

DATA-DRIVEN MODELING, CONTROL AND TOOLS FOR  
CYBER-PHYSICAL ENERGY SYSTEMS

Madhur Behl

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania  
in Partial Fulfillment of the Requirements for the  
Degree of Doctor of Philosophy

2015

---

Rahul Mangharam, Associate Professor of Electrical and Systems Engineering  
Supervisor of Dissertation

---

Alejandro Ribeiro, Associate Professor of Electrical and Systems Engineering  
Graduate Group Chairperson

Dissertation Committee:

George Pappas, Professor of Electrical and Systems Engineering

Ruben Lobel, Assistant Professor of Operations, Information and Decisions

Jin Wen, Associate Professor of Civil, Architecture and Environmental Engineering,  
Drexel University

Rajesh Gupta, Professor of Computer Science and Engineering,  
University of California, San Diego

DATA-DRIVEN MODELING, CONTROL AND TOOLS FOR  
CYBER-PHYSICAL ENERGY SYSTEMS

COPYRIGHT

2015

Madhur Behl

*To my family.*

# Acknowledgments

While I alone am responsible for this thesis, it is nonetheless at least as much a product of years of interaction with, and inspiration by, a large number of friends and colleagues as it is my own work. For this reason, I wish to express my warmest gratitude to all those persons whose comments, questions, criticism, support and encouragement, personal and academic, have left a mark on this work. I also wish to thank those institutions which have supported me during the work on this thesis. Regrettably, but inevitably, the following list of names will be incomplete, and I hope that those who are missing will forgive me, and will still accept my sincere appreciation of their influence on my work.

My deepest gratitude is to my advisor, Dr. Rahul Mangharam. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered. Rahul taught me how to question thoughts and express ideas and to identify problems which will have a big impact. I am grateful to him for holding me to a high research standard and enforcing strict validations for each research result. He challenged me to defend my ideas and taught me the importance of taste and strategy. These are virtues which I will carry with me throughout my life. It is impossible to keep count of the number of opportunities that he has provided for me over the years. Whether it be a chance to present my work in seminars and conferences and in front of nobel prize winners, interact with professors outside of my department and in other universities or initiating a collaboration with the industry and get feedback on how relevant, were the problems we were solving and how useful was our proposed solution. For instance, in the early part of my PhD; being a part of the Department of Energy's energy-efficient buildings hub, exposed me with problems in the energy domain and helped me to identify where the biggest hurdles in this area lie. He has always supported my ideas, interests and plans and it is only because of this reason that I was able to and even encouraged to take courses outside of engineering, be it courses in Statistics, which later played a key role in my thesis or a course in genomics simply out of curiosity. One simply could not wish for a better or friendlier supervisor. I hope that one day I would become as good an advisor to my students as Rahul has been to me.

I am extremely grateful to all the members of my thesis committee, Dr. George Pappas, Dr. Rajesh Gupta, Dr. Jin Wen and Dr. Ruben Lobel. They have guided me well and I have really enjoyed our discussions. I also want to thank you for

letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you. Most impressive to me, however, is that in addition to their encouraging feedback and their role shaping my thinking about my research, they are all friendly, generous, constructive, and good people. If all goes well, I will be collaborating with each of them for many more years.

I thank all my co-authors. Without their contributions and help, it would have been near to impossible for me to complete this work successfully. A big thank you to Dr. Truong X. Nghiem, a friend and a mentor; thank you for the numerous discussions on related topics that helped me improve my knowledge in the area. I thank Willy Bernal for helping me with several simulations, results and publications and for being a great friend. Thank you Santiago Gonzalez and Francesco Smarra for helping with the tool development for my research and for being such excellent collaborators. I thank Achin Jain, for his help with case studies and results, within just a few months after joining the PhD program.

I thank Pengyuan Shen from Penn Design, for helping me getting access to energy data for the Penn campus. Also from Penn design, a thank you to Dr. Yun Yi and Dr. William W. Braham for your support and guidance. Thank you Dr. Mark Allen Hughes and Angela Garcia at the Kleinmen Center for Energy Policy for always being inviting and presenting us with opportunities to collaborate. I would also like to thank Ken Ogawa and the entire team (Benedict Suplick, Andrew Zarynow, Earl Boston, Walter Molishus and others) at Penn Facilities for being so supportive and open to work with us and share Penn's energy data. Your experience and insights have really helped me to identify the "real" problems where significant contributions could be made. Thank you Dr. Yan Lu and Dr. Sanjeev Srivastava for the discussions and feedback during your time at Siemens. Thank you Dr. Girija Parthasarathy for providing me with an opportunity to spend a summer at Honeywell Automation and Control Labs.

I am grateful to all former and current members of mLab for their various forms of support during my graduate study. mLab was my second home and you all have been nothing short of my extended family and I feel glad and fortunate to have met you. It is because of you, my graduate experience has been one that I will cherish forever. Thank you Dr. Miroslav Pajic and soon to be Dr. Zhihao Jiang. Thank you (in no particular order) Yash Vardhan Pant, Kuk Jang, Dr. Houssam Abbas, Dr. Marco Beccani, Matthew O'Kelly, Matthew Brady, Vincent Pacelli, Tao Lei, Praveen Pitchai, Abhijeet Mulay, Harsh Jain, Mansimar Aneja, Utsav Drolia, Srinivas Vemuri, William Etter, Paul Martin, Eric Berdinis, Ross Boczar, Rajib Dutta, Chen Zhen, Parth Chopra and Siddharth Deliwala.

I am also grateful to the former or current staff at Electrical and Systems Engineering, and the staff at the Moore Business Office for their various forms of support during my graduate study and for helping me at the eleventh hour to ensure all administrative matters ran smoothly and without any problems. Thank you Irene Tan, Lilian Wu, Jackie Egitto, Dru Spanner, Nichole Wood, Gail Shannon, Denice Gorte

and others.

Although, my major is in Electrical and Systems Engineering, dozens of people (both former and current) have helped and taught me immensely at the PRECISE center in Computer and Information Science at Penn Engineering and I thank them for their support. Thank you Dr. Insup Lee, Dr. Linh Thi Xuan Phan, Dr. Krishna Venkatasubramanian, Dr. James Weimer, Peter Gebhard, Sanjian Chen, Dr. BaekGuy Kim and Andrew King. A special thanks to Liz for being a "superwoman" administrator and for making sure that the food at each seminar was nothing but the best.

Many friends outside of my work have helped me stay sane and motivated through these years. Their support and care helped me overcome setbacks and stay focused on my graduate study. I greatly value their friendship and I deeply appreciate their belief in me. Thank you Dr. Bilwaj Gaonkar, and Dr. Paramveer Dhillon for being excellent roommates. I have especially enjoyed our profound yet perplexing discussions on all matters of life and the universe. Thank you Pulkit Kapur, Dr. Nipun Sinha and Harsha Battapady for some wonderful initial years of my PhD and for your support and encouragement. Thank you Dr. Aris Sotiras and Dr. Luke Macyszyn.

Most importantly, none of this would have been possible without the love and patience of my family. My immediate family to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to my parents Suman and Anil Behl. They have supported me in every decision throughout my career and my gratitude to them is beyond words. A big thanks and a big hug to Shikha and Preeti for their love and support. You are the best sisters a brother can hope for. My extended family has aided and encouraged me throughout this endeavor and I thank them for their support. Last, but not the least, I thank my two favorite people on this planet, my nephews Arsh and Arjun. You guys are rockstars and are destined for great things !

## ABSTRACT

### DATA-DRIVEN MODELING, CONTROL AND TOOLS FOR CYBER-PHYSICAL ENERGY SYSTEMS

Madhur Behl  
Rahul Mangharam

Energy systems are experiencing a gradual but substantial change in moving away from being non-interactive and manually-controlled systems to utilizing tight integration of both cyber (computation, communications, and control) and physical representations guided by first principles based models, at all scales and levels. Furthermore, peak power reduction programs like demand response (DR) are becoming increasingly important as the volatility on the grid continues to increase due to regulation, integration of renewables and extreme weather conditions. In order to shield themselves from the risk of price volatility, end-user electricity consumers must monitor electricity prices and be flexible in the ways they choose to use electricity.

This requires the use of control-oriented predictive models of an energy systems dynamics and energy consumption. Such models are needed for understanding and improving the overall energy efficiency and operating costs. However, learning dynamical models using grey/white box approaches is very cost and time prohibitive since it often requires significant financial investments in retrofitting the system with several sensors and hiring domain experts for building the model. We present the use of data-driven methods for making model capture easy and efficient for cyber-physical energy systems.

We develop Model-IQ, a methodology for analysis of uncertainty propagation for building inverse modeling and controls. Given a grey-box model structure and real input data from a temporary set of sensors, Model-IQ evaluates the effect of the uncertainty propagation from sensor data to model accuracy and to closed-loop control performance. We also developed a statistical method to quantify the bias in the sensor measurement and to determine near optimal sensor placement and density for accurate data collection for model training and control. Using a real building test-bed, we show how performing an uncertainty analysis can reveal trends about inverse model accuracy and control performance, which can be used to make informed decisions about sensor requirements and data accuracy.

We also present DR-Advisor, a data-driven demand response recommender system for the building's facilities manager which provides suitable control actions to meet the desired load curtailment while maintaining operations and maximizing the economic reward. We develop 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 demand response methods for a large DoE commercial reference building and leads to a significant amount of load curtailment (of 380kW) and over

\$45,000 in savings which is 37.9% of the summer energy bill for the building. The performance of DR-Advisor is also evaluated for 8 buildings on Penn's campus; where it achieves 92.8% to 98.9% prediction accuracy. We also compare DR-Advisor with other data driven methods and rank 2<sup>nd</sup> on ASHRAE's benchmarking data-set for energy prediction.

# Contents

<b>Title</b>	<b>i</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>Abstract</b>	<b>vii</b>
<b>Contents</b>	<b>ix</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.1.1 Modeling challenges for cyber-physical energy systems . . . . .	4
1.2 Research Goals . . . . .	7
1.2.1 Low-cost building model capture . . . . .	9
1.2.2 Data-driven control oriented modeling for demand response . . . . .	10
1.3 Contributions . . . . .	11
1.4 Organization . . . . .	12
<b>2 Building modeling with first principles</b>	<b>15</b>
2.1 White-box modeling . . . . .	16
2.1.1 EnergyPlus . . . . .	17
2.1.2 TRNSYS . . . . .	18
2.2 Grey-box models . . . . .	19
2.2.1 Parameter Estimation (Model Training) . . . . .	22
2.3 Concluding remarks . . . . .	23
<b>3 Low-Cost Building Model Capture</b>	<b>24</b>
3.1 Uncertainty in Building Modeling . . . . .	25
3.2 Model-IQ approach . . . . .	26
3.2.1 Example with TRNSYS Model . . . . .	27
3.3 Input Uncertainty Analysis . . . . .	31

3.4	Model Accuracy vs MPC Performance . . . . .	34
3.4.1	MPC formulation . . . . .	35
3.4.2	State Observer . . . . .	36
3.4.3	Single zone example . . . . .	37
3.5	Sensor Placement and Data Quality . . . . .	41
3.6	Case Study With Real Building Data . . . . .	48
3.6.1	Model-IQ implementation for Suite 210 . . . . .	49
3.6.2	Results . . . . .	51
3.7	Related Work . . . . .	53
3.7.1	Model predictive control related . . . . .	53
3.7.2	Sensitivity analysis related . . . . .	53
3.7.3	Uncertainty related . . . . .	54
3.8	Concluding remarks . . . . .	54
<b>4</b>	<b>Data-driven modeling with regression trees</b>	<b>56</b>
4.1	Learning from data . . . . .	57
4.2	Regression trees . . . . .	58
4.2.1	Node Splitting Criteria . . . . .	61
4.2.2	Stopping Criteria and Pruning . . . . .	62
4.3	Ensemble Methods . . . . .	63
4.4	Model Based Regression Trees . . . . .	64
4.5	Comparison with k-means . . . . .	64
<b>5</b>	<b>Data-Driven Modeling and Control for DR</b>	<b>65</b>
5.1	Problem definition . . . . .	69
5.1.1	DR Baseline Prediction . . . . .	69
5.1.2	DR Strategy Evaluation . . . . .	70
5.1.3	DR Strategy Synthesis . . . . .	70
5.2	Data-driven demand response . . . . .	71
5.2.1	Data-Description . . . . .	72
5.2.2	Weather Data . . . . .	72
5.2.3	Schedule Data . . . . .	74
5.2.4	Building Data . . . . .	74
5.2.5	Data-Driven DR Baseline . . . . .	75
5.2.6	Data-Driven DR Evaluation . . . . .	75
5.3	DR synthesis with regression trees . . . . .	76
5.3.1	Model-based control with regression trees . . . . .	78
5.3.2	DR synthesis optimization . . . . .	80
5.4	The case for using regression trees for DR . . . . .	84
5.5	DR-Advisor:Toolbox design . . . . .	85
5.6	Case study . . . . .	88
5.6.1	Building and Data Description . . . . .	88
5.6.2	Model Validation . . . . .	90

5.6.3	Energy Prediction Benchmarking . . . . .	93
5.6.4	DR Strategy Evaluation . . . . .	97
5.6.5	DR Strategy Synthesis . . . . .	98
5.6.6	Choosing the best tree . . . . .	103
5.7	Related work . . . . .	104
5.8	Concluding remarks . . . . .	106
<b>6</b>	<b>Energy Analytics</b>	<b>107</b>
6.1	Filter attributes at the leaves . . . . .	107
6.2	Query-response case study with real data . . . . .	110
6.2.1	Example queries . . . . .	111
<b>7</b>	<b>Conclusion</b>	<b>113</b>
7.1	Conclusion . . . . .	113
7.2	Future work . . . . .	114
	<b>Bibliography</b>	<b>116</b>

# List of Tables

2.1	List of parameters . . . . .	21
3.1	Wilcoxon's test results for all values of $k$ . . . . .	47
5.1	Model validation with Penn data . . . . .	90
5.2	Comparison of methods on Building 101 data . . . . .	91
5.3	ASHRAE Shootout Competition Results . . . . .	96
6.1	Query-response for College Hall . . . . .	111

# List of Figures

1.1	Volatility in real-time electricity prices in the PJM ISO from January 2014. . . . .	2
1.2	Steep increase of 32 fold in the wholesale electricity prices in July 2015. . . . .	3
1.3	Electricity and natural gas consumption breakdown for commercial buildings in the United States. . . . .	4
1.4	Electricity and natural gas consumption breakdown for commercial buildings in the United States. . . . .	6
1.5	Comparison of modeling methods for cyber-physical energy systems. . . . .	8
2.1	Modeling categories for cyber-physical energy systems . . . . .	16
2.2	White-box building modeling approach for buildings. . . . .	17
2.3	Grey-box building modeling approach for buildings. . . . .	19
2.4	RC lumped-parameter model representation for a thermal zone obtained from information about the zone geometry and usage. . . . .	20
3.1	Model-IQ Toolbox uncertainty analysis for building controls. . . . .	25
3.2	Overview of the Model-IQ input uncertainty analysis . . . . .	27
3.3	Training inputs for single zone . . . . .	28
3.4	Simulation setup in TRNSYS used for data uncertainty evaluation. . . . .	29
3.5	TRNSYS RC model validation . . . . .	30
3.6	TRNSYS input uncertainty analysis . . . . .	32
3.7	Model sensitivity coefficients for different input data streams. . . . .	34
3.8	MPC overview . . . . .	35
3.9	Baseline MPC performance . . . . .	38
3.10	MPC performance with an inaccurate model . . . . .	40
3.11	MPC performance for models of different degrees of accuracy. . . . .	40
3.12	Temperature sensor locations for suite 210 . . . . .	41
3.13	Sensor placement and density . . . . .	42
3.14	All the temperature measurements from different locations in suite 210. . . . .	43
3.15	Thermostat vs mean temperature . . . . .	44
3.16	Scatter plot between the mean temperature and thermostat reading . . . . .	45
3.17	All temperatures Q-Q plot . . . . .	46
3.18	Bland-Altman plots for all 6 sensors . . . . .	46
3.19	Building 101 . . . . .	48

3.20	Training data for Building 101 . . . . .	50
3.21	Predicted and actual zone temperature . . . . .	51
3.22	Input uncertainty analysis for Building 101 . . . . .	52
3.23	Model accuracy sensitivity coefficients for Building 101 . . . . .	53
4.1	Recursive partitioning and regression trees . . . . .	60
5.1	Rule-based vs. DR-Advisor . . . . .	66
5.2	DR-Advisor Architecture . . . . .	67
5.3	Demand Response Time line and Baseline Prediction. . . . .	69
5.4	Demand Response Strategy Evaluation. . . . .	71
5.5	Example training data for DR-Advisor . . . . .	73
5.6	Accuracy of auto-regressive trees for temperature predictions. . . . .	76
5.7	Mixed order regression tree . . . . .	77
5.8	Separation of variables . . . . .	79
5.9	DR-synthesis algorithm overview . . . . .	81
5.10	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. . . . .	82
5.11	Separation of variables accuracy. . . . .	83
5.12	Screenshot of the DR-Advisor MATLAB based GUI. . . . .	85
5.13	DR-Advisor Workflow . . . . .	86
5.14	DRAdvisor model identification tab . . . . .	87
5.15	Building 101 and DoE commercial reference building. . . . .	88
5.16	Penn campus buildings . . . . .	88
5.17	Model validation for the clinical research building at Penn. . . . .	89
5.18	Model validation for Building 101 . . . . .	91
5.19	Resubstitution error in random forest . . . . .	92
5.20	Predictor variables (feature) importance in the random forest ensemble. . . . .	92
5.21	Training data for the ASHRAE Great Energy Predictor Shootout Challenge. . . . .	94
5.22	Whole building electricity (WBE) training data . . . . .	95
5.23	Resubstitution error is shown for the number of trees in the random forest method. . . . .	96
5.24	Comparison between predicted and ground truth values for WBE for the testing data. . . . .	97
5.25	Rule-based strategies used in DR Evaluation. CHSTP denotes Chiller set point and CLGSTP denotes Zone Cooling temperature set point. . . . .	99
5.26	Prediction of power consumption for 3 strategies. DR Evaluation shows that Strategy 1 (S1) leads to maximum power curtailment. . . . .	100

5.27	DR synthesis using the mbCRT algorithm for July 17, 2013. A cur- tailemnt of 380kW is sustained during the DR event period. . . . .	101
5.28	Optimal DR strategy as determined by the mbCRT algorithm. . . . .	102
5.29	The mbCRT algorithm maintains the zone temperatures within the specified comfort bounds during the DR event. . . . .	102
5.30	Zoomed in view of the DR synthesis showing how the mbCRT algo- rithm selects the appropriate linear model for each time-step based on the forecast of the disturbances. . . . .	103
6.1	Attributes at leaf nodes . . . . .	108
6.2	Attributes defined at each leaf node of the tree in a random forest. . .	109
6.3	Comparison between predicted and actual power consumption of the building over the test-set. . . . .	110
6.4	Data structure plot . . . . .	111

# Chapter 1

## Introduction

### 1.1 Motivation

In 2013, a report by the U.S. National Climate Assessment provided evidence that the most recent decade was the nations warmest on record [Melillo et al. 2014] and experts predict that temperatures are only going to rise. In fact, the year 2015 is likely to become the hottest year on record since the beginning of weather recording in 1880 [NOAA 2015]. Heat waves in summer and polar vortexes in winter are growing longer and pose increasing challenges to an already over-stressed electric grid. With the increasing penetration of renewable generation, the electricity grid is 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 volatility due to the mismatch between electricity generation and supply further leads to volatility in the wholesale price of electricity. An example of such price volatility is shown in Figure 1.1 where the fluctuations in electricity price from a single day in January, 2014th are shown for the PJM independent system operator (ISO); one of the largest grid operators in the world. In another example 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/MW h to \$2,680/MW h in a matter of a few minutes [Michael J. Kormos 2014]. Similarly, the summer price spiked 32 fold from an average of \$25/MW h to \$800/MW h in July of 2015 as shown in Figure 1.2. Such events show how unforeseen and uncontrollable circumstances can greatly affect electricity prices that impact (independent system operators) 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 [Goldman 2010]. While energy efficiency

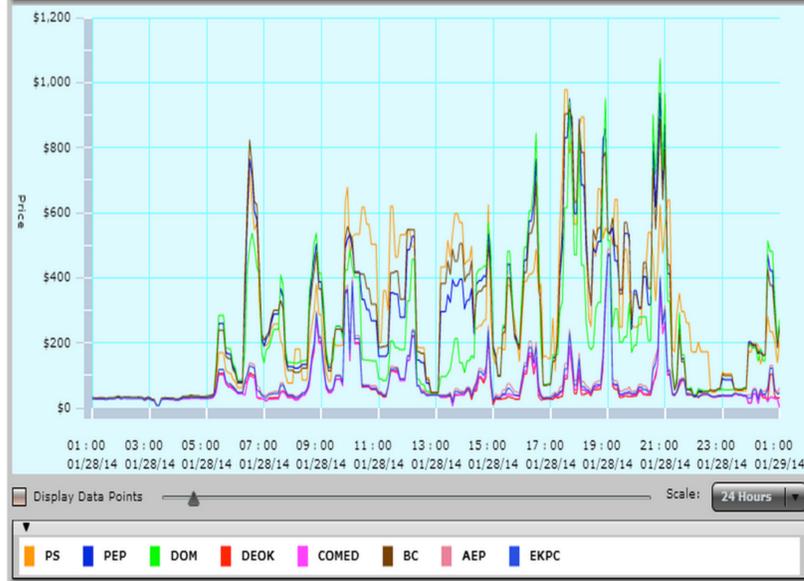


Figure 1.1: Volatility in real-time electricity prices in the PJM ISO from January 2014.

is a prominent component of growing efforts to supply affordable, reliable and clean electric power; most utilities and system operators are increasingly turning to demand response as a cost effective and environmentally responsible way to serve peak load. The potential demand response resource contribution from all U.S. demand response programs is estimated to be nearly 72,000 megawatts, or about 9.2 percent of U.S. peak demand [Commission et al. 2012] making DR the largest virtual generator in the U.S. national grid. The revenue to end-users from DR markets with PJM ISO alone is about \$700 million [Interconnection 2014]. A recent report [Research 2015] estimates that the global commercial and industrial (C&I) DR revenue is expected to reach nearly \$40 billion from 2014 through 2023.

Buildings consume more than one third of the worlds total primary energy [in Buildings and Programme 2013]. They account for 40% of all energy use in the U.S and for 72% of total U.S. electricity consumption [Construction 2010]. Large commercial and industrial buildings, are the biggest consumers of electricity and a significant contributor to peak load conditions on the grid. The largest use of energy consumption in buildings is attributed to the heating, ventilation and air-conditioning (HVAC) systems (Figure 1.3). Therefore, measures directed at the HVAC systems are attractive for load curtailment and energy-efficient building operation. Energy efficiency measures for large buildings refer to permanent changes to electricity usage through installation of, or replacement with, more efficient end-use devices and control systems that reduce the total quantity of energy needed to condition a building. Therefore, a large part of improving energy-efficiency entails infrastructure upgrading and retrofitting expenses. Demand response, on the other hand, refers to intentional changes in electricity usage by end-use customers from their normal consumption patterns in response to real-time changes in the price of electricity, or to incentive

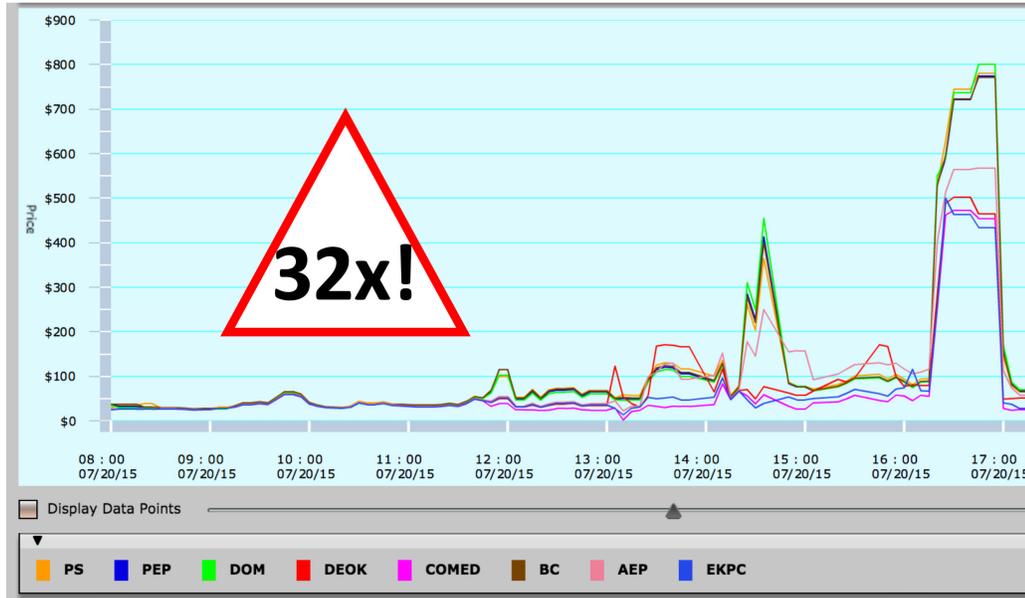
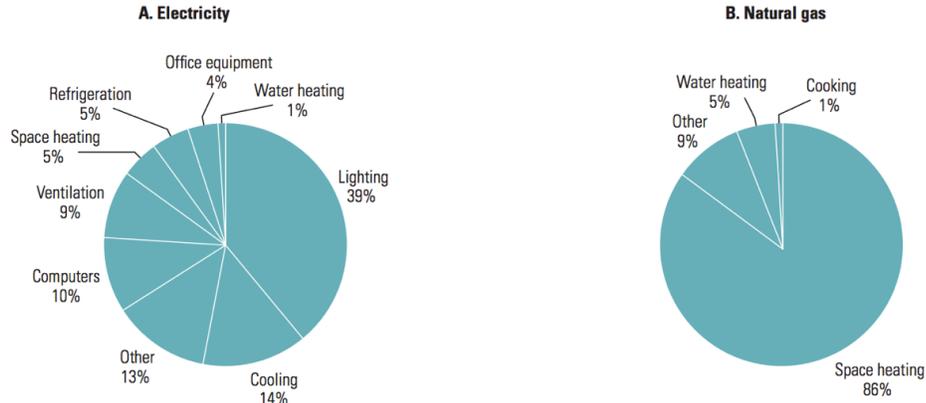


Figure 1.2: Step increase of 32 fold in the wholesale electricity prices in July 2015.

payments designed to induce lower electricity use at times of high market prices or when system reliability is jeopardized. Demand response programs are designed to elicit changes in customers electric usage patterns. Many DR programs vary the price of electricity over time to motivate customers to change their consumption patterns; this approach is termed price-based demand response. Other DR programs reward customers for reducing their electric loads upon request or for giving the electric utility some level of direct control over the customers electricity-using equipment. These are termed incentive or event-based demand response.

The organized electricity markets in the United States all use some variant of real-time locational marginal price for wholesale electricity. For e.g. PJMs real-time market is a spot market where electricity prices are calculated at five-minute intervals based on the grid operating conditions. Electricity costs are the single largest component of a large commercial and industrial building's operating budget. This is because, such customers are often subject to peak-demand based electricity pricing. In this pricing policy, a customer is charged not only for the amount of electricity it has consumed but also for its peak demand over the billing cycle. High peak loads also lead to a higher cost of production and distribution of electricity. Therefore, these peaks are not only operationally inefficient but also extremely expensive for both the utilities and the end-users. 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 [Mul 2014]; 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

Data from the U.S. Energy Information Administration show that cooling, lighting, and ventilation account for 62 percent of electricity use (A), and space heating dominates natural gas use at 86 percent (B).



Note: Insufficient data were available for electric consumption of Cooking equipment; sum may not total 100% due to rounding.

© E Source; data from the U.S. Energy Information Administration

Figure 1.3: Electricity and natural gas consumption breakdown for commercial buildings in the United States.

programs to help manage their electricity costs. Such customers are increasingly looking to DR programs to help manage their electric utility costs.

However, to take advantage of real-time pricing and DR programs, the *C/I/I* consumers must monitor electricity prices and be flexible in the ways they choose to use electricity.

### 1.1.1 Modeling challenges for cyber-physical energy systems

Energy systems are experiencing a gradual but substantial change in moving away from being non-interactive and manually-controlled systems to utilizing tight integration of both cyber (computation, communications, and control) and physical representations guided by first principles, at all scales and levels. These cyber-physical energy systems are characterized by multiple coordinated controllers (spread across a large building or a campus), the inter-working of different energy types (natural gas, electricity, steam & hot water), the interaction with real-time energy pricing markets, integration of renewable energy sources (solar, wind and on-site generation), automated and grid-friendly building operations, knowledge about the system operations due to sensors and data analytics, information technology and security challenges.

A major challenge with such systems is in accurately modeling the dynamics of the underlying physical system. Dynamic models of such systems can help to design for robustness and stability, to estimate and anticipate problems and to optimize the system for performance and efficiency. Models for energy systems serve as problem solving tools that may have different objectives, such as design, energy performance assessment, optimization, development of controls or simply a better insight of the

system. Consequently, the modeling effort should be tailored to suit the needs of a given application.

Control-oriented predictive models of an energy system’s dynamics and energy consumption, are needed for understanding and improving the overall energy efficiency and operating costs. Combining different aspects like physical infrastructure, communication systems, electricity markets, operations and people leads to a heterogeneous system that is complex to model. The complexity arises due to the interaction between a large number of subsystems and their very different nature with respect to physical dynamics, uncertain behavior, timing and size. System identification techniques are usually used to identify parameters of a physics based white-box or grey-box model which attempt to model the system behavior. The preference for building first principles based models, arises due to the fact that the parameters of these models have a physical meaning and that the model structure is suitable for control synthesis. However, the major barrier in modeling energy systems with white box and grey box approaches, is the user expertise, time, and associated high costs required to develop a mathematical model that accurately reflects reality. This includes the installation cost of retrofitting the building with additional sensors, costs related to the training of the engineering, commissioning and service personnel to implement model-based control and the cost of the necessary engineering effort required for constructing a model.

In the context of buildings and their energy systems, an alternative to traditional rule-based building control is model-based control. In rule-based control the operation of the mechanical and electrical systems is purely “reactive” i.e. it is continuously adjusted in response to weather variations and the thermal load due to building occupants. With a reasonably accurate forecast of future weather and building operating conditions, dynamical models can be used to predict the energy needs of the building over a prediction horizon. Once such a model is made available, it can be used to design an optimal controller that balances comfort and energy usage. To achieve building-level energy-optimality, the model should be able to capture the interaction between physically connected spaces in the building, occupancy schedules, and the state and control input constraints. Such model-based control techniques for building operations require high-fidelity white-box or grey-box models of the building’s thermal envelope and equipment.

Learning dynamical models for buildings using grey/white box approaches is very cost and time prohibitive and requires retrofitting the building with several expensive sensors [Sturzenegger et al. 2015]. This is because buildings are dynamical systems with uncertain and time-varying plant and occupant characteristics. The heat transfer characteristics of a building are highly dependent on the ambient conditions. For instance, thermal properties such as convective heat transfer coefficient of peripheral walls is dependent on outside temperature, wind speed and direction. Also, the dynamics of a building is a function of external factors: ambient weather conditions such as radiative heat flux into the walls and windows, outside humidity and internal fac-

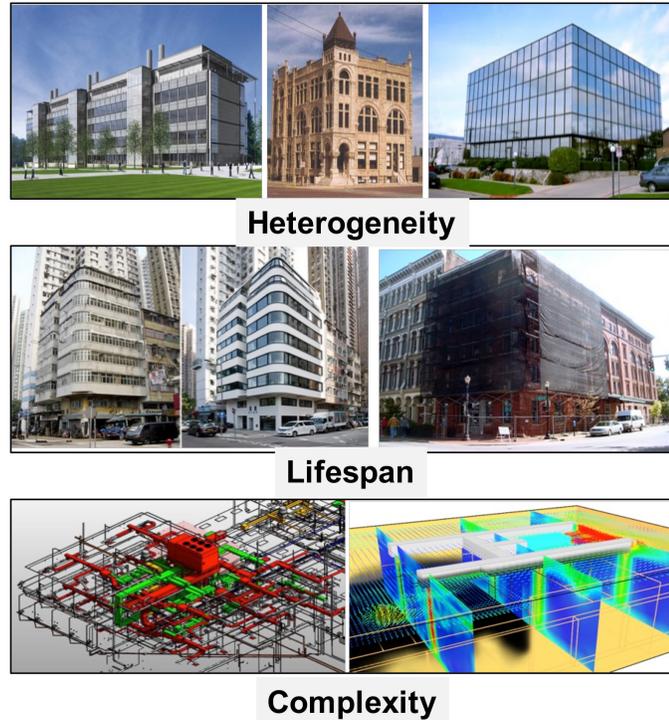


Figure 1.4: Electricity and natural gas consumption breakdown for commercial buildings in the United States.

tors: such as occupancy level, internal heat generation from lighting, and computers. These quantities are highly time-varying and therefore the dynamics of the building and, consequently, parameters of the mathematical model describing the dynamics of the buildings are constantly changing with time.

The modeling difficulty is compounded due to the fact that each building is designed and used in a different way and thus, has to be uniquely modeled, i.e. there is a lot of *heterogeneity* in buildings (Figure 1.4). Unlike the automobiles and aircraft industry, buildings are generally manufactured on the basis of one-off designs and with operational lifetimes of 50 – 100 years. Their operation changes over their long lifetimes and requires remaking the model every-time there is a significant change (Figure 1.4). Even if using white or grey-box models is adequate, there is a tedious task to identify/estimate model parameters from measured data for each subsystem; in other words, to calibrate every subsystem model and its interactions. Another major downside with physics-based modeling is that building construction and materials data is often not easily available and guesses have to be made for nominal parameter values which requires expert know how. Usually a building modeling domain expert will use a software tool (e.g. EnergyPlus [Crawley et al. 2001] and TRNSYS [Klein et al. 2004]) to create the geometry of a building from the building design, add detailed information about material properties, about equipment, operational schedules and then calibrate model parameters to describe the data from the real building.

Furthermore, to attain energy efficiency, control algorithms need to be tailored to the physical properties of the building at hand. Successful experience with one building is usually not repeatable on other buildings, until the domain expert has conducted a few rounds of physical experiments with the new buildings.

The alternative is to use black-box, or completely data-driven modeling approaches, to obtain a realization of the system’s input-output behavior. The primary advantage of using data-driven methods is that it has the potential to eliminate the time and effort required to build white and grey box building models. Listening to real-time data, from existing systems and interfaces, is far cheaper than unleashing hoards of on-site engineers to physically measure and model the building. The key is to identify the key dynamics that help describe the building and its behavior at the macro level. Development of data-driven models for long-term energy consumption prediction, building equipment modeling and occupancy modeling by applying machine learning algorithms such as neural networks and regression has been investigated by several researchers [Edwards et al. 2012, Kialashaki and Reisel 2013b, Vaghefi et al. 2014, Yin et al. 2012, Hong et al. 2013]. However, the benefit obtained from the simplicity of model capture comes at the cost of having non-physical parameters and incapacity for control design and synthesis.

Improved building technology and better sensing is fundamentally redefining the opportunities around smart buildings. Decisions on how best to optimize today’s building operations are becoming so complex, so conflicting and so continuous that advanced algorithms must play a role. But with complexity, comes opportunity. As complex as it can be, it is always best to start simple and learn by listening. Unprecedented amounts of data from millions of smart meters and thermostats installed in recent years has opened the door for systems engineers and data scientists to analyze and use the insights that data can provide, about the dynamics and power consumption patterns of these systems. The challenge now, with using data-driven approaches, is to close the loop for real-time control and decision making in large scale cyber-physical energy systems. Furthermore, providing a technological solution alone is not enough, the solution must provide interpretability and guidance to the system’s facilities managers and domain experts.

## 1.2 Research Goals

The focus of this research is to use data-driven approaches for making model capture easy and efficient for cyber-physical energy systems. The cost and time prohibitive process of grey/white box model capture can be made simpler or eliminated altogether by: (a) reducing the need for sensor retrofits required for estimating the parameters of first principle based models, and (b) relying more on readily available data rather than domain experts for building control-oriented models.

Figure 1.5 shows the comparison between white-box, grey-box and black-box modeling methods for cyber-physical energy systems on the basis of their modeling com-

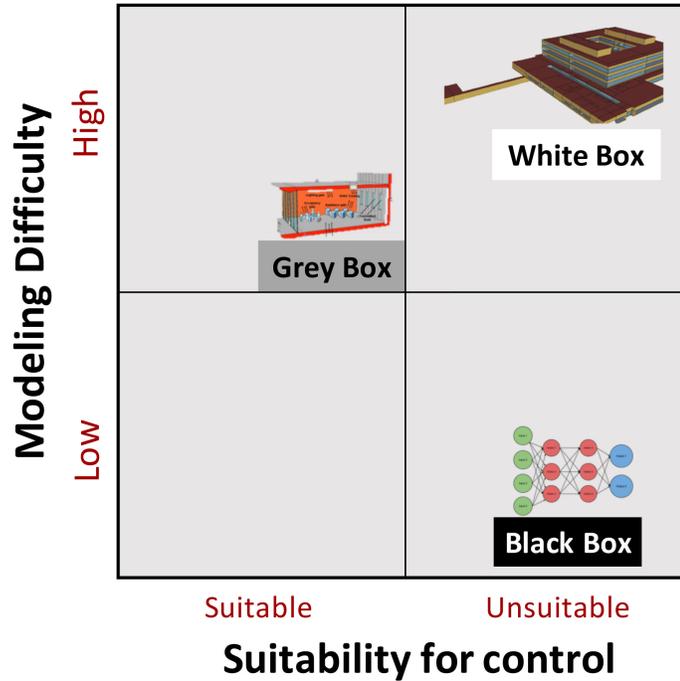


Figure 1.5: Comparison of modeling methods for cyber-physical energy systems.

plexity and their suitability for model based control. As discussed in Section 1.1.1, white box models are extremely cost and time prohibitive to learn and are largely unsuitable for model based control. Therefore, they are in the top right of the comparison chart of Figure 1.5. Although, grey-box models are suitable for model-based control design of buildings they are still expensive to learn and pose significant difficulties in model capture both in terms of modeling time and modeling cost due to additional sensors and domain experts. Lastly, on the bottom right of the comparison chart we have, black-box models. While black-box models are easier to learn, since they only require historical data and the modeling process can be automated, they are not well suited for model-based control design and synthesis. Their use, in the context of buildings and other cyber-physical energy systems is limited to only prediction. Consequently, the goal of this thesis is to devise methods which can make grey-box modeling much easier and low-cost and make black-box modeling more suitable for model-based control synthesis.

Specifically, this thesis answers the following questions:

1. **Low-cost building model capture:** Using a temporary sensor deployment,
  - (a) *Can we determine the benefit obtained from adding additional sensors to a building, on model accuracy and thus, on control performance of a model-based controller ?*
  - (b) *Can we identify the important sensors to be installed and their associated placements for obtaining the best data quality ?*

## 2. Data-driven modeling and control for demand response:

- (a) *Can we develop data-driven methods for building model capture which are suitable for control synthesis ?*
- (b) *Can we provide interpretable energy consumption models for buildings ?*

Small and medium sized buildings constitute more than 90% of the commercial buildings stock in the United States, but only about 10% of such buildings are equipped with a building automation system [Katipamula et al. 2012] and enough sensors to learn first principles based models. The sheer cost of building energy modeling makes it something that is primarily done by researchers and for large projects [Žáčková et al. 2014]. It is not a cost that the retrofit market or most use cases would absorb in the foreseeable future without drastic reductions in the cost of having cheaper, yet accurate model generation. It is the main hurdle towards adopting model-based control design approaches for cyber-physical energy systems and the goal of this thesis is to remove this hurdle.

The goal of this dissertation is to develop data-driven methods of model capture and control for cyber-physical energy systems, in particular for commercial buildings which are subject to time varying electricity prices and demand response tariffs by the electric utility companies. The proposed methods are integrated into ready to use and deployable tools. Several comprehensive case studies in data-driven building modeling, model-based design and demand response for buildings have been conducted to show the potential of the approach. The challenges associated with answering each research question are briefly described below:

### 1.2.1 Low-cost building model capture

Learning mathematical models of buildings from sensor data has a fundamental property that the model can only be as accurate and reliable as the data on which it was trained. Any measurement exhibits some difference between the measured value and the true value and, therefore, has an associated uncertainty. Non-uniform measurement conditions, limited sensor calibration, the amount of sensor data and the amount of excitation of the plant make the measurements in the field vulnerable to errors.

