A Data-driven Approach for Coordinating Air Conditioning Units in Buildings during Demand Response Events

Xiangyu Zhang

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Electrical Engineering

Saifur Rahman

Manisa Pipattanasomporn

Robert Broadwater

Guoqiang Yu

Chang-Tien Lu

December 5, 2018

Arlington, Virginia

Keywords: Smart Grid, Demand Response, HVAC Coordination, Building Thermal Model, Reinforcement Learning

A Data-driven Approach for Coordinating Air Conditioning Units in Buildings during Demand Response Events

Xiangyu Zhang

Abstract

Among many smart grid technologies, demand response (DR) is gaining increasing popularity. Many utility companies provide a variety of programs to encourage DR participation. Under these circumstances, various building energy management (BEM) systems have emerged to facilitate the building control during a DR event. Nonetheless, due to the cost and return on investment, these solutions mainly target homes and large commercial buildings, leaving aside small- and medium-sized commercial buildings (SMCB). SMCB, however, accounts for 90% of commercial buildings in the US, and offer great potential of load reduction during peak hours.

With the advent of Internet-of-Things (IoT) devices and technologies, low cost smart building solutions have become possible for the SMCB; nonetheless, related intelligent algorithms are not widely available. This dissertation work investigates automated building control algorithms, tailored for the SMCB, to realize automatic device control during DR events. To be specific, a control framework for Air-Conditioning (AC) units' coordination is proposed. The goal of such framework is to reduce the aggregated AC power consumption while maintaining the thermal comfort inside a building during DR events.

To achieve this goal, three major components of the framework were studied: building thermal property modeling, AC power consumption modeling and control algorithms design. Firstly, to consider occupants' thermal comfort, a reverse thermal model was designed to predict the indoor temperature of thermal zones

under different AC control signals. The model was trained with supervised learning using coarse-grained temperature data recorded by smart thermostats; thus, it requires no lengthy configuration as a forward model does. The cost efficiency and plug-and-play feature of the model make it appropriate for SMCB. Secondly, a power disaggregation algorithm is proposed to model the power-outdoor temperature relationship of multiple AC units, using data from a single power meter and thermostats. Finally, algorithms based on mixed integer linear programming (MILP) and reinforcement learning (RL) were devised to coordinate multiple AC units in a building during a DR event. Integrated with the thermal model and AC power consumption model, these algorithms minimize occupants' thermal discomfort while restricting the aggregated AC power consumption below the DR limit. The efficiency of these control algorithms was tested, which demonstrate that they can generate AC control schedule in short notice (5 minutes) ahead of a DR event. Verification and validation of the proposed framework was conducted in both simulation and actual building environments. In addition, though the framework is designed for SMCBs, it can also be applied to large homes with multiple AC units to coordinate.

This work is expected to give an insight into the BEM sector, helping the popularization of implementing DR in buildings. The research findings from this dissertation work shows the validity of the proposed algorithms, which can be used in BEM systems and cloud-based smart thermostats to exploit the untapped DR resource in SMCB.

A Data-driven Approach for Coordinating Air Conditioning Units in Buildings during Demand Response Events

Xiangyu Zhang

General Audience Abstract

For power system operation, the demand and supply should be equal at all time. During peak hours, the demand becomes very high. One way to keep the balance is to provide more generation capacity, and thus more expensive and less efficient generators are brought online, which causes higher production cost and more pollution. Instead, an alternative is to encourage the load reduction via demand response (DR): customers reduce load upon receiving a signal sent by the utility company, usually in exchange for some monetary payback. For buildings to participate in DR, an affordable automation system and related control algorithms are needed.

This dissertation proposed a cost-effective, self-learning and data-driven framework to facilitate small- and medium-sized commercial buildings or large homes in air-conditioner (AC) units control during DR events. The devised framework requires little human configuration; it learns the building behavior by analyzing the operation data. Two algorithms are proposed to coordinate multiple AC units in a building with two goals: firstly, reducing the total AC power consumption below certain limit, as agreed between the building owners and their utility company. Secondly, minimizing occupants' thermal discomfort caused by limiting AC operation. The effectiveness of the framework is investigated in this dissertation based on data collected from a real building.

Acknowledgements

To begin with, I would like to thank my advisor, Professor Saifur Rahman, for his kind guidance and endless support during my time at Virginia Tech. In my four-year PhD study, he has provided me many opportunities to collaborate with national labs and other research institutes; this exposure greatly facilitates my professional development. I also have benefit tremendously by learning from Dr. Rahman and I felt so lucky to have him as my mentor. Without his inspiration, I cannot have such a rewarding and meaningful Ph.D. journey. I also want to thank Dr. Manisa Pipattanasomporn, who continuously give me encouragement during these years. Her optimistic and wholesome life attitude has great impact on me and will keep influencing me. Special thanks to Dr. Robert Broadwater, Dr. Guoqiang Yu and Dr. Chang-Tien Lu for their precious time to serve on my Ph.D. committee. It's my great pleasure to present my work to them and listen to their helpful feedbacks.

In addition, I want to thank all my colleagues of the BEMOSS project. Avijit, Kruthika and Warodom have led me into this amazing and challenging project. I had a great time working with Rajendra, Aditya, Mengmeng and Abdullah on developing the BEMOSS source code. Dr. Murat Kuzlu and Yonael Teklu helped me better understand the hardware aspect of this project. Also, I would like to thank all other students at Advanced Research Institute for their company and support.

Moreover, I would like to thank my advisor at Wuhan University, Professor Changhong Deng and my advisor at Tsinghua University, Professor Feng Liu. Without their encouragement and support, I would never reach this far in academic research. Special thanks to Hanchen from UIUC and Yuqing from UMich, our mutual encouragement has finally brought all of us to finish our Ph.D. degree.

Many thanks to Google Nest as well, I learnt a lot during my internship there, which helped me gain more industry experience on my research related area.

Last but not the least, my family is always the endless source for supporting. I would like to thank my parents, Zhenquan and Xiaoling, for their eternal and unconditional love and support. My wife, Mengmeng, is always at my back anytime anywhere, she is the lighthouse when I lost in mist. She also keeps giving me comments and feedbacks on my research. I am also grateful for her family for supporting both of us pursuing our Ph.D. degree.

TABLE OF CONTENTS

List of Figures

List of Tables

1. INTRODUCTION

1.1. Background

With the advent of the smart grid, there has been numerous revolutionary advancements in smart technologies in the areas ranging from electricity generation, transmission, distribution to end-uses. Interdisciplinary researches have studied the integration of renewable energy sources, like solar and wind generation through smart devices. Other developments on information technology, standards[1] and electricity market are also obvious and pushing smart grid technologies to the maturity, as exemplified in Figure 1-1.

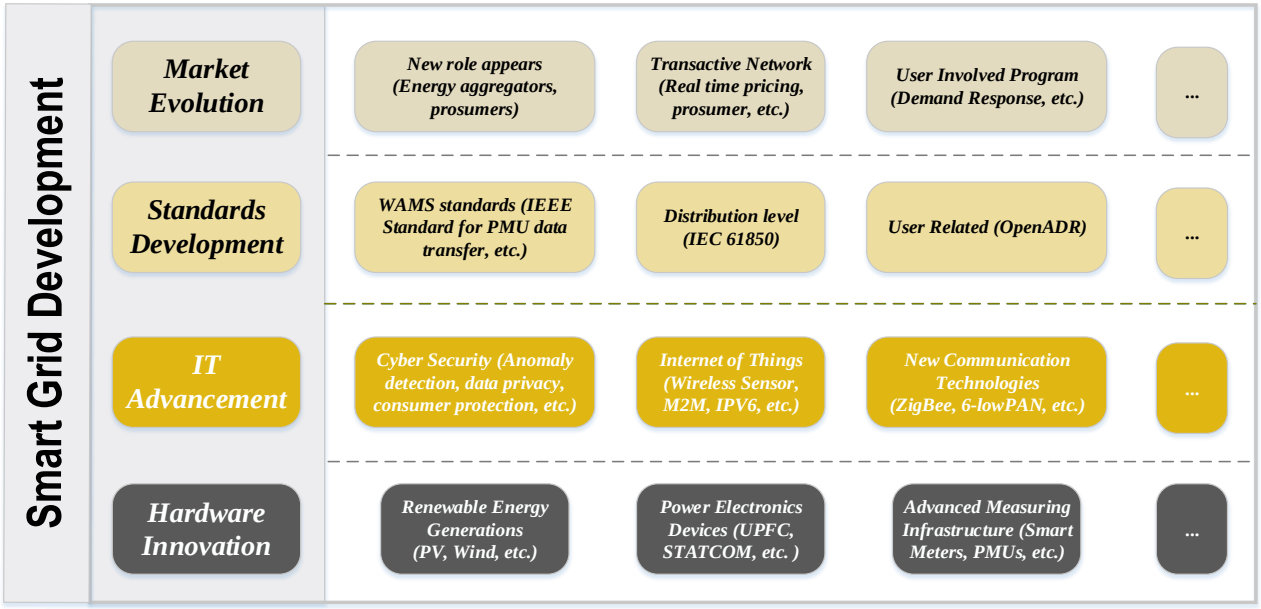


Figure 1-1 Four Major Aspects of Smart Grid Development

Among many differences between the traditional power system and the smart grid, active customer participation is unprecedented. Based on the concept of demand side management (DSM), demand response (DR) puts emphasis on the customer side. DR poses a win-win outcome for both the grid operator and the end-use customers. A previous study [2] has shown the over-burdened transmission lines might trigger a massive blackout. Hence, initiating a DR event can help reduce the stress on the grid and would further reduce the risk of power outage. On the other hand, electricity customers are willing to adjust (reduce or shift) their loads in exchange of some monetary benefits. Additionally, DR can help the market avoid higher cost and less efficient generation units. Although various DR programs exist, customer participation in DR programs remains low. The primary reason is that there is no cost-effective and automated tool suitable for most end-use customers to implement DR programs. Therefore, it is of great importance to develop proper low-cost hardware, software platform and automated control algorithms to encourage customer participation in DR programs.

1.2. Objective and Scope of the Dissertation

1.2.1. Dissertation Objective

In the U.S., buildings, both residential and commercial, consume over 70% of the total electricity usage [3]. Efforts have been made to improve energy efficiency in residential buildings using home energy management (HEM) systems. Homeowners usually pay for their utility bills directly. This incentivizes them to save energy. Commercial buildings, on the other hand, are in a more complicated situation. Major companies, such as Siemens and Johnson Controls, already have years of commercial building control experience and mature building energy management (BEM) systems as products. Unfortunately, not all commercial buildings can afford those proprietary products. Results in [4] reveal that more than 90% of the commercial buildings in the U.S. are 50,000 square feet or less; in addition, more than half of these buildings have less than 5,000 square feet. These small- and medium-sized commercial buildings (SMCB), owned by small businesses, usually cannot afford a sophisticated BEM system and thus lack an appropriate tool to implement building automation.

Fortunately, from the perspective of hardware development, there is a booming development in the area of the Internet of Things (IoT) in the past a few years and the trend remains strong. Figure 1-2 shows the recorded value and the prediction of the world-wide IoT device installed base till 2025. The advent and widespread of cost-effective IoT devices and corresponding operating systems evolves into an ecosystem where the IoT-based BEM system is born. An IoT-based BEM system, featured with low-cost and scalability, is a perfect solution to improve energy efficiency in SMCB.

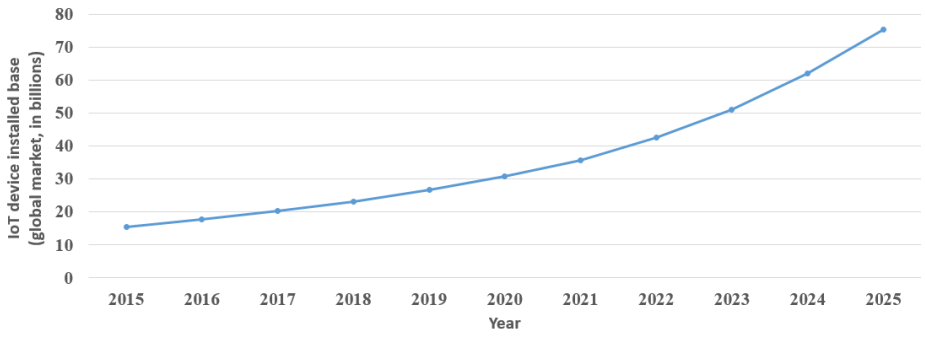


Figure 1-2 Prediction of the IoT Device Installed Base (Global Market) [5]

Besides the hardware advancement, software are also developed, both on the home automation and building automation perspective. In addition, a lot of them are open source software, which means they can be used by the building owners at no cost. Home automation examples are 'Home Assistant' and 'OpenHAB', both are suitable to be run on low-cost hardware, e.g., Raspberry Pi, and supporting hundreds of smart devices and services. 'BEMOSS', developed by the Virginia Tech, is similar open source software with emphasis on commercial buildings.

Although the IoT hardware and software platforms are readily available, control algorithms tailored for a building energy management application are limited. Existing control algorithms are mostly rule-based control on a single/two device(s) level, and thus cannot achieve optimal control at the building level. Hence, the objective of this dissertation is to design control algorithms for a cost-effective IoT-based BEM to fulfill the automated device control in SMCB, especially for the application of demand response control.

