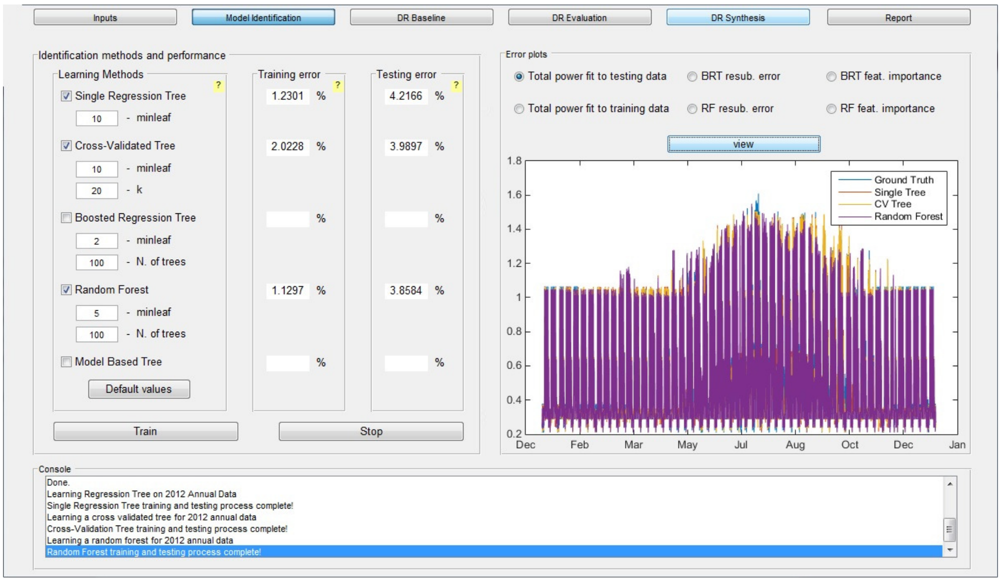

Figure 10: DRAdvisor model identification tab

4. **DR Strategy Evaluation:** In this step of the workflow, the user first has to specify the pre-determined DR strategies which need to be evaluated during the DR event. The user can choose different control variables and specify their value for the duration of the DR event. A group of such control variables constitute the DR strategy. The user may specify several DR strategies, in which different combinations of the control variables take different values. Upon executing the DR evaluation process, DR-Advisor, is capable of selecting the best set of strategies for the DR event based on load curtailment.

5. **DR Strategy Synthesis:** For DR synthesis, two inputs are required: the user needs to provide an electricity/DR rate structure and the user needs to specify which of the variables are the control variables. DR-Advisor then uses the mbCRT (Section 4.1) algorithm to synthesize and recommend a DR strategy for the DR event by assigning suitable values to the control inputs.

6. **Report Generation:** Facilities managers need to log reports of the building's operation during the DR event. DR-Advisor can generate summarized reports of how much load for curtailed and the estimated revenue earned from the DR event. The report also includes plots of what control actions were recommended by DR-Advisor and the comparison between the estimated baseline power consumption and the actual load during the event.

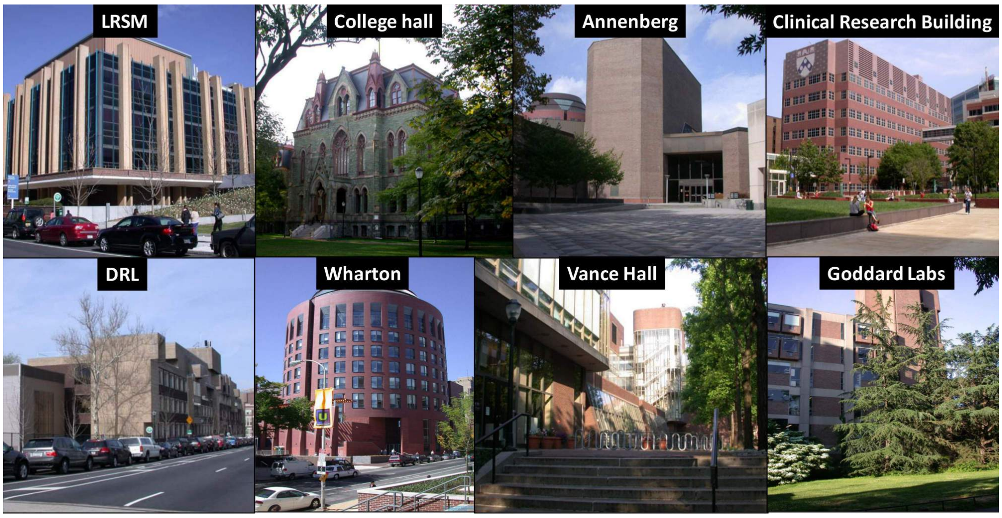

Figure 11: 8 different buildings on Penn campus were modeled with DR-Advisor

7. Case study

DR-Advisor has been developed into a MATLAB toolbox available at `http://mlab.seas.upenn.edu/dr-advisor/`. In this section, we present a comprehensive case study to show how DR-Advisor can be used to address all the aforementioned demand response challenges (Section 2) and we compare the performance of our tool with other data-driven methods.

7.1. Building description

We use historical weather and power consumption data from 8 buildings on the Penn campus (Figure 11). These buildings are a mix of scientific research labs, administrative buildings, office buildings with lecture halls and bio-medical research facilities. The total floor area of the eight buildings is over 1.2 million square feet spanned across. The size of each building is shown in Table 1.