A major challenge to the use of models for buildings controls lies in understanding the impact of uncertainty in the model structure, the estimation algorithm, and the quality of the training data. It is known that the quality of the training data, characterized by uncertainty, depends on factors such as the accuracy of sensors, sensor placement and density, and the assumption that air is well mixed. It is intuitive to assume that installing additional sensors to obtain higher quality training data should result in more accurate models, which will further result in better performance of a model-based controller (e.g., Model Predictive Control (MPC)).

In the case of using sensor data for training inverse models (e.g. grey box or black box), the goal is to provide maximum benefit, in terms of model accuracy, for

the least sensor cost. One approach to obtaining the necessary data for generating a high-fidelity building model involves installing low-cost temporary wireless sensors and measuring the necessary model inputs and outputs for a sufficient period of time to enable training and testing of the building model.

However an understanding of the cost-benefit associated with adding additional sensors to a building is either limited or missing altogether.

### 1.2.2 Data-driven control oriented modeling for demand response

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 [Crawley et al. 2001]) is very cost and time prohibitive and requires retrofitting the building with several sensors [Sturzenegger et al. 2015];
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.
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 assess 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.
4. **Interpretability of modeling and control:** Predictive models for buildings, regardless how sophisticated, lose their effectiveness unless they can be inter-

preted by human experts and facilities managers in the field. Therefore, the required solution must be transparent, human centric and highly interpretable.

Regardless of the specific DR program, upon receiving a notification for a demand response event, the building’s facilities manager must determine an appropriate or optimal control strategy to achieve the required power curtailment level. These control strategies can include adjusting zone temperature set-points, increasing 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 assess 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. Moreover, the performance of any DR strategy will vary due to the outside weather conditions, during the DR event.

The goal with data-driven demand response is to make the best of both worlds; i.e. keep the 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.

### 1.3 Contributions

The contributions of this dissertation, are summarized below:

- **Sensor data quality vs model accuracy:** We have developed, Model-IQ, a method to quantify how the quality of the data from sensors affects the accuracy of a building model. This is based upon our input uncertainty propagation and analysis algorithm. This was published in [Behl et al. 2014b]
- **Model accuracy vs control performance:** Using Model-IQ we empirically establish the effect of model accuracy on the performance of a model-based closed-loop controller for buildings, such as mode-predictive control.
- **Sensor placement and density:** Using non-parametric statistical methods and temporary sensors, we have developed a method to quantify the bias in a sensor measurement due to the location and density of sensors with applications to determine the near optimal sensor positions. This work has been published in [Behl et al. 2014a]
- **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.

- **DR Strategy Evaluation:** We present an approach for building auto-regressive trees and apply it for demand response strategy evaluation. Our models take into account the state of the building and weather forecasts to help choose the best DR strategy among several pre-determined strategies.
- **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.
- **DR-Advisor (Demand Response-Advisor),** Our methods have been integrated into a MATLAB based tool called DR-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. This work has resulted in a series of publications [Behl et al. 2016, Behl and Mangharam 2015a,b, Behl et al. 2015, Behl and Mangharam 2015c].
- **Energy analytics with regression trees:** We present a methodology to preserve interpretability of ensemble regression trees based methods and use the ensembles for providing energy analytics, beyond demand response, to the facilities manager based on data.

## 1.4 Organization

The remaining chapters of this dissertation are organized as follows:

### Chapter 2: Building modeling with first principles

We start by describing the different types of modeling methods used for building modeling, namely white-box, grey-box and black-box modeling. We then present in detail each of the modeling methods and the limitations and challenges associated with each method. We then present the basics of a grey-box modeling method called the RC model of the heat transfer and use it to create a state-space dynamical model of a single zone. This chapter provides the reader with the background of building energy modeling and describes why the modeling techniques are time consuming and expensive.

### Chapter 3: Low-Cost Building Model Capture

This chapter uses the modeling principles described in chapter 2 to address the first research goal of this dissertation i.e. how can we lower the cost of additional sensors required to tune grey-box building models. We begin this chapter with a description of the input uncertainty analysis algorithm in Section 3.2. Given a model structure and an input-output data-set, the algorithm analyzes the trade-off between sensor data quality and model accuracy. This is the basis of the Model-IQ toolbox. We next, describe how we can use empirical methods to evaluate the trade-off between model accuracy and the performance of a closed loop model based controller such as model predictive control in Section 3.4. Section 3.5 then describes a non-parametric statistical approach for computing sensor bias and for sensor placement and density insights. We present a comprehensive case study in Section 3.6 using data from a real office building and show how Model-IQ can be used for understanding sensor deployment trade-offs for grey-box modeling of this building. We conclude the chapter with a summary of related work in Section 3.7 and a short discussion in Section 3.8

#### **Chapter 4: Data-driven modeling with regression trees**

Chapter 4 presents an overview of data-driven modeling using regression tree based algorithms. In this chapter, we formally introduce the principles of data-driven approaches and provide illustrative examples of how regression trees are constructed. The methods described in this chapter form the basis of the data-driven modeling and control approaches described later in Chapter 5. In this chapter we cover the seminal algorithm to construct regression trees, CART, followed by brief descriptions about extensions of that algorithm in the form of cross-validated trees, boosted regression trees, random forests and the M5 model based regression trees.

#### **Chapter 5: Data-Driven Modeling and Control for DR**

This chapter covers the methods which fulfill the second research goal of this dissertation i.e. making data-driven model suitable for close loop control synthesis and apply them to the problem demand response of cyber-physical energy systems. We first present a description of the three challenging problems in demand response in Section 5.1. This is followed by Section 5.2, which presents a description of how the regression trees algorithms presented in Chapter 4 can be utilized to solve the DR challenges. Section 5.3, presents a new algorithm to perform real-time control with regression trees and applied to the DR synthesis problem. Section 5.4 makes a compelling case for regression trees being a suitable choice of data-driven models for real-time DR. Section 5.5 describes the MATLAB based DR-Advisor toolbox. Section 5.6 presents a comprehensive case study with DR-Advisor on data from 8 real buildings on Penn's campus, a large office building, a virtual test-bed and evaluation of the tool on bench-marking data from ASHRAE's energy prediction competition. In Section 5.7, a detailed survey of related work has been presented. We conclude this chapter in Section 5.8 with a summary of our results.

#### **Chapter 6: Energy Analytics**

This chapter describes how the regression trees based models described in Chapters 4 and 5 can also be used for energy analytics in order to assist a facilities manager understand more about the energy usage of his building/campus. In this chapter, we describe the extension to regression trees which allows us to build data-structures which can be searched to provide answers to open ended queries about the system from the facilities manager. At the core of the chapter is the idea of converting the regression tree models into a searchable and interpretable knowledge database. We demonstrate this idea using data from a real building on Penn's campus.

Finally, **Chapter 7** draws the conclusion of the dissertation with a discussion on the possible directions for future work.

## Chapter 2

# Building modeling with first principles

For the simulation and control of buildings, models of appropriate accuracy and complexity are essential. A variety of simulation tools that have been developed for building energy analysis, simulation, control design and building design automation. In this chapter, we present an overview of different techniques for building modeling using first principles of physics and heat transfer. We will present examples and describe tools which are often used for building such models.

The main objective of an HVAC system for air temperature control is to reject disturbances due to outside weather condition and internal heat gain/loss caused by occupants, lighting and plug-in appliances.

Therefore, the building model must accurately capture the thermal response of the building to the different disturbances and due to the hVAC control. Building models for cyber-physical energy systems can be broadly classified into three categories as shown in Figure 2.1:

1. *White-box* models are based on the laws of physics and permit accurate and microscopic modeling of the building system. High fidelity building simulation programs like EnergyPlus and TRNSYS [Crawley et al. 2008] fall into this category. Although such models provide a high degree of accuracy they are unsuitable for control design due to their high level of complexity and a large number of parameters. Furthermore, the process of constructing the model and tuning the parameters with limited data is very time consuming and not cost effective.
2. *Black-box models* are not based on physical behaviors of the system but rely on the available data to identify the model structure. These models are often purely statistical and have a simple structure.
3. *Grey-box models* fall in between the above two categories. A reduced order model structure is chosen loosely based on the physics of the underlying system



Figure 2.1: Modeling categories for cyber-physical energy systems

and the available data is used to estimate the values of the model parameters. These models are suitable for model-based control design and rely on first principles based modeling.

We now describe white-box and grey-box modeling methods for buildings in detail:

## 2.1 White-box modeling

The white-box approach for modeling whole building energy consumption and thermal dynamics involves a domain expert or a building modeler using the software tool they have most experience with to create the geometry of a building, layer it with detailed metrics encoding material properties, adding equipment currently or expected to be in the building, with anticipated operational schedules. For instance, an EnergyPlus building model has  $\sim 3,000$  inputs for a normal residential building with very specific details that most energy modelers do not have the sources of data for. As shown in Figure 2.2, the domain expert must import all the available data about the building’s geometry, construction materials and operation into a building energy simulation software in the form of values of the parameters of the building. Often, all the information required to initialize all the parameters of a model is not available, which leads to the domain expert guessing the values of certain model parameters. This process is further compounded by the fact that there is always a gap between the as-designed and as-built structure. Due to the sources of variance involved in the input process, white-box building models must often be painstakingly tuned manually to

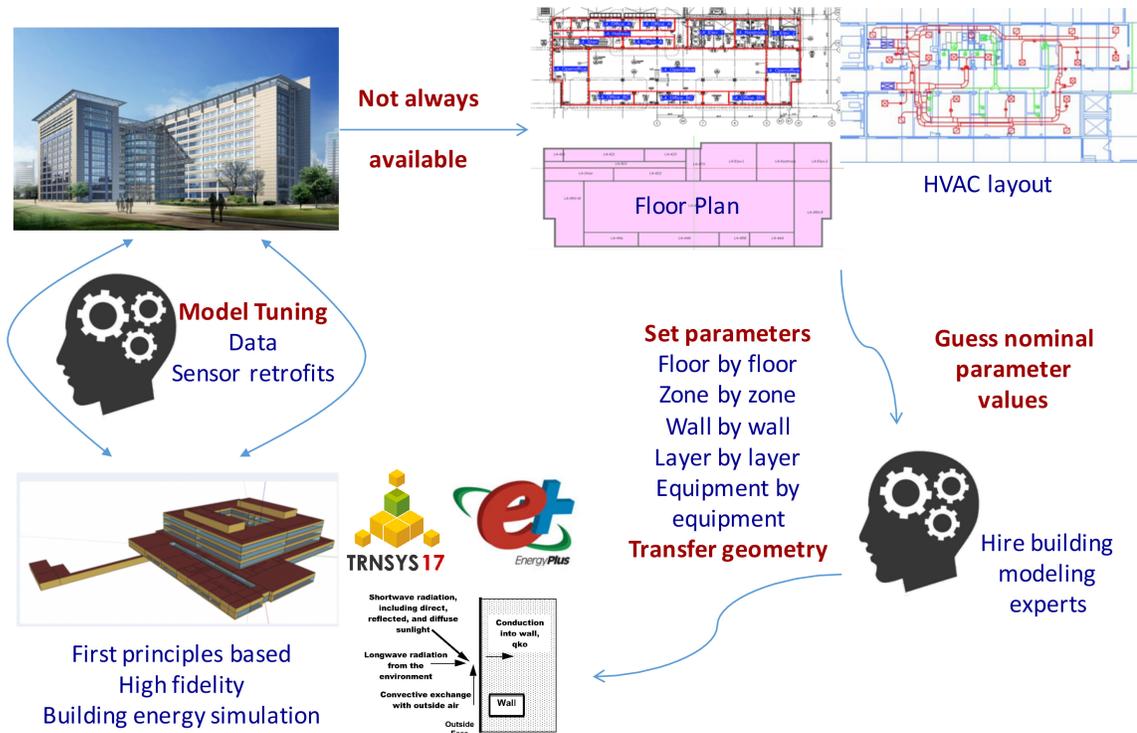


Figure 2.2: White-box building modeling approach for buildings.

match the measured data from the building. This tuning process is highly subjective and repeatable across neither modelers nor software packages.

Furthermore, tuning such a software model also requires that the building be retrofitted with enough sensors which can generate high resolution spatial and temporal data which can be used to tune the parameters of the building model. The two most popular softwares for building energy modeling are EnergyPlus and TRNSYS. We briefly describe them next.

### 2.1.1 EnergyPlus

EnergyPlus is the U.S. Department Of Energy (DOE) whole building energy simulation software for modeling building heating, cooling, lighting, ventilation, other energy flows, and water use [Crawley et al. 2001]. Modeling the performance of a building with EnergyPlus enables building professionals to optimize the building design for energy and water usage. EnergyPlus is typically used by engineers, architects, and researchers to perform building energy analysis. EnergyPlus handles integrated, simultaneous solution where the building response and the primary and secondary systems are tightly coupled and performs iteration when necessary. It also supports sub- hourly, user-definable time steps for the interaction between the thermal zones and the environment, and variable time steps for interactions between the thermal zones and the HVAC systems. EnergyPlus supports ASCII text-based weather, in-

put, and output files including hourly or sub-hourly environmental conditions, and standard and user definable reports. It leverages heat balance-based techniques for building thermal loads that allow for simultaneous calculation of radiant and convective effects at both interior and exterior surface during each time step. Transient heat conduction through building elements such as walls, roofs, floors, etc. using conduction transfer functions are also supported. EnergyPlus also features combined heat and mass transfer model that accounts for moisture absorption/desorption. Thermal comfort models based on activity, inside dry bulb, and humidity as well as anisotropic sky model for improved calculation of diffuse solar on tilted surfaces are available. Detailed and advanced fenestration calculations in EnergyPlus includes controllable window blinds, electrochromic glazings, layer-by-layer heat balances that allow proper assignment of solar energy absorbed by window panes, and a performance library for numerous commercially available windows. Daylighting controls including interior illuminance calculations, glare simulation and control, luminaire controls, and the effect of reduced artificial lighting on heating and cooling are the other features supported by EnergyPlus. EnergyPlus is regarded as a very detailed and high-fidelity whole building simulation tool. This makes EnergyPlus a good choice if the objective of the study is to analyze and compare the effects of some building features. On the other hand, being a very detailed modeling tool, EnergyPlus is not a program of choice if the task is model-based control design.

### 2.1.2 TRNSYS

TRNSYS, is a TRaNsient SYstems Simulation Program with a modular structure [Crawley et al. 2008]. It recognizes a system description language in which the user specifies the components that constitute the system and the manner in which they are connected. The TRNSYS library includes many of the components commonly found in thermal and electrical energy systems, as well as component routines to handle input of weather data or other time-dependent forcing functions and output of simulation results. The modular nature of TRNSYS gives the program tremendous flexibility, and facilitates the addition to the program of mathematical models not included in the standard TRNSYS library. TRNSYS is well suited to detailed analyses of any system whose behavior is dependent on the passage of time. Main applications include: solar installations (solar thermal and photovoltaic systems), renewable energy systems, co-generation, and fuel cells. However, much like EnergyPlus, even a TRNSYS model has thousands of parameters whose nominal values must be provided by the domain expert based on the design, construction and operation of a the real building.

As stated, earlier, there is always a gap between the modeled and the real building and the domain expert must manually tune the model to match the measured data from the building. Recently, there has been efforts, like in [R et al. 2012], to automate the parameter tuning process with white-box building models built in EnergyPlus. The development of an such an autotuning capability to adapt building models to building performance data would significantly facilitate market adoption of

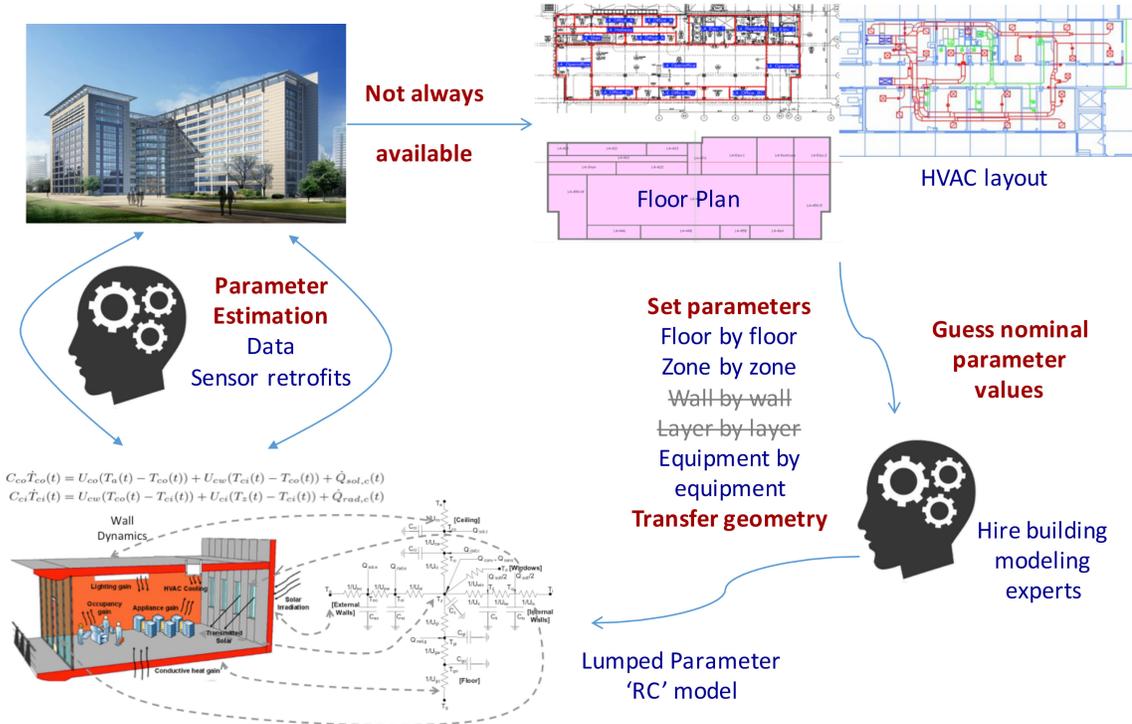


Figure 2.3: Grey-box building modeling approach for buildings.

energy modeling software. However, it would still require one to build the EnergyPlus model first from the building design, construction and operation information. Such approaches are also limited in their impact on reducing the cost of additional sensor retrofits required to tune the models.

## 2.2 Grey-box models

The grey-box building modeling approach is very similar to the white-box modeling approach expect for two key differences:

- Firstly, the domain expert or the building modeler does not need to consider the design , construction details and operation of the building is as much detail as the white box model. Certain details are aggregated or 'lumped' together to reduce model complexity; and
- Secondly, instead of using a software simulation to add parameter values, in grey-box modeling, the model structure itself is built from scratch, using first principles, by the domain expert as shown in Figure 2.3.

However, similar to the case of white-box approaches, the domain expert must manually tune the parameters of the grey-box model to match real data from the building. This process is described in detail next.

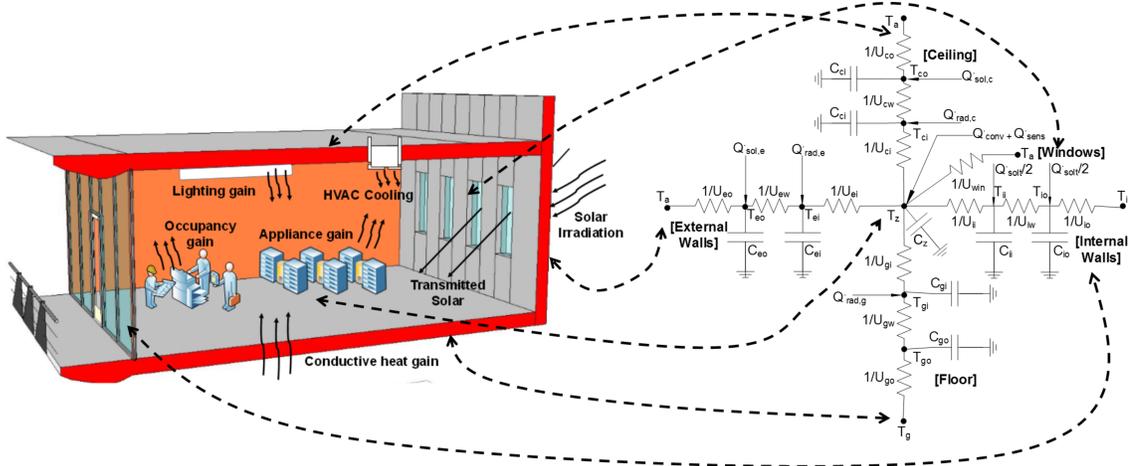


Figure 2.4: RC lumped-parameter model representation for a thermal zone obtained from information about the zone geometry and usage.

The American Society of Heating, Refrigerating, and Air Conditioning (ASHRAE) has established extensive methodologies for calculation of heating and cooling loads in [ASHRAE 2009]. Such methodologies involve heat balance analysis. It is a straightforward physics-based and comprehensive methodology that involves calculating a node-to-node heat balance of a room through consideration of conductive, convective, and radiative heat transfer mechanisms, a node can either be wall or air in the room. A commonly used grey-box representation of the thermal response of a building due, which uses this ASHRAE methodology, is based on a lumped parameter Resistive-Capacitive (RC) network. This approach for modeling buildings represents has been used widely, e.g. in [Braun and Chaturvedi 2002, McKinley and Alleyne 2008, Dewson et al. 1993]. Figure 2.4 shows an example of such a model for a single zone, as used in [Braun and Chaturvedi 2002]. In this representation, the central node of the RC network represents the zone temperature  $T_z$  ( $^{\circ}\text{C}$ ). The geometry of the zone is divided into different kinds of surfaces, each of which is modeled using a ‘lumped-parameter’ branch of the network. For instance, all the external walls of the zone are lumped into a single wall with 3R2C (3 resistances and 2 capacitance) parameters. The same process is applied to the ceiling, the floor and the internal (or adjacent) walls of the zone. The zone is subject to several (heat) disturbances which are applied at different nodes in the network in the following manner:

- (a) solar irradiation on the external wall  $\dot{Q}_{sol,e}$  (W) and the ceiling  $\dot{Q}_{sol,c}$  (W) is applied on the exterior node of the lumped wall,
- (b) incident solar radiation transmitted through the windows  $\dot{Q}_{solt}$  (W) is assumed to be absorbed by the internal and adjacent walls,
- (c) radiative internal heat gain  $\dot{Q}_{rad}$  (W) which is distributed with an even flux to the walls and the ceiling,

Table 2.1: List of parameters

$U_{*o}$	convection coefficient between the wall and outside air
$U_{*w}$	conduction coefficient of the wall
$U_{*i}$	convection coefficient between the wall and zone air
$U_{win}$	conduction coefficient of the window
$C_{**}$	thermal capacitance of the wall
$C_z$	thermal capacity of zone $z_i$
	$g$ : floor; $e$ : external wall; $c$ : ceiling; $i$ : internal wall

(d) the convective internal heat gain  $\dot{Q}_{conv}$ (W) and the sensible cooling rate  $\dot{Q}_{sens}$ (W) is applied directly to the zone air,

(e) the zone is also subject to heat gains due to the ambient temperature  $T_a$ (°C), ground temperature  $T_g$ (°C) and temperatures in other zones which are accounted for by adding boundary condition nodes to each branch of the network.

The list of all parameters in the model and their descriptions is given in Table 2.1. Given this model, the nodal equations for the lumped external wall and the ceiling network are:

$$\begin{aligned}
 C_{eo}\dot{T}_{eo}(t) &= U_{eo}(T_a(t) - T_{eo}(t)) + U_{ew}(T_{ei}(t) - T_{eo}(t)) + \dot{Q}_{sol,e}(t) \\
 C_{ei}\dot{T}_{ei}(t) &= U_{ew}(T_{eo}(t) - T_{ei}(t)) + U_{ei}(T_z(t) - T_{ei}(t)) + \dot{Q}_{rad,e}(t) \\
 C_{co}\dot{T}_{co}(t) &= U_{co}(T_a(t) - T_{co}(t)) + U_{cw}(T_{ci}(t) - T_{co}(t)) + \dot{Q}_{sol,c}(t) \\
 C_{ci}\dot{T}_{ci}(t) &= U_{cw}(T_{co}(t) - T_{ci}(t)) + U_{ci}(T_z(t) - T_{ci}(t)) + \dot{Q}_{rad,c}(t)
 \end{aligned} \tag{2.1}$$

$$\begin{aligned}
 C_{eo}\dot{T}_{eo}(t) &= U_{eo}(T_a(t) - T_{eo}(t)) + U_{ew}(T_{ei}(t) - T_{eo}(t)) \\
 &\quad + \dot{Q}_{sol,e}(t)
 \end{aligned} \tag{2.2a}$$

$$\begin{aligned}
 C_{ei}\dot{T}_{ei}(t) &= U_{ew}(T_{eo}(t) - T_{ei}(t)) + U_{ei}(T_z(t) - T_{ei}(t)) \\
 &\quad + \dot{Q}_{rad,e}(t)
 \end{aligned} \tag{2.2b}$$

$$\begin{aligned}
 C_{co}\dot{T}_{co}(t) &= U_{co}(T_a(t) - T_{co}(t)) + U_{cw}(T_{ci}(t) - T_{co}(t)) \\
 &\quad + \dot{Q}_{sol,c}(t)
 \end{aligned} \tag{2.2c}$$

$$\begin{aligned}
 C_{ci}\dot{T}_{ci}(t) &= U_{cw}(T_{co}(t) - T_{ci}(t)) + U_{ci}(T_z(t) - T_{ci}(t)) \\
 &\quad + \dot{Q}_{rad,c}(t)
 \end{aligned} \tag{2.2d}$$

Similarly, one can write the equations for the dynamics of the nodes of the floor and internal wall network. The law of conservation of energy gives us the following heat balance equation for zone

$$\begin{aligned}
 C_z\dot{T}_z(t) &= U_{ei}(T_{ei}(t) - T_z(t)) + U_{ci}(T_{ci}(t) - T_z(t)) \\
 &\quad + U_{ii}(T_{ii}(t) - T_z(t)) + U_{gi}(T_{gi}(t) - T_z(t)) \\
 &\quad + U_{win}(T_a(t) - T_z(t)) + \dot{Q}_{conv}(t) + \dot{Q}_{sens}(t)
 \end{aligned} \tag{2.3}$$

Differential equations in 2.1 and 2.3 can be combined to give a state space model of the system. Define  $x = [T_{eo}, T_{ei}, T_{co}, T_{ci}, T_{go}, T_{gi}, T_{io}, T_{ii}, T_z]^T$  to be the state vector of all node temperatures. The input  $u$  is a vector of all the inputs to the systems, i.e.  $u = [T_a, T_g, T_i, \dot{Q}_{sol,e}, \dot{Q}_{sol,c}, \dot{Q}_{rad,e}, \dot{Q}_{rad,c}, \dot{Q}_{rad,g}, \dot{Q}_{solt}, \dot{Q}_{conv}, \dot{Q}_{sens}]^T$ . The control input to the zone is the sensible cooling rate  $\dot{Q}_{sens}$ . The cooling rate can be controlled by changing the mass flow rate of cold air which enters the zone (in case of cooling) or by changing the set point of the supply air temperature. The rest of the inputs are disturbances to the zone or non-manipulated inputs. The elements of the system matrices depend nonlinearly on the  $U$  and  $C$  parameters. Let us consider  $\theta = [U_{eo}, U_{ew}, U_{ei}, \dots, C_{io}, C_{ii}]^T$  as a vector of all the parameters of the model. Then the state space equations have the following representation which emphasizes the parameterization of the system matrices.

$$\begin{aligned} \dot{x}(t) &= A_\theta x(t) + B_\theta u(t) \\ y(t) &= C_\theta x(t) + D_\theta u(t) \end{aligned} \tag{2.4}$$

The output matrix  $C_\theta$  and the feed-forward matrix  $D_\theta$  depend on the outputs of interest. For instance, if the output of the model is the zone temperature then  $C_\theta = [0, 0, \dots, 0, 1]$ , which is a row vector with all entries equal to zero except the last entry corresponding to the zone temperature  $T_z$  equal to one. In this case  $D_\theta = 0$ , the null matrix.

This model structure is based on the underlying assumption that the air inside the zone is well mixed and hence it can be represented by a single node. Furthermore, only one-dimensional heat transfer is assumed for the walls and there is no lateral temperature difference. The parameters of the model are assumed to be time invariant.

### 2.2.1 Parameter Estimation (Model Training)

We discretize the continuous-time model in equation 2.4 with the measurement sampling time to obtain a discrete-time state space model:

$$\begin{aligned} x(k+1) &= \hat{A}_\theta x(k) + \hat{B}_\theta u(k) \\ y(k) &= \hat{C}_\theta x(k) + \hat{D}_\theta u(k) \end{aligned} \tag{2.5}$$

The goal of parameter estimation is to obtain estimates of the parameter vector  $\theta$  of the model from input-output time series measurement data. The parameter search space is constrained both above and below by  $\theta_l \leq \theta \leq \theta_u$ . For a given parameter vector  $\theta$ , the model, given by equation 2.5, can be used to generate a time series of the zone air temperature  $T_{z_\theta}$  using the measured time series data for the inputs  $u(k)$ . The subscript  $\theta$  denotes that the temperature value  $T_{z_\theta}$  is the predicted value using the model with parameters  $\theta$  and the inputs  $u$ . This model generated time series  $T_{z_\theta}$  may then be compared with the corresponding observed values of the zone temperature

$T_{z_m}$ , and the difference between the two is quantified by a statistical metric. The metric usually chosen is the sum of the squares of the differences between the two time series. The parameter estimation problem is to find the parameters  $\theta^*$ , subject to  $\theta_l \leq \theta \leq \theta_u$ , which result in the least square error between the predicted and the measured temperature values, i.e.

$$\theta^* = \arg \min_{\theta_l \leq \theta \leq \theta_u} \sum_{k=1}^N (T_{z_m}(k) - T_{z_\theta}(k))^2 \quad (2.6)$$

where the summation is over the N data points of the input-output time series under investigation.

The least square optimization of equation 2.6 is a constrained minimization of a non-linear objective. It is numerically solved using a trust region reflective algorithm [Coleman and Li 1996] such as the Levenberg-Marquardt [Moré 1978] algorithm. A well known problem with non-linear search algorithms is the problem of the solution getting stuck at a local minima. It is desirable that the initial parameter estimates  $\theta_0$  are as close as practicable to their (unknown) optimal (true) values. The initial values of the parameters can be estimated from the details of the constructions and materials used for the building. Generally speaking, the further the initial guess of the parameters is away from the true values, the bigger the search region is for the optimization. As search region grows, it becomes more likely that the estimation process will converge to a local optimum.

## 2.3 Concluding remarks

The grey-box modeling framework is suitable for model-based control design due to the reduced model complexity and less number of model parameters. However, the model development process is not easily scalable. From one building to another, one needs to repeat all the training and calibration. Much like, white-box modeling, the modeling cost and time required to build accurate grey-box models is very high.

# Chapter 3

## Low-Cost Building Model Capture

One of the biggest challenges in the domain of cyber-physical energy systems is in accurately capturing the dynamics of the underlying physical system. In the context of buildings, the modeling difficulty arises due to the fact that each building is designed and used in a different way and therefore, it has to be uniquely modeled. Furthermore, each building system consists of a large number of interconnected subsystems which interact in a complex manner and are subjected to time varying environmental conditions.

Control-oriented models are needed to enable optimal control in buildings. Models for building load requirements and temperature dynamics are particularly important and often the existing sensor measurements available from the building heating ventilation and air conditioning (HVAC) system are not sufficient for use in training and testing a model. Learning mathematical models of buildings from sensor data has a fundamental property that the model can only be as accurate and reliable as the data on which it was trained. Any measurement exhibits some difference between the measured value and the true value and, therefore, has an associated uncertainty. Non-uniform measurement conditions, limited sensor calibration, the amount of sensor data and the amount of excitation of the plant make the measurements in the field vulnerable to errors. With appropriate understanding of the sources of errors and their effect on the operating cost of the controller, the errors associated with some of these sources can be minimized. In the case of using sensor data for building first principles based models (e.g. grey box or white box), the goal is to provide maximum benefit, in terms of model accuracy, for the least sensor cost.

Small and medium sized buildings constitute more than 90% of the commercial buildings stock, but only about 10% of such buildings are equipped with a building automation system [Katipamula et al. 2012]. A proposed approach to obtaining the necessary data for generating a high-fidelity building models involves installing temporary sensors and measuring the necessary model inputs and outputs to enable training and testing of the building model. A recent report by the Department of Energy (DoE)[Butters et al. 2011] also emphasizes a program focused on adapting wireless sensor technology into an inexpensive system for energy-efficiency optimiza-

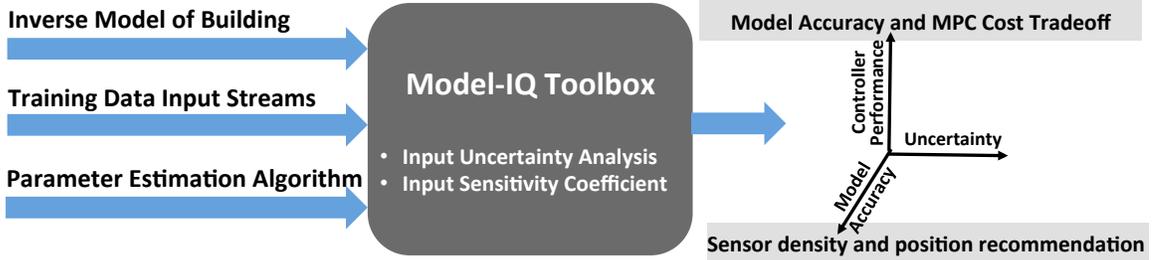


Figure 3.1: Model-IQ Toolbox uncertainty analysis for building controls.

tion of selected zones in commercial buildings. Therefore, there is economic value in performing an uncertainty analysis for such installations in order to better access the tradeoffs involved in sensor installation cost and building performance.

However, a major challenge to the use of models for buildings controls lies in understanding the impact of uncertainty in the model structure, the estimation algorithm, and the quality of the training data. It is known that the quality of the training data, characterized by uncertainty, depends on factors such as the accuracy of sensors, sensor placement and density, and the assumption that air is well mixed. It is intuitive to assume that installing additional sensors to obtain higher quality training data should result in more accurate models, which will further result in better performance of a model based controller (e.g. Model Predictive Control (MPC)). However an understanding of the cost-benefit associated with adding additional sensors to a building is either limited or missing altogether. The reason for this is twofold:

- a) It is not clear how the quality of the data from sensors affects the accuracy of a building model. This is because data quality is only one of several factors that may affect model accuracy.
- b) It is hard to analytically establish the effect of model accuracy on the performance of a model based closed-loop controller for buildings.

The goal of this work is to study the cost-benefit effect of adding temporary low cost wireless sensors to a zone to improve model accuracy and subsequently enable low-cost implementation of advanced control schemes such as Model Predictive Control (MPC) (see 3.1). We first perform an input uncertainty analysis to classify the effect of the quality of training data on the accuracy of a “grey-box” building inverse model. We show how additional sensors can affect the training data quality. We empirically evaluate the tradeoff between model accuracy and MPC performance, i.e. all things being the same, for different model accuracies how MPC performance varies.

### 3.1 Uncertainty in Building Modeling

Uncertainty in modeling the dynamics of the underlying physical system is largely due to (a) the model structure, (b) the performance of the parameter estimation

algorithm and (c) the uncertainty in the training data. In this effort, we assume the first two are fixed and focus on understanding the effect of uncertainty of the training data from the building and environment sensors on the overall performance of a model-based controller for the buildings HVAC systems.

The uncertainty in training data can be characterized in two ways: bias error or random error. Biases are essentially offsets in the observations from the true values. Bias error can also be referred to as the systematic error, precision or fixed error. The bias in the sensor measurement is due to a combination of two reasons. The first reason is the sensor precision. The best corrective action in this case is to ascertain the extent of the bias (using the data-sheet or by re-calibration) and to correct the observations accordingly. The sensor may also exhibit bias due to its placement, especially if it is measuring a physical quantity which has a spatial distribution, e.g. air temperature in a zone. In this case, it is hard to detect or estimate the bias unless additional spatially distributed measurements are obtained. Random error is an error due to the unpredictable (e.g. measurement noise) and unknown extraneous conditions that can cause the sensor reading to take some random values distributed about a mean.

The density and location of sensors in a zone affects the deviation of the measured value from the true value. For instance, a zone thermostat sensor placed too close to the wall, window, supply or return air duct can introduce a bias in the zone temperature measurement. A bias in the zone temperature value can lead to wastage of energy and discomfort with simple zone air control schemes like On-Off and PID control. Both the controllers are *model-independent* i.e. they only used the measurement of the *process variable* (which is the room temperature in this case) to compute the control signal that will either track the set-point (PID) or keep the temperature bounded around the set-point (On-Off). As we proceed to apply *model-based control* schemes for building retrofits, the bias or the uncertainty in the measured data will also influence the accuracy of the model itself which in turn affects the performance of the model based controller.

## 3.2 Model-IQ approach

In this section, we describe the Model-IQ approach for analyzing uncertainty propagation for building grey-box models. The accuracy of the grey-box building model, described in Section 2.2, depends primarily on the following three factors:

- (a) **The structure of the model** which depends on the extent to which the model respects the physics of the underlying physical system,
- (b) **The performance of the estimation algorithm.** As discussed previously, in the case of non-linear estimation the performance of the algorithm depends heavily on the nominal values of the parameters, and

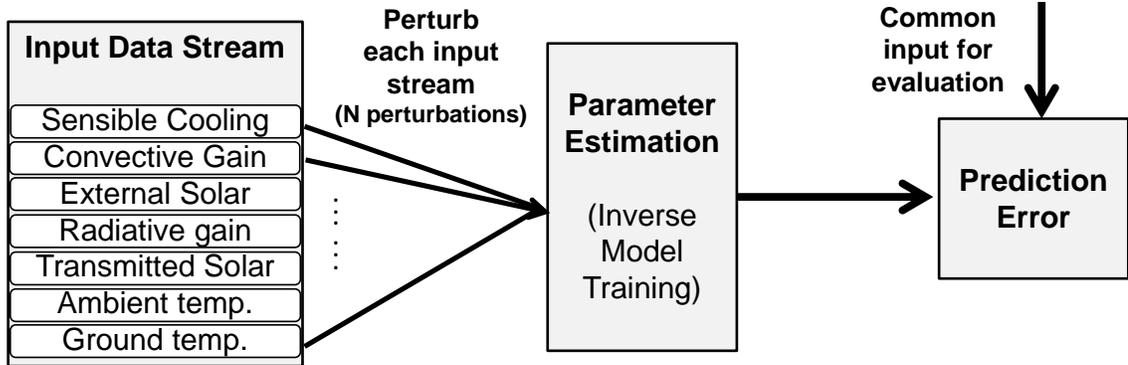


Figure 3.2: Overview of the Model-IQ input uncertainty analysis methodology, an offline method to confirm the influence of each training input on the accuracy of the model.

- (c) **The quality of the training data**, which can be characterized by its uncertainty.

The main premise of the input uncertainty propagation is that once the model structure and the parameter algorithm are fixed, one can study the influence of the uncertainty in the training data on the accuracy of the model using virtual simulations which utilize artificial data-sets. Figure 3.2 shows an overview of the Model-IQ approach. We introduce an uncertainty bias in each of the training data streams in form of bounded perturbations around the unperturbed (nominal) values. This results in the creation of artificial training data sets each of which is similar to the original unperturbed data-set except for one input data stream. For each artificial data-set, we train a new grey-box model and calculate its test error. A common test data set allows us to fairly compare the accuracy of the models in terms of their test root mean square error (RMSE). This allows us to quantify the effect of uncertainty bias in each input stream on the accuracy of the grey-box model. For the remaining part of the section we will use an example of a building modeled in TRNSYS for conducting the input uncertainty propagation analysis.