1.2.2. Dissertation Scope

Lighting, plug load and heating, ventilation and air conditioning (HVAC) systems are three major loads in commercial buildings. To control lightings and plug loads during DR events, it is comparatively straightforward, i.e., a building manager can set up some rules in the IoT-based BEM to automatically turn off low priority loads during a DR event. For the HVAC system, however, since it is directly related to occupants' thermal comfort, this means the HVAC units cannot be arbitrarily turned OFF during DR events. In addition, the package unit of HVAC system, which is commonly used in SMCB, has a cyclic power usage pattern. Without proper coordination, multiple HVAC system operating at the same time will cause a high demand and the corresponding demand charge. Therefore, this dissertation will focus on designing an automated control system with the following two controlling goals:

- Effectively coordinate Air-Conditioning (AC) units to reduce peak load during DR events
- Minimizing occupant thermal discomfort caused by AC control during DR events

Although the proposed algorithms are to be developed in the context of SMCB, they are applicable for deployment in large homes with multiple AC units, or even large commercial buildings with package units, i.e., rooftop unit (RTU), as their HVAC system. In addition, because the DR events are more likely to happen in summer time, this dissertation mainly focuses on the HVAC cooling condition; but the proposed model can be easily adapt to winter cases when necessary.

1.2.3. Proposed Tasks

In order to fulfill the objective of this dissertation, which is to design a cost-effective and automated control system for HVAC system in SMCB during a DR event, the following tasks are proposed:

Task 1: Develop a zone-based indoor thermal property model for the SMCBs using data collected from BEMOSS, an IoT-based BEM software.

a) Participate in BEMOSS installation in a building and manage the operating and maintenance issues, preparing for experiments;
b) Collect indoor temperature data from BEMOSS using the smart thermostats installed in the building, and analyze these data;
c) Develop a self-learning/plug-and-play indoor thermal property model considering possible influencing factors, including occupant activities and outdoor environmental impacts;
d) Implement the proposed indoor thermal property model using different supervised learning models, discuss the model configuration (e.g., choices of hyper-parameters), compare their performances and choose the best model;
e) Further analyze the selected model and discuss its generalization ability;
f) Explain how this model can be used in reality.

Task 2: Develop an AC unit power consumption model with respect to outdoor environment.

a) Collect the power consumption data of each AC unit in the building for several months;
b) Analyze the relationship between AC unit power consumption and weather data, including outdoor temperature and outdoor humidity;
c) Derive an AC power consumption model with respect to outdoor temperature and/or outdoor humidity;
d) Develop a power disaggregate algorithm for automatically generating power consumption models for multiple AC units, using aggregated power meter data measured from a single meter.

Task 3: Develop demand response control algorithms on automated AC units coordinated control utilizing models developed in Task 1 and Task 2.

a) Develop Algorithm I: Formulate the AC units' coordination control problem into a mixed integer linear programing (MILP) solution paradigm;
b) Develop Algorithm II: Formulate the AC units' coordination control problem into a reinforcement learning (RL) solution paradigm;
c) Implement the algorithms and validate the feasibility of the proposed control algorithms using simulation and/or real building control;
d) Discuss the efficacy and suitable scenarios for the proposed algorithms.

1.3. Contributions

1.3.1. Indoor Thermal Property Model

This dissertation proposed an indoor thermal property model learnt from coarse-grained historical temperature data collected by cost-effective smart thermostats. Because no other sophisticated sensor network is needed, the model can be implemented in a very affordable setting. The purpose of the model is to provide building thermal behavior knowledge for the AC coordination control algorithm, in order to consider occupants' thermal comfort. Because of the coarse data granularity, an innovative concept called 'temperature variation speed' is proposed as the core of the thermal model. To effectively learn the thermal model, this dissertation proposed a plug-and-play learning framework, which learns the thermal behavior of different thermal zones directly from the data. Previously, building thermal model are based on physics only and thus requires much domain knowledge for the building engineers to configure. In contrast, as a major contribution, the proposed model is easy-to-use, and removes the complicated system configuration process as well as one of the road blockers of popularizing DR. Experiment results demonstrated later in this dissertation also show the efficacy of the proposed model.

1.3.2. Power Consumption Model of AC Unit and a Power Disaggregating Algorithm

During a DR event, the aggregated AC power consumption should be kept below certain limit. Because the power consumed by each AC unit is time variant, which depends on outdoor environment. Thus, a power consumption model for each AC unit is essential for the control algorithm to estimate the total power. As a result, in this dissertation, I first proposed a linear model to model the single unit power consumption with respect to the outdoor temperature. Then, a power disaggregation algorithm based on data mining is developed. This algorithm is a significant contribution in this dissertation work in that it can disaggregate the power-temperature models of multiple AC units using data from thermostats and a single power meter. The fact that it needs only one meter to identify all AC units' power consumption models shows the cost-efficiency, and further removes the financial burden from the SMCB owners.

1.3.3. Optimization Based AC Units' Coordination Algorithm for DR

To help limiting the building peak demand, this dissertation develops an algorithm to generate a control schedule to coordinate the operation of multiple AC units during a DR event. Utilizing the building thermal model and AC power model, the scheduling problem is formulated as mixed integer linear programming (MILP) with the operation status of the AC units as binary variables and all other constraints properly linearized. This algorithm enables a fully automated control for the AC system during DR events. Experiment results show that the solver, implemented using IBM CPLEX, is able to solve the MILP

problem and generate a control strategy within a very short period of time; hence this enables rapid DR implementation in buildings. Tests also show that the algorithm's computation efficiency makes it suitable for most SMCB, with up to 12 thermal zones. Finally, this algorithm can also be used in the simulations to help the building owners to decide what the optimal DR power limit for their buildings is; this results help their decision-making when choosing the appropriate DR plan with the utility company.

1.3.4. Reinforcement Learning Based AC Units' Coordination Algorithm for DR

This dissertation proposes a second automatic AC units' coordination control algorithm using a reinforcement learning (RL) framework. Since RL is most suitable to help making state-based sequential decision to achieve an optimal control objective, it is appropriate for the AC units scheduling problem discussed in this study. In this case, an optimal AC control agent is trained before the DR event to obtain the optimal policy, and when a DR event starts, no other computation is needed. As a result, this control is based on offline training. The uniqueness of this algorithm lies in that it can provide a stepwise correction when compared with the optimization based control algorithm, this is because the agent will choose an action according to the state it is in at this time step. Since the policy used for action choosing is trained before the event, when a DR signal comes, a building can participate in the DR immediately by following the optimal policy. The comparison of the two algorithms (1.3.3 and 1.3.4) will be discussed in the dissertation and their deployment suitability will be analyzed.

1.4. Summary

In summary, by adopting the models and algorithms developed in this dissertation, the SMCBs' owners can enable automated AC units' coordinated control with consideration of indoor thermal comfort during DR events. Based on the low-cost IoT-based BEM platform, this solution is more affordable to many building owners. By providing such a cost-effective and automated DR solution, this dissertation can help mitigating the barrier of popularizing DR programs in many buildings.

2. LITERATURE SEARCH

2.1. Background Knowledge

2.1.1. Smart Grid

2.1.1.1 Definition

According to the Energy Independence and Security Act of 2007 (also known as the Clean Energy Act of 2007)[6], which appears to be the first official definition of Smart Grid:

"*It is the policy of the United States to support the modernization of the Nation's electricity transmission and distribution system to maintain a reliable and secure electricity infrastructure that can meet future demand growth and to achieve each of the following, which together characterize a Smart Grid:*

(1) Increased use of digital information and controls technology to improve reliability, security, and efficiency of the electric grid.

(2) Dynamic optimization of grid operations and resources, with full cyber-security.

(3) Deployment and integration of distributed resources and generation, including renewable resources.

(4) Development and incorporation of demand response, demand-side resources, and energy-efficiency resources.

(5) Deployment of 'smart' technologies (real-time, automated, interactive technologies that optimize the physical operation of appliances and consumer devices) for metering, communications concerning grid operations and status, and distribution automation.

(6) Integration of 'smart' appliances and consumer devices.

(7) Deployment and integration of advanced electricity storage and peak-shaving technologies, including plug-in electric and hybrid electric vehicles, and thermal storage air conditioning.

(8) Provision to consumers of timely information and control options.

(9) Development of standards for communication and interoperability of appliances and equipment connected to the electric grid, including the infrastructure serving the grid.

(10) Identification and lowering of unreasonable or unnecessary barriers to adoption of smart grid technologies, practices, and services."

An European conception of smart grid is pointed out by [7] that smart grid can “intelligently integrate the actions of all users connected to it-generators, consumers and those that do both – in order to efficiently deliver sustainable, economic and secure electricity supplies.”

2.1.1.2 Smart Grid Technologies

The technologies related to smart grid has been started long before this term, many of the research can be dated back to the 20 century; however, the need to have a reliable and efficient power grid, plus recent breakthrough on many other disciplines have fostered an ideal environment for the smart grid technologies

and their applications. Over the past ten years, the developments of smart grid technologies are marvelous; a review categorizes the smart grid technologies into the following aspects[7]:

A. Control and Communication

In the smart grid, the power flow, grid components, network topology and operation requirement are different from what they were used to be in the legacy power grid. Study in [8] proposes four control methods to introduce more clean energy in the grid: power electronic based control method, multi-agent system based control method, advanced fault management control method. In the electricity transmission and distribution grid, with the maturity of the power electronic based control technologies, the flexible AC transmission system (FACTS) and high voltage DC (HVDC) technology have been applied to increase the grid reliability and reduce the costs on power transmission [9].

Smart grid requires a reliable, two-way and secure communication. According to the application area, many communication technologies, wired or wireless, are available. Currently there are two types of information infrastructure in smart grid: the first data flow is from sensor or appliances to smart meters; the other is from smart meters to the utility's data centers [10]. In general, the first type commonly exists in a house or a building, it can be accomplished by powerline communication, ZigBee, Z-wave, Bluetooth and 6LowPAN. Some commonly used communication technologies for home energy management system are compared in[11]. The second communication happens for longer distance and can be achieved by cellular or the Internet. Authors in [12] proposed a co-simulation platform for smart grid operation, with consideration of communication system.

B. Sensing and Measurement

Sensing technology is indispensable in the smart grid, sensors of different capabilities include phasor measurement unit (PMU), the high speed sensors with sample rate of 48 per AC cycle; remote terminal units in the Supervisory Control and Data Acquisition (SCADA) system; and smart meters measuring the power consumption of end-users every 15 minutes interval. In the transmission network, compared with today's SCADA system, the Wide Area Measurement System (WAMS) is fast enough to track dynamic events as well as capable to monitor vital stability indicators [13]; the control and protection-scheme based on such sensing technology make it possible for the smart grid to be more robust. Authors in [14] analyzed communication schemes for advanced metering infrastructure. On the end-user side, smart metering provides the platform for transactive control [15] and the possibility of having more flexible pricing plans.

C. Plug-in Hybrid Electric Vehicle (PHEV) and Vehicle-to-grid (V2G) technology

PHEV offers many benefits such as zero emission and fast acceleration. Besides, it is capable to interact with the power grid. With proper communication and control techniques, an aggregation of PHEV can respond to the request from the power grid: provide peak load and spinning reserve. Research has shown the V2G technology, by optimally scheduling, can significantly reduce the variance of load curve, smoothing the load profile and thus enhance the power system operation efficiency[16]. Other research includes: EVs interact with renewable energy, impact analysis of vehicle-to-grid technology and charging strategies of electric vehicles on distribution networks, etc.[17–19].

D. Security of Smart Grid

Security of power grid is a crucial issue due to the severe consequences caused by loss of power. Compared with traditional power grid, the smart grid has more participants and is operated under a more deregulated

environment, it is prone to some security attacks. Work in [20] points out some most serious vulnerabilities in the smart grid are:

1. *Customer security*: smart meters collect and send massive amount of customers' data, posing a privacy vulnerability to the end-users.
2. *Increasing number of intelligent devices*: Numerous intelligent devices are interconnected with the smart grid, introducing more attack entry points in the grid.
3. *Physical security*: many smart grid components are out of the utility's premises, and might be in insecure physical locations.
4. *Attacks in communication*: device-to-device communication is prone to the data spoofing attack.
5. *Using Internet Protocol (IP):* Together with convenience and compatibility, the IP-based smart grid components also bring the vulnerability of many IP-based network attacks such as IP spoofing, denial of service, etc.
6. *Human related:* insider attacks is possible with so many stakeholders.

Similarly, author of [20] summarizes the security issues and challenges in the IoT-based smart grid.

E. The Integration Of Renewable Energy

Renewable energy such as solar and wind generation is growing in global installation capacity, system penetration rate and single generator capacity. The largest single wind turbine has reached nearly 10 MW [21]. Distributed renewable generation: building integrated PV (BIPV) system, in the form of roof tiles, has been commercialized and into the market [22]. Power electronic devices also play vital roles in the smart grid and realize the integration of renewable energy in a highly efficient way [23,24].

F. Micro-grids

Micro-grid provides a good solution to integrate the distributed generation and electricity demand, it has two operating mode: interconnected mode under normal situation and islanding mode under some contingencies. Therefore, it improves the power grid local reliability, reduces transmission loss and facilitates the integration of renewable energy. Authors in [25] review the distributed energy resources (DER) and micro-grid related technologies from the perspective of micro-grid architecture, emergency operation, fault detection, safety analysis and market operation. Authors in [26–28] investigated the problem of DER planning in micro-grids. Micro-grid control technologies are summarized and reviewed in [29]. The study [30] also reviews the micro-grid power quality, protection and stability. A convolutional neural network based analyzing tool for micro-grid voltage security is proposed in [31]. Authors in [32] studied how to coordinate DER for active power provision, using a data-driven approach.