We also use the DoE Commercial Reference Building (DoE CRB) simulated in EnergyPlus [18] as the virtual test-bed building. This is a large 12 story office building consisting of 73 zones with a total area of 500,000 sq ft. There are 2,397 people in the building during peak occupancy. During peak load conditions the building can consume up to 1.6 MW of power. For the simulation of the DoE CRB building we use actual meteorological year data from Chicago for the years 2012 and 2013. On July 17, 2013, there was a DR event on the PJM ISO grid from 15:00 to 16:00 hrs. We simulated the DR event for the same interval for the virtual test-bed building.

7.2. Model Validation

For each of the Penn buildings, multiple regression trees were trained on weather and power consumption data from August 2013 to December 2014. Only the weather forecasts and proxy variables were used to train the models. We then use the DR-Advisor to predict the power consumption in the test period i.e., for several months in 2015. The predictions are obtained for each hour, making it equivalent to baseline power consumption estimate. The predictions on the test-set are compared to the actual power consumption of the building during the test-set period. One such comparison for the clinical reference building is shown in Figure 12. The following algorithms were evaluated: single regression tree, k-fold cross validated (CV) trees, boosted regression trees (BRT) and random forests (RF). Our chosen metric of prediction accuracy is the one minus the normalized root mean square error (NRMSE). NRMSE is the RMSE divided by the mean of the data. The accuracy of the model of all the eight buildings is summarized in Table 1. We notice that DR-Advisor performs quite well and the accuracy of the baseline model is between 92.8% to 98.9% for all the buildings.

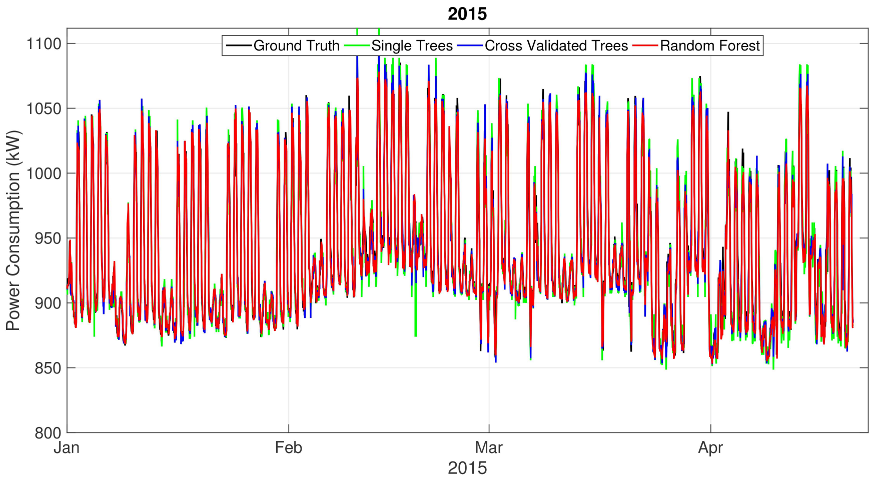

Figure 12: Model validation for the clinical research building at Penn.

Table 1: Model validation with Penn data

Building Name	Total Area (sq-ft)	Floors	Accuracy (%)
LRSM	92,507	6	94.52
College Hall	110,266	6	96.40
Annenberg Center	107,200	5	93.75
Clinical Research Building	204,211	8	98.91
David Rittenhouse Labs	243,484	6	97.91
Huntsman Hall	320,000	9	95.03
Vance Hall	106,506	7	92.83
Goddard Labs	44,127	10	95.07

Table 2: ASHRAE Energy Prediction Competition Results

ASHRAE Team ID	WBE CV	CHW CV	HW CV	Average CV
9	10.36	13.02	15.24	12.87
DR-Advisor	**11.72**	**14.88**	**28.13**	**18.24**
6	11.78	12.97	30.63	18.46
3	12.79	12.78	30.98	18.85
2	11.89	13.69	31.65	19.08
7	13.81	13.63	30.57	19.34

7.3. Energy Prediction Benchmarking

We compare the performance of DR-Advisor with other data-driven method using a bench-marking data-set from the American Society of Heating, Refrigeration and Air Conditioning Engineers (ASHRAE's) Great Energy Predictor Shootout Challenge [19]. The goal of the ASHRAE challenge was to explore and evaluate data-driven models that may not have such a strong physical basis, yet that perform well at prediction. The competition attracted $\sim$ 150 entrants, who attempted to predict the unseen power loads from weather and solar radiation data using a variety of approaches. In addition to predicting the hourly whole building electricity consumption, WBE (kW), both the hourly chilled water, CHW (millions of Btu/hr) and hot water consumption, HW (millions of Btu/hr) of the building was also required to be a prediction output. Four months of training data with the following features was provided: (a) 1. Outside temperature (°F) 2. Wind speed (mph) 3. Humidity ratio (water/dry air) 4. Solar flux (W/m^2) In addition to these training features, we added three proxy variables of our own: hour of day, IsWeekend and IsHoliday to account for correlation of the building outputs with schedule.

Finally, we use different ensemble methods within DR-Advisor to learn models for predicting the three different building attributes. In the actual competition, the winners were selected based on the accuracy of all predictions as measured by the normalized root mean square error, also referred to as the coefficient of variation statistic CV. The smaller the value of CV, the better the prediction accuracy. ASHRAE released the results of the competition for the top 19 entries which they received. In Table 2, we list the performance of the top 5 winners of the competition and compare our results with them. It can be seen from table 2, that the random forest implementation in the DR-Advisor tool ranks 2^{nd} in terms of WBE CV and the overall average CV. The winner of the competition was an entry from David Mackay [20] which used a particular form of bayesian modeling using neural networks.

The result we obtain clearly demonstrates that the regression tree based approach within DR-Advisor can generate predictive performance that is comparable with the ASHRAE competition winners. Furthermore, since regression trees are much more interpretable than neural networks, their use for building electricity prediction is, indeed, very promising.