### 3.2.1 Example with TRNSYS Model

The virtual test-bed used for the input uncertainty analysis is a single zone building modeled in TRNSYS as shown in Figure 3.4. The building is north facing, has 4 external brick walls each of which contains a large window, a concrete ceiling and a floor. For the simulation we use the Philadelphia – TMY2 weather file which provides the ambient temperature and solar irradiation data for modeling. The building is assumed to be equipped with a HVAC system with a maximum cooling power of 3.5kW. In addition to the heat gains due to outside temperature and incident irradiation, the building is also subject to internal heat gains from occupants, appliances and lighting fixtures. Without lack of generality we only consider the case when the building is being cooled. The operation of the heating system would be similar.

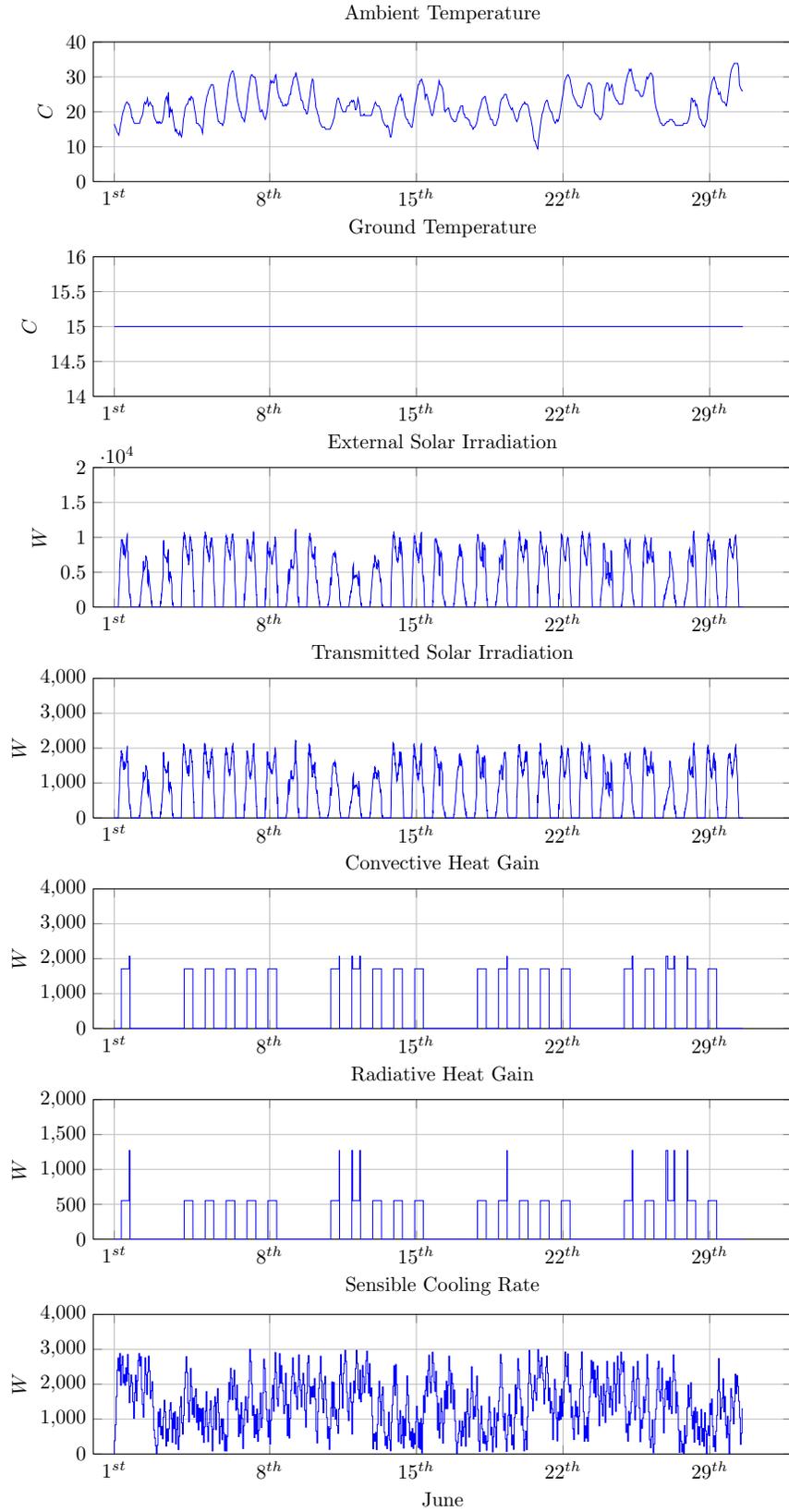


Figure 3.3: Inputs sampled at 2 minutes during the month of June were used for training the single zone model.



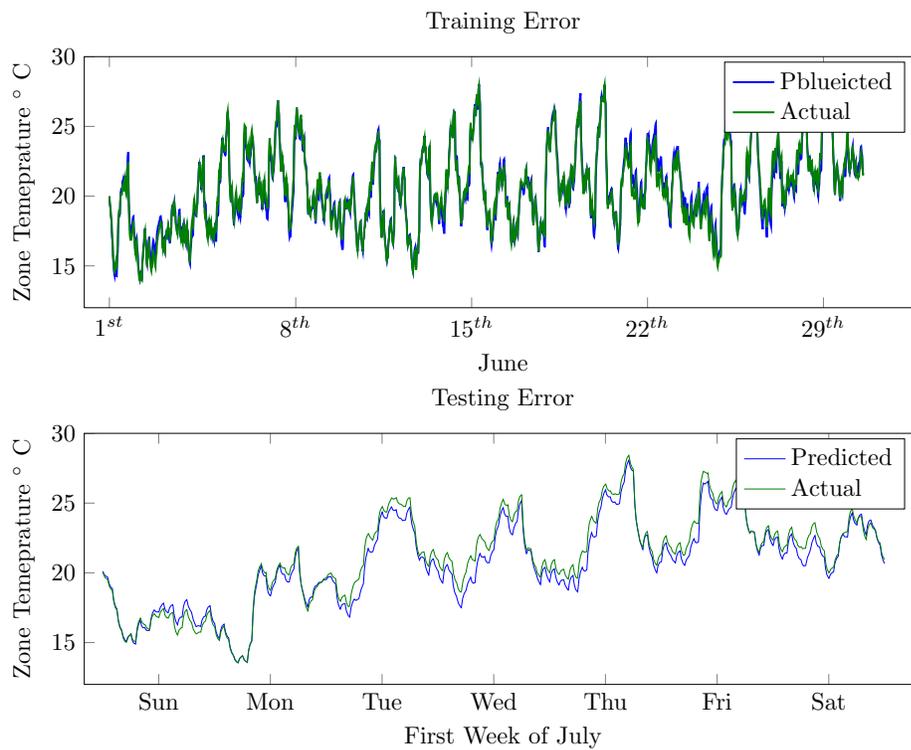


Figure 3.5: The fit between the predicted and actual values of the zone temperature for the training period in June (top figure) and for the testing period during the first week of July (bottom figure).

accuracy of the grey-box model was also tested on a test data-set. The test data-set is the time of the year corresponding to the first week of July. The time-series of inputs for the test period were used with the learned model and the results of the comparison between the predicted model output and the actual zone temperature is shown in 3.5 (bottom). The grey-box model is able to predict the zone temperature with good accuracy for the testing period as well. The RMSE for the testing period was 0.292 with a  $R^2$  value of 0.961. These stats are a better indicator of the accuracy of the model since during the testing period the model is subject to an input data-set that it was not trained on.

### 3.3 Input Uncertainty Analysis

The aim of this analysis is to determine the influence of bias in the training data inputs on the accuracy of the grey-box model and then, to quantify the relative importance of the inputs. First, some notation is introduced for brevity. We consider a model with  $m > 0$  training input data sets denoted by  $U = \{u_1, \dots, u_m\}$ . Note that these are inputs for model training, not the inputs for the model itself, e.g. even though zone temperature is a model output, it is still a required data-set (hence, an input) for model training.  $U_{i,\delta} = \{u_i = u_i + \delta, u_j = u_j | i, j \leq m, j \neq i\}$  denotes the artificial data-set obtained by perturbing input  $u_i$  by an amount  $\delta$  while keeping all other inputs data sets unperturbed.  $U_0$  denotes the data-set in which all the inputs are unperturbed. Now,  $\hat{M}_{U_{i,\delta}}$  is the grey-box model with obtained by training on the data-set  $U_{i,\delta}$  and  $\hat{M}_{U_0}$  is the model obtained by training on a completely unperturbed data-set. We denote the RMSE of the model  $\hat{M}_{U_{i,\delta_k}}$  by  $r(\hat{M}_{U_{i,\delta_k}})$ . The Model-IQ approach for conducting an input uncertainty analysis consists of the following steps:

- (a) Establish a baseline (reference) model: The baseline model,  $\hat{M}_{U_0}$ , is the grey-box model obtained by training on the unperturbed data set  $U_0$ , which is considered as the ground truth.
- (b) Determine which model outputs will be investigated for their accuracy and what are their practical implications.
- (c) Each of the input data streams are then perturbed within some bounds. There are a total of  $N$  perturbations  $\delta_1, \dots, \delta_N$  for each input stream  $u_i, i \leq m$ . This results in  $N$  artificial data-sets  $U_{i,\delta_1}, \dots, U_{i,\delta_N}$  for each input stream  $i$ .
- (d) Corresponding to every perturbation, the grey-box modeling process is run again and a new model  $\hat{M}_{U_{i,\delta_k}}$  is obtained.
- (e) The prediction accuracy of each of the trained model is evaluated on a common input data stream  $U_T$ . The accuracy of the model  $\hat{M}_{U_{i,\delta_k}}$  is measured by the RMSE  $r(\hat{M}_{U_{i,\delta_k}})$  between the predicted and the actual model output values for the common input stream  $U_T$ .

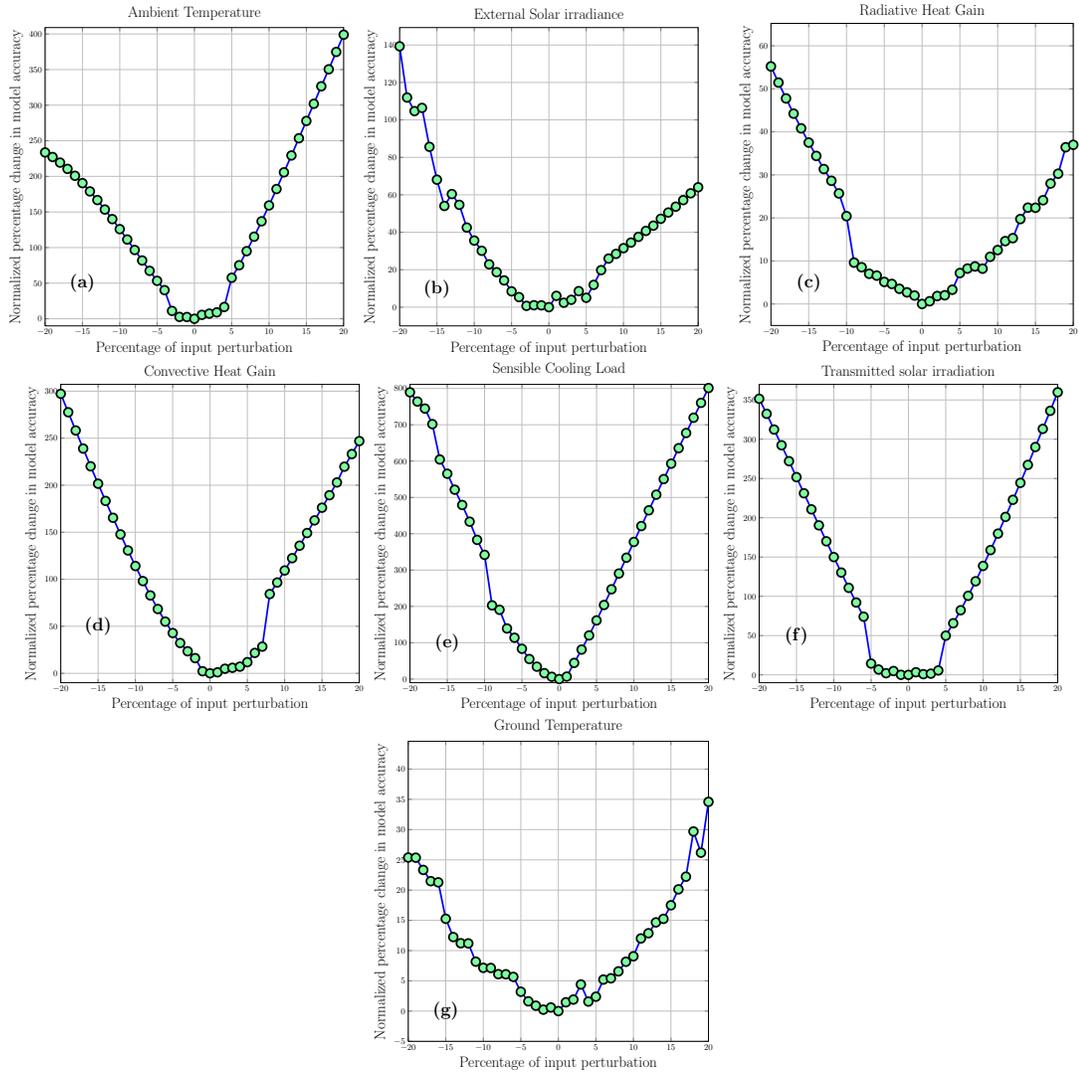


Figure 3.6: Input uncertainty analysis results for a single zone TRNSYS model. The x-axis shows the magnitude of the perturbation in percent change from the unperturbed data. The y-axis is the percent change in model accuracy wrt. the RMSE for the model trained on unperturbed data. Inputs shown: (a) ambient temperature ( $^{\circ}\text{C}$ ); (b) incident solar irradiation on the external walls (W); (c) radiative internal heat gain (W); (d) convective internal heat gain (W); (e) sensible cooling rate (W); (f) solar irradiation transmitted through the windows (W); and, (g) floor (ground) temperature ( $^{\circ}\text{C}$ ).

- (f) Using the RMSE of the fit and the magnitude of the perturbation, determine the sensitivity coefficient for each input training stream.

An overview of the steps for the input uncertainty analysis is shown in 3.2.

For our example, the baseline model is the model trained on unperturbed training data-set. i.e. the original input data-set corresponding to the month of June. The RMSE for the baseline model is denoted by  $r(M_{U_0})$ . The artificial data-sets are created in a normalized manner by adding (and subtracting) a bounded bias to the unperturbed data in the form of the per-cent change from the unperturbed (baseline) value i.e the perturbations  $\delta_k$ 's are in form of per-cent changes around the unperturbed data point. Therefore, each data-point  $x_i$  belonging to the unperturbed input  $u_i$  gets perturbed to a new value of  $\tilde{x}_i = x_i(1 + \delta_k/100)$ . This is done so that every input is treated in the same manner regardless of the scale of the input. One can relate the per-cent change to the absolute value of the change, simply through the mean of the data-set. For e.g. if the mean of unperturbed ambient temperature was 20°C, then the mean of data which was perturbed  $\delta = +10\%$  would be 22°C which is equivalent to a mean absolute bias of 2° degrees in the ambient temperature. Each of the 7 training input data streams (3.2) are perturbed one at a time within  $[-20\%, 20\%]$  around the unperturbed nominal value with increments of 1%. Every perturbation for each of the inputs creates an artificial training set for the grey-box model. Therefore for each of the 7 input streams,  $N = 40$  additional artificial data-sets  $U_{i,\delta_1}, \dots, U_{i,\delta_{40}}$  were created resulting in a total 280 different training data-sets. The grey-box model for the single zone building was trained on each of the artificial data-sets and the accuracy of the model was evaluated in terms of the RMSE on the test data-set. The use of a common test data-set for evaluating the accuracy of the model ensures a fairness in the comparison of the influence of the uncertainty among different inputs on the model accuracy.

Finally the model accuracy sensitivity coefficient is calculated as follows:

$$\gamma_i = \text{mean}_{k=1, \dots, N} \left( \frac{r(\hat{M}_{U_{i,\delta_k}}) - r(M_{U_0}) / r(M_{U_0})}{|\delta_k|} \right) \quad (3.1)$$

It is the mean of the ratio of the normalized change in the model accuracy to that of the normalized change in the magnitude of the input data stream. Both normalization's are with respect to the baseline case. The magnitude of the sensitivity coefficient  $\gamma_i$  can be interpreted as the mean value of the change in the RMSE of the model due to 1% bias uncertainty in the training data stream  $i$ . The sensitivity coefficient is sometimes also referred to as the influence coefficient or point elasticity. The results for the input uncertainty analysis for the TRNSYS building are shown in 3.6. These results align well with the intuition that as the magnitude of the uncertainty bias increases in the input data stream the grey-box model becomes worse and its prediction error increases. This is the case for all the input data streams and it results in the parabolic trend. The shape of the curve varies from input to input, due

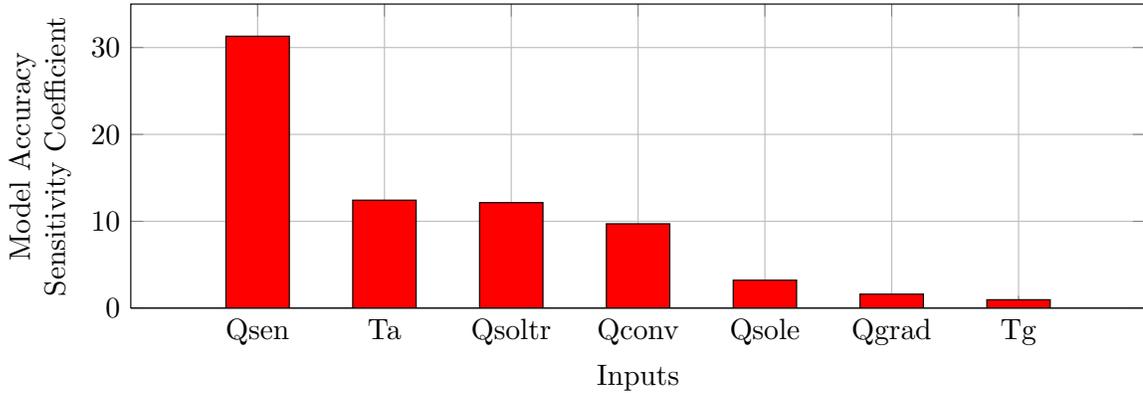


Figure 3.7: Model sensitivity coefficients for different input data streams.

to a different sensitivity coefficient value, and is an indicator of the extend to which a particular input influences the model accuracy.

The model accuracy sensitivity coefficients were calculated for every input data stream and their comparison is shown in 3.7. For the building under consideration, it is clear that the sensible cooling rate, ambient temperature, transmitted solar gain and convective heat gains should be measured accurately in order to learn an accurate grey-box model. One limitation of the input uncertainty analysis is that it assumes independence between training inputs and analyzes them one by one. It may be the case that the model training inputs are not independent. This will require better sampling methods (factorial sampling, latin hypercube sampling) for perturbing multiple inputs at the same time. Although the results presented here assume a particular building and a specific model structure, the Model-IQ approach is general and works for any building inverse model.

### 3.4 Model Accuracy vs MPC Performance

Input uncertainty analysis reveals important insights about the relationship between data quality and model accuracy. It is also necessary to examine if the model accuracy has any direct control performance impact, especially when energy-efficient control algorithms rely on the accuracy of the underlying mathematical model of the building in order to determine optimal control inputs. Installation of additional sensors in a building can yield better quality of data for model training. However there is a trade-off between the accuracy of an inverse model and the cost to obtain it. The trade-offs can be better understood if an end-to-end relationship between the data uncertainty, model accuracy and control performance are known. For model-based control for buildings such as MPC, the accuracy of the building model affects the operation efficiency and cost of the control system. There is significant value in knowing how much the cost of a model predictive controller changes with the model accuracy. This information can be used to provide “target” accuracy levels for the inverse model, which in turn specify the degree of accuracy required on the sensing.

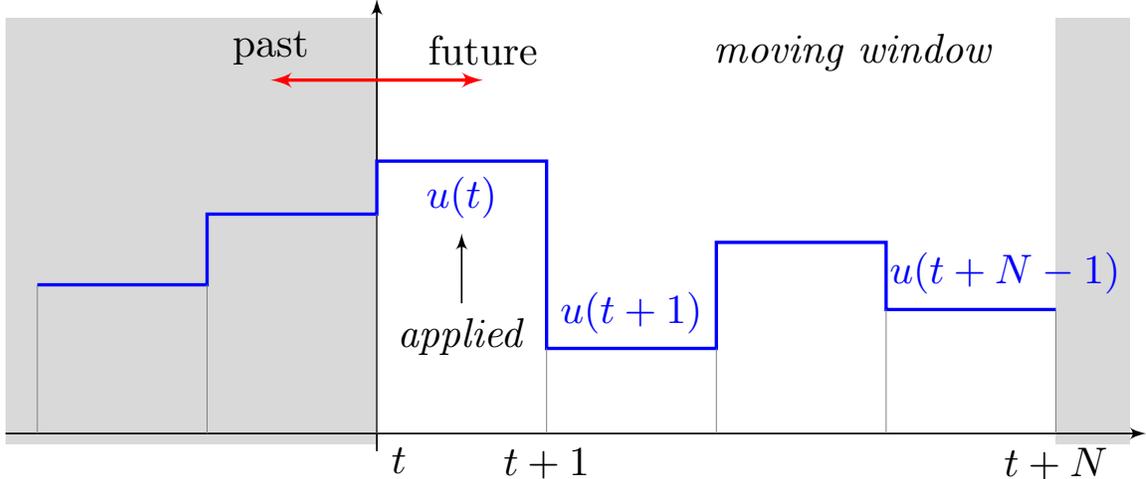


Figure 3.8: Finite-horizon moving window of MPC: at time  $t$ , the MPC optimization problem is solved for a finite length window of  $N$  steps and the first control input  $u(t)$  is applied; the window then recedes one step forward and the process is repeated at time  $t + 1$ .

However, analytically determining the impact of the model accuracy on the MPC performance is a hard problem, due to the complexity of the model structure and the MPC formulation itself. For this reason, to quantify this effect of model accuracy on the performance of a MPC controller, we make use of an empirical analysis with the same single zone TRNSYS model used in Section 3.2. First, a model predictive controller was designed for the single zone TRNSYS building. The MPC simulation is then run for models of different accuracy and the outputs are compared to reveal the trend of the model accuracy’s effect on the MPC cost. We now describe the MPC formulation followed by the results of this analysis.

### 3.4.1 MPC formulation

The MPC formulation involves optimizing a cost function subject to the dynamics of the system and the constraints on the zone temperature, over a finite horizon of time. After an optimal sequence of control inputs are computed, the first input is applied, then at the next step the optimization is solved again as shown in Figure 3.8.

The model in equation 2.5 can also be written as

$$x(k + 1) = \hat{A}_\theta x(k) + \hat{B}_\theta u(k) + \hat{E}_\theta d(k) \quad (3.2)$$

$$y(k) = \hat{C}_\theta x(k) + \hat{D}_\theta u(k) \quad (3.3)$$

where the control input  $u(k)$  is the cooling rate  $\dot{Q}_{sens}$  to the zone, and  $d(k)$  is the vector of all the disturbances to the zone (ambient temperature, heat gains, etc.). To reduce the number of optimization variables we use the move-blocking technique. During each move-blocking window of length  $l$ , the control  $u$  is held constant. So  $u(0) = u(1) = \dots = u(l - 1)$ ,  $u(l) = u(l + 1) = \dots = u(2l - 1)$  and in general

$u(il) = u(il + 1) = \dots = u((i + 1)l - 1)$ , i.e. MPC re-optimizes at integral multiples of the window length  $il$  only.

Let us consider a control horizon  $H$  in terms of move-blocking windows, so the number of time steps is  $Hl$ . At time  $t = il$ , the MPC problem is to minimize

$$\sum_{k=0}^{H-1} \sum_{\sigma=t+kl}^{t+(k+1)l-1} (P_U(\sigma)u(k) + P_T(\sigma) (y(\sigma) - y_{sp}(\sigma))^2) \quad (3.4)$$

subject to

$$x(t) = x_0 \quad (3.5)$$

$$\begin{bmatrix} x(t+kl+1) \\ \vdots \\ x(t+(k+1)l) \end{bmatrix} = \text{diag}(A) \begin{bmatrix} x(t+kl) \\ \vdots \\ x(t+(k+1)l-1) \end{bmatrix} \quad (3.6)$$

$$+ \text{col}(B)u(k) + \text{diag}(E) \begin{bmatrix} d(t+kl) \\ \vdots \\ d(t+(k+1)l-1) \end{bmatrix} \quad (3.7)$$

$$u_{min}(\sigma) \leq u(k) \leq u_{max}(\sigma) \quad (3.8)$$

where the last two constraints hold for all  $k = 0, \dots, H - 1$  and  $\sigma = t + kl, \dots, t + (k + 1)l - 1$ ,  $\text{diag}(\cdot)$  represents a block diagonal matrix of appropriate dimensions, and  $\text{col}(B)$  is the column vector constructed by stacking the columns of matrix  $B$ .  $P_U(\sigma)$  is the price of electricity at time  $\sigma$  and  $P_T(\sigma)$  is the penalty for errors in tracking the desired zone temperature trajectory  $y_{sp}(\sigma)$ . Both the cost and penalty functions vary throughout the day, e.g. the price of electricity can be high during the peak hours of the day as compared to the off peak hours. Similarly, the temperature set-point can change during the day depending on the zone occupancy. Note that in our MPC formulation we only consider soft constraints on the zone temperature. The initial state of the system is  $x_0$ , while  $u_{min}(\sigma)$  and  $u_{max}(\sigma)$  are the lower and upper bounds on the cooling rate which can vary during the day to account for equipment schedules. In order to use the state space model of equation 2.5 for MPC, we also need to design a state observer, which provides estimates  $\hat{x}(k|k)$  of the state of the plant model at every time step.

### 3.4.2 State Observer

In order for us to use the state space model eq. 2.5 for model predictive control, we also need to design a state observer. The state observer is designed to provide the estimates of  $\hat{x}(k|k)$ , the state of the plant model at every MPC time step. The estimates are computed from the measured output  $y_m(k)$  by the linear state observer. The reason for estimating the states of the plant is that the states  $x_1, \dots, x_{n-1}$  are lumped parameter temperatures which are hard (and almost impossible) to measure.

Given the discrete plant model

$$x(k+1) = Ax(k) + Bu(k) + Gw(k) \quad (3.9)$$

$$y(k) = Cx(k) + Du(k) + Hw(k) + v(k) \quad (3.10)$$

where,  $w(k)$  is white process noise and  $v(k)$  is white measurement noise satisfying  $E(w(k)) = E(v) = 0$ ,  $E(ww^T) = Q$ ,  $E(vv^T) = R$ ,  $E(wv^T) = N$ . The estimator has the following state equation:

$$\begin{aligned} \hat{x}(k+1|k) = A\hat{x}(k|k-1) + Bu(k) \\ + L(y(k) - C\hat{x}(k|k-1)) \end{aligned} \quad (3.11)$$

The gain matrix  $L$  is derived by solving a discrete Riccati equation:

$$L = (APC^T + \bar{N})(CPC^T + \bar{R})^{-1} \quad (3.12)$$

where,

$$\bar{R} = R + HN + N^T H^T + HQH^T \quad (3.13)$$

$$\bar{N} = G(QH^T + N) \quad (3.14)$$

The prediction  $\hat{x}(k|k-1)$  is updated using the new measurement  $y(k)$  as:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + M(y(k) - C\hat{x}(k|k-1) - Du(k)) \quad (3.15)$$

where the innovation gain  $M$  is defined as:

$$M = PC^T(CPC^T + \bar{R})^{-1} \quad (3.16)$$

For the simple case,  $E(wv^T) = N = 0$ ,  $D_d = D_u = 0$ ,  $H = 0$ , and  $G = B$ .

### 3.4.3 Single zone example

The MPC described above was implemented for the single zone TRNSYS model. The cooling system of the building is always switched on during the occupancy period from 8 AM to 6 PM on weekdays and remains off during the weekend. The maximum and minimum constraints on the cooling rate were  $u_{max} = 3500\text{W}$  and  $u_{min} = 0\text{W}$ . The temperature set-point of the zone was kept at  $24^\circ\text{C}$  for the occupancy period. The zone temperature is allowed to float during the weekend. The simulation was run for a part of the first week of July. The building is also subject to peak demand pricing: the price of electricity is 10 times the nominal price during the on-peak hours, which are from 1 PM to 5 PM. We compare two different cases. The first is the comparison of the building operation with and without an MPC controller. Without MPC control, the cooling switches on at 8 AM and then tries to supply exactly the amount of cooling energy required to keep the temperature at  $24^\circ\text{C}$  for the occupancy period.

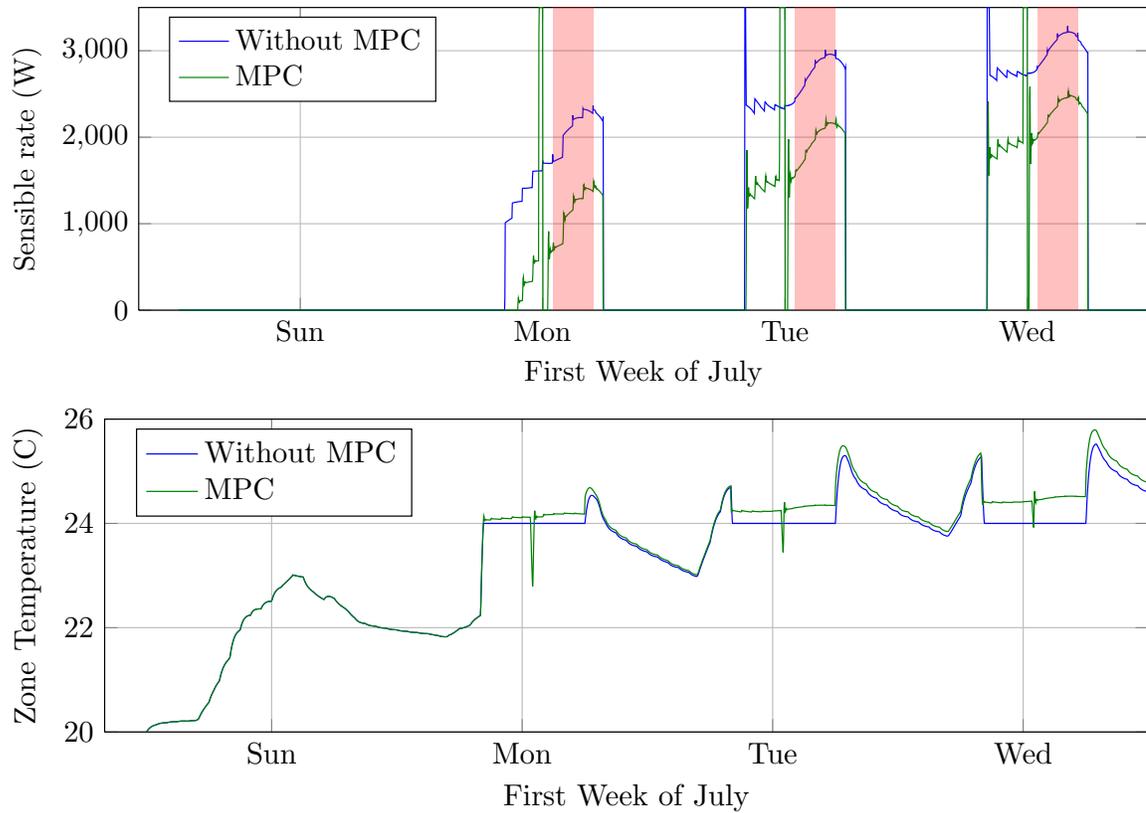


Figure 3.9: Top: comparison between the performance of an MPC controller with the default case without MPC. The peak-pricing period for each day is highlighted in red. Bottom: zone temperature values for both the controllers.

The total energy consumption for the simulation period is 93.71kWh. In this case, the power consumption remains high even during the peak pricing hours, resulting in a total cost of 511.83 units.

The baseline model for MPC is the model with the best RMSE (0.187) for the testing data. This model was trained on unperturbed data and was also used as the baseline for the input uncertainty analysis. The move-blocking step of MPC is 5 minutes and the MPC horizon is 2 hours. Figure 3.9 compares the performance of the MPC controller and the case without MPC. The MPC controller rapidly pre-cools the zone just before the peak pricing period begins at 1 p.m. (regions shown in red in Figure 3.9). This can be seen in both the cooling rate and the zone temperature plots. Consequently, the energy consumption during the peak hours is reduced, which results in an overall lower energy cost. The total energy consumption for this case was 87.29kWh and the total energy cost was reduced to 442.06 units. So there is a 13.63% reduction in the energy cost and a 6.85% reduction in the total energy consumption. The primary reason for the reduction in cost is the pre-cooling of the zone, which shifted part of the cooling demand from on-peak hours to off-peak hours. Another reason for the lower energy consumption is that because the MPC has soft temperature constraints, the zone temperature is slightly above the temperature set-point, requiring it to use less cooling energy.

Having implemented MPC for the baseline model, we now use models trained on perturbed data and compare their performance with the baseline case. This allows us to observe the trend between MPC performance and model accuracy. An example of such a simulation run is shown in Figure 3.10, which compares the baseline model with a relatively inaccurate model (with a much higher RMSE of 0.538 compared to 0.187 for the baseline model). Obviously an inaccurate model performs poorly compared to the “good” baseline case. The total energy consumption was 91.68kWh, a 2.2% saving from the case without MPC. The total energy cost was 492.53 units, only 3.77% reduction compared to 13.63% for the baseline case. Several inverse models with different degrees of accuracy, in terms of their testing RMSE, were run with the MPC controller. Their savings, measured against the case when no MPC was used, are shown in Figure 3.11. The trend of the plot aligns with intuition and shows that MPC performance deteriorates as the underlying model becomes less accurate. It can also be seen that the potential savings of MPC deteriorates quite rapidly as the model accuracy decreases (i.e. test RMSE increases). In the left region of the plot there is a positive cost benefit associated with adding additional sensors to improve the model accuracy from RMSE of 0.331 to 0.187. However, if the model accuracy is in the right region of the plot then there is no cost benefit associated with adding additional sensors to obtain improved models upto a certain RMSE threshold (0.331), beyond which the MPC savings are significant.

We have seen that models can lose their predictive performance if they are trained on uncertain (biased) data. The input uncertainty analysis reveals the extent to which different inputs are responsible for the accuracy of the inverse model. By empirically

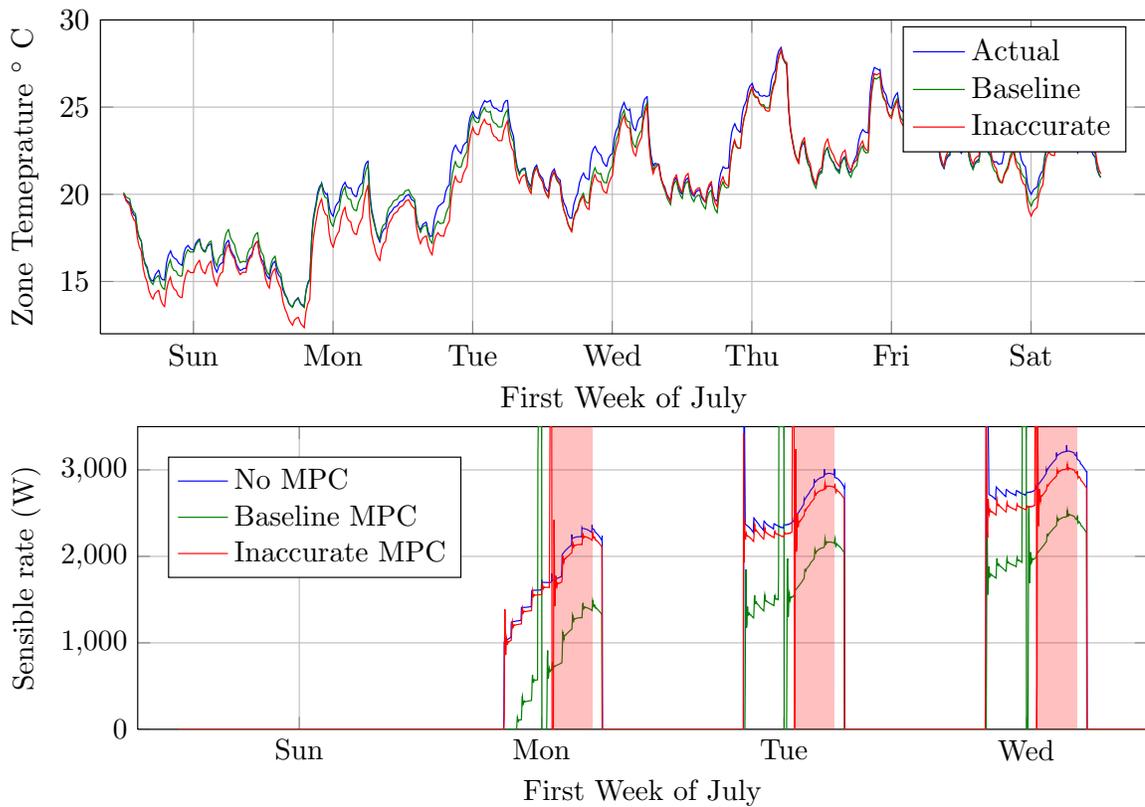


Figure 3.10: Top: comparison of the fit on the test data between the baseline model RMSE (0.187) and an inaccurate model (RMSE 0.538). Bottom: comparison between the MPC performance of the models. The red-regions indicated the peak pricing period for the day.

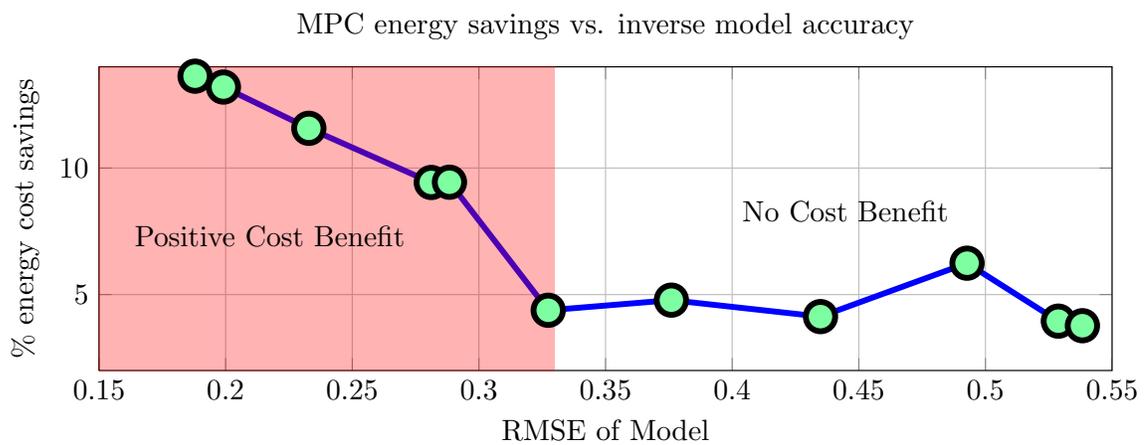


Figure 3.11: MPC performance for models of different degrees of accuracy.