Besides the abovementioned technologies, there are other important researches that is making significant impact on the future of the power grid. In addition, research based on the system or device that is built on these technologies is popular, such as smart buildings and energy management system.

2.1.2. Demand Response

2.1.2.1 Definition

The Federal Energy Regulatory Commission (FERC) gives the following definition to demand response[33]:

"Changes in electric usage by demand-side resources from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized"

According to this definition, demand response will be conducted under two circumstances: the presence of a high wholesale market price and the system has a decreased reliability. An increase on the electricity wholesale market price means the unit cost for producing electricity has raised, and sometimes the generators with low efficiency and heavier pollution need to be turned on. The reduction of load at this time will improve the system efficiency and be more environmental friendly. On the other hand, organized load reduction will enhance the grid reliability, postpone the system upgrade and increase the capacity utilization rate.

2.1.2.2 Benefits of Demand Response

Demand response improves the resource-efficiency of electricity production, which further creates the following four types of benefits [34]:

- Participant financial benefits: the end-user who participates in the DR programs receives bill savings or incentive payments;
- Market-wide financial benefits: the reduction on demand avoids the need to use costly-to-run generators as well as postpones the power system capacity upgrade;
- Grid reliability benefits: by reducing the load, probability for forced-outages also decreases. This prevents the power system blackouts that will cause severe consequences to the society;
- Market performance benefits: demand response mitigates the energy suppliers' ability to exercise market power by raising power prices significantly above production costs.

2.1.2.3 End User Response

Research works [34,35] point out three general actions that can be taken by the end users in the grid when demand response programs start:

a. Customers reduce their peak load for specific time periods without changing the power consumption pattern in other time periods, for example raise the set point of a HVAC thermostat temporarily;
b. Users respond to high electricity prices and shift the load from peak time to off-peak periods when the price is much lower, for example use the washing machine late at night instead of in the afternoon;
c. If possible, the users start their own power generation facility. Such generators enable the customers reduce their power consumption from the grid and meanwhile decrease the impact of the DR events. For example, some factories do not want to entirely stop their production will use their own generation to support.

2.1.2.4 Types of DR Programs

In general, demand response programs can be categorized as incentive based programs (IBP) and price based programs (PBP).

Incentive based Programs

IBP can be further divided as classical, such as direct load control (DLC) and market-based, such as demand bidding and capacity market.

Price based Programs

PBP programs are reflecting the dynamic electricity pricing, in general, price during peak period will be very high to discourage usage of electricity and thus realizing the goal of shaving the peak. These programs are gaining more popularity due to the advancement in measuring and communication technologies. Rates such as critical peak pricing (CPP), time of use (TOU) and real time pricing (RTP) are examples[35].

2.1.2.5 Examples of DR Programs from Utility Companies

Utility companies and load aggregators are very interested in demand response, and many programs have been introduced, as exemplified in [36–40]. Programs from one utility company, the San Diego Gas & Electric (SDG&E), are explained in detail below. Table 2-1 shows three of the available DR programs for commercial buildings from SDG&E, they differs from each other on DR rewards, notification time and limit of occurrence. The customers can choose one from these available DR programs according to their preferences.

Table 2-1 San Diego Gas & Electric(SDG&E) DR Programs (Excerpt) [41]

DR Programs	Reward	Risk	Notification Time
Capacity Bidding Program (CBP)/ May - October (Day Ahead/Day of/Day of 30 Min)	Incentives provided to customers who reduce their energy use when requested the day ahead by SDG&E.	Failure to reduce energy may result in a penalty	Day ahead of event before 3pm / Day of event by 9 am / Day of event 30 minutes prior to start
Critical Peak Pricing (CPP-D) / Year Round	A time of use rate which features increased costs during critical event periods.	Failure to decrease energy use during an event can result in significantly higher prices.	Before 3 pm the day before the event.
Summer Saver / May - October	$9/ton for the 30% cycling option and $15/ton for the 50% cycling option.	During an Event Day the device overrides the owners' thermostats and controls the amount of time their HVAC units' condenser is allowed to operate	Direct Control, no advanced notice

Though these programs are different from each other by how DR is specifically implemented, they are essentially conducting customer side load reduction (in kW). In general, they belong to three types of implementations.

A. Direct load control

This program should be more appropriate to be called as demand side management (DSM) since the utility is managing the load instead of the end-users initiatively respond to some signals. Since the utility companies are compensating their customers who join this program, it is an incentive based program. By installing a control device on the customer’s AC unit, the Summer Saver program enables direct control of

the customer's HVAC units for the utility company, namely SDG&E in this case. The reward is in proportion to the AC unit's cooling capacity, and there are two levels of control intensities: reducing AC run time by 30% or 50%. Although it is simple to implement and at no cost, the customers are compromising their thermal comfort in exchange for the reward, since the utility company will not and cannot consider the occupants' thermal comfort when doing the control.

B. Reducing load below a preset load amount

Capacity bidding programs (CBP) allow the customers make a monthly pledge on power reduction, and when demand response comes, the customers need to reduce the power to fulfill the pledge (electric meters reporting electricity use in 15-minute intervals are needed). Apparently, these programs are incentive-based as well. The incentive is calculated based on the pledged amount of load reduction; that means the more load is reduced, the more payment the customer will get from the utility company. According to different notification time, the programs can be categorized as 1) Day-Ahead, which will notify DR event a day ahead; 2) Day-Of, which will notify DR event on the day of event, before 9 AM and 3) Day-Of (30 minutes), which will notify DR event 30 minutes prior to its start. Each of these DR program has three levels depending on the duration of the DR event: 1) 1 to 4 hours; 2) 2 to 6 hours and 3) 4 to 8 hours. As shown in Table 2-2, the customers will get maximal incentives if the load reduction can be implemented quickly (within 30 minutes) and is able to last long.

Table 2-2 Load Reduction Incentive Payment in August ($/kW-month) [42]

	1 to 4 hours	2 to 6 hours	4 to 8 hours
Day-Ahead	17.56	19.99	20.76
Day-Of	21.08	23.99	24.91
Day-Of (30 Minutes)	24.24	27.59	28.65

Failure to fulfill the pledge will result in the applicable penalty provisions.

C. Reducing load below capacity reserve

The critical peak pricing default (CPP-D) plan is essentially a time-of-use (TOU) tariff. There are four different pricing level as shown in Table 2-3, they are set depending on how congested the power grid is: Off-peak, semi-peak, on-peak and CPP event day. The CPP-D program is a price based program: customers respond to the changing electricity price. According to Table 2-3, during the CPP hours, the electricity price increase tremendously compared with other times. Usually, during these hours, the customers reduces their load to a large extend; or they can avoid such high price by paying the capacity reserve charge (CRC).

Table 2-3 CPP-D Pricing (Secondary price in $/kWh)

	Summer	Winter	CPP Event Day Adder
On-Peak	0.10731	0.09846	1.16815
Semi-Peak	0.09845	0.08401	
Off-Peak	0.07177	0.06411	

CRC is introduced in the CPP-D plan [36,37]:

"The Capacity Reservation Charge option allows customers to self-select an amount of electric usage (in kilowatts) that they want to protect from the high price of electricity during a CPP event, and pay for through a fixed monthly Capacity Reservation Charge (CRC). All usage during a CPP Event that is protected under the customer's capacity reservation will be billed the corresponding energy charges for the time period but not the CPP Event Day Adder. All

usage during a CPP Event that is not protected under the customer's capacity reservation will be billed at the CPP Event Day Adder and the corresponding energy charges for the time period."

Capacity reserve can be viewed as a price protection for certain amount of power, below which the electricity cost will not include the CPP Event Day Adder. Otherwise, the electricity price will be more than 10 times higher according to Table 2-3. However, capacity reserve comes with a price, for instance, CRC at utility secondary level is $5.60/kW per month.

Similar to the load reduction pledge in CBP program, the business enrolling in CPP program has to shift the load so that the remaining load is lower than the reserved amount.

In summary, three types of DR programs in the electricity market are introduced in this section; though these programs given as examples above are from SDG&E, they are typical DR programs and commonly available in many other utility companies with little or no differences. For instance, the Southern California Edison also provides their customers with capacity bidding programs (CBP), but their critical peak pricing programs (CPP), also known as summer advantage incentive (SAI) programs, do not have an option for capacity reserve so that the customer need to reduce their power as low as possible to avoid the extremely high critical peak pricing [43].

Table 2-4 shows some statistics of two abovementioned utility companies: the number of customers enrolled in demand response programs; expected demand savings at the system peak hour assuming all demand response is called; actual demand reduction achieved by DR events.

Table 2-4 Demand Response Implementation in Two Utility Companies in 2016 [44]

		Residential	Commercial	Industrial	Total
San Diego Gas & Electric Co	Customer Enrolled	102,638	123,308	74	226,020
	Potential Peak Saving (MW)	17.8	36.2	8.4	62.4
	Actual Peak Saving (MW)	17.8	36.2	5.1	59.1
Southern California Edison Co	Customer Enrolled	671,300	18,024	1,831	691,155
	Potential Peak Saving (MW)	219.1	1,180.5	165.5	1,565.1
	Actual Peak Saving (MW)	218.6	866.0	70.7	1,155.3

2.1.3. Building Energy Management (BEM)

A building energy management system is a software platform used to monitor the energy usage in the building, with a goal of improving building energy efficiency or implementing intelligent control. Two major benefits from a BEM system is providing the system awareness and building automation: by analyzing the system operation data, the building managers gain more understanding on the energy usage inside their buildings; by implementing scheduling or other rule-based control, BEM can also realize building automation. Until recent years, the BEM supports limited communication protocols, and is not scalable and is hard to install and configure. In this dissertation, these BEMs are called traditional BEM system. Though traditional BEM system can achieve an automated control, and laid the foundation for DR implementation, it is impractical for most of the commercial buildings to have such system due to its

prohibitive price. According to an interview conducted by Intel [45], the average expenditure on a basic BEM system is $2.50 per square foot in the U.S., and this number can be as high as $7.00. That is to say, on average, owners of a commercial building with a floor plan of 10,000 square feet will need to invest $25,000 to make the building smart enough to participate in the DR programs. Therefore, a more cost-effective solution should be developed to facilitate the DR implementation.

2.1.4. Internet-of-Things (IoT)

According to [46], the Internet of Things is "*a paradigm where everyday objects can be equipped with identifying, sensing, networking and processing capabilities that will allow them to communicate with one another and with other devices and services over the Internet to accomplish some objective.*" With the booming development of information and communication technologies, the area of IoT flourishes. As one of the most important research area in the Internet of Things, smart buildings are embracing an increasing number of IoT technologies: an IoT-based framework with smart location-based automated energy control is proposed in [47]. The study [48] reviews the IoT considerations, requirements and architectures for smart buildings from an energy optimization perspective. Research on the energy efficiency problem in IoT based smart building is conducted in [49]. Building Energy Management Open Source Software, developed by the Virginia Tech, is also one of the famous energy management system for small and medium commercial buildings[50]. The researchers also discuss the performance evaluation and deployment experience in [51], and quantifying energy saving by thermostat control using BEMOSS[52].

As a result, many companies provide an IoT-based BEM system as a solution, which usually costs much lower than the traditional ones and provides a more compelling return on investment. With the help from the IoT-based BEM system, DR can be implemented.

2.1.5. Building Retrofit

The advent of IoT-based BEM system has reduced the building retrofit cost to a large extent; beside this, research and policies also help building owners to make a decision. Researchers from the Lawrence Berkeley National Lab have conducted ten million EnergyPlus simulations of small and medium commercial buildings, saving the results to the energy efficiency performance (DEEP) database [53]. Together with a web-based toolkit, DEEP can recommend energy conservation measures and estimate performance on energy savings as well as financial return. The work in [54] analyzes the PACE financing, On-bill repayment and Energy savings Performance contracts in the energy-efficiency retrofits in the commercial buildings. Authors of [55] discuss three common building retrofit business models in Spain: the owner financed model, the utility fixed repayment model and the energy performance model. The market challenge and model evaluation are also studied.

2.1.6. Machine Learning

Machine learning has been around for more than half a century already, and currently being widely applied to solve a great variety of problems. According to Arthur Samuel, 1959, machine learning is the field of study that gives computers the ability to learn without being explicitly programmed [56]. Machine learning is suitable for many use cases: it can simplify the problems for which the existing solutions require many hand-crafted rules; it can get insights about complex problems from great amount of data; also, many machine learning models can adapt to new data which makes it suitable to be applied in a changing environment for online learning. Machine learning models can be categorized for the type of problems it is solving: classification or regression. In addition, it can also be categorized as supervised learning,

unsupervised learning and reinforcement learning three types. Each of these types can be applied in engineering, and solve real-world problems: supervised learning can be used for prediction; unsupervised learning will be used for anomaly detection and reinforcement learning can be used in decision making or control.

The reason machine learning becomes a great problem solving tool is that in recent decades, with the advance of technology, the amount of data collected from all cyber physical system are skyrocketing, which provides a great resource for more profound insights. According to [57], research on the data-driven smart grid management include but not limit to power generation side management, micro-grid and renewable energy management, asset management and collaborative operations and demand side management (demand response); in addition, many companies, such as EnerNOC, C3 IoT, Nest and more are providing the machine learning smart energy management products and services to help their customers to improve the energy efficiency and to participate in demand response programs.

2.1.6.1 Machine Learning In Prediction

Many machine learning models can be used for prediction: regression models, neural network, kernel machine, random forest and more. In the smart grid paradigm, electricity load prediction is one of the most important topics, and has become the best testbed for all the machine learning models ever since. Higher accuracy on the load prediction brings significant benefit from the perspective of environment, economy and energy saving. The paper [58] presents a review and performance comparison between neural network, support vector machine and regression trees. New forecasting models are being proposed [59,60]. Solutions of load prediction problem specific to buildings are reviewed [61,62]. Authors of [63] compared deep learning approach with traditional time series approaches on building load forecasting.