7.4. DR-Evaluation

We test the performance of 3 different rule based strategies shown in Fig. 13. Each strategy determines the set point schedules for chiller water, zone temperature and lighting during the DR event. These strategies were derived on the basis of automated DR guidelines provided by Siemens [21]. Chiller water set point is same in Strategy 1 (S1) and Strategy 3 (S3), higher than that in Strategy 2 (S2). Lighting level in S3 is higher than in S1 and S2.

We use auto-regressive trees (Section 3.3) with order, $\delta = 6$ to predict the power consumption for the entire duration (1 hour) at the start of DR Event. In addition to learning the tree for power consumption, additional auto-regressive trees are also built for predicting the zone temperatures of the building. At every time step, first the zone temperatures are predicted using the trees for temperature prediction. Then the power tree uses this temperature forecast along with lagged power consumption values to predict the power consumption recursively until the end of the prediction horizon.

Fig. 14 shows the power consumption prediction using the auto-regressive trees and the ground truth obtained by simulation of the DoE CRB virtual test-bed for each rule-based strategy. Based on the predicted response, in this case DR-Advisor chooses to deploy the strategy S1, since it leads to the least amount of electricity consumption. The predicted response due to the chosen strategy aligns well with the ground truth power consumption of the building

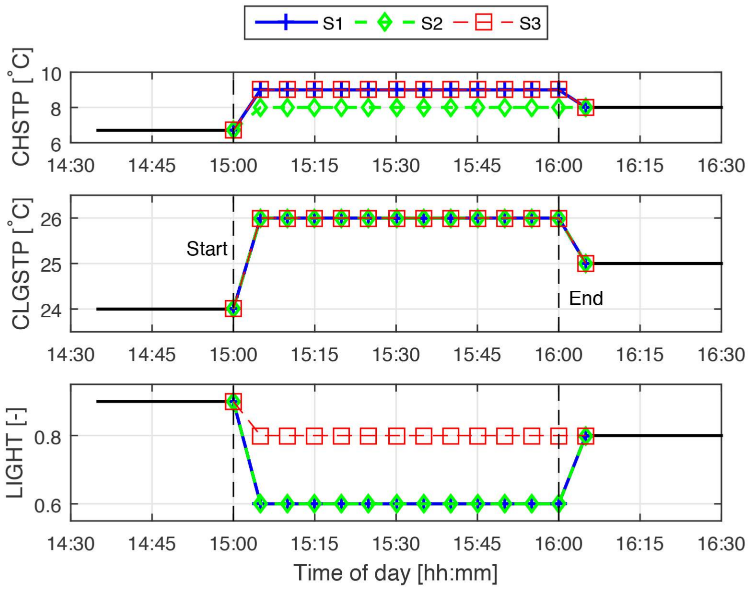

Figure 13: Rule-based strategies used in DR Evaluation. CHSTP denotes Chiller set point and CLGSTP denotes Zone Cooling temperature set point.

due to the same strategy, showing that DR strategy evaluation prediction of DR-Advisor is reliable and can be used to choose the best rule-based strategy from a set of pre-determined rule-based DR strategies.

7.5. DR-Synthesis

We now evaluate the performance of the mbCRT (Section 4.1) algorithm for real-time DR synthesis. Similar to DR evaluation, the regression tree is trained on weather, proxy features, set-point schedules and data from the building. We first partition the set of features into manipulated features (or control inputs) and non-manipulated features (or disturbances). There are three control inputs to the system: the chilled water set-point, zone air temperature set-point and lighting levels. At design time, the model based tree built (Algorithm 1) has 369 leaves and each of them has a linear regression model fitted over the control inputs with the response variable being the power consumption of the building.

In addition to learning the power consumption prediction tree, 19 additional model based trees were also built for predicting the different zone temperatures inside the building. When the DR event commences, at every time-step (every 5 mins), DR-Advisor uses the mbCRT algorithm to determine which leaf, and therefore, which linear regression model will be used for that time-step to solve the linear program (Eq 5) and determine the optimal values of the control inputs to meet a sustained response while maintaining thermal comfort.

Figure 15 shows the power consumption profile of the building using DR-Advisor for the DR event. We can see that using the mbCRT algorithm we are able to achieve a sustained curtailed response of 380kW over a period of 1 hour as compared to the baseline power consumption estimate. Also shown in the figure is the comparison between the best rule based fixed strategy which leads to the most curtailment in Section 7.4. In this case the DR strategy synthesis outperforms the best rule base strategy (from Section 7.4, Fig. 14) by achieving a 17% higher curtailment while maintaining thermal comfort. The rule-based strategy does not directly account for any effect on thermal comfort. The DR strategy synthesized by DR-Advisor is shown in Figure 16. We can see in Figure 17 how the mbCRT algorithm is able to maintain the zone temperatures inside the building within the specified comfort bounds. These results demonstrate the benefit of synthesizing optimal DR strategies as opposed ot relying on fixed rules and pre-determined strategies which do not account for any guarantees on thermal comfort. Figure 18 shows a close of view of the curtailed response. The leaf node which is being used for the power consumption constraint at every time-step is also shown in the plot. We can see that the model switches several times during the event, based on the forecast of disturbances.