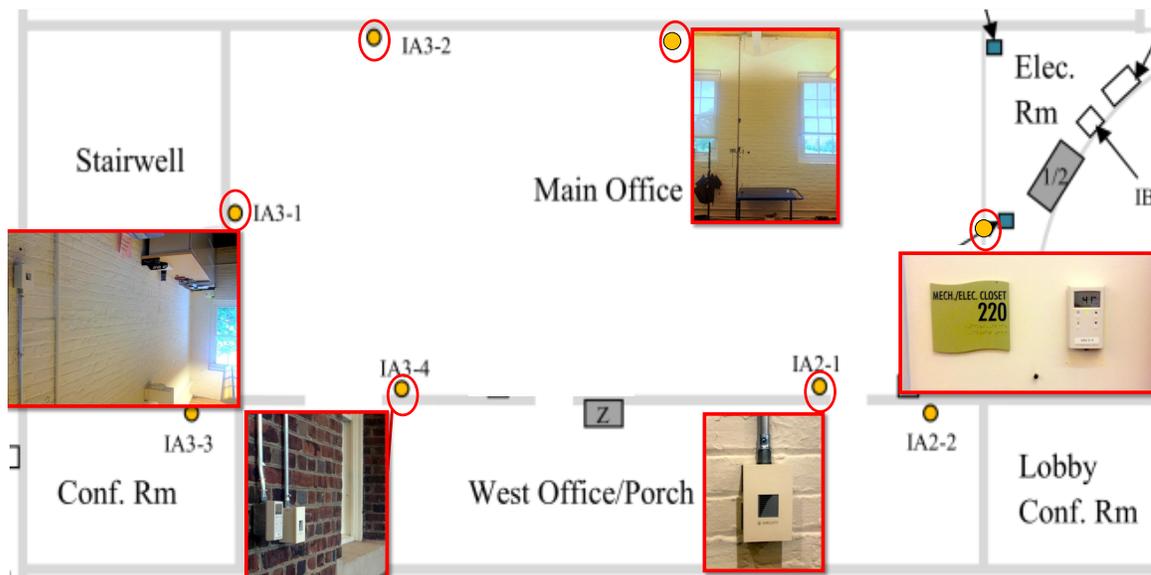


Figure 3.12: Temperature sensor locations for suite 210. The thermostat is located on the right wall. The location of 4 IAQ temperature data loggers and the portable temperature sensor cart is also shown.

establishing a relationship between model accuracy and MPC performance, one can take informed decisions about the investment on additional sensors and the associated cost benefit for improving the data quality. In the next section we apply the Model-IQ toolbox on a model for a real building using real sensor data.

### 3.5 Sensor Placement and Data Quality

So far, we have shown the adverse effects of having uncertainty in the training data on the accuracy of the building inverse model which in turn influences the performance of a model predictive controller. We now show how the location of a sensor can affect the quality of measured data and also present a statistical method to determine the optimal sensor placement and density for obtaining high quality data. We compared the thermostat measurement of suite 210 in building 101 with the mean of several temperature measurements made in the same zone but at different locations. A single point temperature measurement of a zone is based on the assumption that the air inside the zone is well mixed. Our aim was to analyze the temperature data from suite 210 to determine if there is any significant location bias in the thermostat reading and to study how adding additional temporary sensors to a location changes the accuracy of the data.

Suite 210 at building 101 contains several sensors which log air temperature at different locations in the zone. The layout of the zone and the location of the tem-

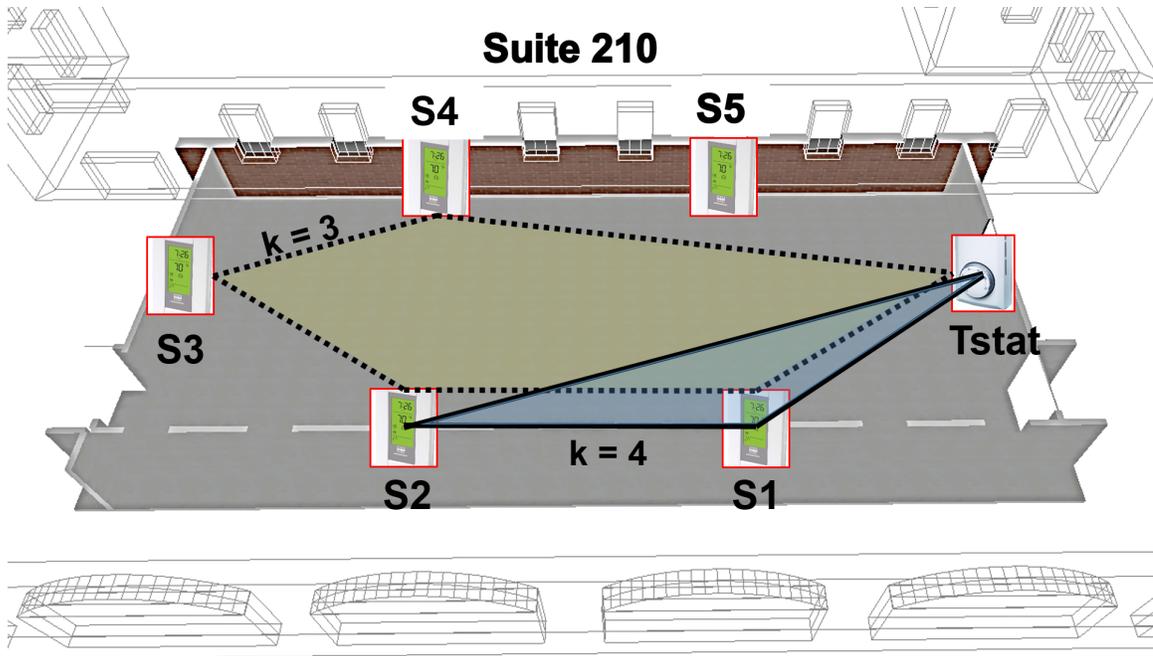


Figure 3.13: Temperature sensor locations for suite 210. The thermostat ( $T_{stat}$ ) is located on the south (right) wall.

perature sensors is shown in Figure 3.12. There are a total of six different locations ( $S_1, S_2, S_3, S_4, S_5$  and  $T_{stat}$ ) in suite 210 where air temperature is logged, as shown in Figure 3.13. The zone thermostat ( $T_{stat}$ ) is placed on the south wall. There are 4 indoor air quality (IAQ) sensors which also measure zone temperature placed on the west, north and the east wall. An additional source of temperature measurement is a portable cart which measures temperatures at 8 different height levels. The location of the cart was not changed for almost an entire year therefore its data can be treated as data measured from the same location. Figure 3.15 shows the plot of air temperature at each location from January 2013 to August 2013 for suite 210. The true value of the temperature of a zone (air volume) is extremely hard to determine. Since the different temperature sensors are located around the zone in a uniform manner, the mean of all six temperature measurements is a better representation of the zone temperature and is regarded as the *true* temperature (denoted by  $T_{tr}$ ).

The mean temperature value is compared with the thermostat measurement in Figure 3.15. The values of the residuals are plotted. It can be seen that the difference between the thermostat and the mean temperature can be upto  $4^\circ$ . This suggests that the reading of the thermostat may be biased due to its location.

Another way to compare the two data-sets is through a scatter plot between the mean temperature and the thermostat reading. Figure 3.16 shows such a comparison along with a histogram plot for each axis. Two main inferences can be drawn from this plot. First, the spread of the data reveals how much the thermostat reading deviates from the mean temperature. A lower spread indicates that the two measurements are

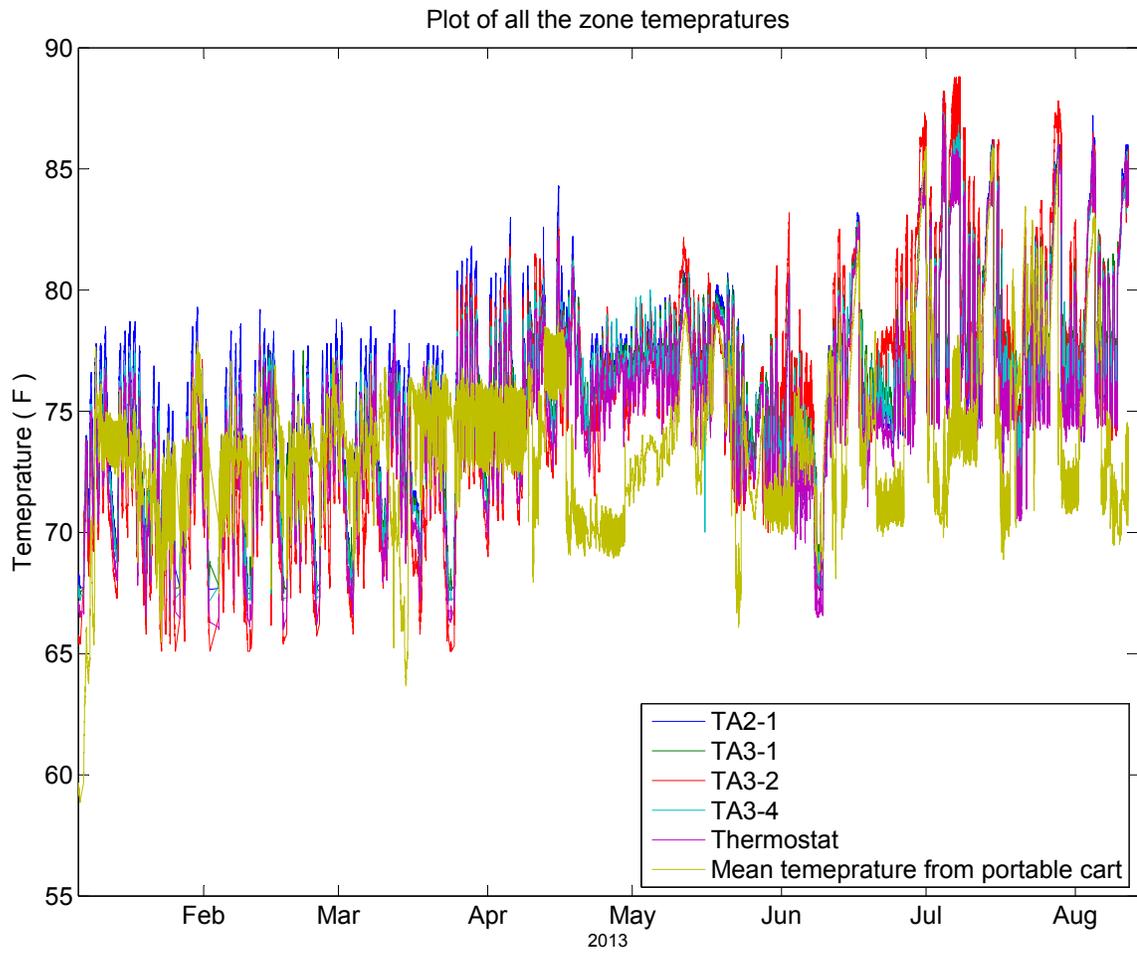


Figure 3.14: All the temperature measurements from different locations in suite 210.

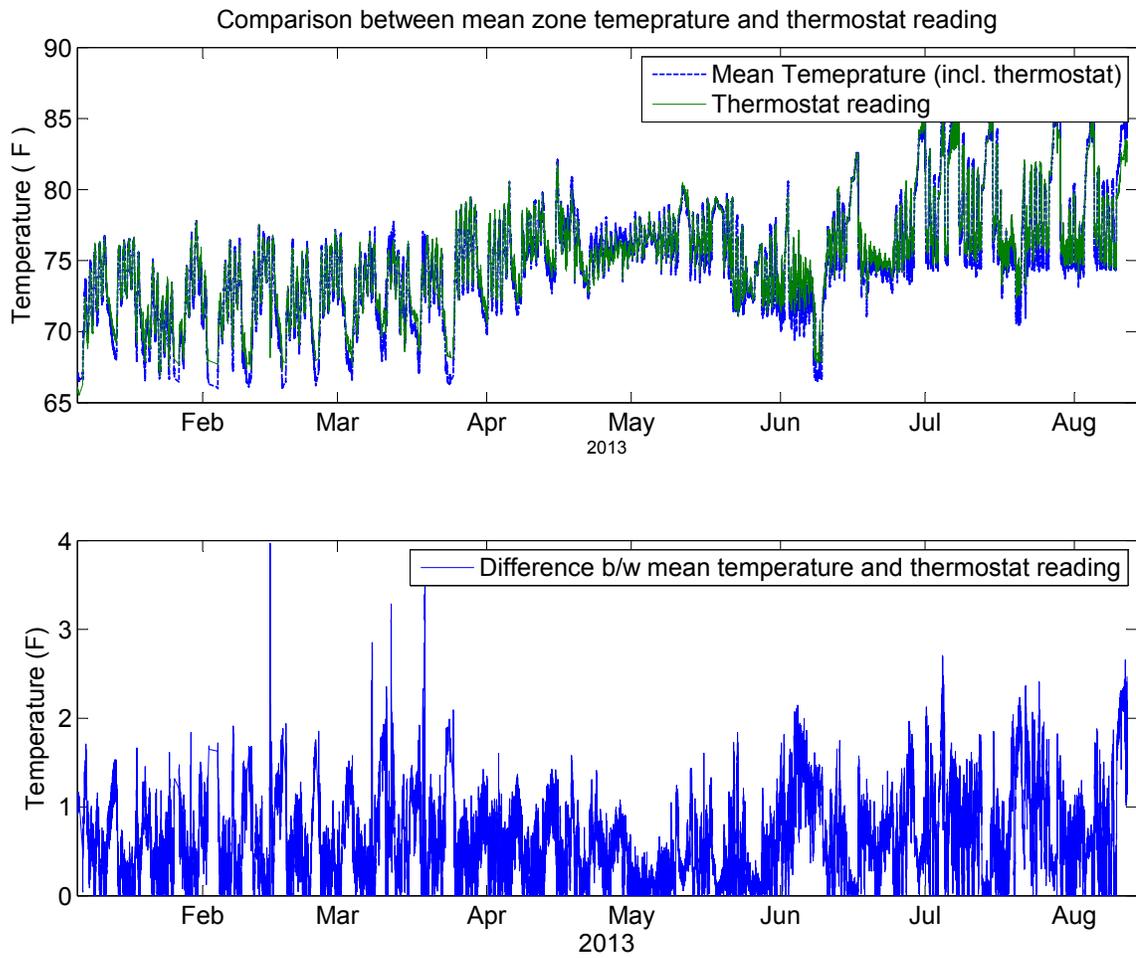


Figure 3.15: Comparison of thermostat reading and the mean temperature reading for suite 210. The bottom figure plots residuals between the two data-sets.

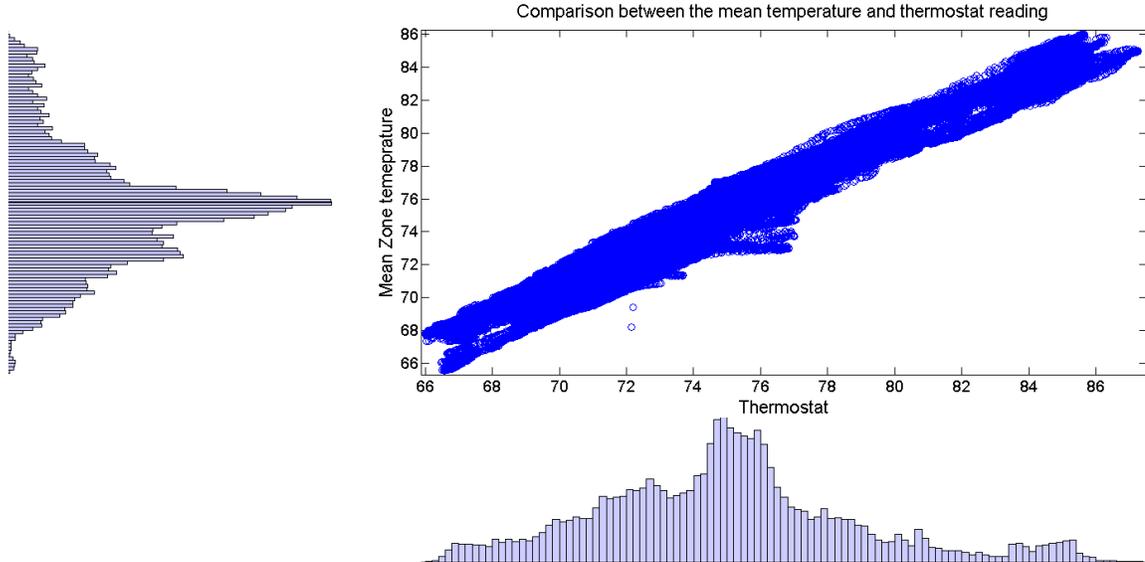


Figure 3.16: Scatter plot between the mean temperature measurement (y-axis) and the thermostat reading (x-axis). The spread of a data indicates the bias between the thermostat and the mean temperature measurement.

in agreement and that the well mixed assumption holds well for the zone. Second, the histogram of the data-sets reveals that the thermostat data has a much larger variance than the mean temperature measurement.

It should be pointed out that the thermostat reading is the one which is used for controlling the zone temperature so any biases/errors in the temperature measurement are not desired.

We wanted to estimate the bias due to sensor location and to determine the best sensor placement and density. For this, we select  $k$  sensors ( $k = 1, 2, \dots, 5$ ) out of the six available sensors and compare the average temperature  $T_k$  of the selected  $k$  sensors with the *true* temperature  $T_{tr}$  through hypothesis testing.

We first check the normality of the temperature data from each sensor using a quantile-quantile (Q-Q) plot. The Q-Q plot is used to check the validity of a distributional assumption for a data-set. The idea is to compute the theoretically expected value for each sample based on the distribution in question. If the temperature data follow a normal distribution, then the points on the normal Q-Q plot will fall on a straight line. Figure 3.17 shows the Q-Q plot for the data from all six sensors against the theoretical samples obtained from normal distribution. The plots suggest that the temperature data is not normally distributed and is likely to follow a distribution with thicker tails than the normal distribution. This implies that the t-test is no longer the best test for comparing any two data-sets as it assumes that the data is normally distributed.

To overcome this problem, we use non-parametric statistics for comparing temperature data-sets against each other. In particular, we use the Wilcoxon rank-sum statistic. The Wilcoxon test is valid for data from any distribution, whether normal

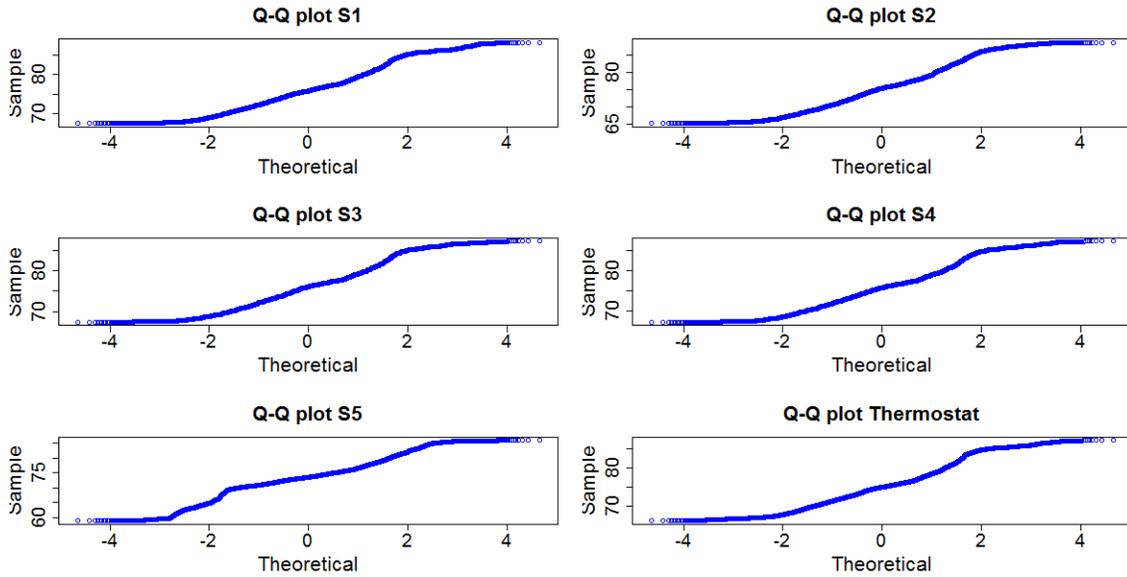


Figure 3.17: Q-Q plot for the temperature data from all the locations show that the temperature data is not likely to be normally distributed.

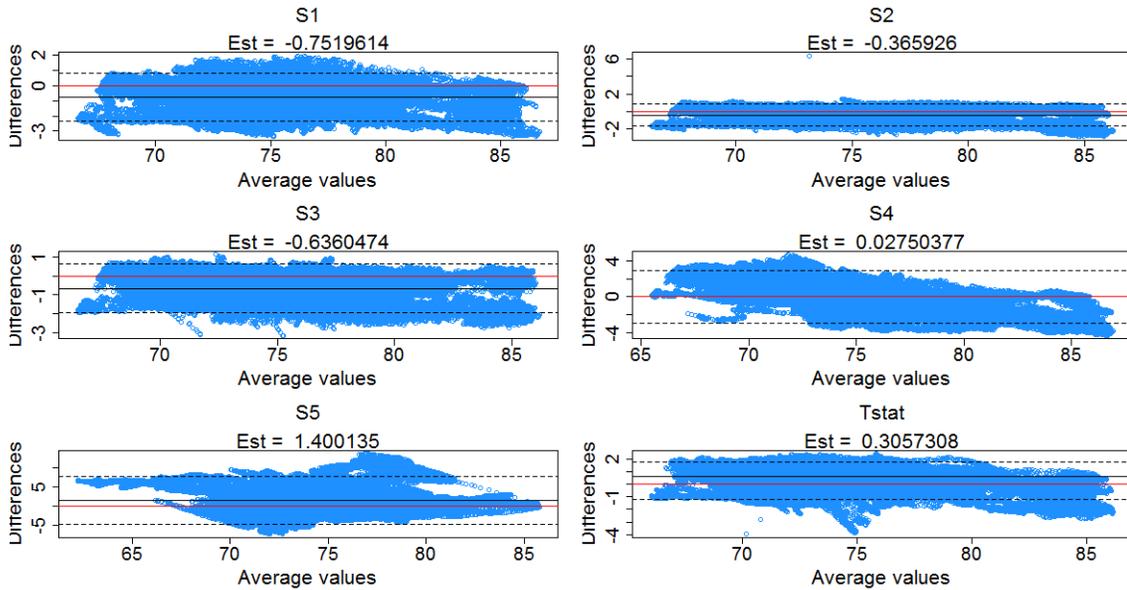


Figure 3.18: Bland-Altman plots for all 6 sensors i.e  $k = 1$ . The solid red line indicates the mean difference between the *true* temperature and the data-set. The estimated bias is obtained through the Wilcoxon's test.

Table 3.1: Wilcoxon’s test results for all values of  $k$ 

k	Min. bias subset $T_k$	Bias Estimate $\mu_k$
1	$S_4$	0.0275
2	$S_3, S_4$	-0.0106
3	$S_1, S_2, Tstat$	0.00708
4	$S_1, S_3, S_4, Tstat$	0.22
5	$S_1, S_3, S_4, S_2, Tstat$	-

or not, and is much less sensitive to outliers than the t-test. It is suitable for testing differences between paired data-sets, for e.g. comparing the air temperature measurements from two different sensors sampling at the same rate. While comparing any data-set  $T_k$  to the *true* temperature  $T_{tr}$  we check the hypothesis that the two data-sets are originating from the same distribution and that the median difference between pairs of observations is zero. The Wilcoxon’s test provides an estimate  $\mu_k$  (*Hodges-Lehmann estimate*) and the 95% confidence interval (C.I.) of the bias between the two data-sets.

An intuitive, but informal, method of comparing the *true* temperature with another data-set by visual inspection is through the Bland-Altman plot also known as Tukey’s mean difference plot. In this method, the difference of the two paired data-sets is plotted against their mean. Figure 3.18 shows Bland-Altman plots for the case  $k = 1$ , i.e. each sensor measurement ( $T_{S_1}, T_{S_2}, T_{S_3}, T_{S_4}, T_{S_5}$  and  $Tstat$ ) is compared with the *true* temperature  $T_{tr}$ . The estimate of the bias  $\mu_k$  between the data-sets and the *true* temperature obtained through the Wilcoxon’s test is also indicated for each data-set. For the case  $k = 1$ , there are 6 possible comparisons with the *true* temperature. It turns out that the sensor location  $S_4$  is the closest to the *true* temperature with an estimated bias of only 0.027°C. The thermostat measurement  $Tstat$  has an estimated bias of 0.588°C with respect to the *true* temperature. This means that if we were to place just one sensor in the zone to estimate the *true* temperature, it should be placed at the location of  $S_4$ . The next section evaluates whether the bias in the zone thermostat is enough to affect the model accuracy.

The same method is repeated for each value of  $k = (1, 2, \dots, 5)$ . For each  $k$ , all  $\binom{6}{k}$  sensor combinations are enumerated and the mean temperature  $T_k$  of the  $k$  selected sensors is compared with the *true* temperature using the same techniques as described above. The combination with the minimum bias estimate  $\mu_k$  is selected as the best sensor subset for each value of  $k$ . The results of these comparisons are summarized in Table 3.1.

The results indicate that adding multiple sensors to a zone tends to improve the accuracy of the quantity being measured. However, it is not always the case that adding additional sensor will always lead to an improvement in data accuracy. This can be seen from Table 3.1 where the bias in the combined measurement of the data obtained from 3 sensors is much less than the bias due to the combined measurement obtained from 4 different sensors. The minimum bias sensor subset for  $k = 3$  and

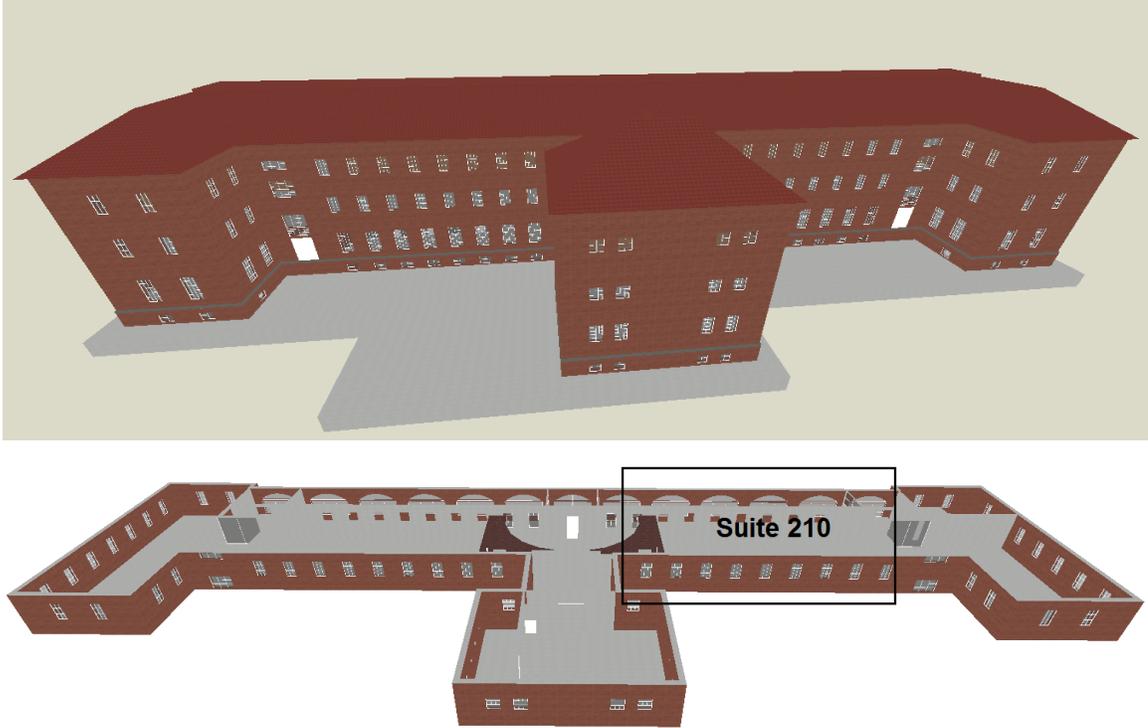


Figure 3.19: 3D view of Building 101, the site chosen for the case study and the location of suite 210 in the north-wing of the building.

$k = 4$  is also shown in Figure 3.13.

### 3.6 Case Study With Real Building Data

In this section we present the results of applying the Model-IQ approach, described in Section 3.2, to real sensor data. First, we calculate the bias in the input data due to sensor placement. We then perform an input uncertainty analysis on the training data and the building inverse model.

The Model-IQ approach described in Section 3.2 was applied to real sensor data. The site chosen for analysis is called Building 101. Building 101, located in the Navy Yard in Philadelphia, is the temporary headquarters of the U.S. Department of Energy’s Energy Efficient Building Hub [US Department of Energy 2013]. It is a highly instrumented commercial building where the acquired data is continuously stored and is made available to Hub researchers. The building (see Figure 3.19, top), is comprised of offices, a lunchroom, mechanical spaces, and miscellaneous spaces. For the case study, we focus on suite 210, a large office space on the second floor of the north-wing of the building as shown in Figure 3.19 (bottom). This zone has a single external wall on the east side with 8 windows, a large interior wall on the west side which is adjacent to the porch area on the north-wing and two more adjacent walls on the north and the south side. In July 2013, functional tests were run from

00:00, 20-07-2013 to 22:29, 20-07-2013, on the air handling unit serving suite 210 as a part of an ongoing Hub project. During a functional test, the supply air temperature is changed rapidly so there is enough thermal excitation in the zone to generate a rich data-set for learning its dynamical model.

### 3.6.1 Model-IQ implementation for Suite 210

We first created the lumped parameter RC-network model for suite 210. The model has 9 states, 9 inputs and 1 output. There are a total of 22 RC parameters in the model structure for this zone.

The temperature inputs to the model were the ambient temperature  $T_a(^{\circ}\text{C})$ , boundary condition for the floor  $T_f(^{\circ}\text{C})$  given by the temperature of the zone on the first floor underneath suite 210, boundary condition for the ceiling  $T_c(^{\circ}\text{C})$  given by the temperature of the zone on the third floor above suite 210 and temperature of the adjacent porch area  $T_p(^{\circ}\text{C})$ . The external solar irradiation  $Q_{sole}$  incident on the east wall is logged by a pyranometer. For the internal heat gain calculation, we consider 3 different heat sources: occupants, lighting and appliances. The number of people in the zone at different times during the functional test period was estimated using data from the people counter. We assume, using ISO standard 7730, that in a typical office environment the occupants are seated, involved in light activity and emit 75 (W) of total heat gain, 30% of which is convective and 70% is radiative gain. Using the power rating of the lighting fixtures and their efficiency, one can calculate the heat gain due to lighting. In this zone, lights contribute about 13 ( $\text{W}/\text{m}^2$ ) with a 40% – 60% split between the convective and the radiative part. A constant heat gain due to the electrical appliances and computers is also assumed. The total internal convective heat gain  $Q_{conv}$  was obtained by adding the convective gain contributions from the three different heat gain sources. The total internal radiative heat gain was obtained in a similar way. The total internal radiative heat gain is further split into the radiative gain on the external wall  $Q_{qgrade}$  and the radiative gain on the ceiling  $Q_{qgradc}$  and applied as two separate inputs. The sensible cooling rate  $Q_{sen}$  was calculated using the temperature and mass flow rate measurements for the supply and the return air.

The sampling rate of the data was 1 minute. The total available data was split into a training set (80% of the data) and a test set (remaining 20% data). All the inputs for training the inverse model are shown in Figure 3.20. The output of the inverse model is the zone temperature  $T_z$ . After completion of the training process, the zone temperature predicted by the model is compared with the actual value of the zone temperature for both the training and the test period. The results of the inverse model training are shown in Figure 3.21. The RMSE for the training data-set was 0.062 with  $R^2$  equal to 0.983 (Figure 3.21, top) while the RMSE and  $R^2$  values for the test set were 0.091 and 0.948 respectively (Figure 3.21, bottom).

After successfully training the inverse model, we conducted an input uncertainty analysis on the input-output training data-set. The model trained on unperturbed

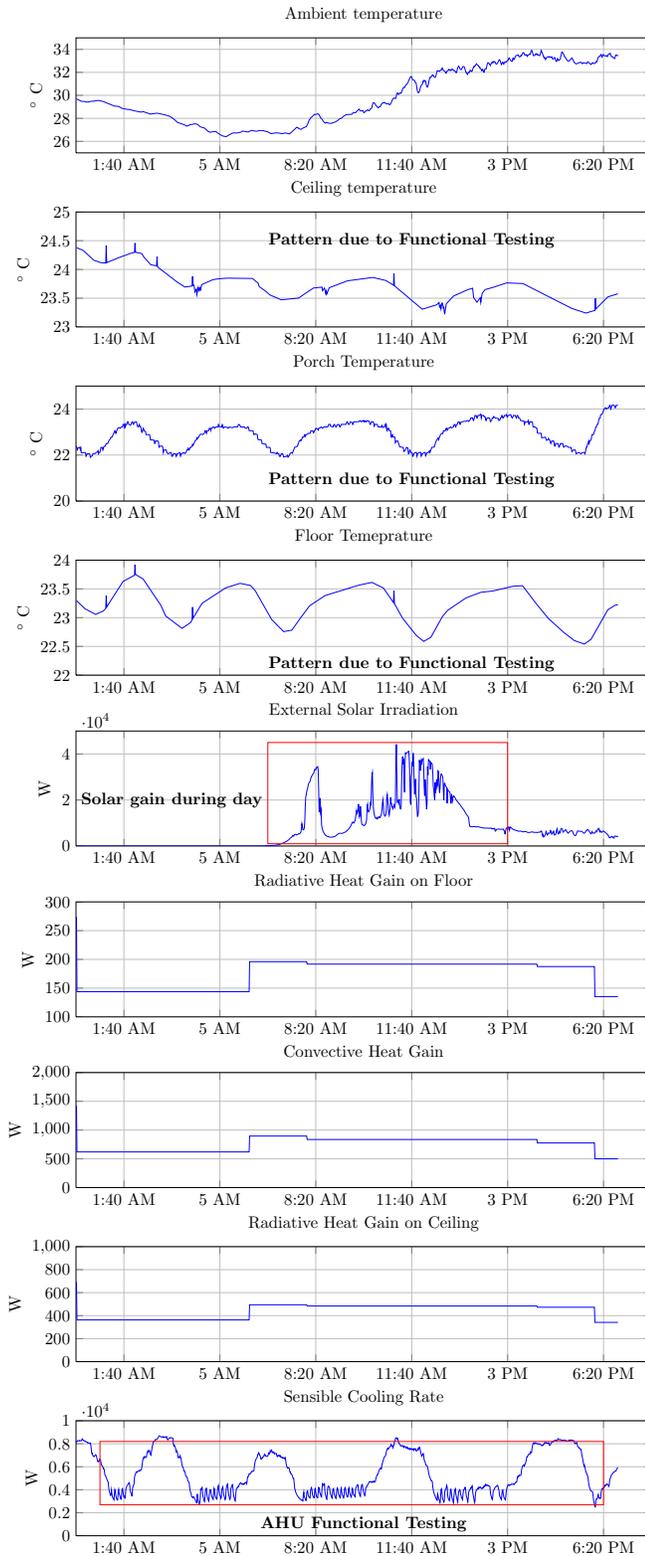


Figure 3.20: Training data for suite 210 of Building 101. The data obtained by running a functional test on the zone's air handling unit from 20-07-2013 00:00 to 20-07-2013 22:29

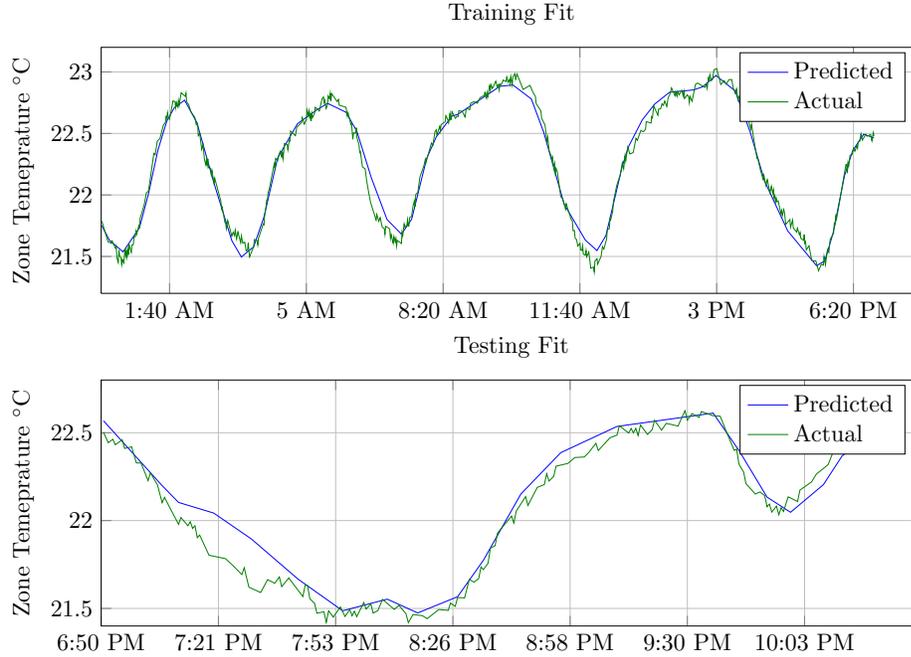


Figure 3.21: Fit between the predicted and actual zone temperature in suite 210. Top: for the training period, with  $RMSE = 0.062$  and  $R^2 = 0.983$ . Bottom: for the test period with  $RMSE = 0.091$  and  $R^2 = 0.948$ .

data serves as the baseline model for the uncertainty analysis. Similar to the case of the single-zone TRNSYS model, we created artificial data-set from the training data by perturbing each input data stream within  $[-20\%, 20\%]$  of the unperturbed values in increments of 1%. For this case study, we also wanted to characterize the influence of uncertainty in the output of the model, the zone temperature, on the accuracy of the model. Therefore, in addition to the 9 aforementioned model inputs, perturbations were also introduced in the output training data-set i.e. in  $T_z$ . With 40 additional data-sets each, there were a total of 400 artificial data-sets. Each of these data-sets were used again for model training and the resulting model was evaluated for its accuracy in terms of the RMSE on the test-set.