2.1.6.2 Machine Learning In Control

Reinforcement learning depicts a problem solving agent making sequential decision on what action to take at different states with a goal of maximizing future discounted reward. Many work have been done on solving problems in the energy industry. An optimal transformer tap setting using batch reinforcement learning is proposed in [64]. Authors in [65] use reinforcement learning to develop a controller aiming at providing comfort in buildings with minimal energy consumption. The study [66] proposes a new energy management system formulation to deal with DR in residential and small commercial buildings, using device-based reinforcement learning. By pointing out the MPC approach usually very complicated, authors of [67] present a reinforcement learning control for the low exergy buildings, with renewable energy technologies. Researchers in [68] propose the Consumer Automated Energy Management System (CAES) to decrease the energy cost in residential houses and smooth energy usage based on real-time pricing. The study [69] engage thermostatically controlled loads to provide short term ancillary service to the grid based on a classic Markov decision process, producing similar results from a MPC approach in [70].

2.2. In-depth Review

2.2.1. Research on Building-Grid Interaction

2.2.1.1 Building-Grid Interconnection

Buildings, as a major electricity consumer, act passively in the past and consumes energy in a unidirectional manner. With the advance of the distributed renewable energy generation, smart metering and other

information technology, buildings are able to participate into the operation of smart grid more actively, and thus become smarter. According to the authors of [71], the term of 'smart buildings' has been public for many decades; today, the smart buildings are based on earlier concepts and advance information, sensor and communication technologies. The authors also summarized that the sensors, actuators, controllers, user interface, communication network and smart meter together make up the smart building. A literature review is conducted in [72] about the significance of buildings in the framework of the smart grid. As mentioned in the paper, a smart building should meet with four requirements in order to interact with the smart grid: 1. Incorporation of smart metering; 2. Capable of demand response; 3. with distributed architecture; 4. Interoperability.

Some work focus on making a building more demand responsive and more controllable. For example, [73] uses PV and ice storage to optimize the electricity usage by the building, and enable them to provide service to the grid. A method based on custom power device to transform a building into a dispatchable load or generator is presented in [74]. Integrated with Li-Po batteries and PV module, such configuration allows the building to control the amount of active and reactive power being imported from or exported to the grid. Authors in [75] propose an interactive control strategy for bringing in commercial buildings to the smart grid optimization problem. A simplified building thermal storage model is developed to predict the demand alteration potential. A distributed demand-side energy management scheme in residential smart grids, with consideration of different household appliances is proposed in [76]. Besides the optimization of grid and building operation, enabling buildings to participate in the ancillary service market has also been proved significant: The work in [77] designs an agent-based building energy management system for the interaction between the operation of smart grid and smart building, the results demonstrate this system can significantly enhance the voltage profile on the low voltage feeder while maximizing building comfort and energy efficiency. Reference [78,79] propose an easy and low-cost solution to provide substantial frequency regulation service to the power grid by controlling HVAC fans in the commercial buildings. According to [78], the potential of regulation reserve in the U.S. by using this proposed method is estimated to be 4GW.

2.2.1.2 Building Energy Consumption Model

Load forecast in buildings is important to an effective building control. The most common load forecast techniques can be categorized as engineering methods, statistical techniques [80] and artificial intelligence techniques. The performance of various models have been reviewed in [81,82]. Similarly, the work in [58] presents a review of several load forecast models, which are based on regression. Comparisons of models are demonstrated on day ahead hourly load forecast and daily peak demand forecast. Conclusions of this study include finding out the best performer: artificial neural network with Bayesian regulation, and almost all the models yield better accuracy when predicting a campus load than that of a single building load. Not only are there forecasting models for predicting energy usage in buildings, but also a meta-learning approach to identify the best and most suitable model for a specific building [83].

2.2.2. Demand Response and New Requirements

According to the Table 2-2 from Section 2.1.2.5, utility company pays more to the customers who participates in the DR programs with less notification and preparation time. That means the faster the user respond, the more value the DR provides. Shorter response time gives the grid operator more ability to deal with potential contingency; as mentioned in [84], "Providers should be able to respond to an instruction from us within a maximum of 240 minutes, although response times within 20 minutes are preferable". Authors in [85] mention DR resources are capable of providing ancillary services in the electricity market for their superior response time when compared with generators.

However, as of now, the customers need more time to prepare their load reduction. The study in [40] points out that data from the Short Term Operating Reserve (STOR) market in the United Kingdom concludes over 85% of DR participants have less than 10 minutes to respond, which poses a major limiting factor preventing load shifting demand response. The reason is obvious: most of the customers do not have the ability for a fast demand response. The question of how to facilitate buildings to implement a fast DR is not well studied, though there are limited papers work on this topic: Authors of [86] develop a control strategy for chiller in the building to implement a fast DR, making the building act with the role of 'operating reserves'; the authors also shows considerable amount of power can be reduced with little sacrifice on occupants' comfort.

The implementation of demand response includes two phases: decision making on a control strategy and the execution of such control commands. With the help of a BEM system, both these two steps can be optimized and automated. In addition, an appropriate control algorithm is needed to accelerate these processes to realize a fast DR.

2.2.3. Building Participation in DR

2.2.3.1 DR Potential in Buildings

Each building, due to the difference on its structure, operating schedule, utility and types of appliances, has different saving potential for participating in DR programs. Before signing contract with the utility company, the facility managers of buildings would like to evaluate such potential.

A data-driven approach, using data collected from temperature/humidity/light sensors, carbon dioxide sensors, passive infrared sensors and smart plugs, evaluates the ability of a building to participate in DR [87]. The limitation of such method is the hardware investment: even before saving money, the building owner has to invest a lot on building a sophisticated sensor network. In addition, after the sensors are deployed, more than two months, as proposed in the paper, have passed before the data size is big enough for analysis. So, a method without expensive hardware investment and based on ready to use data is needed. By studying the electric load data from commercial and industrial customer (peak load larger than 200 kW) with 15-minute interval, authors of [88] propose a methodology helps the building owners to discover the building's potential for participating DR, improving energy efficiency, eliminating energy waste and peak load management.

2.2.3.2 Device Control in Buildings during DR

Three major loads in commercial buildings are lighting, plug load, and HVAC systems. Researches in both industry and academia have made progress on involving such loads into demand response controls.

There are two approaches for reducing power consumption for lighting system: turning off, which is suitable for unnecessary lights such as decorative lights, and dimming them, when adequate hardware is installed to implement dimming. With dimmable ballast or LED dimming driver, lights can be dimmed ranging from 0~100%. In fact, DR events mostly happens in a sunny summer afternoon, when the sunshine is the brightest. BEMOSS proposed a closed loop lighting control with the help from a light sensor, with a goal of maintaining fixed Lux level indoor [89]. A literature review on the usage of daylighting instead of artificial lighting in office buildings is presented in [90]. Similarly, the study in [91] investigates the potential to reduce energy consumption by artificial lights in the buildings by using shading systems. Authors of [92] present a literature review for energy saving potential for electric lighting in office buildings. Control strategies and techniques for demand response used in commercial building both for HVAC system

and lighting system is also studied in [93]. Works on managing plug loads for demand response have also been done[94,95]. Authors of [95] build an Integrated Plug Load Control (IPLC) system, which consists of a central building controller, an energy information gateway and the smart power strips in a hierarchy structure; and therefore turns the plug loads in a building as demand response resources.

Besides the control of three major loads in buildings, higher level of control flexibility can be achieved with the assistance of renewable energy sources or energy storage device in buildings. By strategically deploy renewable energy generation and ice storage in the building, the study in [73] proposes a method to reduce peak demand and over energy usage in demand responsive buildings. Authors of [96] investigate a control algorithm for heterogeneous devices in buildings, with real time pricing in consideration and a goal of peak load reduction and energy efficiency improvement. Similar work on smart home environment is investigated in [97].

2.2.3.3 HVAC Control in Buildings during DR

The HVAC system usually consumes around 30% of the total building electricity usage[98]. In this section, the approaches for controlling HVAC system during DR events are summarized.

A. Pre-cooling

Due to the thermal inertia in the building thermal mass, shifting the cooling task to an earlier time before a DR event starts is an effective way for reducing load during the event. The study in [99] studies the peak load reduction effect by using pre-cooling in an office building and demonstrates effective and significant load reduction. The authors also mentioned there is insufficient evidence to demonstrate the benefits of nocturnal pre-cooling. Authors of [100] present a result that pre-cooling provides much demand relief but consumes more energy and cost more. A demand-limiting strategy using pre-cooling is compared with the night-setup control in [101]. To implement pre-cooling, the techniques, such as start cooling how long before, are needed and will change with different buildings. Based on simulation, authors of [102] discuss how to optimize pre-cooling strategies, and demonstrate by using pre-cooling, it is able to reduce peak demand as expected on a DR event day.

B. Fan control

Fan and ventilation system accounts for nearly 50% of HVAC system's electricity usage, according to the Commercial Buildings Energy Consumption Survey in 2012 [98]. Due to the fast response feature, controlling fans is one effective approach to help implement load reduction. The work in [79] demonstrates the ability of using fans in HVAC system to provide substantial frequency regulation service without damaging the indoor comfort and at no additional cost. Similar studies are [103,104]. Authors in [78] demonstrate to provide ancillary power flexibility by modulating the supply duct static pressure (SDSP), and estimate that more than 4 GW of regulation reserve is available in the commercial buildings in the US. The study in [105] evaluates the DR potential of HVAC fans in Nordic countries with consideration of indoor climate conditions and renewable energy source. Admittedly, special care is needed for designing fan control strategy as [93] points out insufficient airflow will increase the pollutant level in the conditioned space.

C. Global Temperature Adjustment (GTA)

The study in [106] conducted simulation using 2012 US Energy Information Administration data, showing that the over-cooling problem in the commercial buildings in the US is causing tremendous energy, environmental and financial costs. Raising the set point is one of the most obvious way to reduce the power

of HVAC system. Authors of [107] propose a peak-load reduction computing tool to individually control each zone's set point. It also summarizes that global temperature adjustment method is the best performer among many HVAC based DR strategies [108–110]; and the work in [111] showcases the savings in different studies using this method. Researchers in [112] present a systematic approach for quantifying the influence of many factors on HVAC energy consumption in office buildings; these factors include set points, dead band, occupants schedule and more.

D. Model Predictive Control (MPC)

According to [113], the advantages of MPC for HVAC system control include but not limit to:

- By using a system model, the control actions are anticipatory instead of corrective;
- With disturbance model for disturbance rejection;
- Being able to consider constraints and uncertainties;
- With a cost function for achievement of multiple objectives;
- Being able to use advanced optimization algorithms for control strategy computation.

As a result, MPC method is widely used in building energy management area [114,115]. Authors of [116] investigate a peak demand reduction method using MPC and with consideration of hourly-based electricity tariff; a result of significantly reduced peak load is shown.

E. Cost-effective Solutions:

As discussed in Section 2.1.3, small business with small- and medium-sized commercial buildings usually do not have sufficient budget for a very sophisticated solution. To target these buildings, cost-effective solutions with minimum investment for retrofit are needed, and is also an area for research. The study in [117] describes a supervisory control strategy for limiting peak load in the small and medium commercial buildings. Such software-only retrofit enables the peak load reduction by controlling air conditioning and refrigeration system. Authors of [118] describe a cost effective retrofit technology which is able to conduct peak load shaving by collectively control multiple rooftop AC units in small and medium commercial buildings. A proof of concept is given by deploying the prototype in a building and an estimation of energy saving shows that the installation fee can be paid back within a year's energy saving.

2.2.3.4 Building Control Considering Indoor Thermal Comfort

A. Indoor temperature prediction

Indoor temperature prediction is of vital importance in HVAC system control since the forecast results will provide key information for the BEM system to control the AC units while maintaining the indoor thermal comfort. Many research works have been conducted for designing accurate temperature prediction models using traditional thermodynamic, simple regression [119,120], artificial neural network [121–125], grey-box models[126] and more. In [122], a realistic multi-zone temperature prediction model for a large building is proposed. The model is based on ANN and is capable to consider the coupled effects between thermal zones. Researchers in [121] form an indoor thermal comfort regulation framework by combining a neural network predictor and a fuzzy logic controller. Four difference building thermal behavior prediction models are compared in [127].

However, most of the prediction models rely on the inputs from a sophisticated sensor network [119,120,123–127], for instance, carbon dioxide sensor is used to model occupancy level, sun irradiance for calculating solar heat gain. Such a variety of inputs will definitely increase the prediction accuracy with

no doubt; nonetheless, it is usually impractical in real world SMCB application scenarios when installation cost is one of the major concerns of building owners.

Without sophisticated and designated sensors for measuring specific factors in SMCB, some of them can be indirectly obtained: for most of the building, the occupancy level reflects a changing distribution and therefore can use time of the day and day of the week to represent. This practice is common in previous research according to [125].

To solve the problem of insufficient hardware investment, Researchers in [128,129] propose multiple AC units coordinated control using only thermostat reading, which is a great workaround since the thermostats are the only 'sensor' that need to be installed and eliminate other hardware as well as system configuration costs.