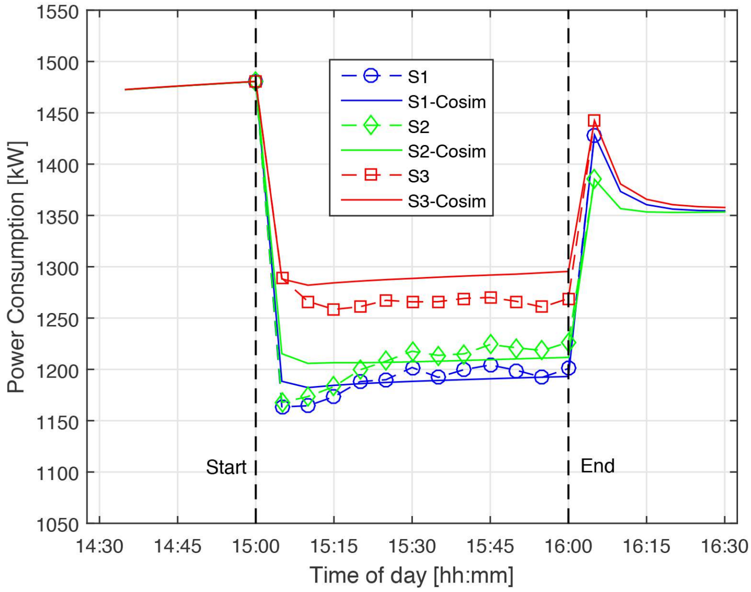

Figure 14: Prediction of power consumption for 3 strategies. DR Evaluation shows that Strategy 1 (S1) leads to maximum power curtailment.

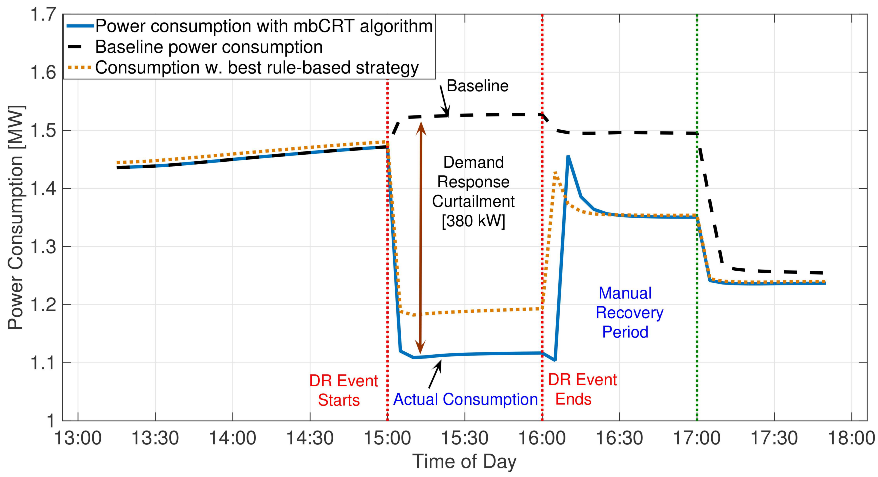

Figure 15: DR synthesis using the mbCRT algorithm for July 17, 2013. A curtailemnt of 380kW is sustained during the DR event period.

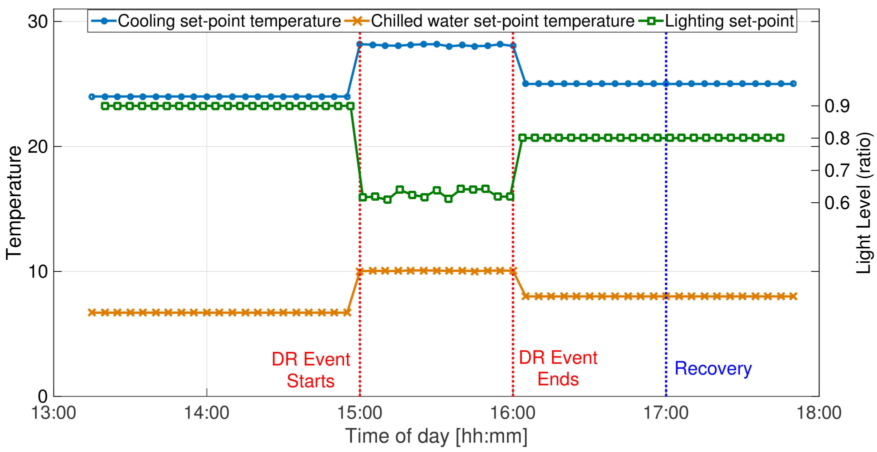

Figure 16: Optimal DR strategy as determined by the mbCRT algorithm.

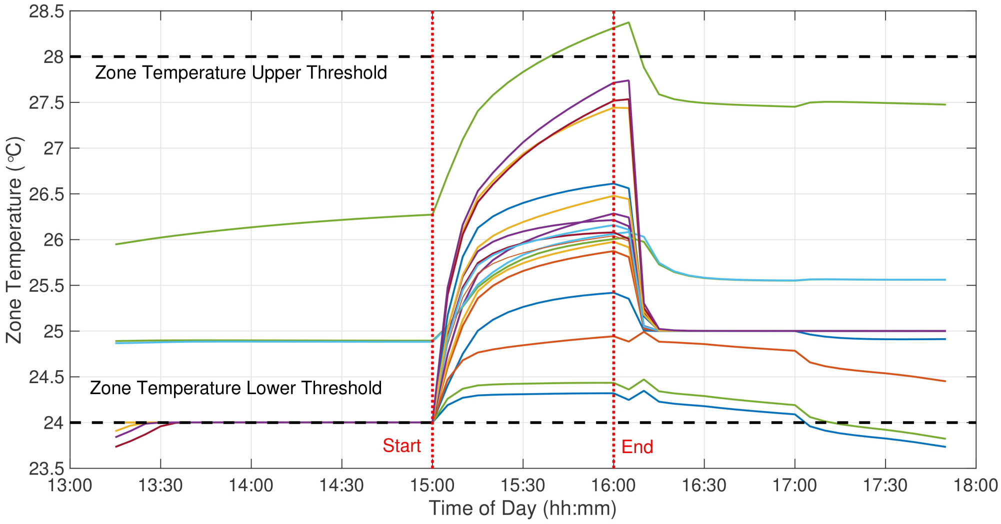

Figure 17: The mbCRT algorithm maintains the zone temperatures within the specified comfort bounds during the DR event.