### 3.6.2 Results

The results of the input uncertainty analysis for suite 210 in Building 101 are shown in Figure 3.22. Yet, again we see the parabolic trend obtained as a result of “artificial” uncertainty in the training data for each of the training data-sets. The sensitivity coefficients for the different training inputs were calculated. Figure 3.23 shows the comparison of the model accuracy sensitivity coefficients for the inverse model for suite 210. It is seen that the zone temperature has the largest model accuracy sensitivity coefficient suggesting that the accuracy of the model is quite sensitive to the zone temperature measurement. We saw earlier in Section 3.5 that the thermostat

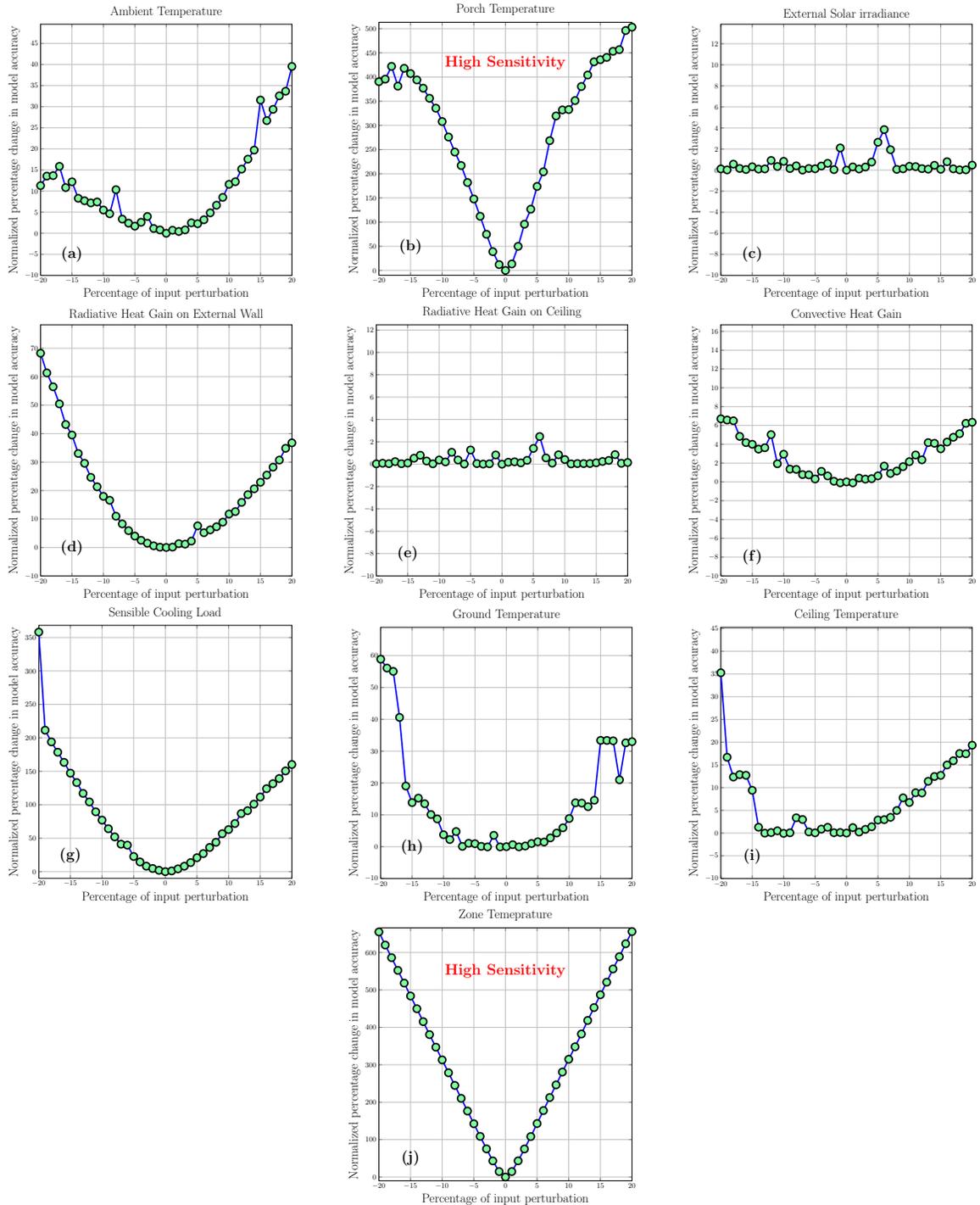


Figure 3.22: Input uncertainty analysis results for Building 101 inverse model. The x axis shows the magnitude of the perturbation in percent change from the unperturbed data while the y axis is the percent change in the model accuracy compared to the RMSE for the model trained on unperturbed data. The following inputs are shown: (a) ambient temperature ( $^{\circ}\text{C}$ ); (b) porch temperature ( $^{\circ}\text{C}$ ); (c) incident solar irradiation on the external walls (W); (d) and (e) radiative internal heat gain on external wall and ceiling (W); (f) convective internal heat gain (W); (g) sensible cooling rate (W); (h) floor temperature ( $^{\circ}\text{C}$ ); (i) ceiling temperature ( $^{\circ}\text{C}$ ), and (j) zone temperature ( $^{\circ}\text{C}$ )

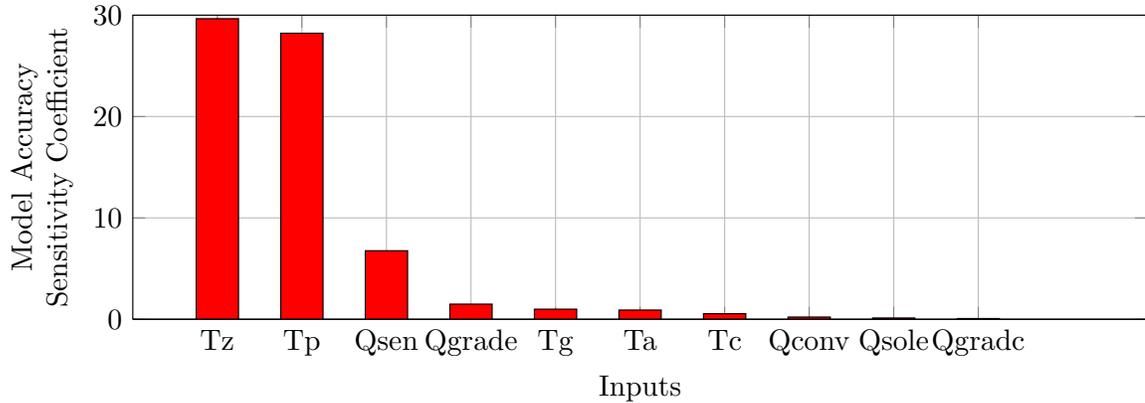


Figure 3.23: Model accuracy sensitivity coefficients for Building 101

measurement has an uncertainty bias of about 1%. We see that this can effect the model accuracy by upto 20%.

This suggests that for this zone, it would be better to deploy additional low-cost wireless sensors just during the model training phase and get a better estimate of the zone temperature for training the inverse model. Also, the mean value obtained by adding more sensors could be used to re-calibrate or correct the thermostat reading for location bias, resulting in data which can yield an inverse model which can better represent the dynamics of the zone.

## 3.7 Related Work

### 3.7.1 Model predictive control related

The treatment and analysis of the implementation of model based control schemes like MPC and optimal control for buildings have been very thorough. [Gyalistras et al. 2010, Ma et al. 2012] describe the implementation of MPC for energy efficient operation of buildings, supported by strong case studies. In [Oldewurtel et al. 2010] the authors consider uncertainty in the prediction of disturbances and propose a stochastic version of MPC. In [Kim and Braun 2012], a reduced order model has been used for MPC. [Prívvara et al. 2012] advocates the use of simpler building models based on the physical description of the building. The authors highlight the building modeling process as a crucial part for building predictive control.

### 3.7.2 Sensitivity analysis related

Parametric sensitivity analysis of a model reveals the important parameters of the model which most significantly affect the model output. In [Lam and Hui 1996], important input design parameters are identified and analyzed from points of view of annual building energy consumption, peak design loads and building load profiles.

In [Eisenhower et al. 2012], the authors extend traditional sensitivity analysis and increase the size of analysis by studying the influence of about 1000 parameters.

### 3.7.3 Uncertainty related

It is only recently in [Bengea et al. 2011, Candanedo et al. 2013] and [Slaven Peles 2012], that researchers have analyzed the uncertainty in modeling for close loop control. In [Bengea et al. 2011], the authors acknowledge that the performance of advanced control algorithms depends on the estimation accuracy of the parameters of the model. They design an MPC algorithm using a control model that is structurally identical to the plant model but has perturbed parameters. The closed loop system is simulated and the impact of the parameter perturbations on the energy cost is evaluated. Although, this methodology bears some similarity with the Model-IQ approach, there are some key differences. First, for a fixed model structure, the model parameters can change either due to the estimation process or due to the quality of data. The cause of the parameter change has not been addressed in their work. So although one can identify which parameters should be estimated well, it is not clear how can one get a good estimate for that parameter. Second, the use of the same model as the control and the plant model is debatable. Realistically, the control model can only be an approximation of the plant dynamics but can never be exactly the same as the plant model. Which is why we used the TRNSYS building as the plant model in our MPC simulation to make it more realistic. In [Candanedo et al. 2013], the authors discuss the development of a control-oriented simplified modeling strategy for MPC in buildings using virtual simulations [Slaven Peles 2012] presents a methodology to automate building model calibration and uncertainty quantification using large scale parallel simulation runs. The method considers global sensitivity analysis using probabilistic data while we consider a fixed bias error.

## 3.8 Concluding remarks

We introduced Model-IQ, a methodology for analysis of uncertainty propagation for building inverse modeling and controls. Model-IQ enables the modeling framework to incorporate uncertainty to a level that enables end-users understanding of the limits of their models and controls. Given a plant model and real input data, Model-IQ evaluates the effect of the uncertainty propagation from sensor data to model accuracy to controller performance. Through analysis with a high fidelity virtual building modeled in TRNSYS and a case study with real measurements from an office building, we show:

- (a) Uncertainty bias present in the training input adversely effects the accuracy of the building inverse model. The extent of the influence of uncertainty in each training data stream on the model accuracy can be quantified through an input uncertainty analysis.

- (b) We evaluate the relationship between model accuracy and performance of a MPC controller. Our empirical treatment of this analytically hard problem is both new and realistic compared to related work. We demonstrate that an accurate building inverse model can result in a MPC cost reduction of more than 13% while a bad model will barely reduce the cost (3%).
- (c) We run the Model-IQ toolbox on a data-set obtained from a real building. We show that the density and placement of sensors are responsible for introducing a location based bias in the measured data. We observe that a bias of 1% degrades the model accuracy by 20%.
- (d) We also presented a statistical method to quantify the bias in the sensor measurement (if any) and to determine near optimal sensor placement and density for accurate data measurements.

# Chapter 4

## Data-driven modeling with regression trees

Techniques and algorithms that have stemmed from the field of machine learning have indeed now become a powerful tool for the analysis of complex and large data, successfully assisting scientists in numerous breakthroughs of various fields of science and technology. Public and famous examples include the use of boosted decision trees in the statistical analysis that led to the detection of the Higgs boson at CERN [Chatrchyan et al. 2012], the use of random forests for human pose detection in the Microsoft Kinect [Criminisi and Shotton 2013] or the implementation of various machine learning techniques for building the IBM Watson system [Ferrucci et al. 2010], capable to compete at the human champion level on the American TV quiz show Jeopardy.

Formally, machine learning can be defined as the study of systems that can learn from data without being explicitly programmed. However, the goal of machine learning is not only to produce algorithms making accurate predictions, it is also to provide insights on the predictive structure of the data [Breiman et al. 1984]. For practitioners, which are not experts in machine learning, interpretability is indeed often as important as prediction accuracy. It allows for a better understanding of the phenomenon under study, a finer exploration of the data and an easier self-appropriation of the results. By contrast, when an algorithm is used completely as a black box, yielding results seemingly out of nowhere, it may indeed be difficult to trust or accept if it cannot be understood how and why the procedure came to them.

Unfortunately, the current state-of-the-art in machine learning often makes it difficult for non-experts to understand and interpret the results of an algorithm. Likewise, few of them actually provide clear and insightful explanations about the results they generate. In this context, the goal with data-driven building modeling is to provide an interpretable model of the building's dynamics which is also suitable for model based control. Our choice of machine learning algorithms to achieve this goal is based on regression trees. While regression trees based methods have proven to be a robust, accurate and successful tool for solving countless of machine learning tasks,

including classification, regression, density estimation, manifold learning or semi-supervised learning their use for fine-grained building modeling and their extension for close loop control synthesis is new.

## 4.1 Learning from data

The objective of any data-driven modeling method is to find a systematic way of predicting a phenomenon given a set of measurements. In machine learning terms, this goal is formulated as the supervised learning task of inferring from collected data a model that predicts the value of an output variable based on the observed values of input variables. As such, finding an appropriate model is based on the assumption that the output variable does not take its value at random and that there exists a relation between the inputs and the output (i.e. the problem is *well-posed*).

To give a more precise formulation, let us assume as set of cases or objects taken from a universe or sample space  $\Omega$ . Let us arrange the set of measurements on a case in a pre-assigned order, i.e., take the input values to be  $x_1, x_2, \dots, x_p$ , where  $x_j \in \chi_j$  (for  $j = 1, \dots, p$ ) corresponds to the value of the input variable  $X_j$ . Together, the input values  $(x_1, x_2, \dots, x_p)$  form a p-dimensional input vector  $\mathbf{X}$  taking its values in  $\chi_1 \times \dots \times \chi_p = \chi$ , where  $\chi$  is defined as the input space. Similarly, let us define  $y \in \mathbb{Y}$  the value of the output variable  $Y$ , where  $\mathbb{Y}$  is defined as the output or the response space. By definition, both the input and the output spaces are assumed to respectively contain all possible input vectors and all possible output values. Note that input variables are sometimes known as *features*, input vectors as *instances* or *samples* and the output variables as *target* or *response* variable.

Among different variables, we distinguish between two general types. The first correspond to quantitative variables whose values are integer or real numbers, such as temperature, humidity, wind speed, power consumption etc.. The other correspond to qualitative variables whose values are symbolic, day of the week, month etc..

**Definition 4.1.** A variable  $X_j$  is ordered if  $\chi_j$  is a totally ordered set. In particular,  $X_j$  is said to be numerical if  $\chi_j = \mathbb{R}$ .

**Definition 4.2.** A variable  $X_j$ , is categorical if  $\chi_j$  is a finite set of values without any natural order.

In a typical supervised learning task, past observations are summarized by a dataset called the *learning* or *training* set. It consists of the set of observed input vectors together with their actual output value and formally defined as follows:

**Definition 4.3.** A learning set  $\mathcal{L}$  is a set of  $N$  pairs of input vectors and output values  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ , where  $\mathbf{x}_i \in \chi$  and  $y_i \in \mathbb{Y}$ .

Equivalently, a set of p-input vectors  $\mathbf{x}_i$  (for  $i = 1, \dots, N$ ) can be denoted by a  $N \times p$  matrix  $\mathbf{X}$  whose rows  $i = 1, \dots, N$  correspond to input vectors  $\mathbf{x}_i$  and columns

$j = 1, \dots, p$  to input variables  $X_j$ . Similarly, the corresponding output values can be written as a vector  $\mathbf{y} = (y_1, \dots, y_N)$ .

In this framework, the supervised learning task can be stated as a learning function  $\phi : \chi \mapsto Y$  from a learning set  $\mathcal{L} = (\mathbf{X}, \mathbf{y})$ . The objective is to find a model such that its prediction  $\phi(\mathbf{x})$ , also denoted by the variable  $\hat{Y}$ , are as accurate as possible. When  $Y$  is a numerical variable, the the learning task is a regression problem i.e. :

**Definition 4.4.** A regressor is a function  $\phi : \chi \mapsto Y$ , where  $Y = \mathbb{R}$

## 4.2 Regression trees

Building regression trees based models has always been driven by the ambition to understand and uncover complex relations in data. That is, to find models that can not only produce accurate predictions, but also be used to extract knowledge in an intelligible way. Tree-based methods stand as one of the most effective and useful method, capable to produce both reliable and understandable results, on mostly any kind of data.

Historically, the appearance of decision trees is due to [Morgan and Sonquist 1963], who first proposed a tree-based method called automatic interaction detector (AID) for handling multi-variate non-additive effects in the context of survey data. Building upon AID, methodological improvements and computer programs for exploratory analysis were then proposed in the following years by several authors [Sonquist 1970, Messenger and Mandell 1972, Gillo 1972, Sonquist et al. 1974]. Without contest however, the principal investigators that have driven research on the modern methodological principles are [Breiman et al. 1984], [Friedman 1977, 1979] and [Quinlan et al. 1979, Quinlan 1986] who simultaneously and independently proposed very close algorithms for the induction of tree-based models. Most notably, the unifying work of [Breiman et al. 1984], later complemented with the work of [Quinlan 1996], have set decision trees into a simple and consistent methodological framework, which largely contributed in making them easy to understand and easy to use by a large audience in a variety of applications.

Linear regression is one of the most widely used statistical techniques. The problem of regression consists of obtaining a functional model that relates the value of a target continuous variable  $Y$  with the values of the predictor variables  $X_1, X_2, \dots, X_m$ . A linear form is assumed for the unknown regression function and the parameters of the model are estimated using a least squares criterion. Predictors like linear or polynomial regression are global models, where a single predictive formula is assumed to hold over the entire data space. When the data has lots of features which interact in complicated, nonlinear ways, assembling a single global model can be difficult, and hopelessly confusing when you do succeed. Global parametric approaches, like regression, are widely used and give good predictive results when the assumed model correctly fits the data.

An alternative approach to nonlinear regression is to sub-divide, or partition, the data space into smaller regions, where the interactions are more manageable. We then partition the subdivisions 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. Therefore, in this case, the global model has two parts: one is just the recursive partition, the other is a simple model for each cell of the partition. Regression trees belong to the class of recursive partitioning algorithms. The seminal algorithm for learning regression trees from data is the CART algorithm as described in [Breiman et al. 1984]. We first provide a brief overview of how trees are learned, using a conceptual example adapted from [Hastie et al. 2009].

Tree-based methods partition the feature space into a set of rectangles (more formally, hyper-rectangles) and then fit a simple model (e.g. like a constant) 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 4.1 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 matters to only consider recursive binary partitions, like the ones shown in the top right plot of Figure 4.1. 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 4.1, 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 original data-space into the five regions  $R_1, R_2, \dots, R_5$  shown in the figure. The corresponding regression tree model predicts  $Y$  with a constant  $c_i$  in region  $R_i$ , that is,

$$\hat{T}(X) = \sum_{i=1}^5 c_i I \{(X_1, X_2) \in R_i\} \quad (4.1)$$

**Definition 4.5.** *A tree is a graph  $G = (V, E)$  in which any two vertices (or nodes) are connected by exactly one path.*

**Definition 4.6.** *A rooted tree is a tree in which one of the nodes has been designated as the root. We additionally assume that a rooted tree is a directed graph, where all the edges are directed away from the root.*

**Definition 4.7.** *If there exists an edge from  $t_1$  to  $t_2$  (i.e., if  $(t_1, t_2) \in E$ ) then node  $t_1$  is said to be the parent of node  $t_2$  while node  $t_2$  is said to be a child of node  $t_1$ .*

**Definition 4.8.** *In a rooted tree, a node is said to be internal if it has one or more children and terminal if it has no children. Terminal nodes are also known as leaves.*

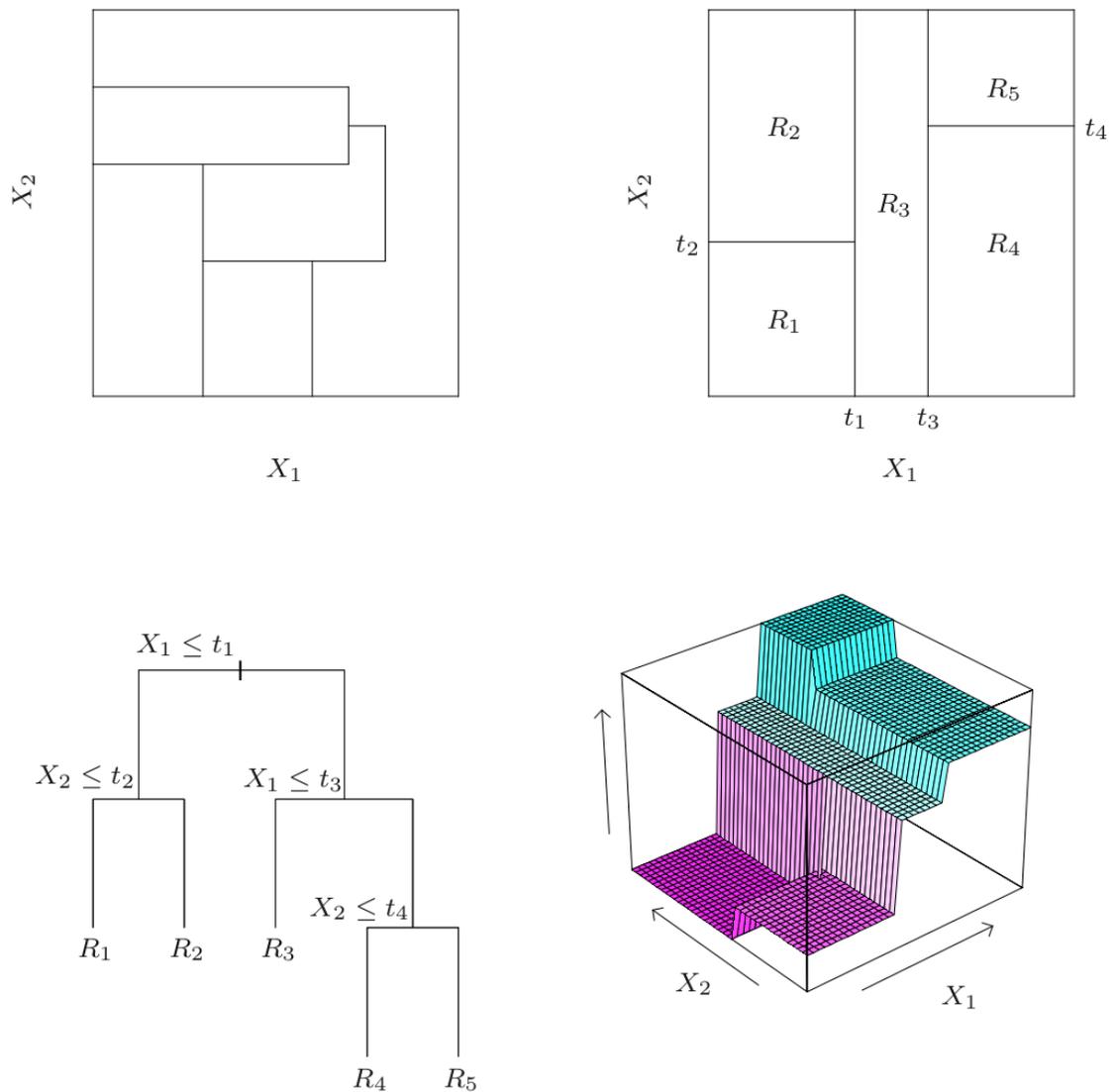


Figure 4.1: Recursive partitioning and regression trees. Top right image shows a partition of a two-dimensional feature space by recursive binary splitting, as used in CART, applied to some conceptual data. Top left figure shows a general partition that cannot be obtained from recursive binary splitting. Bottom left image shows the tree corresponding to the partition in the top right partition, and a perspective plot of the prediction surface appears in the bottom right image. Example adapted from [Hastie et al. 2009]

**Definition 4.9.** A binary tree is a rooted tree where all internal nodes have exactly two children.

This data partitioning model described earlier model can be represented by the binary tree shown in the bottom left of Figure 4.1. The full dataset sits at the top or the root of the tree. Observations satisfying the condition at each junction 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  $R_1, R_2, \dots, R_5$ . The bottom right plot of Figure 4.1 shows the perspective plot of the regression surface obtained as a result of building a regression tree.

### 4.2.1 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.

$$\sum (y_i - \hat{T}(x_i))^2 \quad (4.2)$$

For this criteria it is easy to see that the best response  $c_i$  (from euquation 4.1 for  $y_i$  from partition  $R_i$  is just the average of output samples in the region  $R_i$  i.e

$$c_i = \text{avg}(y_i | x_i \in R_i) \quad (4.3)$$

Finding the best binary partition in terms of minimum sum of squares is generally computationally infeasible. Therefore, 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

$$\begin{aligned} R_L(j, s) &= \{X | X_j \leq s\}, \\ R_R(j, s) &= \{X | X_j > s\} \end{aligned} \quad (4.4)$$

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

$$\min_{j,s} \left[ \min_{c_L} \sum_{x_i \in R_L(j,s)} (y_i - c_L)^2 + \min_{c_R} \sum_{x_i \in R_R(j,s)} (y_i - c_R)^2 \right] \quad (4.5)$$

where, for any choice of  $j$  and  $s$ , the inner minimization in equation 4.5 is solved using

$$\begin{aligned} c_L &= \text{avg}(y_i | x_i \in R_L(j, s)) \\ c_R &= \text{avg}(y_i | x_i \in R_R(j, s)) \end{aligned} \quad (4.6)$$

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. Since multiway splits can be achieved by a series of binary splits, the latter are preferred.

## 4.2.2 Stopping Criteria and Pruning

Every recursive algorithm needs to know when its done, i.e it requires a stopping criteria. For regression trees this means when to stop splitting the nodes. Obviously nodes which contain only one data point cannot be split further, but giving each observation its own leaf is unlikely to generalize well. Clearly 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 would be to split tree nodes only if the decrease in sum-of-squares due to the split exceeds some threshold. This strategy is too short-sighted, however, since a seemingly worthless split might lead to a very good split below it. A better and 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, described next

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,

$$\begin{aligned} N_i &= \# \{x_i \in R_i\}, \\ \hat{c}_i &= \frac{1}{N_i} \sum_{x_i \in R_i} y_i, \\ Q_i(T) &= \frac{1}{N_i} \sum_{x_i \in R_i} (y_i - \hat{c}_i)^2 \end{aligned} \tag{4.7}$$

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:

$$C_\alpha(T) = \sum_{i=1}^{|T|} N_i Q_i(T) + \alpha |T| \tag{4.8}$$

The goal is to find, for each  $\alpha$ , 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  $\alpha$  one can show that there is a unique smallest subtree  $T_\alpha$  that minimizes  $C_\alpha(T)$  [Ripley 1996]. Estimation of  $\alpha$  is achieved by cross-validation.

### 4.3 Ensemble Methods

Regression trees obtain good predictive accuracy in many domains. However, the simple models used in their leaves have some limitations regarding the kind of functions they are able to approximate. The problem with trees is their high variance and that they can over fit the data. It is the price to be paid for estimating a simple, tree-based structure from the data. Often a small change in the data can result in a different series of splits. The main reason for this instability is the hierarchical nature of the process: the effect of an error in the top split is propagated down to all of the splits below it. While pruning and cross validation can help reduce over fitting, we use ensemble methods for growing more stable trees.

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability and robustness over a single estimator. Two families of ensemble methods are usually distinguished:

- (a) In averaging methods, the driving principle is to build several estimators independently and then to average their predictions. On average, the combined estimator is usually better than any of the single base estimator because its variance is reduced.
- (b) By contrast, in boosting methods, base estimators are built sequentially and one tries to reduce the bias of the combined estimator. The motivation is to combine several weak models to produce a powerful ensemble.

#### Random Forests

Random forests or tree-bagging are a type of ensemble method which makes predictions by averaging over the predictions of several independent base models. The essential idea is to average many noisy but approximately unbiased trees, and hence reduce the variance. Injecting randomness into the tree construction can happen in many ways. The choice of which dimensions to use as split candidates at each leaf can be randomized, as well as the choice of coefficients for random combinations of features. Another common method for introducing randomness is to build each tree using a bootstrapped or sub-sampled data set. In this way, each tree in the forest is trained on slightly different data, which introduces differences between the trees. Not all estimators can be improved by shaking up the data like this. However, highly non-linear estimators, such as trees, benefit the most. For a more comprehensive review we refer the reader to [Breiman 2001].

## Boosted Regression Trees

In boosting, trees are fitted iteratively to the training data, using appropriate methods gradually to increase emphasis on observations modeled poorly by the existing collection of trees. A boosted regression tree (BRT) model can be understood as an additive regression model in which individual terms are simple trees, fitted in a forward, stage-wise fashion [Elith et al. 2008]. Stage-wise implies that existing trees are left unchanged as the model is enlarged. Only the fitted value for each observation is re-estimated at each step to reflect the contribution of the newly added tree.

## 4.4 Model Based Regression Trees

One of the key aspects of the previously described algorithms for growing trees is that they make the use of point predictions in the leaves of the tree i.e. the prediction of the response variable is simply obtained by averaging the samples in the partition corresponding to the leaf e.g. shown in 4.3. In model based regression trees, the definition of the leaf of a tree is extended to allow for simple functions, other than averaging, in the leaves which predict the response. The use of linear regression functions in the leaves of the tree, or local linear regression, has been presented in [Quinlan et al. 1992] in an algorithm called M5. [Friedman 1991] describes another variant of this idea, where instead of linear regression in the leaves one can fit regression splines over the predictor variables. This is called the multivariate adaptive regression splines (or MARS) algorithm for growing a regression tree.

## 4.5 Comparison with k-means

One of the most comprehensible non-parametric methods is k-nearest-neighbors or k-means clustering: find the points which are most similar to you, and do what, on average, they do. There are two big drawbacks to it: first, "similar" is defined entirely in terms of the inputs, not the response; second, k is constant everywhere, when some points just might have more very-similar neighbors than others. Trees get around both problems: leaves correspond to regions of the input space (a neighborhood), but one where the responses are similar, as well as the inputs being nearby; and their size can vary arbitrarily. Regression trees are adaptive nearest-neighbor methods.

# Chapter 5

## Data-Driven Modeling and Control for DR

To take advantage of real-time pricing and DR programs, the commercial, industrial and institutional *C/I/I* consumers must monitor electricity prices and be flexible in the ways they choose to use electricity. The challenge for these large consumers of electricity is to be able to predict their aggregate power consumption accurately and at a fast time scale in order to take suitable load curtailment control actions.

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 [Crawley et al. 2001]) is very cost and time prohibitive and requires retrofitting the building with several sensors [Sturzenegger et al. 2015]; 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 [R et al. 2012].
2. **Limitations of rule-based DR:** The building's operating conditions, internal thermal disturbances and environmental conditions must all be taken into

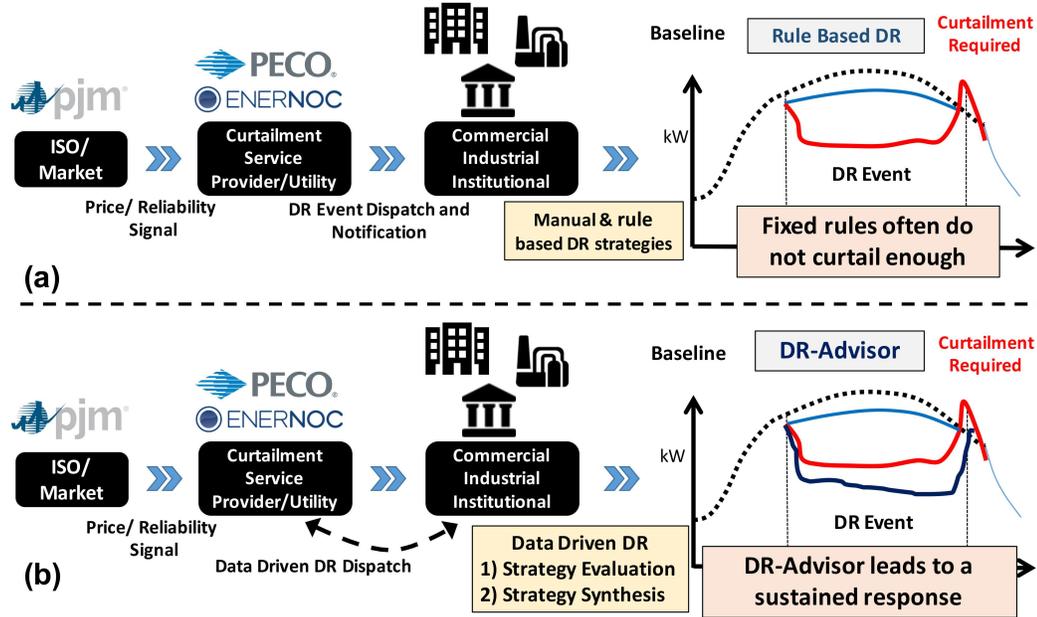


Figure 5.1: Majority of DR today is manual and rule-based. (a) The fixed rule based DR is inconsistent and could under-perform compared to the required curtailment, resulting in DR penalties. (b) Using data-driven models DR-Advisor uses DR strategy evaluation and DR strategy synthesis for a sustained and sufficient curtailment.

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. 5.1(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 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. The challenge is the increasing complexity of possible scenarios. There is a limit as to what can be pre-programmed and only a finite number of operations can be managed using this approach. There are also some operations that cannot be fully managed with a rules-based approach.

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.

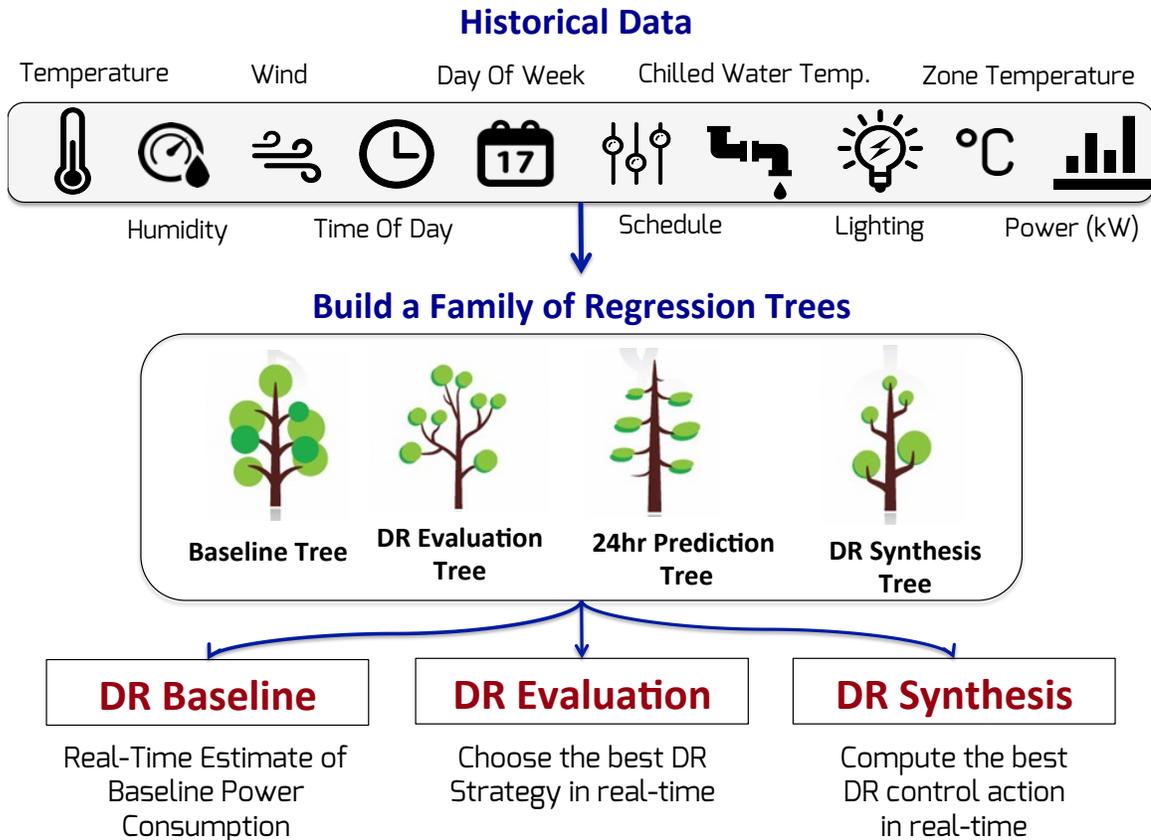


Figure 5.2: DR-Advisor Architecture

In a large building, it is difficult to assess 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 problem for the *C/I/I* consumers is further compounded by the fact that there are thousands of different rate plans provided by the utilities. For comparison, in the United States, there are only 5 major telecoms carriers which serve 99.999% of the

entire market with 62 different rate plans, as opposed to over 3,100 utility companies serving the commercial and industrial consumers with 18,300 rate classes. Therefore any DR curtailment strategy must also account for the utility tariff structure and financial reward while responding to a demand response request.

Regardless of the specific DR program, upon receiving a notification for a demand response event, the building’s facilities manager must determine an appropriate or optimal control strategy to achieve the required power curtailment level. These control strategies can include adjusting zone temperature set-points, increasing 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 assess 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. Moreover, the performance of any DR strategy will vary due to the outside weather conditions, during the DR event.

The goal with data-driven demand response is to make the best of both worlds; i.e. keep the 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.

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 5.2). 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. By using modified model based regression trees, we can also determine good demand response control policies for the duration of the DR event that achieve the required curtailment, without adversely affecting the system and causing any kick-backs.

This chapter is organized as follows: Section 5.1 describes the three big challenges with real-time demand response. In Section 5.2, we describe how regression trees based data-driven methods can be used for the DR challenges. Section 5.3, presents a new algorithm to perform real-time control with regression trees and applied to the DR synthesis problem. Section 5.4 makes a compelling case for regression trees being a suitable choice of data-driven models for real-time DR. Section 5.5 describes the MATLAB based DR-Advisor toolbox. Section 5.6 presents a comprehensive case study with DR-Advisor on data from 8 real buildings on Penn’s campus, a large office building, a virtual test-bed and evaluation of the tool on bench-marking data from ASHRAE’s energy prediction competition. In Section 5.7, a detailed survey of related work has been presented. We conclude this chapter in Section 5.8 with a summary of our results and a few concluding remarks.

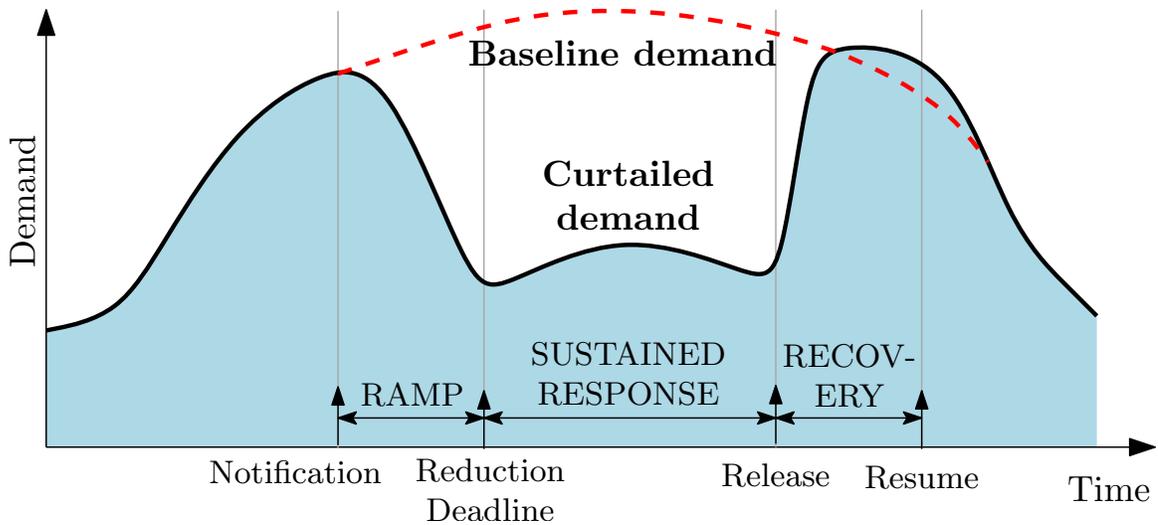


Figure 5.3: Demand Response Time line and Baseline Prediction.

## 5.1 Problem definition

The time line of a typical DR event consists of three periods (Figure 5.3). The main period during which the demand needs to be curtailed is the *sustained response period*. The start of this period, i.e. the time by which the target reduction must be achieved, is the *reduction deadline*. Prior to that deadline, an *event notification* will be issued, at the notification time. The period between this time and the reduction deadline is the *ramp period*, during which the demand transitions from the normal level to the reduced level. The end of the sustained response period – the *release time* – is when the main curtailment is released. It would take the demand side some time to resume the normal operation, during the *recovery period*. A possible phenomenon during this period is the DR rebound, when loads consume more electricity than normal to recover. Large DR rebound is undesirable, and DR strategies should be designed to mitigate this rebound. The DR event ends at the end of the recovery period.

We focus on three challenging problems of demand response

- (i) Demand response baseline prediction,
- (ii) Demand response strategy evaluation, and
- (iii) Demand response strategy synthesis.

Each of these problems is described below:

### 5.1.1 DR Baseline Prediction

A baseline is an estimate of the electricity that would have been consumed by a customer in the absence of a demand response event. Typical demand response programs rely upon financial incentive for customers based on the extent to which they reduce

their energy consumption and therefore require a reliable system to measure the energy reduction. For this reason the measurement and verification of demand response is the most critical component of any DR program. The baseline is the primary tool for measuring curtailment during a DR event. Baselines enable grid operators and utilities to measure performance of DR resources and determine financial paybacks. Baselines are established in form of estimates of what the customer did not do, but would have done, had there not been a DR event. As shown in Figure 5.3, actual meter data is compared with the baseline demand to determine the curtailment achieved by the customer. The goal is to learn a predictive model  $g()$  which relates the baseline power consumption estimate  $Y_{base}^{\wedge}$  to the forecast of the weather conditions and building schedule for the duration of the DR-event i.e.  $Y_{base}^{\wedge} = g(\text{weather}, \text{schedule})$

### 5.1.2 DR Strategy Evaluation

During a DR event notification, there are several options available to a buildings manager in the form of a control actions. These include setbacks in the zone temperature set-point, increasing 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 and switching off large electrical systems like elevators, if required. A DR strategy refers to what control actions, and at what times, a system (lighting, HVAC or plug loads) will actuate. A strategy has one or more steps (control actions at times relative to the onset), and a time validity window for when it can be used. Furthermore, there could be several of such fixed DR strategies, but only one specific strategy can be used at a time. This brings us to our question, *how can we choose good DR strategies from a pre-determined set of strategies ?*

This is the problem of demand response strategy evaluation. Instead of predicting the baseline power consumption  $Y_{base}^{\wedge}$ , in this case we want the ability to predict the actual response of the building  $Y_{kW}^{\wedge}$  due to any given strategy. For example, in Fig. 5.4, 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 ( $\in \{i, j, \dots k \dots N\}$ ) which leads to the largest load curtailment. The resulting strategy could be a combination of switching between the available set of strategies.