B. Energy Management Considering Indoor Comfort

Energy management with consideration of human comfort is important: Authors in [130] design a multi-agent building energy management system based on occupant behaviors. The study [131] analyzes the indoor thermal comfort problem in a single thermal zone building, presenting five model based prediction control algorithms, all with focus on thermal comfort and/or energy saving. Multi-zone indoor environment control is also studied, a multi-agent control system using particle swarm optimization (PSO) is proposed in [132] to achieve maximum thermal comfort level under the circumstance of energy shortage. The work [133] does the similar research as [131], minimizing the energy usage while maintaining the thermal comfort, which is assessed by predicted mean vote (PMV). The predictive model used is implemented by radial basis function (RBF) neural network. Another similar low computational cost technique is investigated in [134].

C. Indoor Thermal Comfort vs. Productivity

Inappropriate thermal condition will cause both physical and psychological influences to the occupants and thus impact their productivity. Authors in [135] propose a method to describe the personalized dynamic thermal comfort in an office environment, based on parameters estimated by online voting data in contrast to the PMV model. By studying a call center for a year, authors in [136] find a linear relationship between indoor temperature and workers' productivity: for instance, increase temperature by 1.0 ℃ from 25.0 ℃ to 26.0 ℃ will results in 1.9% production loss. Subjective experiments are conducted by authors of [137], which shows the optimum temperature range for the tested study is 22 ℃ to 26 ℃. A detailed review on the relationship between thermal comfort and productivity can be found in [138].

2.3. Conclusions and Knowledge Gaps

2.3.1. Practical and Cost-Effective Modeling Method of Indoor Thermal Property

Much effort has been made on developing building thermal property models. Most of these models set out from thermal dynamic theory, which are known as forward models [139]. That is, by constructing a detailed building physical model, the indoor thermal property can be fully simulated. Nevertheless, these models, though they have excellent performance in building simulation, prove themselves to have little practical usage in real buildings. In practice, each building and thermal zone is different and thus to study any building using these methods, a detailed modeling work has to be conducted for a specific building. Moreover, the parameters needed for model development usually cannot easily be gathered, such as building construction materials, orientation, size of each window, etc. Hardly can building managers

without the domain knowledge and access towards the building construction blueprint information be able to set up such a model. As a result, an alternative model with an easy configuration is needed.

With the advancement of IoT technologies, data collected by thermostats can be easily archived. Hence, it is now possible to utilize these data to build a model to represent a building. As most of the building parameters are fixed, including building construction materials, size of windows, orientation, etc., the influence from each of these parameters are of a certain fixed pattern, which can be learnt from historical temperature profiles of a building. In fact, learning building thermal characteristics from historical data is not unprecedented. Many studies have been conducted to accurately predict the indoor temperature of buildings. However, most of the current approaches are based on time-series data that the temperature changing pattern are learnt from historical profiles. To be able to do this, it is required that the temperature profile is accurate enough and this is the reason that most of these studies get the data either from simulation, or from a real world sophisticated sensor network.

Depending on a sophisticate sensor network, existing methods require additional hardware investments as well as installation expenditures. These extra expenses discourage building owners to retrofit their buildings. In fact, besides designated temperature sensors, smart thermostats themselves are able to measure the temperature and the cost is affordable. As a result, learning building thermal properties from thermostat data is a cost-effective alternative. Nonetheless, most affordable smart thermostats in the market can only measure indoor temperatures with 0.5 °F or 1.0 °F granularity; such a coarse-grained dataset is however inappropriate to be used in time-series models, which require higher resolution data.

In summary, a building thermal property model which is based on coarse-grained thermostat data are needed to depict the pattern of indoor temperature.

2.3.2. HVAC Units' Coordination Algorithm for Small- and Medium-sized Buildings

According to the literature research, the reasons that hinder most of the SMCB to participate in DR programs are:

a. Lack of a building retrofit solution that has high return on investment (ROI).
b. Too much detailed configuration burden on building managers for some existing model solutions.
c. Absence of an effective control algorithm targeting to optimize multiple AC units' operation in buildings during DR programs.

To tackle these challenges, a BEM system that has the following features is needed:

1. Considering the indoor thermal comfort.
2. Able to automatically generate an optimal control strategy efficiently.
3. Suitable for typical existing DR programs (demand limiting and CPP)
4. Easy to use, minimum configuration work needed from a building manager
5. Practical and cost-effective

However, among much research work, only one paper discusses about a system similar to the one proposed above, to the best of my knowledge. The authors propose a control algorithm for AC units coordination suitable for SMCBs. However, the algorithm proposed is not for a DR purpose and uses a time-series temperature model based on accurate and simulated temperature data. As a result, an algorithm utilizing coarse-grained thermostat data is still absent.

In summary, at present time, the study on a fast and optimal control framework (platform + algorithm) -- that are suitable for SMCB, can fulfill the DR requirements and simultaneously considering occupant comfort -- is not well-studied. In this dissertation, an automated control algorithm that will help achieve these goals will be studied.

3. BEM System Setup and Problem Formulation

In this chapter, the IoT-based BEM system used in this study is introduced; the problem of HVAC units' coordination during DR events is formulated and how the algorithms can be integrated to the BEM platform is exemplified.

3.1. BEM System for Small- and Medium-sized Commercial Buildings (SMCB)

As previously mentioned in Chapter 2, most SMCB does not have BEM system installed due to the limited options and high cost; nevertheless, the booming development of IoT-based BEM provides many cost-effective solutions to those buildings.

3.1.1. IoT-based Building Energy Management System

With the fast development in the Internet of Things (IoT) and the decrease of hardware cost, many IoT-based BEMs enabled by IoT devices are emerging as affordable solutions to building owners. These solutions are gaining popularity among many SMCBs for their low-cost, flexibility and scalability features.

In general, these BEMs are capable of providing the following three major services, which are essential for the automated building control, as shown in Figure 3-1:

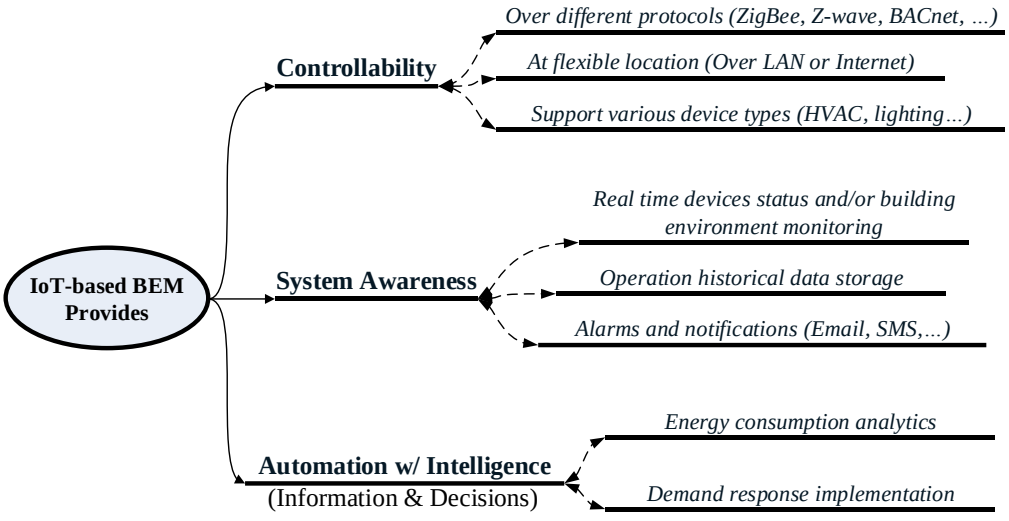


Figure 3-1 Benefits Provide by IoT-based BEM [140]

a. Controllability
   Device control is one fundamental function of a BEM system, building engineers are able to control various devices in building either locally via LAN or remotely via WAN; the automated control by the platform is also possible, such as scheduling or rule-based automated control.
b. System Awareness
   Without a BEM system, building engineers have no clue about how energy is consumed in the building. After installing one, devices' operating conditions are displayed in real time and historical data are archived for future reference or analysis, which increased the system awareness to a large extent.

c. Automation with intelligence
   With BEM as a platform, many previously isolated devices can interact together, based on either rules or other more complicated applications. A rule-based control example is: light sensor from Manufacturer A measures and reports light intensity in a room every 30 seconds interval and a close loop control can be implemented for a dimmable light from Company B. Optimal load control during a DR event is an example based on a more complicated application.

3.1.2. Building Energy Management Open Source Software (BEMOSS)

BEMOSS is an open source building energy management software based on the IoT technologies. Its target users are the SMCB which do not have traditional building energy management system installed. Funded by the U.S. Department of Energy, the development of this software started in 2014 and has ended in 2017. The latest release is its Version 3.5, which is available on GitHub [141].

BEMOSS offers the following features:

a. ***Interoperability:*** BEMOSS supports various different communication technologies including Ethernet, serial interface, ZigBee and Wi-Fi and is capable to communicate using different data exchange protocols, e.g., BACnet, Modbus, Web.
b. ***Ability for remote control and monitoring:*** building owners can remotely monitor the building operation and even change settings/control devices when away from the buildings.
c. ***Product neutral:*** BEMOSS has an open architecture with manufacturer agnostic feature, meaning any product is able to be integrated on BEMOSS as long as the manufacturer has provided an open application programming interface (API) for such product.
d. ***Plug & Play:*** BEMOSS has built-in discovery process that will automatically discover smart devices in the building, very little configuration is needed from the users.
e. ***Cost effective:*** BEMOSS is designed to be operated even on single board computers, including Odroid XU4 [142], Cubieboard4 [143] and Wandboard [144]. This further reduces the hardware investment for retrofitting building using BEMOSS.
f. ***Scalability:*** multiple single board computers can be used in a master-slave multi-node paradigm to form a BEMOSS cluster.
g. ***Platform for intelligent control algorithm:*** BEMOSS lays the groundwork for hosting intelligent algorithm as applications.

During the project, three buildings have installed BEMOSS for testing purpose and eventually demonstrate good performance. Therefore, the algorithms proposed later in this dissertation are designed as applications on BEMOSS platform.

Though BEMOSS is the platform used in this dissertation, the algorithms proposed in this study are also applicable for any other BEM system, with little or no modification.

3.2. Problem Formulation

3.2.1. Demand Response Programs

According to the examples of demand response programs from different utility companies given in Section 2.1.2, in general, every customer participated in such DR program, needs to reduce their load under certain limit when the event starts. This limit is enforced differently in different programs: in CBP program, a preset demand limit represents the maximum power this customer can consume, otherwise the customer

will be penalized; in CPP program, the capacity reserve also represent a safe limit, exceeding which will yield much higher electricity cost.

In the rest of this dissertation, there stands an assumption that the building which is controlled already participates in a DR program with a demand limit during the DR events. The goal for the controlling is to reduce the air-conditioning load below this limit.

3.2.2. Goal for the Proposed Smart Building Control

As mentioned earlier, during a DR event, the control of lightings and plug loads are straightforward: turn off all the interruptible loads until the end of the DR event. Assuming the power consumed by all uninterruptible lightings and plug loads is P_{fix} and the overall demand limit is P_{DR}, that means a control strategy is needed to limit all the AC units' power under P_{AC} at any time, thus there is:

$$P_{AC} = P_{DR} - P_{fix} \tag{3-1}$$

Single AC unit's power profile demonstrates a cyclic feature due to the dead-band control: when the room temperature is higher than the cool set point, the thermostat will control the unit to run and consumes high level of power; after the indoor temperature has reached the target, and AC unit will stop running and consumes little power. An example of a single unit's power consumption over 24 hours is shown in Figure 3-2.

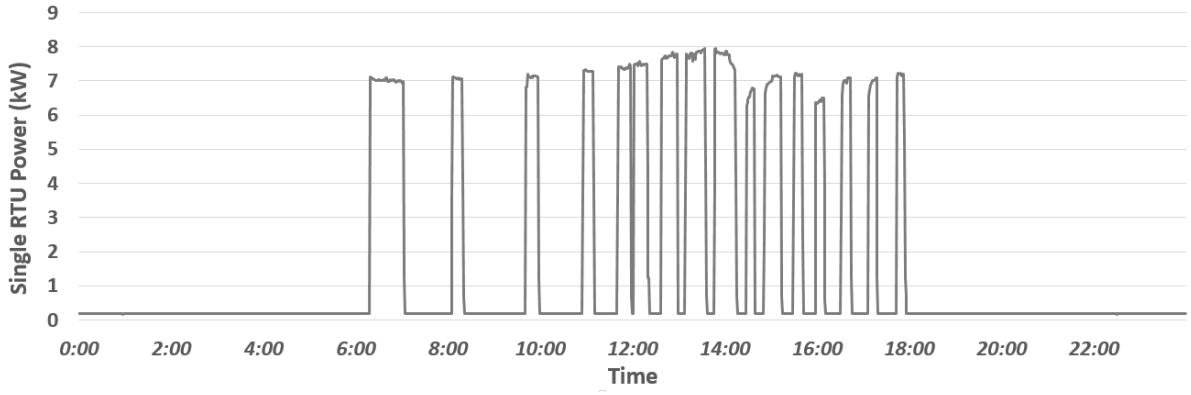


Figure 3-2 AC Unit Power Profile within 24 Hours

When there are multiple AC units present in a building, they need to be coordinated so that the power limit can be efficiently used and high power consumption due to units concurrent operating can be avoided. In the simplest case, where only two AC units with the same rated power are there in the building, they can be coordinated as run in turn: only one unit runs at any time. However, when there are many more AC units exist in the building, the problem become complicated.

In addition, limiting the power quota for all AC units in the building means they can no longer run as frequent as usual. This, inevitably, yields thermal discomfort to the occupants. Therefore, when solving the AC units' coordination problem, the indoor thermal comfort should be a vital.