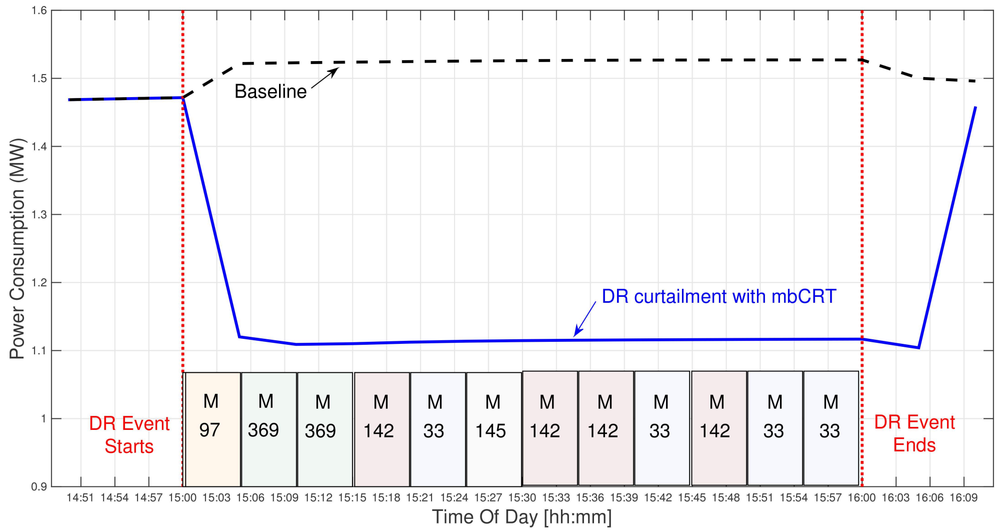

Figure 18: Zoomed in view of the DR synthesis showing how the mbCRT algorithm selects the appropriate linear model for each time-step based on the forecast of the disturbances.

These results show the effectiveness of the mbCRT algorithm to synthesize DR actions in real-time while utilizing a simple data-driven tree-based model.

7.5.1. ***Revenue from Demand Response***

We use Con Edison utility company's commercial demand response tariff structure [22] to estimate the financial reward obtained due to the curtailment achieved by the DR-Advisor for our Chicago based DoE commercial reference building. The utility provides a \$25/kW per month as a reservation incentive to participate in the real-time DR program for summer. In addition to that, a payment of \$1 per kWh of energy curtailed is also paid. For our test-bed, the peak load curtailed is 380kW. If we consider $\sim$ 5 such events per month for 4 months, this amounts to a revenue of $\sim \$45,600$ for participating in DR which is 37.9% of the energy bill of the building for the same duration ($\$120,317$). This is a significant amount, especially since using DR-Advisor does not require an investment in building complex modeling or installing sensor retrofits to a building.

8. Related work

There is a vast amount of literature ([23, 24, 25, 26]) which addresses the problem demand response under different pricing schemes. However, the majority of approaches so far have focused either on rule-based approaches for curtailment or on model-based approaches, such as the one described in [24]; in which model predictive control is used for DR based on a grey-box model of a building. [23] uses a high-fidelity physics based model of the building to solve a problem similar to the DR evaluation problem. [26] uses model predictive control for closed-loop optimal control strategy for load shifting in a plant that is charged for electricity on both time-of-use and peak demand pricing. One of the seminal studies of application of model predictive control on real buildings for demand response and energy-efficiency operation came from the Opticontrol project [10]. After several years of work on using grey-box and white box models for demand response control design, the authors state that the usefulness of any model based controller must be measured by not only its benefits and savings but also its incurred costs, such as the necessary hardware and software and the systems design, implementation, and maintenance effort. They further conclude that the biggest hurdle to mass adoption of intelligent building control is the cost and effort required to capture accurate dynamical models of the buildings. Since DR-Advisor only learns an aggregate building level model and combined with the fact that weather forecasts from third party vendors are expected to become cheaper; there is little to no additional sensor cost of implementing the DR-Advisor recommendation system in large buildings. The difficulties in identifying models for buildings is also highlighted in [27]. The authors observe that while model creation is mentioned only marginally in majority of the academical works dealing with model predictive control, these usually assume that the model of the system is either perfectly known or found in literature, the task is much more complicated and time consuming in case of a real application and sometimes, it can be even more complex and involved than the controller design itself. There are ongoing efforts to make tuning and identifying white box models of buildings more autonomous [11].

There is recent work, which has explored aspects of modeling, implementation and implications of demand response buildings [28, 29, 30, 31], however, their focus has mainly been on the residential sector. [30] shows that in general demand response contributes to a lower cost, higher reliability, and lower emission level of power system operation and highlights the societal value of DR. In [31] authors study the short term and long term affects of DR on residential electricity consumers through an elaborate empirical study. A reduced order physics based, grey-box modeling technique for simulating residential electric demand is presented in [29]. The ability to determine the correct response for large commercial buildings (from DR evaluation or DR synthesis) on a fast time scales (1-5 min) using purely data-driven methods makes both our approach and tool, novel.