### 5.1.3 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

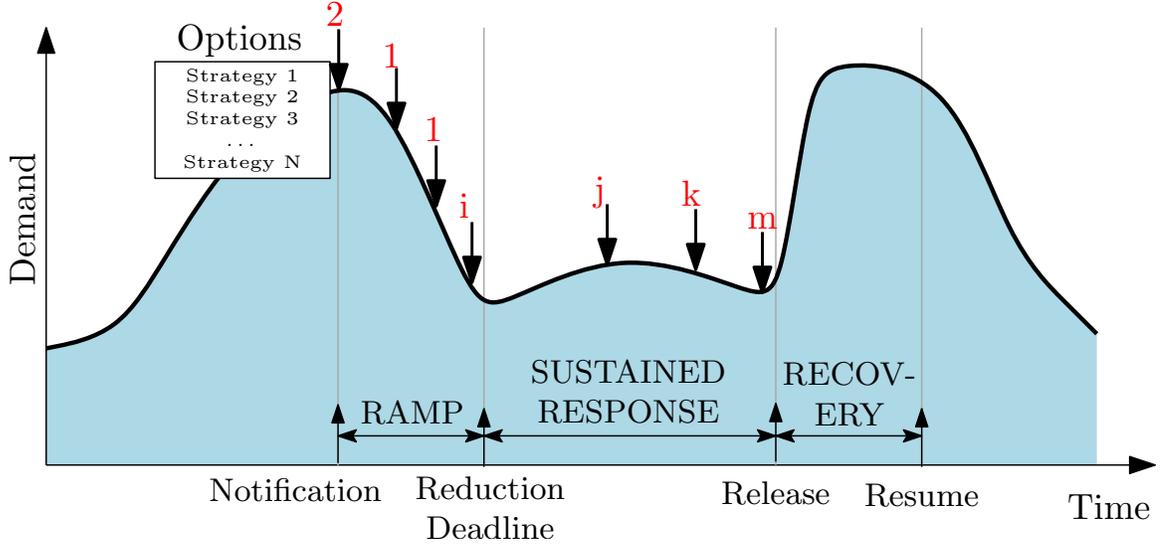


Figure 5.4: Demand Response Strategy Evaluation.

set of control variables,  $\mathbb{X}_c$ , such that

$$\begin{aligned}
 & \underset{\mathbb{X}_c}{\text{minimize}} && f(\hat{Y}_{kW}) \\
 & \text{subject to} && \hat{Y}_{kW} = h(\mathbb{X}_c) \\
 & && \mathbb{X}_c \in \mathbb{X}_{safe}
 \end{aligned} \tag{5.1}$$

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-based 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.

In all the three problems, a common recurring theme is the ability to predict the power consumption of the building. Any data driven approach which intends to solve these problems must have the capability to predict the power consumption of the building under different circumstances and due to different control actions.

## 5.2 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  $[X_1, X_2, \dots, X_m]$  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 [Breiman et al. 1984] and in Chapter 4, Section 4.2.

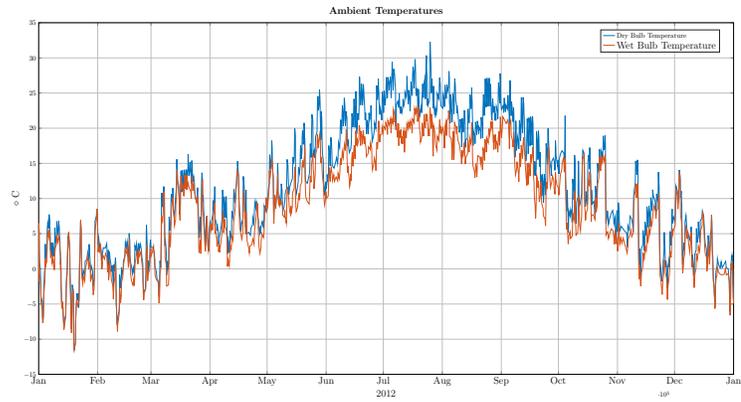
Regression trees based approaches are our choice of data-driven models for DR-Advisor. The primary reason for this modeling choice is that regression trees are highly interpretable, by design. Interpretability is a fundamental desirable quality in any predictive model. Complex predictive models like 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 [Giraud-Carrier 1998], one that is essential for incorporating human centric models in cyber-physical energy systems. Additional advantages of using regression trees for solving these Dr challenges is presented in Section 5.4.

### 5.2.1 Data-Description

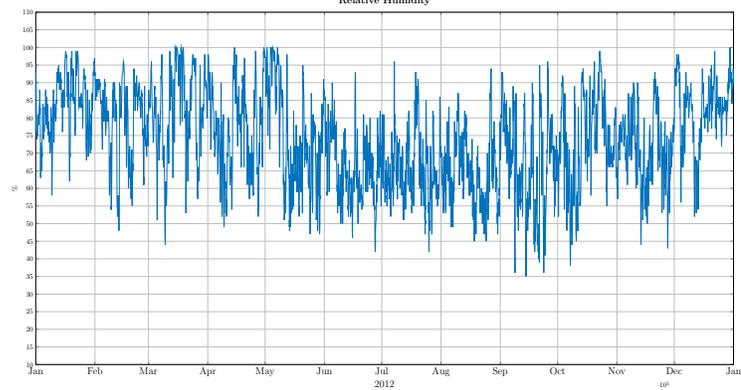
In order to build a regression tree which can predict the power consumption of the building, we need to train on time-stamped historical data. The data that we use can be divided into three different categories as described below:

### 5.2.2 Weather Data

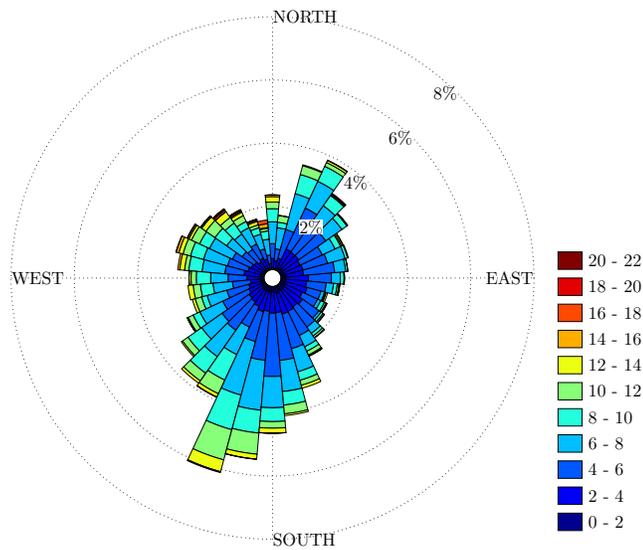
First and foremost, we need historical data which describes the weather. Weather data includes measurements of the dry bulb temperature, wet bulb temperature, relative humidity and wind characteristics. Sometimes, solar irradiation information can also be included, if available. Figure 5.5 shows the weather data from 2012 used for training the regression tree. Fig 5.5a shows the dry and the wet bulb temperature measurements, Fig 5.5b shows the relative humidity and Fig 5.5c shows a wind rose plot of the wind data. A wind rose plot gives a very succinct but information laden view of how wind speed and direction are distributed at the location of the building. Presented in a circular format, the wind rose shows the distribution of winds blowing from different directions. The length of each "spoke" around the circle is the frequency



(a) Temperature



(b) Humidity



(c) Wind

Figure 5.5: Actual meteorological weather data used for building the regression tree. Fig 5.5a shows the dry and the wet bulb temperature measurements, Fig 5.5b shows the relative humidity and Fig 5.5c shows a wind rose plot which depicts the distribution of wind speed (m/s) and wind direction.

of time that the wind blows from that direction. In addition to that each spoke is broken down into discrete categories that show the speed of the wind (in m/s).

### 5.2.3 Schedule Data

Using time-stamp information in the building power consumption data, we can create *proxy* predictor variables which can capture patterns of electricity consumption for e.g. due to occupancy or electrical equipment schedules. No direct measurement of the occupancy is required. Specifically, we use the following three *proxy* predictors:

1. **Day of Week:** This is a categorical predictor which takes values from 1 – 5 depending on the day of the week. This variable can partition the data space on patterns which occur on specific days of the week. For instance, there could be a big auditorium which is only used for fixed days of the week.
2. **Weekends and Holidays:** For most buildings the equipment schedule and occupancy patterns change significantly over weekends and holidays. For our virtual test-bed, weekends, special days and holidays are treated equivalently therefore they can be represented by a single binary predictor which takes the values  $\{1, -1\}$ . In case, weekend operations are different from holiday/vacation operations, one could simply use more binary predictors for each type of special days.
3. **Time of Day:** This is quite an important predictor as it can adequately capture daily patterns in power consumption due to occupancy, lighting and appliance use without directly measuring any one of them. It can also capture changes in set-points throughout the day and the correlation between these changes and the overall power consumption. For example, the building may have a pre-cooling schedule during the early morning, or a set-point reset schedule for the evening which can be accounted for using the time of day *proxy* predictor.

Besides using proxy schedule predictors, sometimes, actual building schedules can also be used as training data for building the trees. The prime candidate for obtaining actual schedules, if possible, are temperature set-points schedules of chilled water supply, supply air temperature and zone air temperature on the HVAC side and lighting schedules. Using actual schedule information can greatly improve the accuracy of the power consumption prediction. While using schedule information is not necessary for demand response baselining, it is necessary for performing DR strategy evaluation, since a fixed strategy involves changing all or a few of these set-points.

### 5.2.4 Building Data

Lastly, since we are trying to predict the power consumption of the building, we require historical time-stamped power consumption (or meter) data. The power consumption is the response variable of the regression tree. For DR baselining, the

knowledge of the state of the building is not necessary. This is because the baseline problem is to estimate what the building *would have done*, rather than predict what the building will do. The state of the building is required for DR strategy evaluation and synthesis. The state of the building includes

- (i) Chilled Water Supply Temperature
- (ii) Hot Water Supply Temperature
- (iii) Zone Air Temperature
- (iv) Supply Air Temperature
- (v) Lighting power or fraction of peak lighting use.

A measurement of all, or a subset of these predictors is necessary if we are to learn and predict the actual power response of the building.

### 5.2.5 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 [Breiman 2001]. 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 [Elith et al. 2008].

### 5.2.6 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

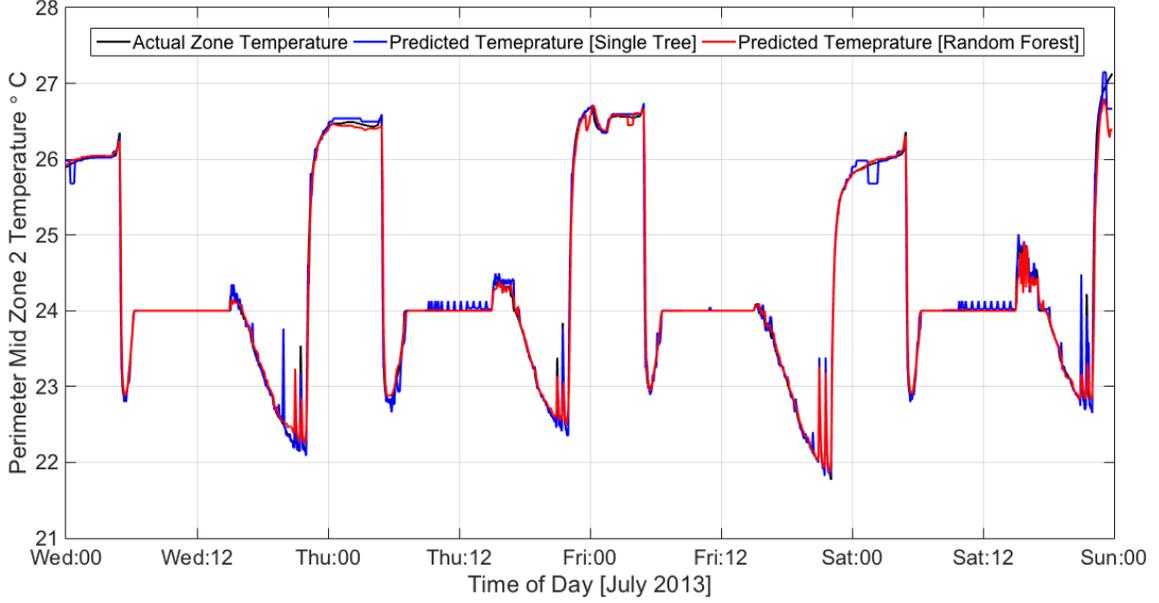


Figure 5.6: Comparison of actual and predicted zone temperature values using auto-regressive single and ensemble trees.

for the regression tree i.e. the tree structure is learned to approximate the following function:

$$Y_{kW}^{\hat{}}(t) = f([X_1, X_2, \dots, X_m, Y_{kW}(t-1), \dots, Y_{kW}(t-\delta)]) \quad (5.2)$$

where the predicted power consumption response  $Y_{kW}^{\hat{}}$  at time  $t$ , depends on previous values of the response itself  $[Y_{kW}(t-1), \dots, Y_{kW}(t-\delta)]$  and  $\delta$  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. Figure 5.6 shows the comparison between predicted and actual zone temperature values for one of the building zones. The accuracy of the random forest auto-regressive tree is 97.54%. This example shows that the auto-regressive trees are well suited for temperature predictions as well.

### 5.3 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 5.1.3. This is one of the primary contributions of this thesis.

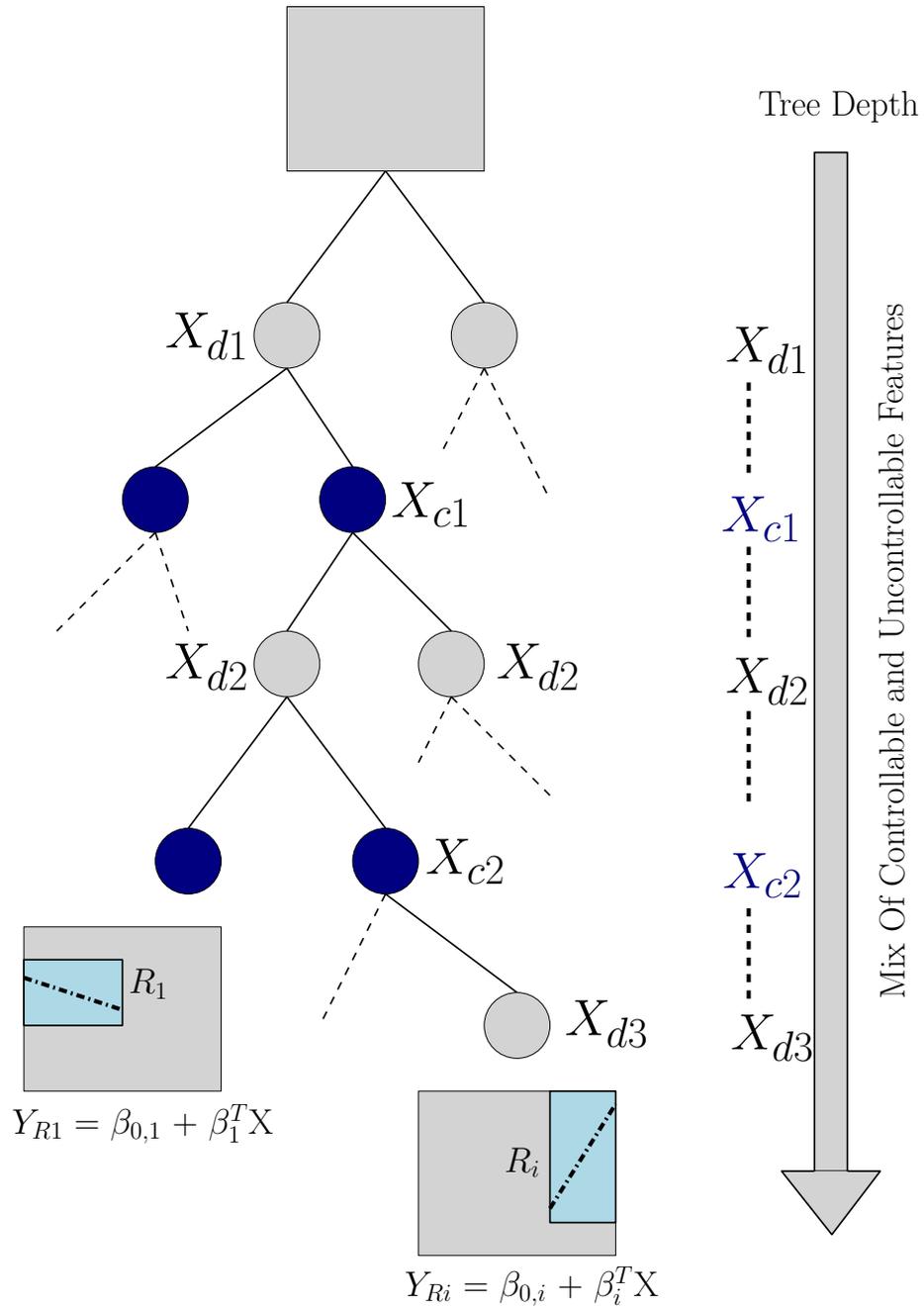


Figure 5.7: Example of a regression tree with linear regression model in leaves. Not suitable for control due to the mixed order of the controllable  $X_c$  (solid blue) and uncontrollable  $X_d$  features.

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  $X_1, X_2, \dots, X_m$ ; i.e.  $Y = f([X_1, X_2, \dots, X_m])$ . Given a forecast of the features  $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_m$  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.

### 5.3.1 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 presented briefly in Section 4.4 and in [Quinlan et al. 1992, Friedman 1991] to *model based control with regression trees (mbCRT)*.

Figure 5.7 shows an example of how manipulated and non-manipulated features can get distributed at different depths of model based regression tree which uses the a linear regression function in the leaves of the tree:

$$\hat{Y}_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X} \quad (5.3)$$

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 5.8) In every region at the leaves of the “disturbance” tree a linear model is fit but only on the control variables  $\mathbb{X}_c$ :

$$Y_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X}_c \quad (5.4)$$

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

$$Y = f(X_1, X_2, \dots, X_m)$$

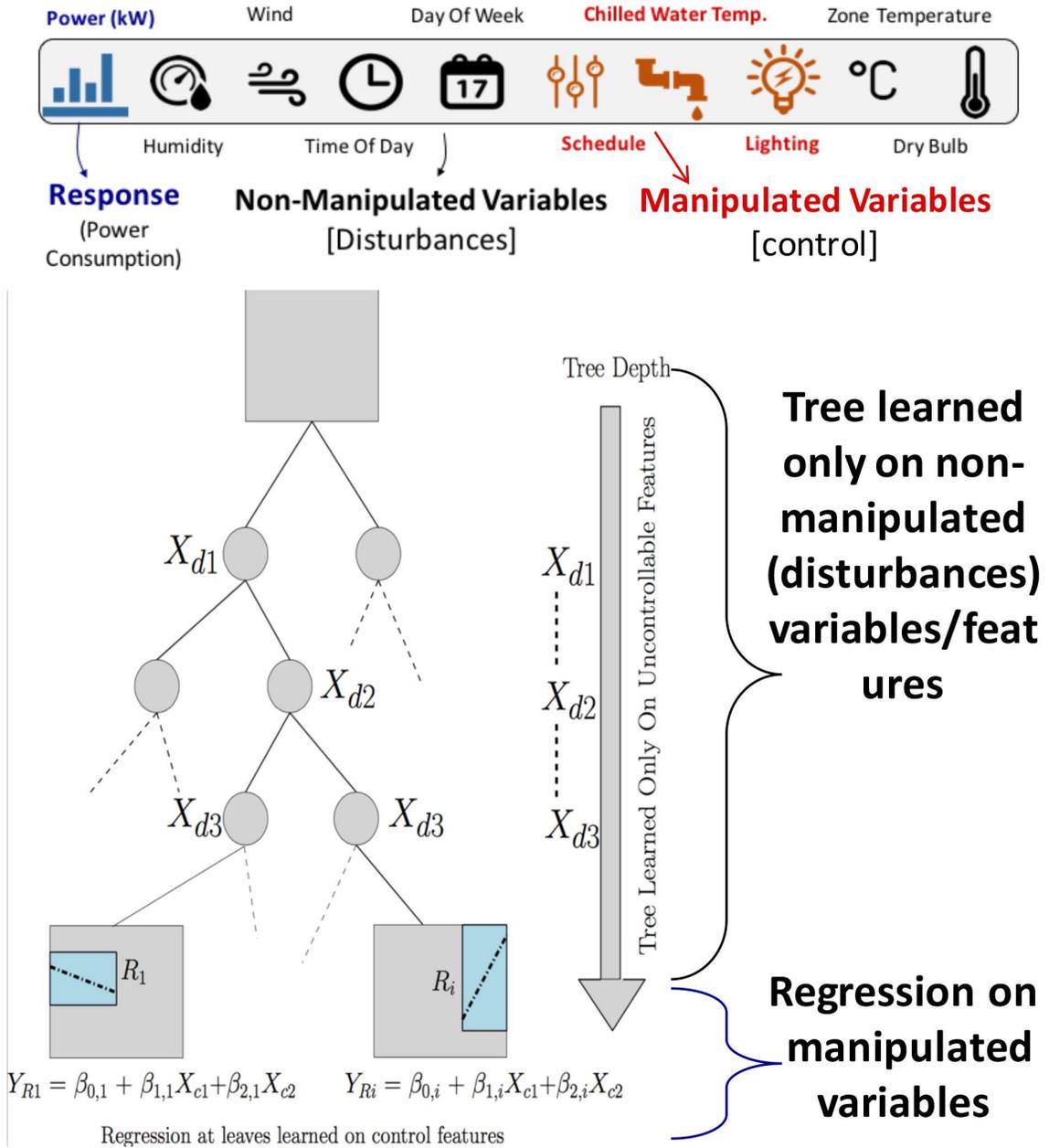


Figure 5.8: Example of a tree structure obtained using the mbCRT algorithm. The separation of variables allows using the linear model in the leaf to use only control variables.

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

---

- 1: DESIGN TIME
  - 2: **procedure** MODEL TRAINING
  - 3:     *Separation of Variables*
  - 4:     Set  $\mathbb{X}_c \leftarrow$  non-manipulated features
  - 5:     Set  $\mathbb{X}_d \leftarrow$  manipulated features
  - 6:     Build the power prediction tree  $T_{kW}$  with  $\mathbb{X}_d$
  - 7:     **for all** Regions  $R_i$  at the leaves of  $T_{kW}$  **do**
  - 8:         Fit linear model  $k\hat{W}_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X}_c$
  - 9:         Build  $q$  temperature trees  $T1, T2 \cdots Tq$  with  $\mathbb{X}_d$
  - 10:     **for all** Regions  $R_i$  at the leaves of  $Ti$  **do**
  - 11:         Fit linear model  $\hat{T}i = \beta_{0,i} + \beta_i^T \mathbb{X}_c$
  - 12: RUN TIME
  - 13: **procedure** CONTROL SYNTHESIS
  - 14:     At time  $t$  obtain forecast  $\hat{\mathbb{X}}_d(t+1)$  of disturbances  $\hat{X}_{d1}(t+1), \hat{X}_{d2}(t+1), \cdots$
  - 15:     Using  $\hat{\mathbb{X}}_d(t+1)$  determine the leaf and region  $R_{rt}$  for each tree.
  - 16:     Obtain the linear model at the leaf of each tree.
  - 17:     Solve optimization in Eq5.5 for optimal control action  $\mathbb{X}_c^*(t)$
- 

### 5.3.2 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 5.9 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

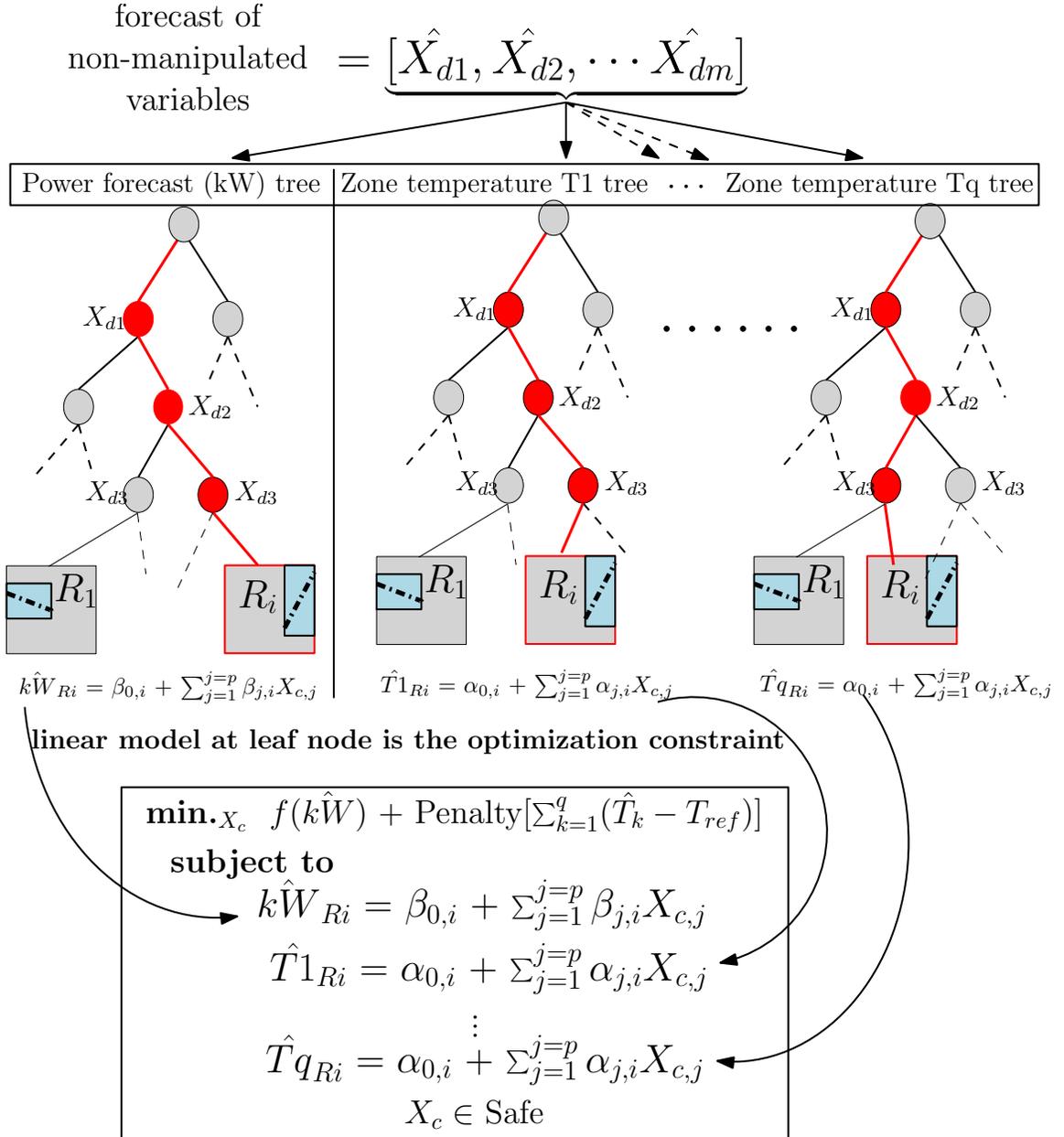


Figure 5.9: DR synthesis with thermal comfort constraints. Each tree is responsible for contributing one constraint tot the demand response optimization.

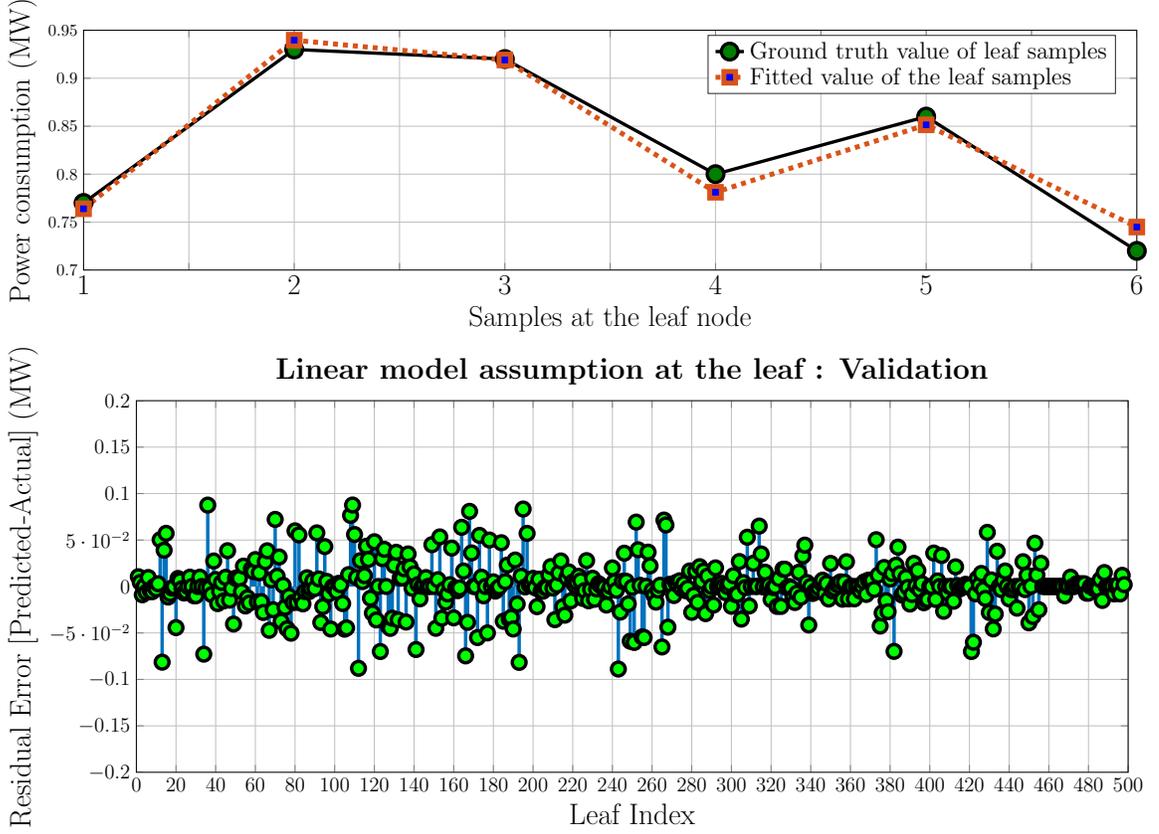


Figure 5.10: Linear model assumption at the leaves. The top figure shows the comparison between fitted values and ground truth values of power consumption for one of the leaves in the power consumption prediction tree. The bottom figure shows the residual error between fitted and actual power consumption values for all the leaf nodes of the tree.

temperatures and the manipulated variables. We can then solve the following DR synthesis optimization problem to obtain the values of the manipulated variables  $\mathbb{X}_c$ :

$$\underset{\mathbb{X}_c}{\text{minimize}} f(k\hat{W}) + \text{Penalty}\left[\sum_{k=1}^q (\hat{T}_k - T_{ref})\right]$$

subject to

$$\begin{aligned} k\hat{W} &= \beta_{0,i} + \beta_i^T \mathbb{X}_c \\ \hat{T}1 &= \alpha_{0,1} + \beta_1^T \mathbb{X}_c \\ &\dots \\ \hat{T}d &= \alpha_{0,q} + \beta_q^T \mathbb{X}_c \\ \mathbb{X}_c &\in \mathbb{X}_{safe} \end{aligned} \tag{5.5}$$

The linear model between the response variable  $Y_{Ri}$  and the control features  $\mathbb{X}_c$  is

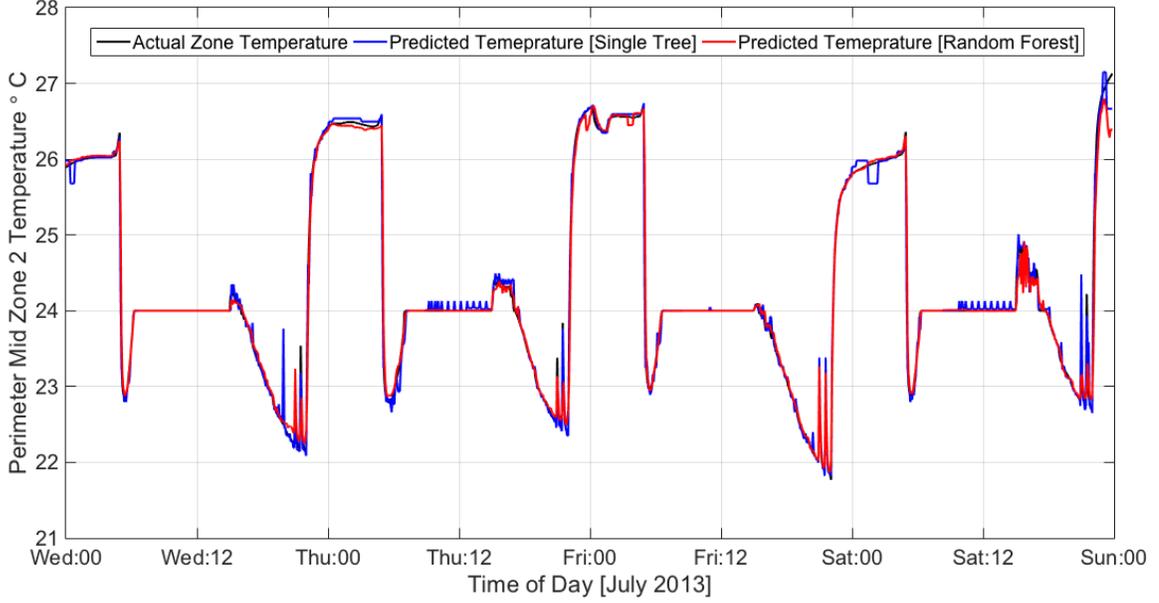


Figure 5.11: Comparison between actual and predicted power consumption values for the DoE commercial reference building. The comparison is between a single regression tree built with all the features and a regression tree with separation of variables.

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 5.10 shows that the linear model assumption in the leaves of the tree is a valid assumption.

Figure 5.11 shows the comparison between the predicted values and actual values of power consumption using two different regression trees. The first prediction is obtained from a regression tree built using all the features. It results in an accuracy of 98.21%. This is compared to the prediction accuracy of 97.66% obtained by using a regression tree in which non-manipulated variables are separated from manipulated variables. This comparison shows that there is a slight decrease in prediction accuracy due to the separation of variables. However, at the expense of the decreased accuracy we gain the ability of performing closed loop control with regression trees. Therefore, there is a trade-off between prediction accuracy and tree error and the control synthesis ability with the regression tree. 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.

## 5.4 The case for using regression trees for DR

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 5.1. 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  $pN\log N$  operations for an initial sort for each predictor, and typically another  $pN\log N$  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.
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

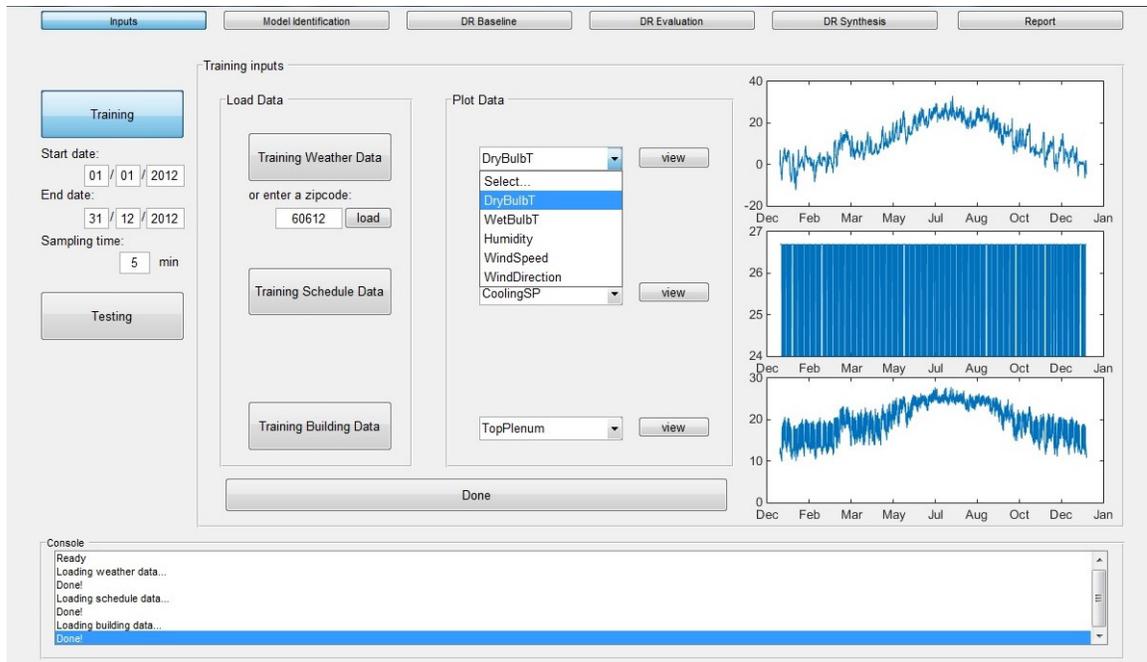


Figure 5.12: Screenshot of the DR-Advisor MATLAB based GUI.

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.

## 5.5 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 5.12) 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 5.13. 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 5.12) is displayed. Here the user can upload and specify any sensor data from the

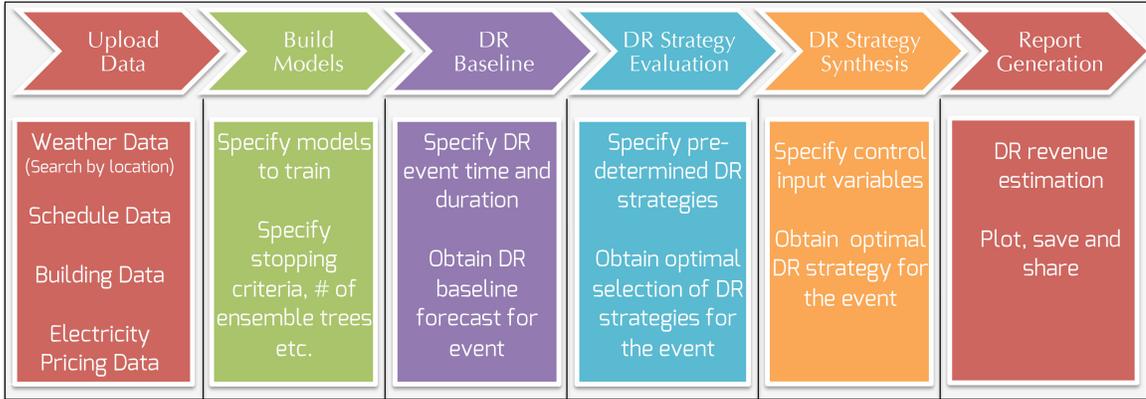


Figure 5.13: DR-Advisor Workflow

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 5.14. 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.
4. **DR Strategy Evaluation:** In this step of the workflow, the user first has to

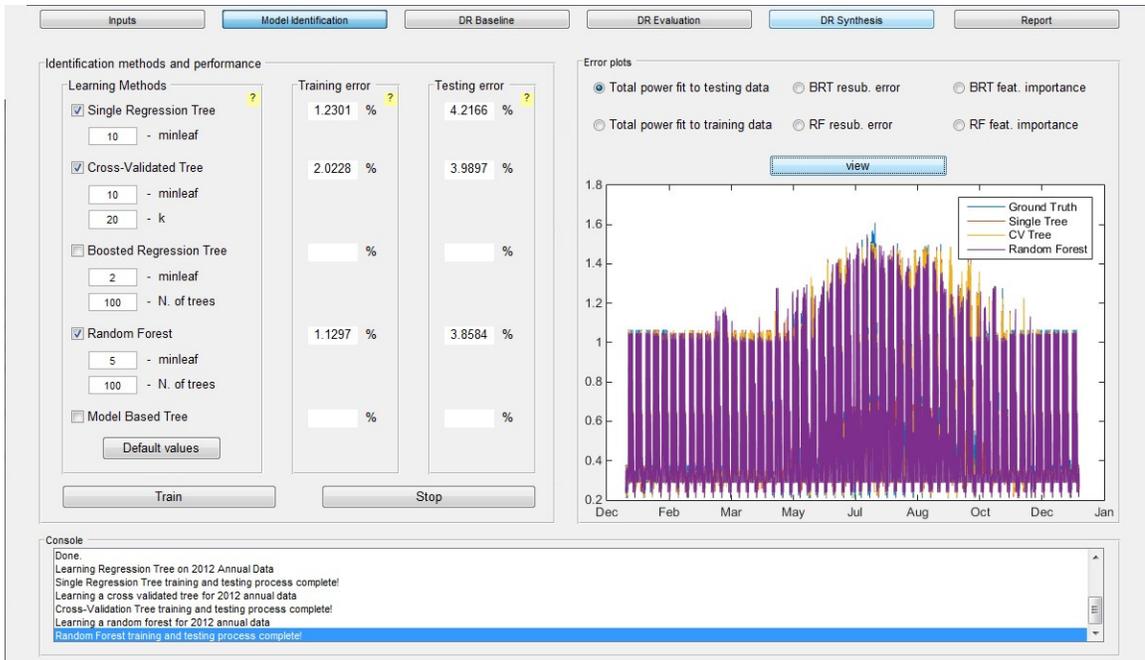


Figure 5.14: DRAdvisor model identification tab

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 5.3.1) 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 5.15: Left: Building 101 in Philadelphia; Right: 3D rendering of the DoE commercial reference building in EnergyPlus.