Thus, the goal of this study is to develop an intelligent AC control algorithm so that when the DR event arrives, it is capable for coordinating multiple AC units in the building, guaranteeing the following three requirements:

a. The total AC unit power consumption is under the limit P_{AC}, as defined in (3-1);
b. The indoor environment in each thermal zone of the building is not intolerable (minimizing thermal discomfort);
c. (Optional) Using minimal amount of energy to achieve the goals.

It is worth noting that the Requirement c above is optional: when the DR program has capacity reserve, e.g., the CPP-D plan in Table 2-1, the CPP Price Adder will not be applied as long as the power consumption is under the reserved limit. In this case, a rational customer might not care about energy usage; on the other hand, if the program does not have capacity reserve, the user will consider energy saving since the critical peak price is extraordinarily high. In Chapter 6 and Chapter 7 below, both these two scenarios will be studied when designing the control algorithms.

In summary, during the DR period, the smart building controller will make sequential decisions upon which unit to turn on (equivalent to which thermal zone to cool) and which to be off at each time step. The aim is to minimize the overall occupants' thermal discomfort given limited power allowance. To realize this, a building thermal property model, an AC unit power consumption model and AC units coordinated control algorithms are needed.

3.2.3. Control Strategy

Based on the description above, the optimal AC control problem can be formulated in a way similar to power system unit commitment: a scheduling decision is made on the status of many generators in power system at each control interval in a period of time in the future, while meeting all security constraints and optimizing a target. Likewise, in this AC units coordination problem, the status of multiple AC units at each time interval in the demand response period are determined by the algorithm with consideration on occupants' thermal comfort and demand response power limit. Finally, before the start of DR event, a control strategy, in a form of status timetable, e.g., Table 3-1, is generated and sent to the BEM for execution. By executing this schedule, the building will fulfill the demand reduction goal and minimizing the discomfort for the occupants.

Table 3-1 Algorithm Generated Control Strategy Example

	T1	T2	T3	…	END
AC Unit 1	ON	OFF	OFF	…	ON
AC unit 2	OFF	OFF	ON	…	OFF
…	…	…	…	…	…

3.2.4. Algorithm Implementation

The proposed algorithms in this paper will be designed as an application of BEMOSS and implemented in Python, the same programming language BEMOSS is built with. As shown in Figure 3-3, the proposed algorithms will take BEMOSS historical data and building manager's configuration as input and generate an optimal control strategy for BEMOSS to execute.

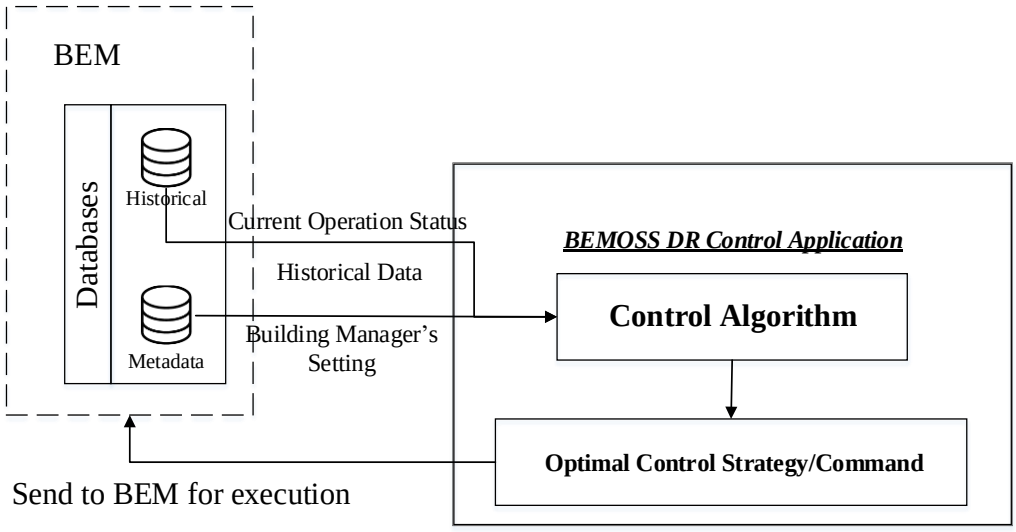


Figure 3-3 Relationship between BEMOSS and the Proposed Algorithms in This Dissertation

4. MODELING OF BUILDING THERMAL PROPERTY

In this chapter, a framework for learning building thermal property model from data collected by smart thermostats is proposed. Without using other sophisticated sensor network, this model provides a cost-effective approach to model the building thermal dynamics and makes it possible to predict indoor temperature based on certain inputs. In addition, the model can be easily utilized by the control algorithms discussed in later chapters.

The work in this chapter was originally proposed in [140], but it includes more extensive results.

4.1. General Concepts

In order to control AC units in a building, especially limiting their operation during a DR event, it is essential to have a good knowledge of each zone's thermal behavior. Namely, it is important to predict how indoor temperature changes given different AC status (ON/OFF).

According to study [145] and [146], (4-1) and (4-2) can be used to calculate the indoor temperature T_{i+1}, given the current indoor temperature T_i, AC status S_i and other parameters. All parameters with an overhead bar means they are constant parameters for a specific room, e.g., $\overline{V_{zone}}$ represents the volume of a thermal zone and its value doesn't change.

$$T_{i+1} = T_i + \Delta t \cdot \frac{G_i}{\overline{C_{air}} \cdot \overline{V_{zone}}} + \Delta t \cdot \frac{\overline{C_{HVAC}} \cdot S_i}{\overline{C_{air}} \cdot \overline{V_{zone}}} \tag{4-1}$$

G_i is the heat gain rate, it can be calculated using (4-2):

$$G_i = \left(\frac{\overline{A_{ceiling}}}{\overline{R_{ceiling}}} + \frac{\overline{A_{window}}}{\overline{R_{window}}} + \frac{\overline{A_{wall}}}{\overline{R_{wall}}}\right) \cdot (T_{out,i} - T_i) + \overline{SHGC} \cdot \overline{A_{south\_window}} \cdot H_{solar} \cdot \frac{3.412 Btu/Wh}{10.76\, ft^2/m^2} + H_p \tag{4-2}$$

In (4-1), Δt is the time interval in hour; $\overline{C_{HVAC}}$ is the cooling capacity, a negative number in Btu/h, $\overline{C_{air}}$ is heat capacity of air in the thermal zone (typical value is $0.0195\ Btu/(ft^3 \cdot °F)$), $\overline{V_{zone}}$ is the house volume in ft^3. In (4-2), the room's ceiling, window and wall have the area in ft^2 of $\overline{A_{ceiling}}$, $\overline{A_{window}}$ and $\overline{A_{wall}}$, additionally the window facing south has the area of $\overline{A_{south\_window}}$. Their heat resistance are $\overline{R_{ceiling}}$, $\overline{R_{window}}$ and $\overline{R_{wall}}$ (in $ft^2 \cdot °F \cdot h/Btu$) respectively. $T_{out,i}$ is the outdoor temperature at step i. $SHGC$ is the window's solar heat gain coefficient (no unit). H_{solar} is the solar radiation in W/m^2 and H_p describes the heat gain from occupants (Btu/h).

Apparently, the model shown in (4-1) and (4-2) reveals the physical thermal dynamics, it can be used to accurately model the building thermal dynamics and it has good performance in building simulation. However, in real life application, using this model puts a heavy burden on the building engineers during system commissioning. This is because the model parameters usually are either not readily accessible to a building engineers (especially for old/existing buildings) or additional sensors are required to capture such information (e.g., for solar radiation). Thus, (4-1) is hard to be configured and integrated to an indoor

temperature prediction model in practice. As a result, in this chapter, a Plug & Play and self-learning black-box model is developed to effectively model the building thermal dynamics.

With a simple manipulation on (4-1), a variable called temperature variation speed is defined in (4-3), with approximation:

$$\frac{dT_i}{dt} \approx \frac{T_{i+1} - T_i}{\Delta t} = \frac{G_i + \overline{C_{HVAC}} \cdot S_i}{\overline{C_{air}} \cdot \overline{V_{zone}}} = \begin{cases} \dfrac{G_i}{\overline{C_{air}} \cdot \overline{V_{zone}}} & if\ S_i = 0 \\ \dfrac{G_i - \left|\overline{C_{HVAC}}\right|}{\overline{C_{air}} \cdot \overline{V_{zone}}} & if\ S_i = 1 \end{cases} \quad (\forall i > 0) \tag{4-3}$$

From (4-3), it shows the temperature variation speed depends only on G_i and S_i. When Δt is small (less than an hour), the variables that determines G_i, namely $T_{out,i}$, T_i and H_{solar}, can be assumed to be constant over this short period. As a result, the temperature variation speed can be considered constant during short period. Empirical observations also substantiate this assumption: One example is illustrated in Figure 4-1, which shows the temperature measurements in a building under study vary linearly.

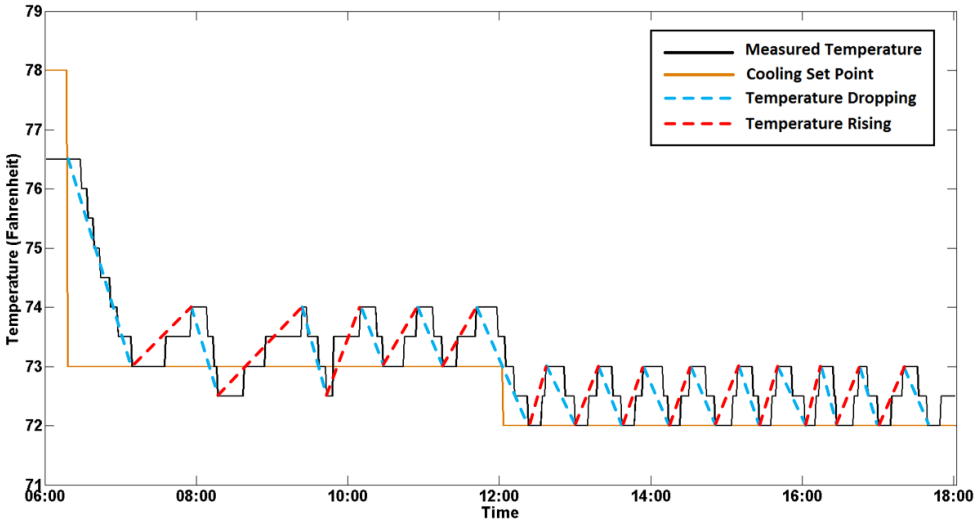


Figure 4-1 Historical Data of A Thermostat in A Building in Blacksburg, VA on July 7th, 2016

As a result, the indoor temperature of the next step can be calculated as a function of last step temperature and the temperature variation speed. For instance, how fast the room is cooled when AC is ON is determined by the temperature decreasing speed (TDS), and the room temperature increases with a temperature increasing speed (TIS):

$$T_{i+1} = \begin{cases} T_i + TIS_i \cdot \Delta t & if\ S_i = 0 \\ T_i - TDS_i \cdot \Delta t & if\ S_i = 1 \end{cases} \quad (\forall i > 0) \tag{4-4}$$

Therefore, the indoor temperature prediction problem is transformed into a temperature variation speed prediction problem: 1) predict the temperature variation speed (TIS/TDS) 2) predict indoor temperature based on TIS/TDS using (4-4).

Because (4-3) is not practical to be used in real world, an alternative is to find an approach to accurately predict TDS and TIS, using historical building operation data. In fact, in a building, most of the parameters, such as the architectural and constructional facts, typically remain constant. Therefore, their influences to indoor temperature variation are fixed and already implied in the historical data. As a result, for a specific building, the indoor temperature variation speed (TDS and TIS) can be modeled as only affected by some variable influencing factors (e.g., outdoor temperature). To model the relationship between these factors and the indoor temperature variation speed, a supervised learning problem is formed and will be discussed in a later section. Before that, the data collection and preparation are discussed.

4.2. Hardware Requirement and Data Preparation

4.2.1. Hardware Requirement

To collect the data for learning the building thermal property model, some types of hardware are needed. Traditional standalone thermostats which are still widely used today in many buildings and homes are not capable for gathering and archiving temperature data. In contrast, smart thermostats based on Wi-Fi, Z-Wave or other technologies are able to collect real time data and send them over LAN or WAN for further data storage. Such hardware upgrade is necessary and cost-effective: for instance, Radio Thermostat CT50, which is one of the thermostats supported by BEMOSS, costs $76.95 on Amazon.com at this writing time [147]. That is to say, less than $1,000 on thermostat upgrade is needed for most SMCBs, implying an easily achievable return on investment (ROI). In addition, there are many utility companies sponsored incentives and on-bill financing (OBF) plans available for the building owners when considering retrofitting as discussed in Section 2.1.5. To summarize, the upgrade on thermostats is needed and affordable.

For the purpose of reducing the cost for retrofitting, other hardware systems, such as the network of occupancy sensors, sophisticated temperature sensors and solar radiation sensors, are not considered. Therefore, the machine learning models in this chapter are developed with the consideration of the absence of these additional sensor networks, though data collected by them can improve the accuracy of the model predictions. Certainly, the proposed model can be easily modified to consider new data features.

4.2.2. Data Collection

An office suite, named as Suite 1, in an office building on Virginia Tech campus in Blacksburg, VA, USA has installed BEMOSS together with smart thermostats. The thermostat used in this suite is Radio Thermostat CT-50, which has a granularity of 0.5 °F and a control dead-band of 1 °F. Data used in this study are collected via BEMOSS from May, June, July, August and September in 2016. Thermostat is monitored by BEMOSS device agent every 20 seconds, when there is any change in the status, new status will be recorded in the Cassandra database, as exemplified in Table 4-1.