Several machine learning and data-driven approaches have also been utilized before for forecasting electricity load. We already compared the performance of DR-Advisor against several data-driven methods in Section 7.3. In [32], seven different machine learning algorithms are applied to a residential data set with the objective of determining which techniques are most successful for predicting next hour residential building consumption. [33] uses artificial neural networks and regression models for modeling the energy demand of the residential sector in the U.S.. A forecasting method for cooling and electricity load demand is presented in [34], while a statistical analysis of the impact of weather on peak electricity demand using actual meteorological data is presented in [35]. In [36] a software architecture using parallel computing is presented to support data-driven demand response optimization. The shortcoming of work in this area is twofold: First, the time-scales at which the forecasts are generated ranges from 15-20 mins to hourly forecast; which is too coarse grained for DR events and for real-time price changes. Secondly, the focus in these methods is only on load forecasting but not on control synthesis, whereas the mbCRT algorithm presented in this paper enables the use of regression trees for control synthesis for the very first time.

9. Conclusions and ongoing work

We present a data-driven approach for modeling and control for demand response of large scale energy systems which are inherently messy to model using first principles based methods. We show how regression tree based methods are well suited to address challenges associated with demand response for large *C/I/I* consumers while being simple and interpretable. We have incorporated all our methods into the DR-Advisor tool - `http://mlab.seas.upenn.edu/dr-advisor/`.

DR-Advisor achieves a prediction accuracy of **92.8%** to **98.9%** for eight buildings on the University of Pennsylvania's campus. We compare the performance of DR-Advisor on a benchmarking data-set from AHRAE's energy predictor challenge and rank $2^n d$ among the winners of that competition. We show how DR-Advisor can select the best rule-based DR strategy, which leads to the most amount of curtailment, from a set of several rule-based strategies. We presented a model based control with regression trees (mbCRT) algorithm which enables control synthesis using regression tree based structures for the first time. Using the mbCRT algorithm, DR-Advisor can achieve a sustained curtailment of **380kW** during a DR event. Using a real tariff structure, we estimate a revenue of ~ $**45,600** for the DoE reference building over one summer which is **37.9%** of the summer energy bill for the building. The mbCRT algorithm outperforms even the best rule-based strategy by **17%**. DR-Advisor bypasses cost and time prohibitive process of building high fidelity models of buildings that use grey and white box modeling approaches while still being suitable for control design. These advantages combined with the fact that the tree based methods achieve high prediction accuracy, make DR-Advisor an alluring tool for evaluating and planning DR curtailment responses for large scale energy systems.

10. Acknowledgments

This work was supported in part by STARnet, a Semiconductor Research Corporation program, sponsored by MARCO and DARPA and by NSF MRI-0923518 grants.

References

[1] J. M. Melillo, T. Richmond, G. W. Yohe, Climate change impacts in the united states: the third national climate assessment, US Global change research program 841.
[2] N. N. C. for Environmental Information, State of the climate: Global analysis for august 2015, [published online September 2015, retrieved on October 15, 2015 from http://www.ncdc.noaa.gov/sotc/global/201508.].
[3] P. I. Michael J. Kormos, Pjm response to consumer reports on 2014 winter pricing.
[4] C. Goldman, Coordination of energy efficiency and demand response, Lawrence Berkeley National Laboratory.
[5] F. E. R. Commission, et al., Assessment of demand response and advanced metering.
[6] P. Interconnection, 2014 deamand response operations markets activity report.
[7] N. Research, Demand response for commercial & industrial markets market players and dynamics, key technologies, competitive overview, and global market forecasts.
[8] Energy price risk and the sustainability of demand side supply chains, Applied Energy 123 (0) (2014) 327 – 334.
[9] D. B. Crawley, L. K. Lawrie, et al., Energyplus: creating a new-generation building energy simulation program, Energy and Buildings 33 (4) (2001) 319 – 331.
[10] D. Sturzenegger, D. Gyalistras, M. Morari, R. S. Smith, Model predictive climate control of a swiss office building: Implementation, results, and cost-benefit analysis.
[11] J. R. New, J. Sanyal, M. Bhandari, S. Shrestha, Autotune e+ building energy models, Proceedings of the 5th National SimBuild of IBPSA-USA.
[12] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and regression trees, CRC press, 1984.
[13] C. Giraud-Carrier, Beyond predictive accuracy: what?, Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation (1998) 78–85.
[14] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
[15] J. Elith, J. R. Leathwick, T. Hastie, A working guide to boosted regression trees, Journal of Animal Ecology 77 (4) (2008) 802–813.