Figure 5.16: 8 different buildings on Penn campus were modeled with DR-Advisor

## 5.6 Case study

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 5.1) and we compare the performance of our tool with other data-driven methods.

### 5.6.1 Building and Data Description

We use historical weather and power consumption data from 8 buildings on the Penn campus (Figure 5.16). 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 5.1.

We also use data from Building 101, a large office building located in Philadelphia.

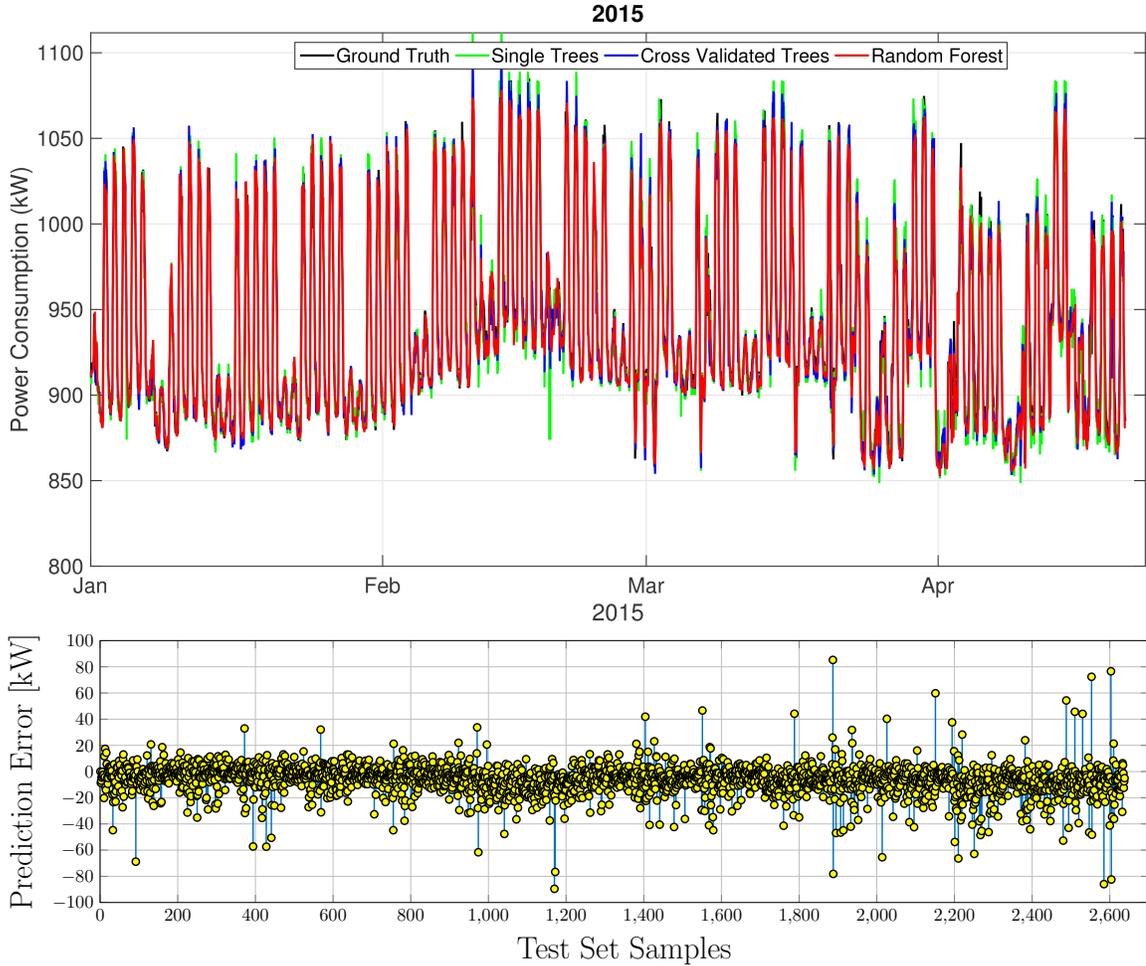


Figure 5.17: Model validation for the clinical research building at Penn.

This building is the temporary headquarters of the U.S. Department of Energy’s consortium for buildings energy innovation. It is a three floored building with a gross building floor area of 75,156 sq-ft (Figure 5.15(Left)). There are a total of 27 conditioned zones in the building served by 3 air handling units.

The third building is the DoE Commercial Reference Building (DoE CRB) simulated in EnergyPlus [Deru et al. 2010]. This virtual test-bed is a large 12 story office building consisting of 73 zones with a total area of 500,000 sq ft (Figure 5.15 (Right)). There are 2,397 people in the building during peak occupancy. The building has 2 electric water-cooled chillers, variable air volume (VAV) supply air terminals with reheat and plenum zones and a single gas based boiler. During peak load conditions the building can consume up to 1.6 MW of power. EnergyPlus provides typical meteorological year data files for many sites which are generated as averages of different weather characteristics across the past 15-30 years. For the simulation of the DoE CRB building we use Actual Meteorological Year (AMY) data from Chicago for the years 2012 and 2013.

Table 5.1: 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

## 5.6.2 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 5.17. 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 5.1. 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.

For Building 101, multiple regression trees were trained on weather and power consumption data from the year 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. February 2015. The predictions are obtained in time-steps of 5 minutes. The predictions on the test set are compared to the actual power consumption of the building during the test-set period. This comparison is shown in Figure 5.18. The following algorithms were evaluated: single regression tree, k-fold cross validated trees, boosted regression trees, random forests and model based regression trees (M5) (Section 4.3).

The accuracy values for Building 101 are listed in Table 5.2. For this data set the boosted regression tree (NRMSE 3.16%) algorithm and random forests (NRMSE 3.41%) can predict the power consumption with very high accuracy ( $\sim 97\%$ ).

For the ensemble methods we grow 500 trees each for boosted regression trees and random forests. The out of bag re-substitution error for the random forests are shown in Figure 5.19. We can see how the RMSE settles to a low value with a few hundred trees. We then use the best forest with the lowest RMSE and the corresponding

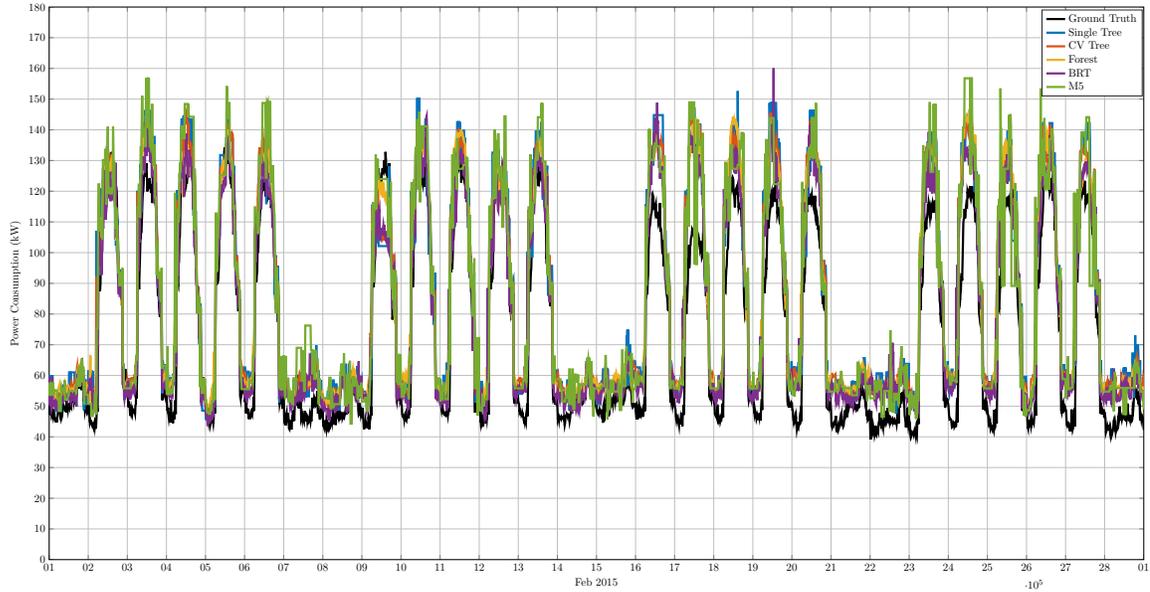


Figure 5.18: Model validation for Building 101. Comparison between the the actual power consumption of the building (ground truth) and the power consumption prediction obtained from DR-Advisor

Method	Accuracy %
Single Tree	95.18
Cross-Validated Tree	96.31
Boosted Regression Tree	<b>96.84</b>
Random Forest	96.59
M5 Model Based RT	95.12

Table 5.2: Comparison of methods on Building 101 data

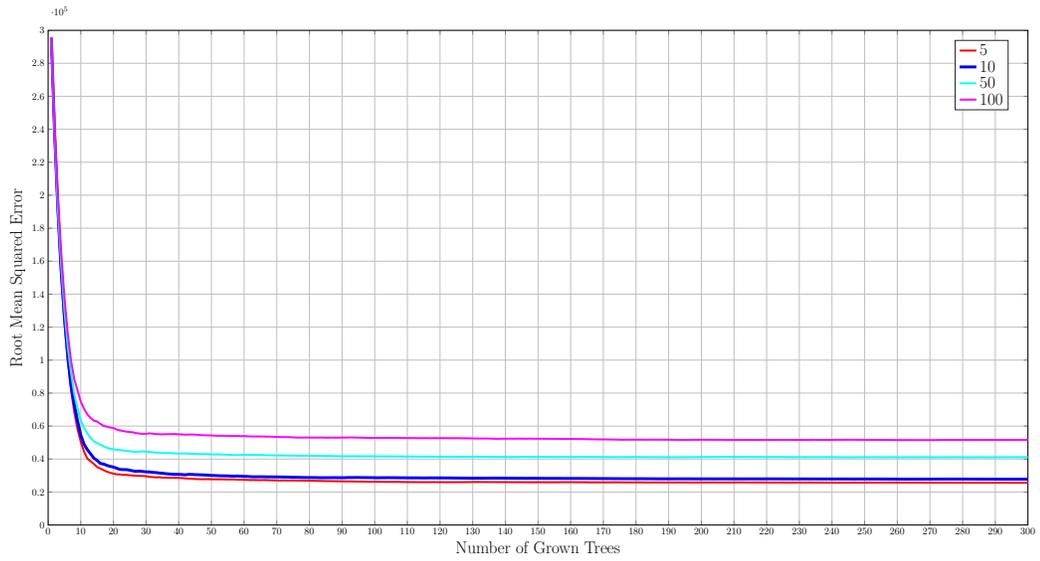


Figure 5.19: Root mean square error obtained on the training data as a function of the number of trees in the random forest. Each curve corresponds to a different threshold of the MinLeaf criteria.

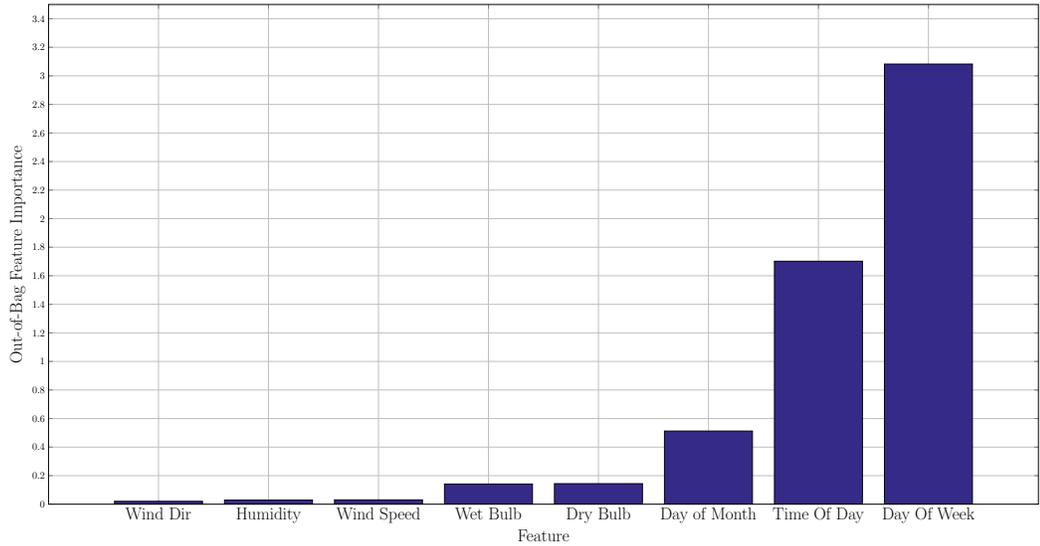


Figure 5.20: Predictor variables (feature) importance in the random forest ensemble.

lowest **MinLeaf** for prediction on the test data-set. By growing a large number of trees for the ensemble methods, we can get a good estimate of the importance of the predictor variables as shown in Figure 5.20.

### 5.6.3 Energy Prediction Benchmarking

”*The Great Energy Predictor Shootout - The First Building Data Analysis and Prediction Competition*” was held in 1993-94 by ASHRAE [Kreider and Haberl 1994].

Two distinct data sets were provided for prediction (training and testing). Contestants were given these two sets of independent variables with the corresponding values of dependent variables, e.g., energy usage. The accuracy of predictions of the dependent variables from values of independent variables from the test data set was the criteria for judging this competition.

The following criteria was used by the organizers for assessing the respective accuracies of the entries when analyzing the testing set:

$$\text{Coefficient of Variation, CV} \\ CV = \frac{\sqrt{\{\sum_{t=1}^n (\hat{y}_t - y)^2\} / n}}{\bar{y}}$$

The coefficient of variation is the same as the normalized root mean square error (NRMSE). The goal of the ASHRAE challenge was to explore and evaluate simpler 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. The winner of the competition was an entry from David Mackay [MacKay et al. 1994]. Mackay’s algorithm was based on Bayesian modeling using neural networks, with an ”Automatic Relevance Determination” (ARD) prior to help select the relevant variables from the large number of possible inputs. Although this algorithm won the competition by some margin, a large fraction of the other highly ranked algorithms were also based on some form of neural network. We use the same data-set from ASHRAE for evaluating the performance of the tree based approaches in the DR-Advisor tool. This allows us to validate the performance of our algorithms and benchmark them against other data-driven methods.

#### ASHRAE Data Description

The training data is a time record of hourly chilled water, hot water and whole building electricity usage for a four-month period in an institutional building. Weather data and a time stamp were also included. The hourly values of usage of these three energy forms was to be predicted for the two following months. The testing set consisted of the two months following the four-month training period. The training data had approximately 3000 samples taken hourly during Sep - Dec 1989. The following information was provided for each time step:

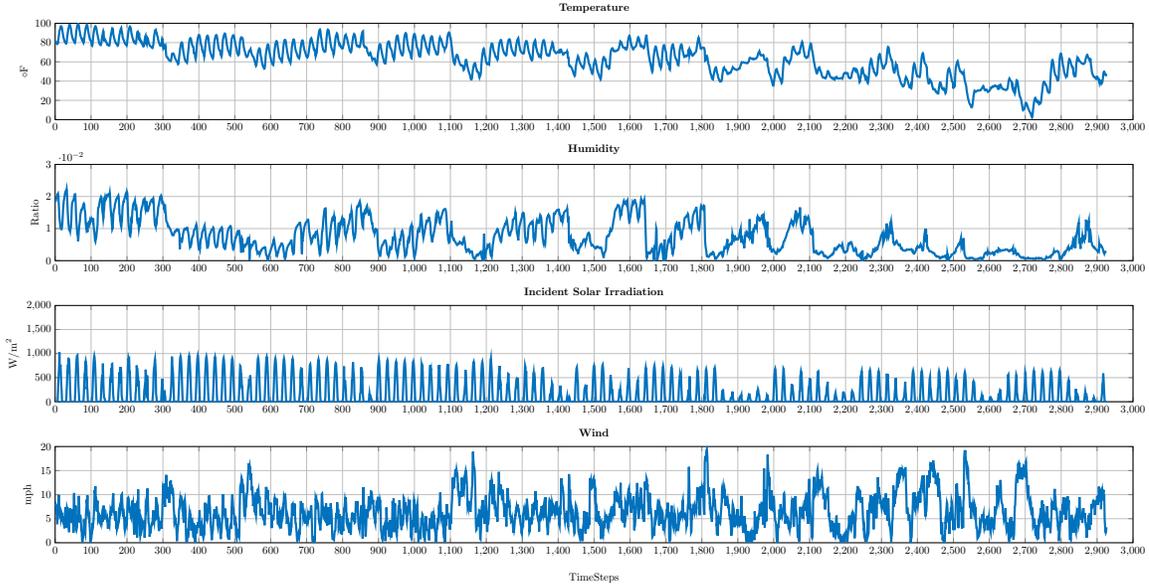


Figure 5.21: Training data for the ASHRAE Great Energy Predictor Shootout Challenge.

1. Outside temperature ( $^{\circ}\text{F}$ )
2. Wind speed (mph)
3. Humidity ratio (water/dry air)
4. Solar flux ( $\text{W}/\text{m}^2$ )
5. Hour of Day
6. Whole building electricity, WBE (kWh/hr)
7. Whole building chilled water, CHW (millions of Btu/hr)
8. Whole building hot water, HW (millions of Btu/hr)

The corresponding date is also provided. The training data features or the independent variables are shown in Figure 5.21. One of the dependent or response variable (WBE: Whole Building Electricity) is plotted in Figure 5.22.

In addition to the variables which are provided, the date for each sample point allows us to define *proxy* variables of our own. We define three such proxy variables:

1. Day of Week: This is a number which takes values from 1 to 7 based on what day of the week it is. The intuition behind introducing this variable is that any repeated building use patterns from a specific day will be captured by this variable.
2. IsWeekend: This is a boolean variable which takes the value **TRUE** for Saturdays and Sundays and **FALSE** otherwise.

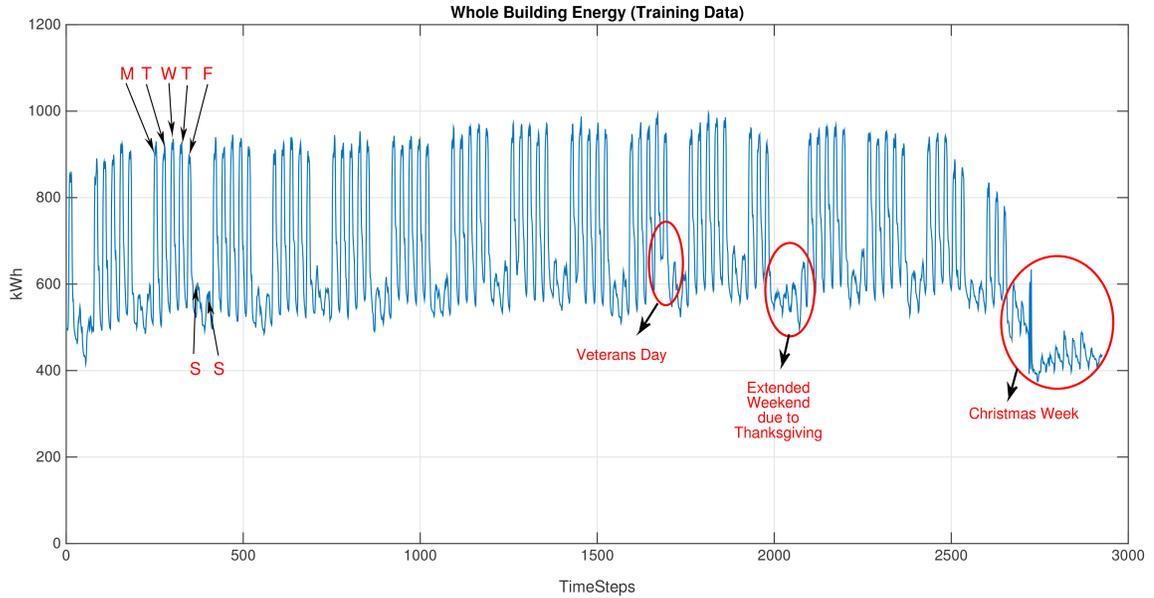


Figure 5.22: Plot of the whole building electricity (WBE) training data. It can be seen that special days and weekends can be easily identified in the data.

3. **IsHoliday:** Sometimes the building might follow a modified schedule on a weekday due to a holiday or a special day. By referring to the 1989 calendar or the training data duration, we are able to identify days in the training and testing data which are special days or holidays.

The motivation of introducing the proxy variables can be understood from Figure 5.22. One can note how weekday consumption patterns appear similar, except on certain holidays where the power consumption is irregular or seems to follow a weekend schedule (e.g. Thanksgiving). Likewise, the building has an irregular power consumption profile during the entire Christmas week, probably because of a vacation schedule.

### ASHRAE Predictor Shootout Results

We use the different ensembles methods in the DR-Advisor tool on the training data to learn three different models, one each for predicting the whole building energy WBE, chilled water consumption, CHW and hot water consumption HW. The test data consists of 1282 samples of weather information and date from the first two months of 1990. Our objective is to predict the WBE, CHW and HW for these two months.

The comparison between the predicted WBE values and the ground truth are shown in Figure 5.24. The coefficient of variation is 11.72%. The re-substitution error (on the training data) is plotted for a different number of trees used by the random forest in Figure 5.23. In the actual competition, the winners were selected based on the accuracy of all predictions as measured by the coefficient of variation statistic CV. The

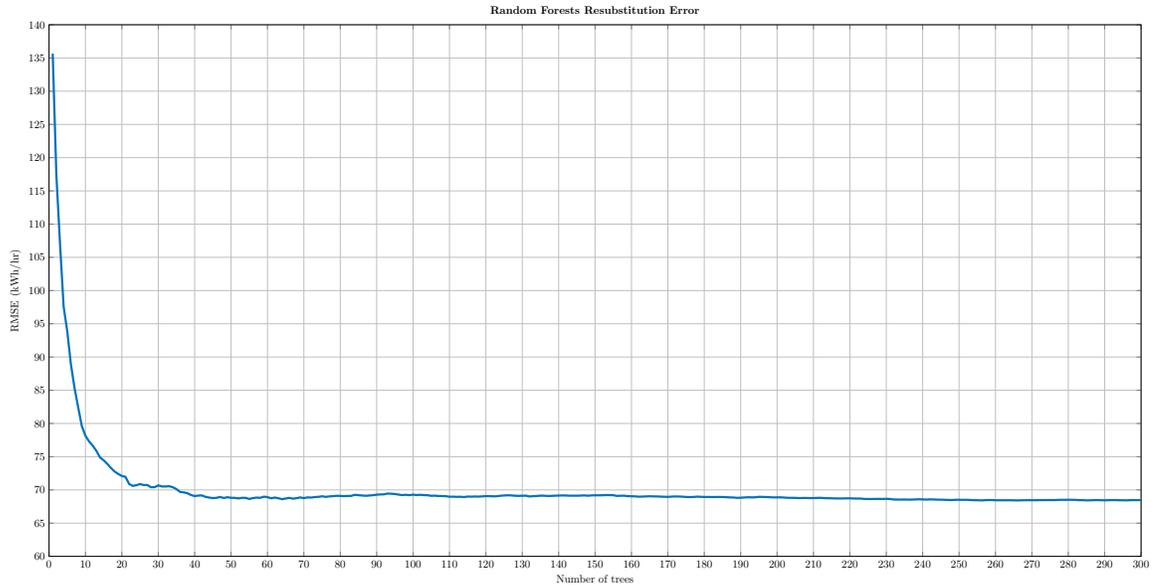


Figure 5.23: Resubstitution error is shown for the number of trees in the random forest method.

Table 5.3: ASHRAE Shootout Competition Results

ASHRAE Team ID	WBE CV	CHW CV	HW CV	Average CV
9	10.36	13.02	15.24	12.87
<b>DR-ADvisor:Random Forests</b>	<b>11.72</b>	<b>14.88</b>	<b>28.13</b>	<b>18.24</b>
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

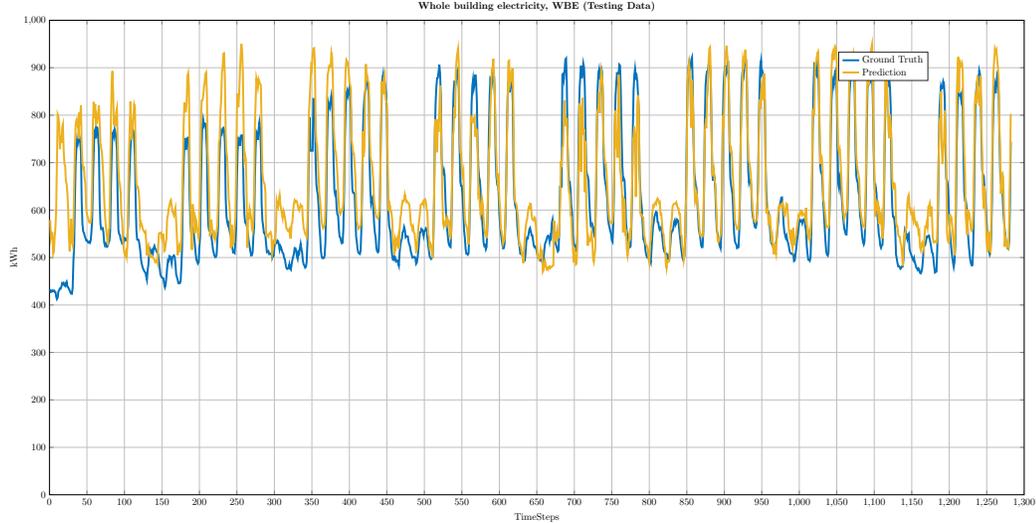


Figure 5.24: Comparison between predicted and ground truth values for WBE for the testing data.

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 5.3, we list the performance of the top 5 winners of the competition and compare our results with them. It can be seen from table 5.3, that the random forest implementation in the DR-Advisor tool ranks 2<sup>nd</sup> in terms of WBE CV and the overall average CV. The other algorithms in the table are (9) Mackays Bayesian Non-Linear Modeling, (6) Ohlssons Feedforward Multi-layer Perceptron, (2) Feustons Neural Network with Pre and Post Processing. Algorithm (9) (Mackay) generates the best results overall, beating many other neural network implementations in the competition, which suggests that it is some particular feature of this implementation, e.g., input preprocessing, network architecture, or training approach, that is important.

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

### 5.6.4 DR Strategy Evaluation

DR strategy evaluation involves choosing good DR strategies from several fixed strategies, in real time (Section 5.1.2). The challenge is to predict the power consumption profile of the building in real-time due to several policies. DR-Advisor then chooses the best control action and repeats this criteria throughout the event. We evaluate the accuracy of the power consumption profile prediction due to a fixed demand response policy chosen by DR-Advisor among several fixed policies. Unlike DR baseline prediction estimates, here we need additional training data in addition to just

power consumption, weather, proxy variables and set-point schedules. Specifically, data about the state of the building is also required. For the DoE CRB building, in addition to the weather, proxy and set-point variables, we also train on previous zone temperature data (for 19 zones), chilled water supply temperature and boiler temperature. There are a total of 34 predictor variables used for training the different trees.

Upon receiving the notification of the DR event at 1600 hrs, there are several pre-determined strategies that can be executed by the facilities manager. For simplicity, we only consider strategies in which two set-points are changed, the zone air temperature set-point and the chilled water supply set-point. We test the performance of 3 different rule based strategies shown in Fig. 5.25. 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 [Siemens 2011]. 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 5.2.6) 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. 5.26 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 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.

### 5.6.5 DR Strategy Synthesis

We now evaluate the performance of the mbCRT (Section 5.3.1) 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

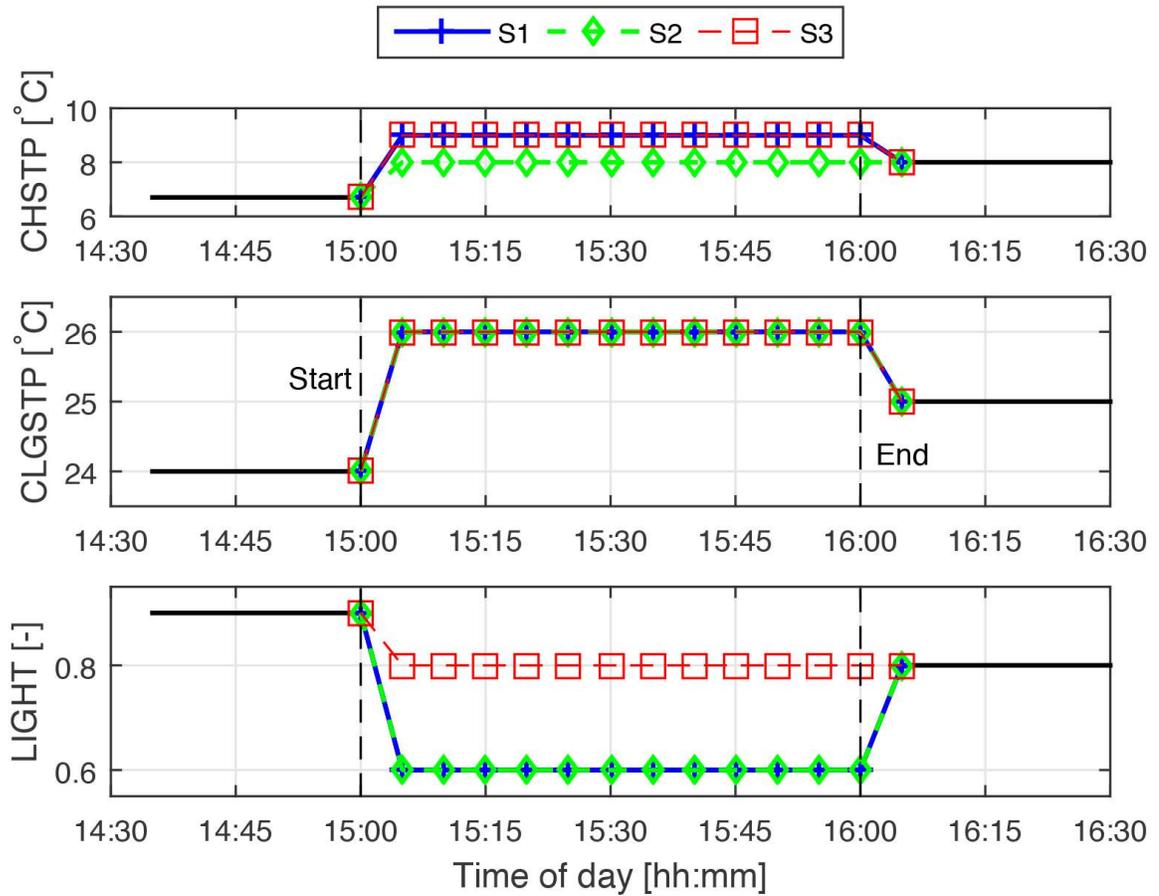


Figure 5.25: Rule-based strategies used in DR Evaluation. CHSTP denotes Chiller set point and CLGSTP denotes Zone Cooling temperature set point.

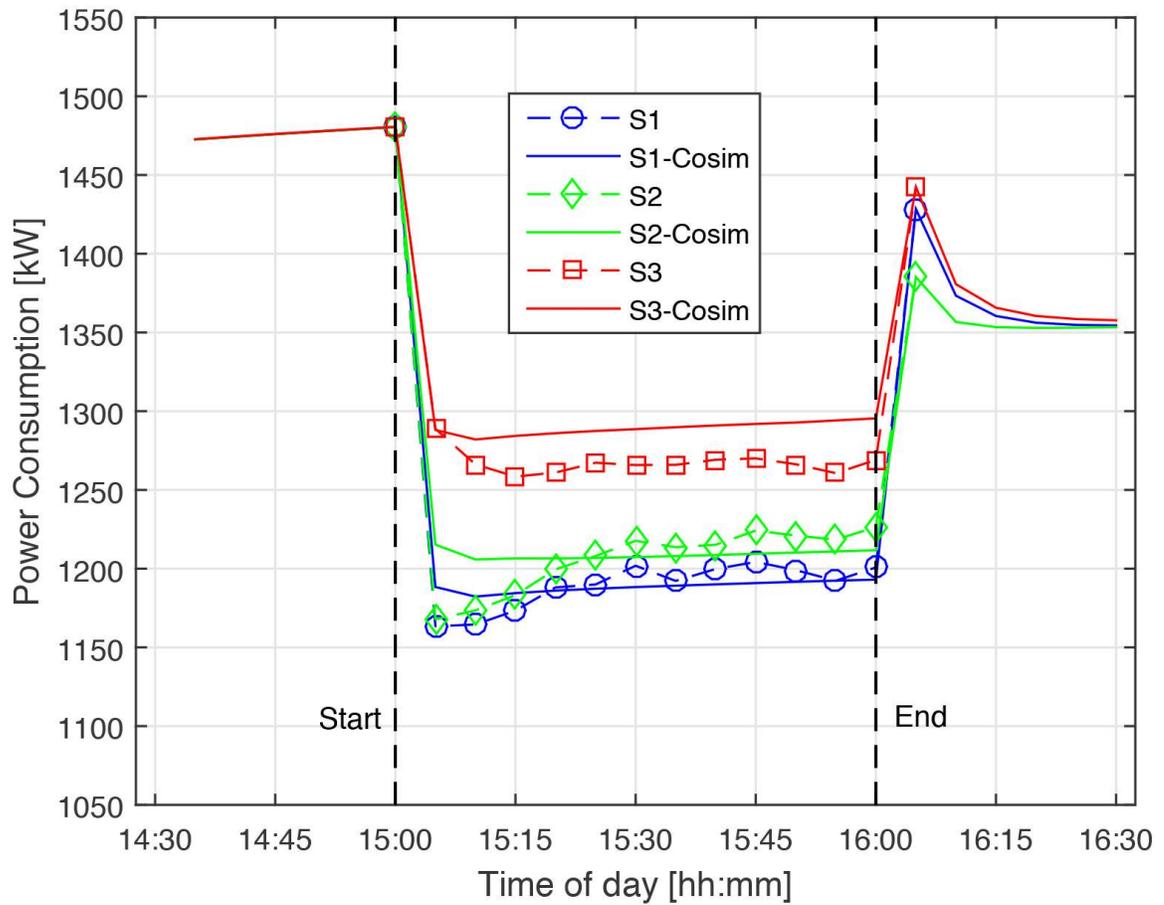


Figure 5.26: Prediction of power consumption for 3 strategies. DR Evaluation shows that Strategy 1 (S1) leads to maximum power curtailment.

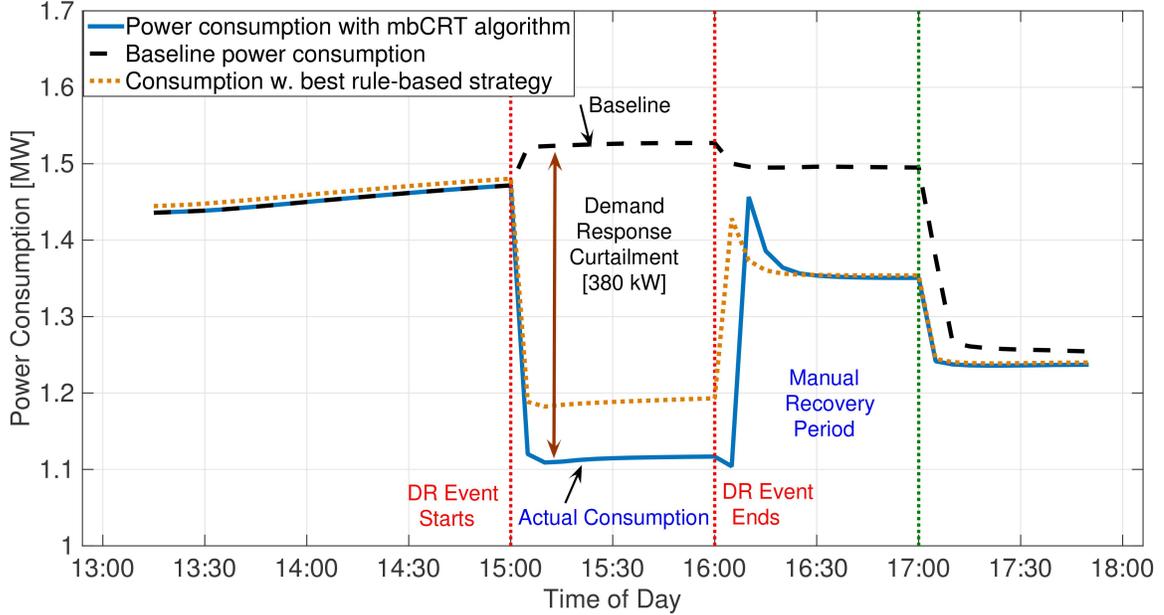


Figure 5.27: DR synthesis using the mbCRT algorithm for July 17, 2013. A curtailment of 380kW is sustained during the DR event period.

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 5.5) and determine the optimal values of the control inputs to meet a sustained response while maintaining thermal comfort.

Figure 5.27 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 5.6.4. In this case the DR strategy synthesis outperforms the best rule base strategy (from Section 5.6.4, Fig. 5.26) 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 5.28. We can see in Figure 5.29 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 to relying on fixed rules and pre-determined strategies which do not account for any guarantees on thermal comfort. Figure 5.30 shows a close of view of the curtailed response. The leaf node which is being used for the power

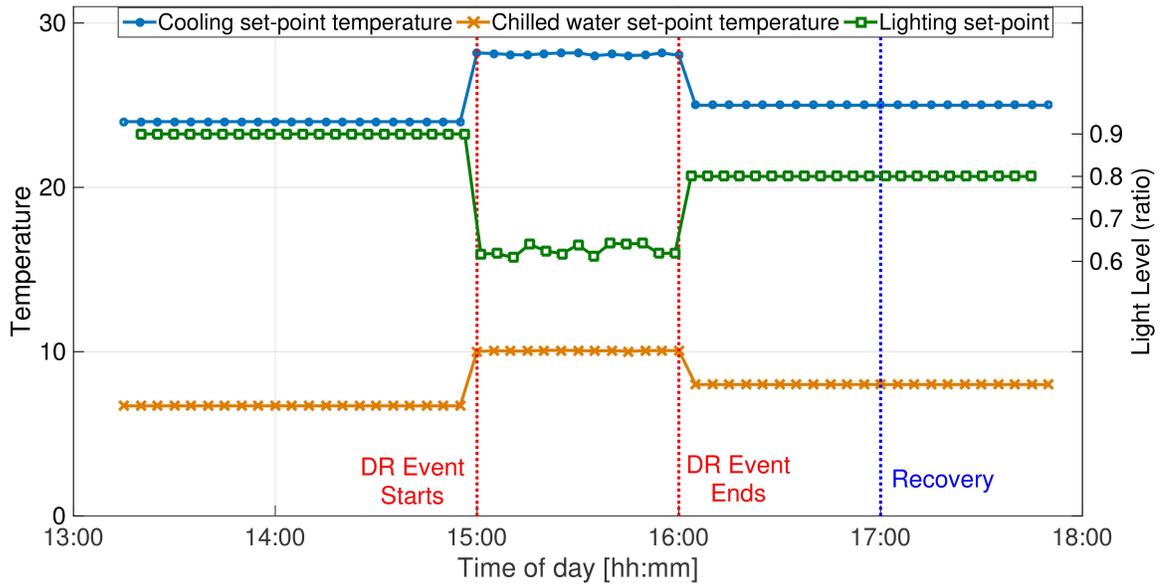


Figure 5.28: Optimal DR strategy as determined by the mbCRT algorithm.