Table 4-1 Example Thermostat Data Collected by the BEMOSS System

Device ID	Date	Time	Set Point	Temperature	AC Mode	AC Status
1TH*******b149	2016-06-10	20:57:20-0400	73.0	73.5	COOL	OFF
1TH*******b149	2016-06-10	21:19:40-0400	73.0	74.0	COOL	COOL
1TH*******b149	2016-06-10	21:27:40-0400	73.0	73.5	COOL	COOL
1TH*******b149	2016-06-10	21:30:40-0400	73.0	73.0	COOL	OFF
…	…	…	…	…	…	…

Columns in the table are:

- Device ID: The MAC address of the thermostat being monitored.
- Date & Time: Local date and time, when the data is recorded.
- Set Point: Target temperature for the thermal zone in Fahrenheit.
- Temperature: Indoor temperature in Fahrenheit, measured by the smart thermostats.
- AC Mode: COOL/HEAT, indicating whether the AC is running on cooling or heating mode.
- AC Status: ON/OFF, indicating whether the compressor in the AC unit is actively cooling or not.

Weather information including outdoor temperature, outdoor humidity and weather conditions are also logged, using information from the Weather Underground [148] online service.

4.2.3. Data Preprocessing

There are two steps of data preprocessing before they can be ingested by the models:

a. Calculate temperature variation speed from temperature time series

The time series data collected by the smart thermostats contains the trends of temperature variation under different environmental conditions. Due to the cost and sensitivity, most of the thermostats available in the market do not have very accurate temperature sensors and thus give only discrete temperature measurement with granularity of 0.5 °F or 1 °F. To calculate the temperature variation speed (°F/second) using these coarse-grained indoor temperature data, one approach is to find out the slope of the blue (temperature decreasing speed) and red (temperature increasing speed) dash lines shown in Figure 4-2. The variation of these slopes can reflect the building thermal model.

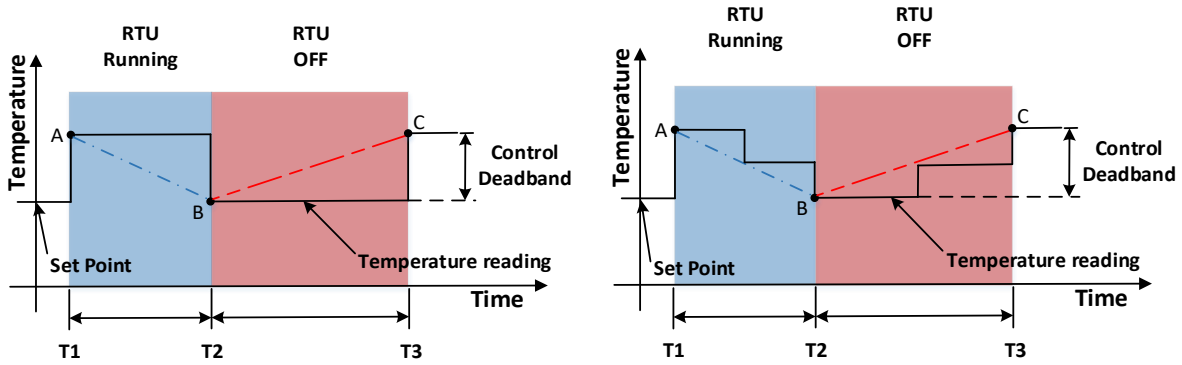
For temperature decreasing cases (e.g., T1~T2):

$$temperature\ decreasing\ speed(TDS) = \frac{control\ deadband}{T2 - T1} \tag{4-5}$$

For temperature increasing cases (e.g., T2~T3):

$$temperature\ increasing\ speed(TIS) = \frac{control\ deadband}{T3 - T2} \tag{4-6}$$

Since most thermostats have a control dead-band of 1°F, this implies that as for calculating the temperature variation speed, both thermostats with 1 °F granularity and 0.5 °F granularity have the same level of accuracy.



(a) 1 °F granularity (b) 0.5 °F granularity

Figure 4-2 Schematic Diagrams for Calculating the Indoor Temperature Variation Speed

b. Clean the data, discard inappropriate data points.

When calculating the temperature variation speed, some results are discarded. For example, if the temperature variation speed is calculated to be larger than 0.05 °F/s, this bad data is discarded: the indoor temperature changes 1 °F within 20 seconds is less likely to happen than a flickering of the thermostat temperature sensor.

4.3. Feature Selection

In this section, the factors that have impact on indoor temperature variation speed is discussed. Previous studies demonstrate that the indoor temperature is mainly influenced by heat gain from the outdoor environment and indoor activities. Therefore, both the outdoor and indoor factors are discussed below. However, some variables (e.g., solar radiation) requires additional dedicated sensors, which makes system more complicate and more expensive to build. In this case, these variables will not be used directly, but approximated inexplicitly using some easily accessible variables.

4.3.1. Outdoor Environments:

Outdoor environments, such as temperature, humidity and direct sunshine, have direct impacts on how soon a zone is cooled or how fast the zone's temperature increases due to heat gain, etc. Outdoor temperature and humidity information are readily available via online sources. Determining the heat gain from direct sunshine without the sensors is not straightforward. However, the direct sunlight depends upon weather condition (sunny/cloudy/rainy), time of the day and building orientation: for example, a room facing west has direct sun radiance in a sunny afternoon. Since the position and orientation of a room is fixed all the time, the solar radiation can be approximated using the weather conditions and time of the day.

Among many available online sources about weather information, this study utilizes the historical and forecasted weather information from Weather Underground (WU) [148], due to its convenient open application programming interface (API). Some typical summer weather conditions used by this service are categorized in Table 4-2 according to whether there might be direct sunlight presents.

Table 4-2 Categories of Weather Conditions from Weather Underground

Category	Description	Weather Conditions from WU
Class 1	Will have direct sunlight into room during some time frame	Clear
Class 2	Might have direct sunlight into room during some time frame	Scattered Clouds, Partly Cloudy, Mostly Cloudy
Class 3	No direct sunlight	Overcast, Light Drizzle, Drizzle, Heavy Drizzle, Light Rain, Rain, Heavy Rain, Mist, Fog, Haze
Class 4	Invalid value	Unknown

4.3.2. Indoor Activity

Occupant indoor activities contribute to internal heat gain which has a crucial impact on how fast the zone temperature increases or decreases. With more occupants in the room, there are more heat emission from human bodies, their appliances such as PC and their drinks like coffee. Indoor activities can be measured using information from occupancy sensors (e.g., occupancy status) and plug loads (e.g., appliance usage status). However, for a more general case where there exists neither occupancy sensor nor smart plug, indoor activities level can be approximated using day of week and time of day, especially in buildings with regular and specific schedule/occupancy pattern.

4.3.3. Machine Learning Model Formation

With all the above mentioned features considered, an example of the preprocessed data is shown in Table 4-3.

Table 4-3 Example of Preprocessed Data for Learning the Building Thermal Model

$\frac{dT_i}{dt}$ (°F /second)	T_i (°F)	T_{out} (°F)	*time*	*dow*	H_{out} (%)	*w*
9.009*10-4	73.0	72.1	11	1	73.0	2
7.936*10-4	73.0	72.5	12	1	69.0	3
…	…	…	…	…	…	…

The goal is to train a machine learning model that can model a relationship shown by (4-7):

$$\frac{dT_i}{dt} = f(T_i, T_{out}, time, dow, H_{out}, w) \tag{4-7}$$

In (4-7), T_i and T_{out} represent indoor and outdoor temperatures in Fahrenheit, respectively; and H_{out} is outdoor humidity in percentage. These are numeric variables that can be used directly. *time* represents hour of day in the 24 hour format; similarly, *dow* represents day of week with day=1~7 meaning Monday to Sunday; and w is the weather class number (Class 1~4 as shown in Table 2).

Before using these features directly, feature engineering is needed:

a) Continuous features:
Features with continuous values (T_i, T_{out} and H_{out}) remain the same.

b) Cyclic ordinal features:
Time and day of week are ordinal features (e.g., 16:00 is later than 12:00), however, the relationship is also cyclic: (01:00 is closer to 23:00 than 12:00). According to [149], a common practice to consider such features is a trigonometric transformation as shown in (4-8) and (4-9).

$$time1, time2 = -\cos(\frac{time}{24} \cdot 2\pi), \sin(\frac{time}{24} \cdot 2\pi) \tag{4-8}$$

$$dow1, dow2 = -\cos(\frac{dow}{7} 2\pi), \sin(\frac{dow}{7} 2\pi) \tag{4-9}$$

c) Categorical features:
The weather feature is a categorical feature, dummy variable encoding will be used to preprocess the feature before put into the learning model.

Table 4-4 shows the same entries of data in Table 4-3, but they are after the feature transformation and encoding. According to this table, the number of feature column is ten. For simplicity, these ten features are denoted as $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}]$ in the following sections, and the temperature variation speed is denoted by y.

Table 4-4 Example of Final Machine Learning Model Inputs (Target & Features)

Target	***Features***									
y	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
9.009*10-4	73.0	72.1	73.0	0.966	0.259	-0.623	0.782	0.0	1.0	0.0
7.936*10-4	73.0	72.5	69.0	1.0	0.0	-0.623	0.782	1.0	0.0	0.0
…	…	…	…	…	…	…	…	…	…	…

Finally, it is worth noting that the temperature increasing and temperature decreasing are two separate dynamic processes. As a result, it is necessary to have two different machine learning models to represent the relationships: for the same features, the temperature increasing speed predictor should output a positive y value while the temperature decreasing speed predictor should output a negative one.

4.4. Machine Learning Model Choices

Five commonly used and good-performing supervised learning models are presented in this section and will be trained for predicting temperature variation speed. The result from these five models will be analyzed and compared. These five models are polynomial regression (PR), support vector regression (SVR), random forest (RF), extreme gradient boosting (XGB), and the multi-layer perceptron (MLP). Four of them are implemented using the Python scikit-learn library, and the gradient boosting method is implemented using the XGBoost library.

4.4.1. Polynomial Regression (PR) Model

Polynomial regression model is simple and has very good interpretability. For example, the 1-order polynomial regression model is shown as (4-10). μ is the bias term.

$$y = \mathbf{x} \cdot \boldsymbol{\theta} + \mu \tag{4-10}$$

However, the relationship between these features and the temperature variation speed might be non-linear, thus, higher order model should also be tested. In these higher order models, the interaction between features are considered. In addition, Lasso regularization is used for two reasons:

1. According to (4-2) and (4-3), the indoor temperature has linear relationship with its variation rate. When using the higher order model with regularization, the higher order term regarding the indoor temperature will have a coefficient of zero;
2. In general, regularization can help avoid overfitting, especially when there are many interaction terms and only some of them have impact on the target values.

With the Lasso regularization, the objective function for minimizing the fitting loss is shown as below:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f_{pr}(\mathbf{x}^i; \deg; \boldsymbol{\theta}) - y^i)^2 + \alpha \left\| \boldsymbol{\theta} \right\|_1 \tag{4-11}$$

N is the number of training data instances, and $f_{pr}(\mathbf{x};\deg;\boldsymbol{\theta})$ is the polynomial regression model.

There are two hyper-parameters to be determined as shown in Figure 4-3: degree(deg) and Lasso penalty multiplier α. Degree defines the complexity of the model; higher degree model performs better than linear model if the relationship to be learnt is non-linear. The Lasso penalty multiplier controls the model complexity as well; it fights the model overfitting. With a higher multiplier value, the model is less prone to overfitting. In addition, unlike L2 regularization, Lasso regularization can bring coefficients of those trivial features to zero, which has the effect of feature selection.

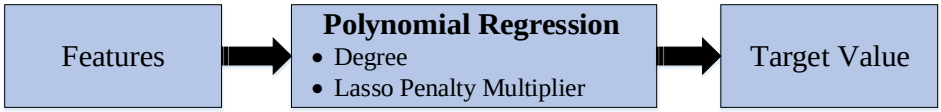


Figure 4-3 Polynomial Regression Model's Hyper-parameters

4.4.2. Support Vector Regression (SVR) Model

Support Vector Regression (SVR) are considered to be one of the best learners and are widely used in many domains. It has the following merits:

- Good mathematical interpretation
- Global optimum guaranteed by solving a quadratic optimization problem
- Easily handle non-linear problems by using the kernel tricks

The mathematics behind the support vector regression is a margin maximization problem, which can be transform into a dual problem to solve, detail of it can be found in the Chapter 12 of [150] and will not be discussed here. One advantage of SVR is: by using the kernel trick, SVR has a great capability dealing with non-linearity. A kernel function can help mapping the non-linear data to a much higher dimension (where data points are sparser), then use a soft margin to divide the more 'linear separable' data points.

For the SVR model, there are four major hyper-parameters to be determined prior to the model learning, which also determine the model performance, as shown in Figure 4-4.