[16] J. R. Quinlan, et al., Learning with continuous classes, in: 5th Australian joint conference on artificial intelligence, Vol. 92, Singapore, 1992, pp. 343–348.
[17] J. H. Friedman, Multivariate adaptive regression splines, The annals of statistics (1991) 1–67.
[18] M. Deru, K. Field, D. Studer, et al., U.s. department of energy commercial reference building models of the national building stock.
[19] J. F. Kreider, J. S. Haberl, Predicting hourly building energy use: The great energy predictor shootout–overview and discussion of results, Tech. rep., American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., Atlanta, GA (United States) (1994).
[20] D. J. MacKay, et al., Bayesian nonlinear modeling for the prediction competition, ASHRAE transactions 100 (2) (1994) 1053–1062.
[21] Siemens, Automated demand response using openadr: Application guide (2011).
[22] Con Edison, Demand response programs details.
URL http://www.coned.com/energyefficiency/pdf/proposed-program-changes.pdf
[23] D. Auslander, D. Caramagno, D. Culler, T. Jones, A. Krioukov, M. Sankur, J. Taneja, J. Trager, S. Kiliccote, R. Yin, et al., Deep demand response: The case study of the citris building at the university of california-berkeley.
[24] F. Oldewurtel, D. Sturzenegger, G. Andersson, M. Morari, R. S. Smith, Towards a standardized building assessment for demand response, in: Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, IEEE, 2013, pp. 7083–7088.
[25] P. Xu, P. Haves, M. A. Piette, J. Braun, Peak demand reduction from pre-cooling with zone temperature reset in an office building, Lawrence Berkeley National Laboratory.
[26] A. J. Van Staden, J. Zhang, X. Xia, A model predictive control strategy for load shifting in a water pumping scheme with maximum demand charges, Applied Energy 88 (12) (2011) 4785–4794.
[27] E. Žáčeková, Z. Váňa, J. Cigler, Towards the real-life implementation of mpc for an office building: Identification issues, Applied Energy 135 (2014) 53–62.
[28] A. Kialashaki, J. R. Reisel, Modeling of the energy demand of the residential sector in the united states using regression models and artificial neural networks, Applied Energy 108 (0) (2013) 271 – 280. doi:http://dx.doi.org/10.1016/j.apenergy.2013.03.034.
URL http://www.sciencedirect.com/science/article/pii/S0306261913002304
[29] M. Muratori, M. C. Roberts, R. Sioshansi, V. Marano, G. Rizzoni, A highly resolved modeling technique to simulate residential power demand, Applied Energy 107 (0) (2013) 465 – 473. doi:http://dx.doi.org/10.1016/j.apenergy.2013.02.057.
URL http://www.sciencedirect.com/science/article/pii/S030626191300175X
[30] B. Dupont, K. Dietrich, C. D. Jonghe, A. Ramos, R. Belmans, Impact of residential demand response on power system operation: A belgian case study, Applied Energy 122 (0) (2014) 1 – 10. doi:http://dx.doi.org/10.1016/j.apenergy.2014.02.022.
URL http://www.sciencedirect.com/science/article/pii/S0306261914001585
[31] C. Bartusch, K. Alvehag, Further exploring the potential of residential demand response programs in electricity distribution, Applied Energy 125 (0) (2014) 39 – 59. doi:http://dx.doi.org/10.1016/j.apenergy.2014.03.054.
[32] R. E. Edwards, J. New, L. E. Parker, Predicting future hourly residential electrical consumption: A machine learning case study, Energy and Buildings 49 (2012) 591–603.
[33] A. Kialashaki, J. R. Reisel, Modeling of the energy demand of the residential sector in the united states using regression models and artificial neural networks, Applied Energy 108 (2013) 271–280.
[34] A. Vaghefi, M. Jafari, E. Bisse, Y. Lu, J. Brouwer, Modeling and forecasting of cooling and electricity load demand, Applied Energy 136 (2014) 186–196.
[35] T. Hong, W.-K. Chang, H.-W. Lin, A fresh look at weather impact on peak electricity demand and energy use of buildings using 30-year actual weather data, Applied Energy 111 (2013) 333–350.
[36] W. Yin, Y. Simmhan, V. K. Prasanna, Scalable regression tree learning on hadoop using openplanet, in: Proceedings of third international workshop on MapReduce and its Applications Date, ACM, 2012, pp. 57–64.
[37] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, R. Tibshirani, The elements of statistical learning, Vol. 2, Springer, 2009.
[38] B. D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, 1996.

Appendix A. Building regression trees

We explain how regression trees are built using an example adapted from [37]. Tree-based methods partition the feature space into a set of rectangles (more formally, hyper-rectangles) and then fit a simple model in each one. They are conceptually simple yet powerful. Let us consider a regression problem with continuous response Y and inputs X_1 and X_2, each taking values in the unit interval. The top left plot of Figure A.19 shows a partition of the feature space by lines that are parallel to the coordinate axes. In each partition element we can model Y with a different constant. However, there is a problem: although each partitioning line has a simple description like $X_1 = k$, some of the resulting regions are complicated to describe. To simplify things, we can restrict ourselves to only consider recursive binary partitions, like the ones shown in the top right plot of Figure A.19. We first split the space into two regions, and model the response by the mean of Y in each region. We choose the variable and split-point to achieve the best prediction of Y. Then one or both of these regions are split into two more regions, and this process is continued, until some stopping rule is applied. This is the recursive partitioning part of the algorithm. For example, in the top right plot of Figure A.19, we first split at $X_1 = t_1$. Then the region $X_1 \leq t_1$ is split at $X_2 = t_2$ and the region $X_1 > t_1$ is split at $X_1 = t_3$. Finally, the region $X_1 > t_3$ is split at $X_2 = t_4$. The result of this process is a partition of the data-space into the five regions $R_1, R_2, \cdots, R_5$. The corresponding regression tree model predicts Y with a constant c_i in region