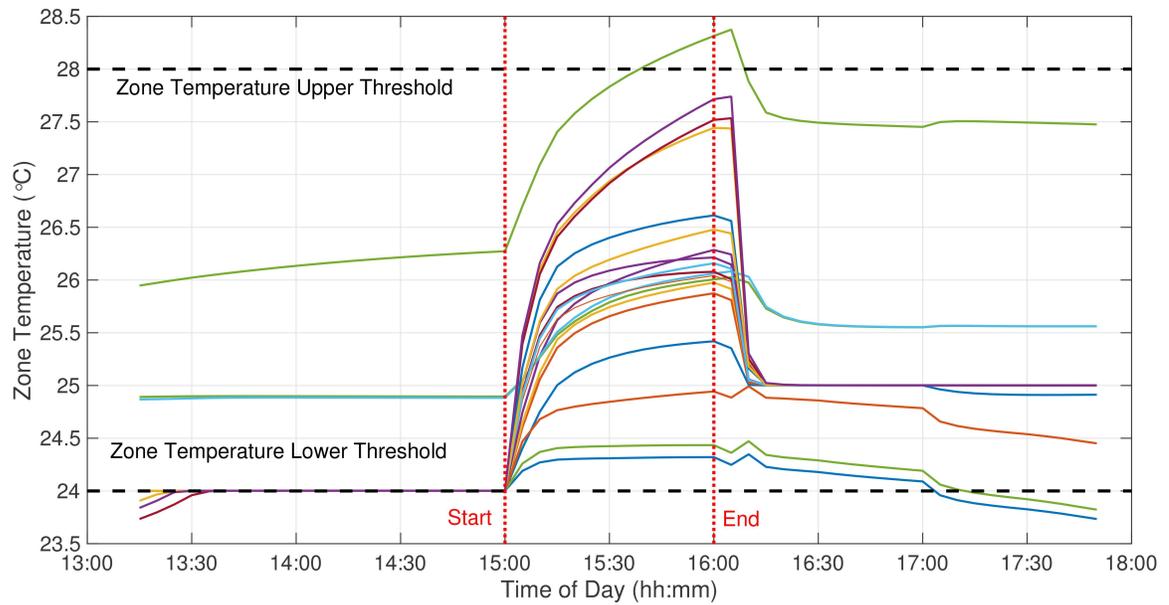


Figure 5.29: The mbCRT algorithm maintains the zone temperatures within the specified comfort bounds during the DR event.

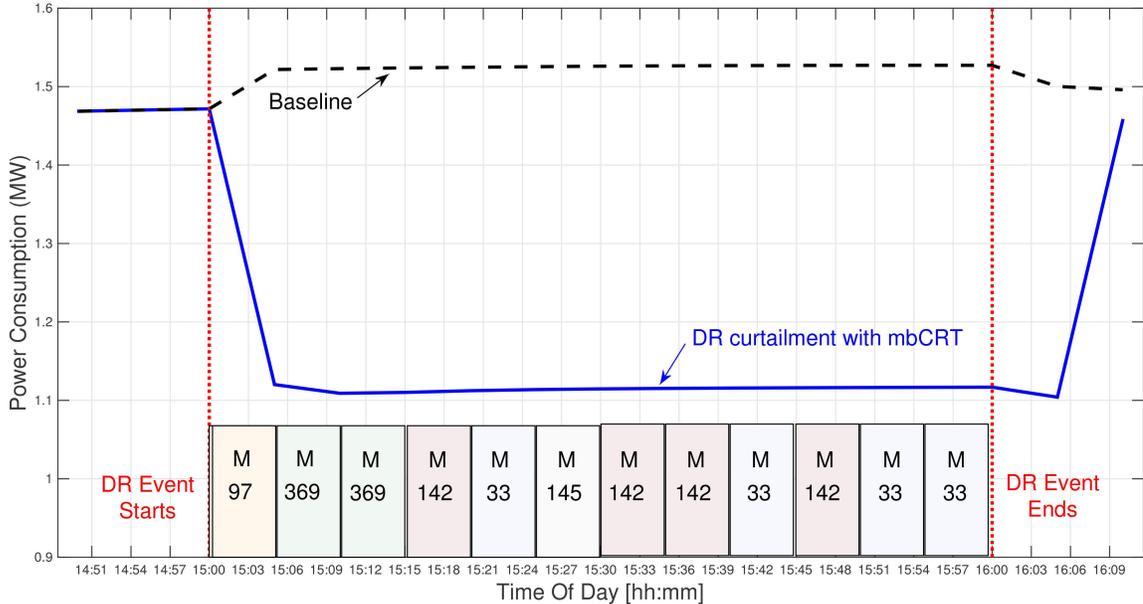


Figure 5.30: Zoomed in view of the DR synthesis showing how the mbCRT algorithm selects the appropriate linear model for each time-step based on the forecast of the disturbances.

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.

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.

## Revenue from Demand Response

We use Con Edison utility company’s commercial demand response tariff structure [Con Edison] 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.

### 5.6.6 Choosing the best tree

One can notice from the results of prediction accuracy from and that the tree which works well for one DR challenge might not perform the same for another DR challenge.

This is due to the dependency of the tree on the underlying data-set. Furthermore, in all the results presented hitherto, we always had the ground truth power consumption data available from the virtual test-bed. In an actual DR event, each of the trees will provide a power consumption prediction and then the question becomes which *tree's prediction is reliable during the event* ?

The question of which regression tree to choose for making power consumption predictions is an open question. Several heuristics are available at one's disposal to answer this question. A simple method is to choose the tree which has consistently performed well over several data-sets. However, at times, even the most accurate historical tree could make mistakes, which is a very undesirable effect. Upon closer observation we find that for each of the results, there are always at least 2 or 3 trees with similar prediction values or consensual predictions. This behavior can also be incorporated while making the predictions. If two or more methods predict similar responses their response could be used for the prediction. Lastly, one also can combine the predictions from the different trees into one single prediction. We can consider the prediction from each tree based method as a prediction from an expert. We then seek to build a *mixture of experts* to provide a prediction and confidence intervals for that prediction. The historical consistency of individual tree methods could be used to weight the expert advices.

## 5.7 Related work

There is a vast amount of literature ([Auslander et al., Oldewurtel et al. 2013, Xu et al. 2004, Van Staden et al. 2011]) 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 [Oldewurtel et al. 2013]; in which model predictive control is used for DR based on a grey-box model of a building. [Auslander et al.] uses a high-fidelity physics based model of the building to solve a problem similar to the DR evaluation problem. [Van Staden et al. 2011] 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 [Sturzenegger et al. 2015]. 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 fore-

casts 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 [Žáčková et al. 2014]. 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 [R et al. 2012].

There is recent work, which has explored aspects of modeling, implementation and implications of demand response buildings [Kialashaki and Reisel 2013a, Muratori et al. 2013, Dupont et al. 2014, Bartusch and Alvehag 2014], however, their focus has mainly been on the residential sector. [Dupont et al. 2014] 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 [Bartusch and Alvehag 2014] 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 [Muratori et al. 2013]. 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 5.6.3. In [Edwards et al. 2012], 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. [Wytock and Kolter 2013] and [Kolter and Ferreira Jr 2011] describes a graphical method for residential energy forecasting. [Kialashaki and Reisel 2013b] 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 [Vaghefi et al. 2014], while a statistical analysis of the impact of weather on peak electricity demand using actual meteorological data is presented in [Hong et al. 2013]. [Kolter and Jaakkola 2012] presents a method of energy disaggregation for residential homes using data-driven methods. In [Yin et al. 2012] 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.

## 5.8 Concluding remarks

We present a data-driven approach for modeling and control of large scale cyber-physical 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  $2^{nd}$  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  $\sim$  **\$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 cyber-physical energy systems.

# Chapter 6

## Energy Analytics

Intuition usually fails in higher dimensions. When the data has a lot of features or when we use ensemble methods with several hundred trees (as is the case with random forest), it leads to a loss of interpretability. By assigning additional attributes to the regression tree construction, we can restore the interpretability of ensemble methods like random forests and boosted regression trees and create interactive filters which allows the facilities manager to obtain insights into building use and electricity consumption. In addition to using regression trees to solve challenges associated with DR, we have also designed the capabilities of using regression trees for energy analytics.

Facility managers are demanding more control over their own energy consumption. They prefer insights into performance and usage across their portfolios to make more effective decisions. Implementing energy analytics with regression trees allows one to get near real-time, detailed information about the energy usage. It helps determine if the system is operating efficiently and lets the facilities manager investigate areas for improvement and evaluate energy efficiency upgrades.

### 6.1 Filter attributes at the leaves

The key to implementing energy analytics using regression trees is in defining certain attributes for the trees which can be later used to answer energy analytics related queries to the system. These attributes help convert the regression tree model into a knowledge database which can be mined for insights and recommendations using a query-response systems.

We define the following searchable attributes at the leaf nodes of every regression tree.

1. Prediction
2. Support
3. Expected error or confidence intervals

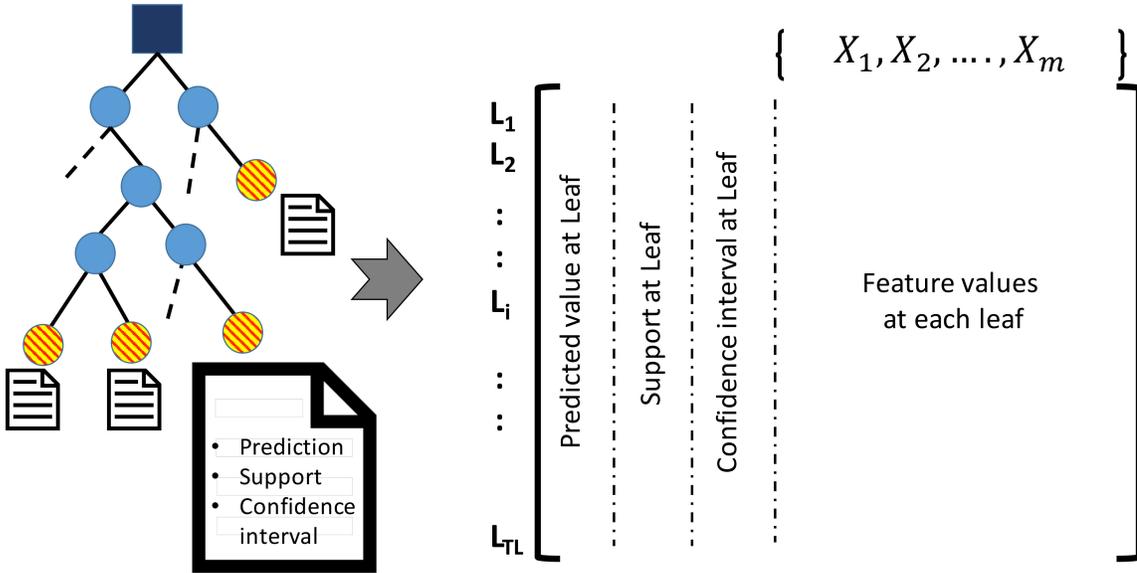


Figure 6.1: The data-structure defined represents the regression tree as a knowledge based which is searchable over the leaf attributes.

**Definition 6.1.** *The prediction value at a leaf node of a tree is the absolute value of the response variable which will be predicted model response  $\hat{Y}$  if the particular leaf node is used for prediction.*

**Definition 6.2.** *The support value at leaf node of each regression tree is the ratio of the number of samples which contribute to the leaf node to that of the total number of samples at the root node of the tree.*

In other words, the support of a leaf node is a measure of how frequently does the predicted response fall in to the data partition cell defined by that leaf node.

**Definition 6.3.** *The confidence interval at a leaf node is the 95% interval around the predicted value of the response variable  $\hat{Y}$  at that leaf node.*

These attributes can be easily calculated after the tree is built by using the values of the sample data points which fall into the leaf node of each tree.

Each attribute specifies a control knob for a filter and changing the value of the attribute determines which rules and branches of the tree are used to answer the high level query in the tree. E.g. we can only look for those leaves of the tree in which the prediction value is 700 kW. This would be equivalent to answering the following query: *under what conditions does the building consume 700 kW ?*

Filtering by support is simply filtering the leaves of the tree using percentage of training data that a given branch received. This filter is useful for identifying branches that were well supported by training data.

Finally, filtering by expected error or confidence intervals is a measure of certainty for the given prediction on a given branch. This filter is useful for identifying

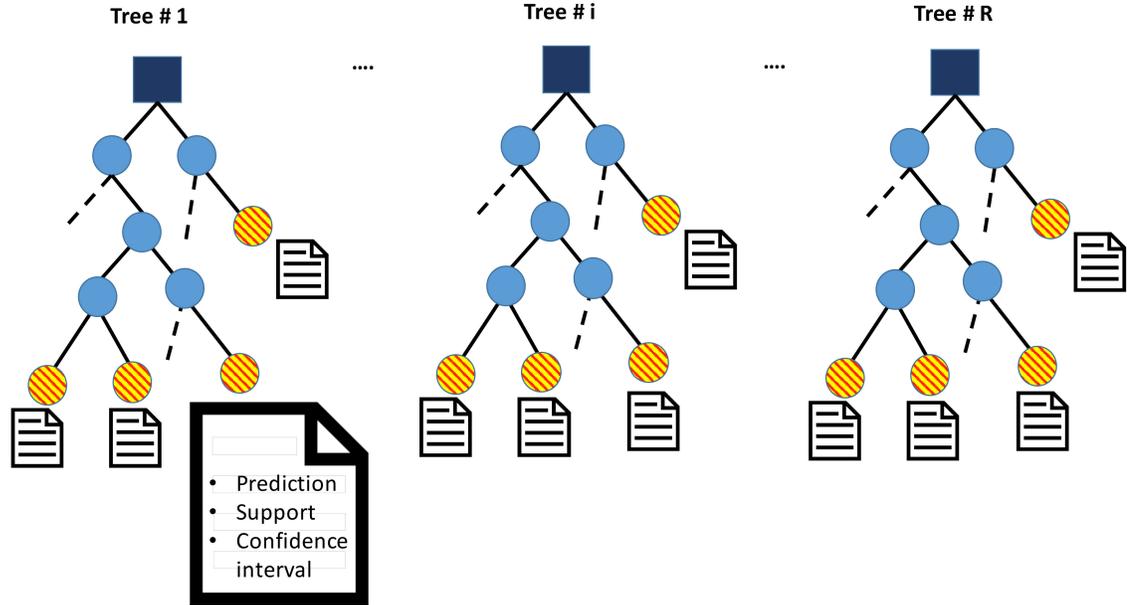


Figure 6.2: Attributes defined at each leaf node of the tree in a random forest.

branches that consistently contain certain outcomes, which in turn lead to predictions that are very likely given the training data.

*Rare branches:* The Rare branch filter can select all the branches of the tree that have very less support, while having a tighter confidence interval. These "rare" branches do not have as much support, but have high confidence. These might be interesting patterns of behavior in the data that are worth exploring. In the case of building models, like those built by DR-Advisor, these might correspond to sensor faults or anomalous power consumption behavior. Likewise, the *Frequent Branches* button will filter for branches that have support above a certain threshold and denote the most prevalent state of the building's electricity consumption.

As shown in Figure 6.1, for each leaf of the tree, the tree attributes are calculated and stored in a matrix, such that each row of the matrix corresponds to each leaf of the tree ( $L_1, L_2, \dots, L_{TL}$ ), where  $TL$  is the total number of leaf nodes in the tree. For each row of the matrix, the first three columns correspond to the values of the three attributes at that leaf node. The remaining columns of the matrix store the values of the data samples of the features  $[X_1, X_2, \dots, X_m]$  which lie in the data partition cell defined by the leaf node. The attributes of the leaf node are also calculated based on the values of the data samples which contribute to the leaf node.

In the case of an ensemble method such as random forest, the three attributes are defined for every leaf node for each tree in the forest (Figure 6.2). Corresponding to each tree the data-structure is built as explained above.

Once the data-structure is built, the query posed by the facilities manager is decomposed into a search over the three attributes and the average response is computed. This is explained with the help of an example in the next section.

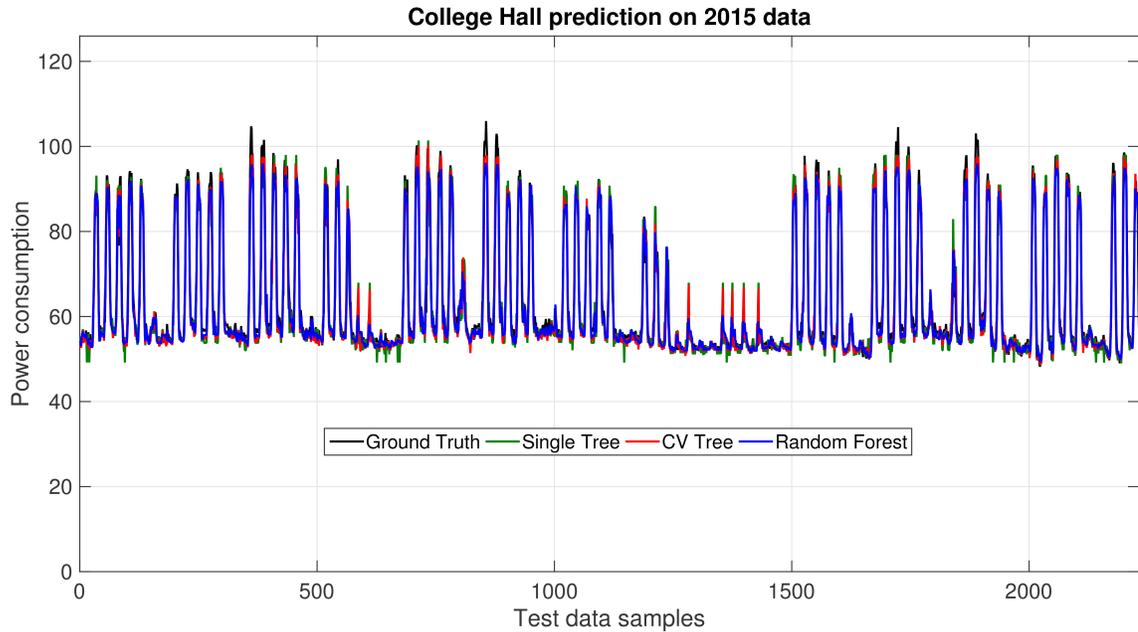


Figure 6.3: Comparison between predicted and actual power consumption of the building over the test-set.

## 6.2 Query-response case study with real data

The building under consideration is College Hall, a large administrative building comprising mainly of office spaces. The building has 6 floors and a total floor area of 110,266 sq ft.

First, a suite of regression tree algorithms are built using hourly historical data, which consists of the following features:

1. Month
2. Day of Month
3. Wet bulb temperature °C
4. Dry bulb temperature °C
5. Humidity %
6. Wind Speed m/s
7. Wind gusts m/s
8. Wind direction
9. Time of Day
10. Solar Irradiation W/m

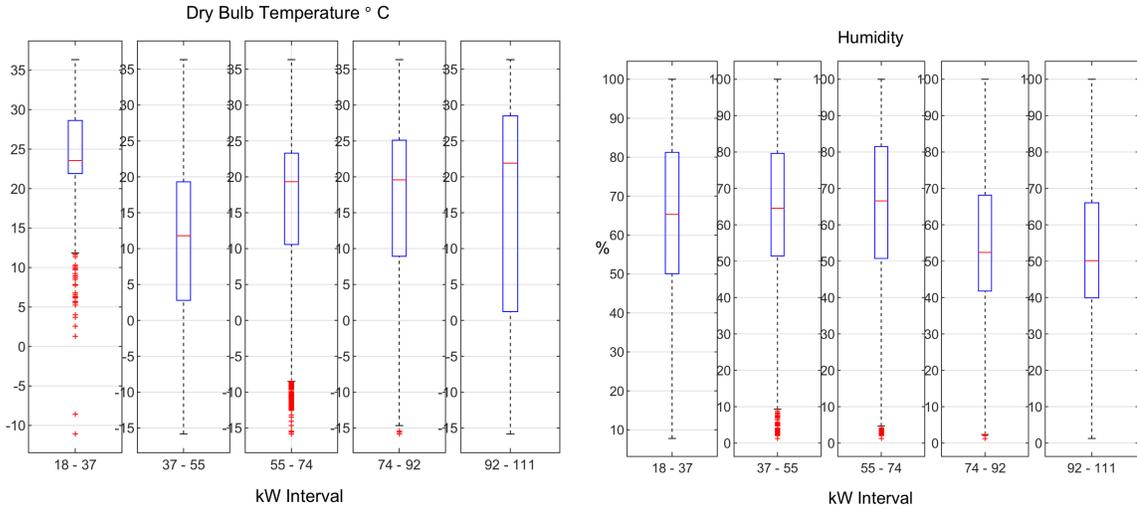


Figure 6.4: Example plot of how the values of the dry bulb temperature and humidity varies across different power values for the entire random forest.

Table 6.1: Query-response for College Hall

Dry Bulb Temperature 22.6°C	Wet Bulb Temperature 6.4°C	Humidity 50.2%
Wind Speed 085m/s	Wind Gusts 4.72m/s	Solar Irradiation 552W/m
HDD 1.8	Wind Direction 36°	CDD 0.6
Either a Tuesday or Thursday	Time between 1300-1600 hrs	During July

11. Heating degree days
12. Cooling degree days
13. Day of week

Figure 6.3 shows the comparison between the predicted and the actual power consumption value for the test-set data form 2015 using different regression tree algorithms, including an ensemble method , i.e. random forest.

Using the attributes defined in Section 6.1, we build the data structure over the random forest model for this building. The random forest consists of 500 trees, with a total of 334,230 leaf nodes in the forest spread across the different trees.

In Figure 6.4, we show how the prediction attribute varies across the random forest model for different values of dry bulb temperature and humidity levels. The box plots of Figure 6.4 can be interpreted as follows.

## 6.2.1 Example queries

Consider the following query by the facilities manager: *Under what conditions College Hall consume more that 90kW ?*

Using data from Figure 6.4, one can see that the average value of the dry bulb temperature across all the leaf nodes for which the predicted response was greater than

90kW is 22.6°C, and average humidity levels are 50.2%. A similar search can be done for all the other features which belong to the leaf node with predicted power output greater than 90kW. The complete response is tabulated in Table 6.1. Note that for categorical variables such as time of day, day of week etc., instead of averaging the value of the variable across all the appropriate leaf nodes, a majority poll is conducted to obtain the average response.

The following are some more examples of the kind of queries supported by the energy-analytics system:

- *What is the leading cause of peak power consumption of the building compared to the baseline ?*
- *Are there any anomalous building electricity consumption patterns ?*
- *Which buildings on campus consistently consume the most power ?*
- *At what time is the peak expected to occur for Building A ?*
- *What will be the effect of changing set-point  $S$  in building A ?*

# Chapter 7

## Conclusion

### 7.1 Conclusion

We presented the problem of learning accurate dynamical models of large scale buildings for use in model-based control design and synthesis. A major challenge with such large scale systems is in accurately modeling the dynamics of the underlying physical system. We show how learning white-box and grey-box models of a building is cost and time prohibitive making it something that is primarily done by researchers and for large projects but is not something which could widely adopted by the majority commercial building stock. It is the main hurdle towards adopting model-based control design approaches for cyber-physical energy systems.

In order for making model capture easy and efficient for cyber-physical energy systems we need to reduce the need for and the associated cost for sensor retrofits required for estimating the parameters of first principle based models and use data-driven methods for building control-oriented models.

In order to reduce the additional sensor cost associated with learning grey-box models of buildings, we first acknowledge that uncertainty in sensor data affects all aspects of building performance: from the identification of models, through the implementation of model-based control, to the operation of the deployed systems. Learning models of buildings from sensor data has a fundamental property that the model can only be as accurate and reliable as the data on which it was trained. We developed Model-IQ, a methodology for analysis of uncertainty propagation for building inverse modeling and controls. Given a grey-box model structure and real input data from a temporary set of sensors, Model-IQ evaluates the effect of the uncertainty propagation from sensor data to model accuracy and to closed-loop control performance. We also developed a statistical method to quantify the bias in the sensor measurement and to determine near optimal sensor placement and density for accurate data collection for model training and control. We considered the end-to-end propagation of uncertainty both in the form of fixed bias and random bias in the sensor data. Using a real building test-bed, we show how performing an uncertainty analysis can reveal

trends about inverse model accuracy and control performance, which can be used to make informed decisions about sensor requirements and data accuracy.

We also considered the problem of data-driven end-user demand response for large buildings which involves predicting the demand response baseline, evaluating fixed rule based demand response strategies and synthesizing demand response control actions. Demand response 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 cost and time prohibitive to build. 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. Using historical data from the building, we build a family of regression trees and learn data-driven models for predicting the power consumption of the building in real-time. We developed 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 for a large DoE commercial reference building and leads to a significant amount of load curtailment (of 380kW) and over \$45,000 in savings which is **37.9%** of the summer energy bill for the building. 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. While regression trees are a popular choice for prediction, this is the first time they are used in the context of demand response. This is also the first time regression tree based methods are used for controller synthesis. The performance of DR-Advisor was evaluated for 8 buildings on Penn’s campus; and it achieves 92.8% to 98.9% prediction accuracy. We compare DR-Advisor with other data driven methods and rank 2<sup>nd</sup> on ASHRAE’s benchmarking data-set for energy prediction. 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 cyber-physical energy systems.

## 7.2 Future work

- As covered in Chapter 5, DR-Advisor has been developed into a MATLAB based toolbox which is capable of implementing all the data-driven algorithms for DR as described in this dissertation. We plan to continue developing the DR-Advisor as a software. Our goal is to further develop DR-Advisor into a plug-and-play software which plugs into virtually any building management system in new or existing buildings, interfaces with utility Automated Demand

Response programs via OpenADR protocols and does not require any system upgrades or new sensors.

- So far, the focus of the data-driven demand response effort has been on the end-user DR challenges. DR participation from these electricity customers can be unreliable and unpredictable unless the right tools, processes, economic incentives, and training are in place. In the future, the DR-Advisor research will be extended to account for the electricity supply side as well. DR-Advisor can automatically inform utilities of the future load profiles of buildings enrolled in DR programs allowing utilities to better plan grid operations before a critical peak event occurs. Specifically, the load aggregator or curtailment service providers could use DR-Advisor to solve the problem of optimal price bidding and DR dispatch under a voluntary DR contract with the end-user. In this setting, the aggregator tries to maximize its own profit while offering the best possible price incentive to the end-user for load curtailment.
- The model-based control with regression trees (mbCRT) algorithm, is a promising method for applying data-driven control synthesis to large scale systems, even beyond energy systems. Doing control synthesis with interpretable regression trees model is novel and our future work involves investigating applications of such methods to other cyber-physical systems.

# Bibliography

- Energy price risk and the sustainability of demand side supply chains. *Applied Energy*, 123(0):327 – 334, 2014. ISSN 0306-2619.
- FUNIP ASHRAE. Fundamentals handbook. *IP Edition*, 2009.
- David Auslander, Domenico Caramagno, David Culler, Tyler Jones, Andrew Krioukov, Michael Sankur, Jay Taneja, Jason Trager, Sila Kiliccote, Rongxin Yin, et al. Deep demand response: The case study of the citris building at the university of california-berkeley.
- Cajsa Bartusch and Karin Alvehag. Further exploring the potential of residential demand response programs in electricity distribution. *Applied Energy*, 125(0):39 – 59, 2014. ISSN 0306-2619. doi: <http://dx.doi.org/10.1016/j.apenergy.2014.03.054>.
- Madhur Behl and Rahul Mangharam. A data-driven demand response recommender system. In *Submitted to the the journal of Applied Energy*, 2015a.
- Madhur Behl and Rahul Mangharam. Sometimes, money does grow on trees: Real-time demand response with dr-advisor. In *2nd ACM International Conference on Embedded Systems For Energy-Efficient Built Environments (BuilSys)*, Seoul, South Korea, 2015b.
- Madhur Behl and Rahul Mangharam. Sometimes, money does grow on trees: Dr-advisor, a data driven demand response recommender system. In *Semiconductor Research Corporation (SRC) TECHCON*, Austin, Texas, USA, 2015c. [Best in session award].
- Madhur Behl, Truong Nghiem, and Rahul Mangharam. IMpACT: Inverse model accuracy and control performance toolbox for buildings. 2014a.
- Madhur Behl, Truong Nghiem, and Rahul Mangharam. Model-IQ: Uncertainty propagation from sensing to modeling and control in buildings. *ACM/IEEE International Conference on Cyber-Physical Systems*, 2014b.

- Madhur Behl, Truong Nghiem, and Rahul Mangharam. Dr-advisor: A data driven demand response recommender system. In *CISBAT International Conference: future buildings and districts sustainability from nano to urban scale*, EPFL, Lausanne, Switzerland, 2015.
- Madhur Behl, Achin Jain, and Rahul Mangharam. Data-driven modeling, control and tools for cyber-physical energy systems. In *Submitted to the International Conference on Cyber-Physical Systems (ICCPs)*, 2016.
- S. Bengea, V. Adetola, K. Kang, M. J. Liba, D. Vrabie, R. Bitmead, and S. Narayanan. Parameter estimation of a building system model and impact of estimation error on closed-loop performance. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5137–5143, 2011.
- James E. Braun and Nitin Chaturvedi. An inverse gray-box model for transient building load prediction. *HVAC and R Research*, 8(1):73–99, 2002. doi: 10.1080/10789669.2002.10391290.
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.
- Rolf Butters, Alan Schroeder, George Hernandez, Peter Fuhr, Timothy McIntyre, and Wayne Manges. US Department of Energy Advanced Sensing and Controls for Energy Efficient Buildings: A Cross Cutting Project. In *ACEEE Summer Study on Energy Efficiency in Industry*, 2011.
- José A Candanedo, Vahid R Dehkordi, and Phylroy Lopez. A control-oriented simplified building modelling strategy. In *13th Conf. Intl. Building Perf. Simulation Ass.*, 2013.
- Serguei Chatrchyan, Vardan Khachatryan, Albert M Sirunyan, A Tumasyan, W Adam, E Aguilo, T Bergauer, M Dragicevic, J Erö, C Fabjan, et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B*, 716(1):30–61, 2012.
- Thomas F Coleman and Yuying Li. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM Journal on optimization*, 6(2):418–445, 1996.
- Federal Energy Regulatory Commission et al. Assessment of demand response and advanced metering. 2012.
- Con Edison. Demand response programs details. URL <http://www.coned.com/energyefficiency/pdf/proposed-program-changes.pdf>.

- McGraw-Hill Construction. *Energy efficiency trends in residential and commercial buildings*. US Department of Energy, Energy Efficiency and Renewable Energy, 2010.
- D. B. Crawley, J. W. Hand, M. Kummert, and B. T. Griffith. Contrasting the Capabilities of Building Energy Performance Simulation Programs. *Building & Environment*, 43(4):661–673, 2008.
- Drury B Crawley, Linda K. Lawrie, et al. Energyplus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4):319 – 331, 2001.
- Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.
- M. Deru, K. Field, D. Studer, et al. U.s. department of energy commercial reference building models of the national building stock. 2010.
- T. Dewson, B. Day, and A.D. Irving. Least Squares Parameter Estimation of a Reduced Order Thermal Model of an Experimental Building. *Building and Environment*, 28(2):127–137, 1993.
- B. Dupont, K. Dietrich, C. De Jonghe, A. Ramos, and R. Belmans. Impact of residential demand response on power system operation: A belgian case study. *Applied Energy*, 122(0):1 – 10, 2014. ISSN 0306-2619. doi: <http://dx.doi.org/10.1016/j.apenergy.2014.02.022>. URL <http://www.sciencedirect.com/science/article/pii/S0306261914001585>.
- Richard E Edwards, Joshua New, and Lynne E Parker. Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings*, 49:591–603, 2012.
- Bryan Eisenhower, Zheng O’Neill, Vladimir A Fonoberov, and Igor Mezić. Uncertainty and sensitivity decomposition of building energy models. *J. Building Perf. Simulation*, 5:171–184, 2012.
- Jane Elith, John R Leathwick, and Trevor Hastie. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813, 2008.
- David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3): 59–79, 2010.
- Jerome H. Friedman. A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers*, (4):404–408, 1977.

- Jerome H Friedman. A tree-structured approach to nonparametric multiple regression. In *Smoothing techniques for curve estimation*, pages 5–22. Springer, 1979.
- Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991.
- MW Gillo. Maid, a honeywell 600 program for an automatized survey analysis, 1972.
- Christophe Giraud-Carrier. Beyond predictive accuracy: what? *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, pages 78–85, 1998.
- Charles Goldman. Coordination of energy efficiency and demand response. *Lawrence Berkeley National Laboratory*, 2010.
- D Gyalistras, A Fischlin, M Morari, CN Jones, F Oldewurtel, A Parisio, F Ullmann, C Sagerschnig, and AG Gruner. Use of weather and occupancy forecasts for optimal building climate control. Technical report, Technical report, ETH Zürich, 2010.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.
- Tianzhen Hong, Wen-Kuei Chang, and Hung-Wen Lin. A fresh look at weather impact on peak electricity demand and energy use of buildings using 30-year actual weather data. *Applied Energy*, 111:333–350, 2013.
- EBC:Energy in Buildings and Communities Programme. Ebc annex 53 total energy use in buildings: Analysis & evaluation methods, 2013.
- PJM Interconnection. 2014 demand response operations markets activity report. 2014.
- Srinivas Katipamula et al. Small-and Medium-Sized Commercial Building Monitoring and Controls Needs: A Scoping Study. Technical report, Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2012.
- Arash Kialashaki and John 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):271 – 280, 2013a. ISSN 0306-2619. doi: <http://dx.doi.org/10.1016/j.apenergy.2013.03.034>. URL <http://www.sciencedirect.com/science/article/pii/S0306261913002304>.
- Arash Kialashaki and John R Reisel. Modeling of the energy demand of the residential sector in the united states using regression models and artificial neural networks. *Applied Energy*, 108:271–280, 2013b.

- D. Kim and J. E Braun. Reduced-order building modeling for application to model-based predictive control. *SimBuild, National Conference of IBPSA-USA*, 2012.
- SA Klein, WA Beckman, JW Mitchell, JA Duffie, NA Duffie, TL Freeman, JC Mitchell, JE Braun, BL Evans, JP Kummer, et al. Trnsys 16—a transient system simulation program, user manual. *Solar Energy Laboratory. Madison: University of Wisconsin-Madison*, 2004.
- J Zico Kolter and Joseph Ferreira Jr. A large-scale study on predicting and contextualizing building energy usage. 2011.
- J Zico Kolter and Tommi Jaakkola. Approximate inference in additive factorial hmms with application to energy disaggregation. In *International conference on artificial intelligence and statistics*, pages 1472–1482, 2012.
- Jan F Kreider and Jeff S Haberl. Predicting hourly building energy use: The great energy predictor shootout—overview and discussion of results. Technical report, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., Atlanta, GA (United States), 1994.
- Joseph C Lam and Sam Hui. Sensitivity analysis of energy performance of office buildings. *Building & Environment*, 31:27–39, 1996.
- Yudong Ma, Francesco Borrelli, Brandon Hancey, Brian Coffey, Sorin Bengea, and Philip Haves. Model Predictive Control for the Operation of Building Cooling Systems. *Control Systems Technology, IEEE Transactions on*, 20(3):796–803, 2012.
- David JC MacKay et al. Bayesian nonlinear modeling for the prediction competition. *ASHRAE transactions*, 100(2):1053–1062, 1994.
- Thomas L McKinley and Andrew G Alleyne. Identification of Building Model Parameters and Loads using On-site Data Logs. *Third National Conference of IBPSA-USA*, 2008.
- Jerry M Melillo, TC Richmond, and Gary W Yohe. Climate change impacts in the united states: the third national climate assessment. *US Global change research program*, 841, 2014.
- Robert Messenger and Lewis Mandell. A modal search technique for predictive nominal scale multivariate analysis. *Journal of the American statistical association*, 67(340):768–772, 1972.
- PJM Interconnection Michael J. Kormos. Pjm response to consumer reports on 2014 winter pricing. 2014.
- Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. *Numerical analysis*, pages 105–116, 1978.

- James N Morgan and John A Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 58(302):415–434, 1963.
- Matteo Muratori, Matthew C. Roberts, Ramteen Sioshansi, Vincenzo Marano, and Giorgio Rizzoni. A highly resolved modeling technique to simulate residential power demand. *Applied Energy*, 107(0):465 – 473, 2013. ISSN 0306-2619. doi: <http://dx.doi.org/10.1016/j.apenergy.2013.02.057>. URL <http://www.sciencedirect.com/science/article/pii/S030626191300175X>.
- NOAA. National centers for environmental information- state of the climate: Global analysis for august 2015, 2015. [published online September 2015, retrieved on October 15, 2015 from <http://www.ncdc.noaa.gov/sotc/global/201508>.].
- F. Oldewurtel, A. Parisio, C.N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth. Energy efficient building climate control using stochastic model predictive control and weather predictions. In *American Control Conference*, pages 5100–5105, 2010.
- Frauke Oldewurtel, David Sturzenegger, Goran Andersson, Manfred Morari, and Roy S Smith. Towards a standardized building assessment for demand response. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 7083–7088. IEEE, 2013.
- Samuel Prívarva, Jiří Cigler, Zdeněk Váňa, Frauke Oldewurtel, Carina Sagerschnig, and Eva Žáčková. Building modeling as a crucial part for building predictive control. *Energy and Buildings*, 2012.
- J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
- J. Ross Quinlan. Improved use of continuous attributes in c4. 5. *Journal of artificial intelligence research*, pages 77–90, 1996.
- J Ross Quinlan et al. *Discovering rules by induction from large collections of examples*. Expert systems in the micro electronic age. Edinburgh University Press, 1979.
- John R Quinlan et al. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, pages 343–348. Singapore, 1992.
- Joshua R, Jibonananda Sanyal, Mahabir Bhandari, and Som Shrestha. Autotune e+ building energy models. *Proceedings of the 5th National SimBuild of IBPSA-USA*, 2012.
- Navigant Research. Demand response for commercial & industrial markets market players and dynamics, key technologies, competitive overview, and global market forecasts. 2015.

- B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996. ISBN 0-521-46086-7.
- Siemens. Automated demand response using openadr: Application guide, 2011.
- Satish Narayanan Slaven Peles, Sunil Ahuja. Uncertainty quantification in energy efficient building performance simulations. In *International High Performance Buildings Conference*, 2012.
- John A Sonquist. Multivariate model building. 1970.
- John A Sonquist, Elizabeth L Baker, and James N Morgan. Searching for structure: an approach to analysis of substantial bodies of micro-data and documentation for a computer program. 1974.
- D. Sturzenegger, D. Gyalistras, M. Morari, and R.S. Smith. Model predictive climate control of a swiss office building: Implementation, results, and cost-benefit analysis. *Control Systems Technology, IEEE Transactions on*, PP(99):1–1, 2015. ISSN 1063-6536. doi: 10.1109/TCST.2015.2415411.
- US Department of Energy. EEB Hub: Energy efficient buildings hub, October 2013. URL <http://www.eebhub.org/>.
- A Vaghefi, MA Jafari, Emmanuel Bisse, Y Lu, and J Brouwer. Modeling and forecasting of cooling and electricity load demand. *Applied Energy*, 136:186–196, 2014.
- Adam Jacobus Van Staden, Jiangfeng Zhang, and Xiaohua Xia. A model predictive control strategy for load shifting in a water pumping scheme with maximum demand charges. *Applied Energy*, 88(12):4785–4794, 2011.
- Matt Wytock and Zico Kolter. Sparse gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1265–1273, 2013.
- Peng Xu, Philip Haves, Mary Ann Piette, and James Braun. Peak demand reduction from pre-cooling with zone temperature reset in an office building. *Lawrence Berkeley National Laboratory*, 2004.
- Wei Yin, Yogesh Simmhan, and Viktor K Prasanna. Scalable regression tree learning on hadoop using openplanet. In *Proceedings of third international workshop on MapReduce and its Applications Date*, pages 57–64. ACM, 2012.
- Eva Žáčková, Zdeněk Váňa, and Jiří Cigler. Towards the real-life implementation of mpc for an office building: Identification issues. *Applied Energy*, 135:53–62, 2014.