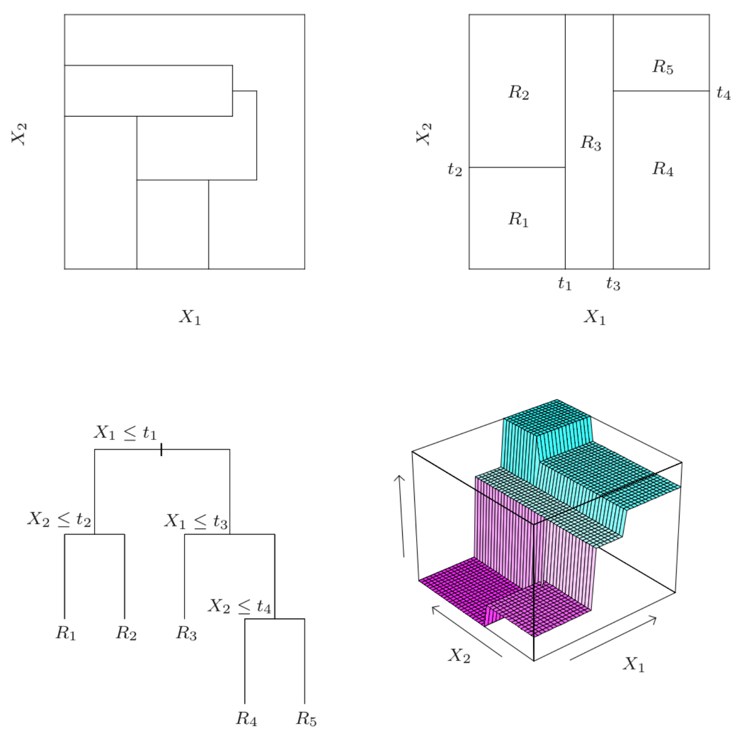

Figure A.19: Top right: 2D feature space by recursive binary splitting. Top left: partition that cannot be obtained from recursive binary splitting. Bottom left: tree corresponding to the partition. Bottom right: perspective plot of the prediction surface.

R_i i.e.,,

$$\hat{T}(X) = \sum_{i=1}^{5} c_i I\{(X_1, X_2) \in R_i\} \tag{A.1}$$

This same model can be represented by the binary tree shown in the bottom left of Figure A.19. The full data-set sits at the top or the root of the tree. Observations satisfying the condition at each node are assigned to the left branch, and the others to the right branch. The terminal nodes or leaves of the tree correspond to the regions $R_1, R_2, ..., R_5$.

Appendix A.1. Node splitting criteria

For regression trees we adopt the sum of squares as our splitting criteria i.e a variable at a node will be split if it minimizes the following sum of squares between the predicted response and the actual output variable.

$$\sum (y_i - \hat{T}(x_i))^2 \tag{A.2}$$

It is easy to see that the best response c_i (from equation A.1 for y_i from partition R_i is just the average of output samples in the region R_i i.e

$$c_i = avg(y_i | x_i \in R_i) \tag{A.3}$$

Finding the best binary partition in terms of minimum sum of squares is generally computationally infeasible. A greedy algorithm is used instead. Starting with all of the data, consider a splitting variable j and split point s, and define the following pair of left (R_L) and right (R_R) half-planes

$$\begin{aligned} R_L(j, s) &= \left\{X | X_j \leq s\right\}, \\ R_R(j, s) &= \left\{X | X_j > s\right\} \end{aligned} \tag{A.4}$$

The splitting variable j and the split point s is obtained by solving the following minimization:

$$\min_{j,s} \left[\min_{c_L} \sum_{x_i \in R_L(j,s)} (y_i - c_L)^2 + \min_{c_R} \sum_{x_i \in R_R(j,s)} (y_i - c_R)^2 \right] \tag{A.5}$$

where, for any choice of j and s, the inner minimization in equation A.5 is solved using

$$\begin{aligned} c_L &= \text{avg}(y_i | x_i \in R_L(j, s)) \\ c_R &= \text{avg}(y_i | x_i \in R_R(j, s)) \end{aligned} \tag{A.6}$$

For each splitting variable X_j, the determination of the split point s can be done very quickly and hence by scanning through all of the inputs (X_i's), the determination of the best pair (j, s) is feasible. Having found the best split, we partition the data into the two resulting regions and repeat the splitting process on each of the two regions. Then this process is repeated on all of the resulting regions.

Rather than splitting each node into just two regions at each stage, we might consider multiway splits into more than two groups. While this can sometimes be useful, it is not a good general strategy. The problem is that multiway splits fragment the data too quickly, leaving insufficient data at the next level down. Hence we would want to use such splits only when needed. Also multiway splits can be achieved by a series of binary splits.

Appendix A.2. Stopping criteria and pruning

Every recursive algorithm needs to know when it's done, i.e it requires a stopping criteria. For regression trees this means when to stop splitting the nodes. A very large tree might over fit the data, while a small tree might not capture the important structure. Tree size is a tuning parameter governing the models complexity, and the optimal tree size should be adaptively chosen from the data. One approach is to split tree nodes only if the decrease in sum-of-squares due to the split exceeds some threshold. However, this strategy is myopic, since a seemingly worthless split might lead to a very good split below it. A preferred, strategy is to grow a large tree, stopping the splitting process only when some minimum number of data points at a node (`MinLeaf`) is reached. Then this large tree is pruned using cost-complexity pruning methods.

Define a subtree $T \subset T_0$ to be any tree that can be obtained by pruning T_0, i.e. collapsing any number of its non-terminal nodes. Let node i corresponding to the partition R_i. $|T|$ denotes the number of terminal nodes in T Define,

$$\begin{aligned} N_i &= \#\{x_i \in R_i\}, \\ \hat{c}_i &= \frac{1}{N_i} \sum_{x_i \in R_i} y_i, \\ Q_i(T) &= \frac{1}{N_i} \sum_{x_i \in R_i} (y_i - \hat{c}_i)^2 \end{aligned} \tag{A.7}$$

where N_i is the number of samples in the partition R_i, $\hat{c}_i$ is the estimate of y within R_i and $Q_i(T)$ is the mean square error of the estimate $\hat{c}_i$. The cost complexity criteria is then defined as:

$$C_\alpha(T) = \sum_{i=1}^{|T|} N_i Q_i(T) + \alpha|T| \tag{A.8}$$

The goal is to find, for each α, the subtree $T_\alpha \subset T_0$ to minimize $C_\alpha(T)$. The tuning parameter $\alpha \geq 0$ governs the trade off between tree size and its goodness of fit to the data. For each α one can show that there is a unique smallest subtree T_α that minimizes $C_\alpha(T)$ [38]. Estimation of α is achieved by cross-validation.