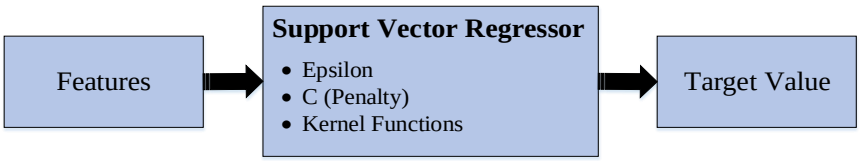


Figure 4-4 Support Vector Regressor's Hyper-parameters

- Epsilon (ε)
  Epsilon determines a loss free band, its value should be properly set to correspond to the scale of output y's value. The value of Epsilon affects the smoothness of the trained model and number of support vectors.
- Penalty Factor C
  The penalty factor C penalizes those instances out of the loss free band. It is a regularization term, which balances the tradeoff between training error and model complexity [151].
- Kernel Functions

In this study, two popular Kernel functions will be used:
Polynomials kernel functions:

$$K(x^t, x) = (\gamma \cdot x^T x^t + r)^d \tag{4-12}$$

Radial-basis functions (RBF):

$$K(x^t, x) = e^{-\frac{\left\|x^t - x\right\|^2}{2s^2}} = e^{-\gamma\left\|x^t - x\right\|^2} \tag{4-13}$$

The parameters of a Kernel function will determine its performance. For example, s in the radial-basis functions defines the radius, when s is too small, namely Gamma is too large, and the problem of overfitting is more likely to occur.

4.4.3. Random Forest (RF) Model

Random forest is an ensemble learning model taking advantage of **B**ootstrap and **agg**regat**ing**, Bagging in short. The output of the model is the output aggregation of many weak learners. This is due to the philosophy behind these ensemble methods: it is easy to build an above-average learner but hard to build a top learner, however, the aggregation of many above-average leaners can yield results as accurate as those from a top learner. As a result, above-average learner such as a decision tree, featuring with low bias and high variance, can be aggregated to form random forest so that higher output accuracy can be achieved. Detail of random forest model and its theory can be found in the Chapter 15 of [150].

In this study, three hyper-parameters are used to determine the random forest regressor, as shown in Figure 4-5:

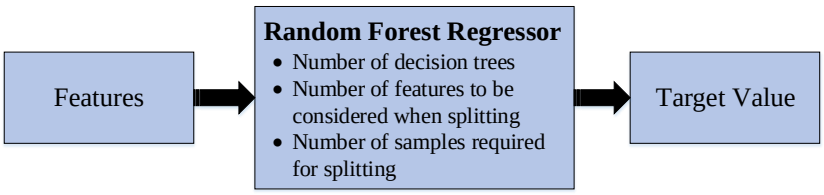


Figure 4-5 Random Forest Regressor's Hyper-parameters

- Number of decision trees in the forest
  Generally speaking, the increase of decision trees' number in the forest will not cause an overfitting issue, therefore, the more the better. Nevertheless, more trees imply longer computation time and the results will not be significantly better after a certain threshold.
- Number of features to be considered when splitting
  When the problem has a large number of correlated features, it is usually helpful to consider a subset of all the features in each tree: this is also known as de-correlating the trees. Because of this, this hyper-parameter defines the number of randomly selected features. Two common choices are: using all N features and randomly choose $\sqrt{N}$ features.
- Number of samples required for splitting
  To prevent overfitting, the number of samples needed for splitting is defined. Generally, the smaller this value is, the model is more prone to overfitting.

4.4.4. Extreme Gradient Boosting (XGB) Model

The philosophy of boosting models is very similar to that of bagging models: aggregate the result of multiple weak learning to form a strong learner. The difference lies in the way of aggregation: bagging models aggregate weak learner in a parallel way while boosting model do this sequentially. Following learners learn the prediction error from the previous learner, and together all learners make an additive model. A popular library called XGBoost is used to implement the gradient boosting tree algorithm. There are four hyper-parameters for the model:

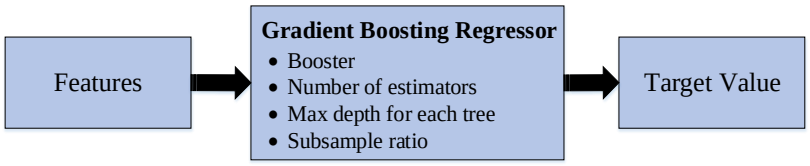


Figure 4-6 Gradient Boosting Regressor's Hyper-parameters

Hyper-parameters:

- Booster: gradient boosting tree or DART (Dropouts meet multiple Additive Regression Trees)
- Number of estimators: Number of boosted trees to fit. Unlike the random forest model, the increase of estimator number in gradient boosting trees will cause an increase of the overfitting probability
- Subsample: Typically it is a float between 0 and 1, representing the subsample ratio of training data used to grow the tree.
- Max_depth: Maximum tree depth of the base learner.

4.4.5. Neural Network Multi-Layer Perceptron (MLP) Model

According to [152], the definition of neural network is:

'Artificial neural networks are massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with the objects of the real world in the same way as biological nervous systems do'

The 'simple elements' in the definition above are the basic building bricks of the neural network: the artificial neurons. As early as 1943, [153] proposed a 'M-P neuron model' which is widely used today, this model abstract the neuron as the following mathematical representation:

$$y = f(\sum_{i=1}^{n} w_i x_i - \theta) \tag{4-14}$$

y is the output of the neuron and $(x_1, x_2, \cdots, x_n)$ are the n inputs of the neuron. The function $f(\mathrm{x})$ is called activation function.

With the interconnection of many such simple elements comes the neural network. Multi-layer perceptron (MLP) is one class of the neural network. In general, the structure of a multilayer perceptron determines the model performance. Its structure is defined by the following hyper-parameters, as shown in Figure 4-7.

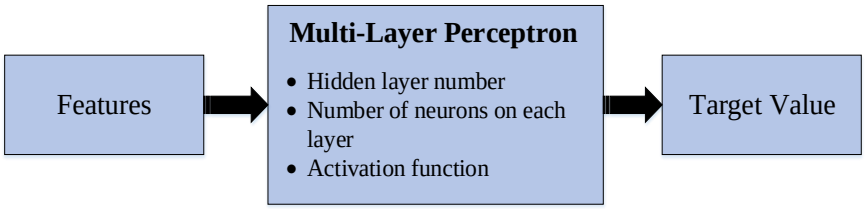


Figure 4-7 Multi-layer Perceptron's Hyper-parameters

- Number of hidden layers and neurons on each layer
  Number of hidden layers and neurons on each layer determines the structure of the multilayer perceptron: with the increase of these numbers the network become more complicated and can represents more complex relationship. But the structure of the neural network should be corresponding to the actual problem at hand: a model with too simple structure cannot map the correct relationship while one with too complex structure will cause problem known as overfitting and thus decrease the prediction accuracy. So far, there is no decisive method for determining the best number of hidden layers as well as the neurons on each layer.
- Type of activation functions
  Activation function is needed in each of the neuron, it will introduce non-linear feature; without the activation function, the neural network can only output the linear combination of the inputs. Sigmoid function is one of the commonly used activation function. Other activation functions such as the ReLU (Rectifier Linear Unit) are also popular since they are less prone to the gradient vanishing issue in the deep learning problems.

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{4-15}$$

$$relu(x) = \max(0, x) \tag{4-16}$$

4.5. Models Selection and Evaluation

Not only there are different options on the supervised learning model, but also the model hyper-parameters can be various. To find out the supervised learning model that suits the best for the problem under investigation, a process of model selection is needed in the first place, via cross validation and grid search.

4.5.1. Cross-validation (CV)

Cross-validation is an effective approach to assess how well a statistical analysis will generalize given different training and testing data. There are many types of cross-validation, the one will be used in this study is *k-fold cross-validation*. To do this, first, the original data set is randomly split into k sub data sets with similar number of samples; then, for k times, each time one of the sub data sets is chosen as testing data set while the rest k-1 sub data sets are used for training, and the inference error measurement index (e.g., mean squared error, mean absolute error) is calculated; next the error measurement indices of the k train/test cases are averaged and this average value is used as the performance indicator for the model being studied. In practice, the splitting number k is 5 or 10 typically. In this study, if the data set is large (more than 300 samples in the original data set), k equals 10 will be used; otherwise, the data is split into 5 parts. Also, mean squared error is used for error measurement in this study.

4.5.2. Grid Search

Grid search is one of the hyper-parameter optimization approaches, also known as parameter sweep. It conducts an exhaustive searching to find the optimal hyper-parameters setting that minimizes a predefined performance metric. For instance, the hyper-parameters for polynomial regression includes $degree \in \{1,2,3\}$ and $\alpha \in \{0.0001,\ 0.001,\ 0.01,\ 0.05\}$ as shown in Table 4-6, then the Cartesian product of $H = degree \otimes \alpha$ contains all the hyper-parameters setting grid search needs to train the model with. For each combination, the cross-validation is conducted and the mean squared error is calculated. Finally, the best hyper-parameters set can be found, namely the hyper-parameters combination that leads to the least mean squared error.

4.5.3. Procedure for Model Selection, Training and Testing

According to Section 4.2.2, building thermal data are collected from May to early September, they are divided into four training/testing data set, as shown in Figure 4-8: Data from the 1st to the 31st of each month are used for training while the first 100 data for the next month is used for testing. The reason for selecting consecutive training and testing data set is based on an assumption: the approximation relationship between solar radiation and time/weather stays constant only for recent period. For instance, the solar radiation at 2:00 pm in a May sunny day can be considered similar with the solar radiation in an early June sunny afternoon; but it might be different from the solar radiation in Mid-August, with the same time and weather type. Same assumption goes with the occupancy approximation. Similarly, later in production environment, the model is likely to use the past 30-days of data for training and then use the model to make predictions in the next a few days.

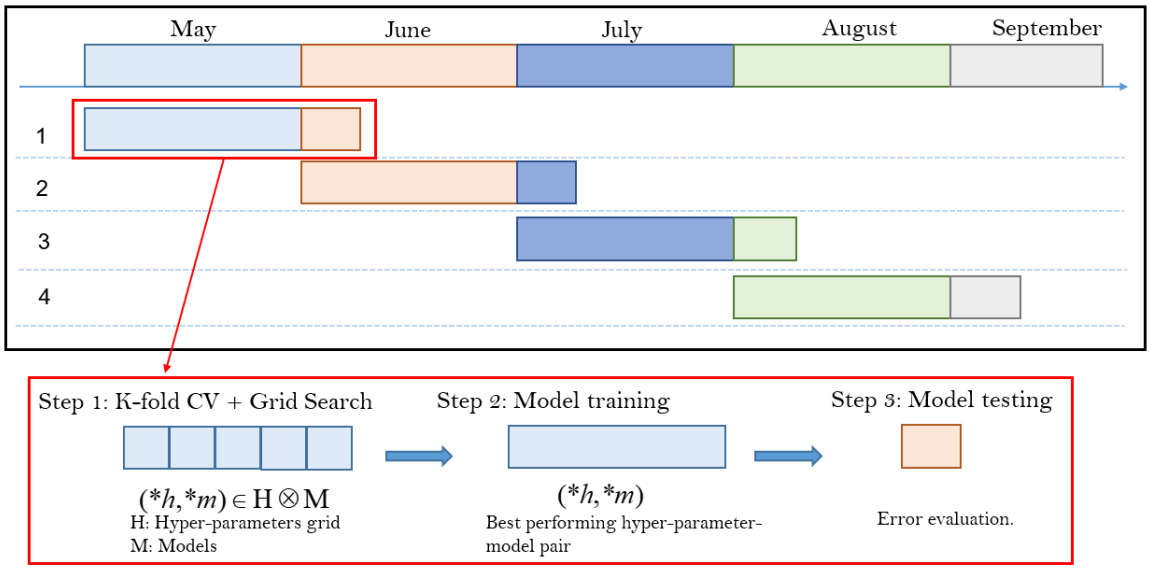


Figure 4-8 Data Sets and Procedure for Model Selection, Training and Testing

In this study, for each training/testing data set, the following steps are taken to evaluate different machine learning models:

Step 1: Conducting K-fold cross validation using the training data on each model with every possible hyper-parameters set ($H \otimes M$).Finding those models with the best performance, namely those with the least mean squared error.

Step 2: Training the best performing estimator selected from Step 1 with the training data.

Step 3: Evaluate the trained model using the testing data.

After these three steps, the best performers are identified for four months and their performance metrics are calculated for comparison.

4.6. Case Study

In the case study, two sections are discussed:

In the first section, the correlations between the temperature variation speed and the proposed six influencing factors are explored, validating these are effective factors to predict temperature variation speed. In the second section, the results for predicting indoor temperature increasing/decreasing speed are analyzed and discussed from three perspectives:

1. Which one is the best performers among the five machine learning models?
2. What is the prediction error for the best estimator?
3. How will error accumulate for multi-step indoor temperature prediction?
4. Is this error level acceptable for the HVAC control during a DR event?

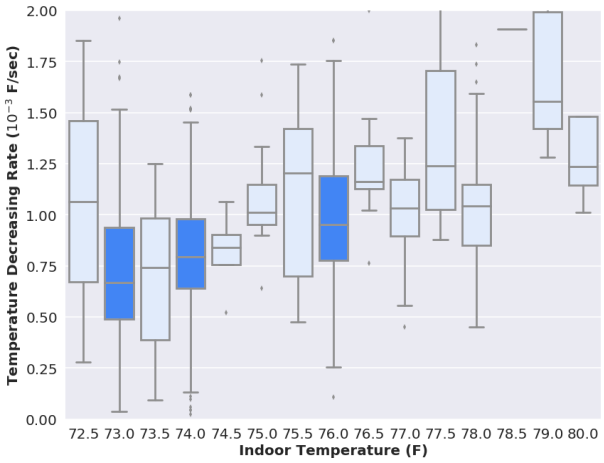
4.6.1. Validation of Predicting Features

In Section 4.3, six features, related to outdoor meteorological condition and indoor occupants' activity are discussed to predict the temperature variation speed. In this section, all data from four months (May to August, 2016) are used to show how each of these factors impact the indoor temperature variation speed.

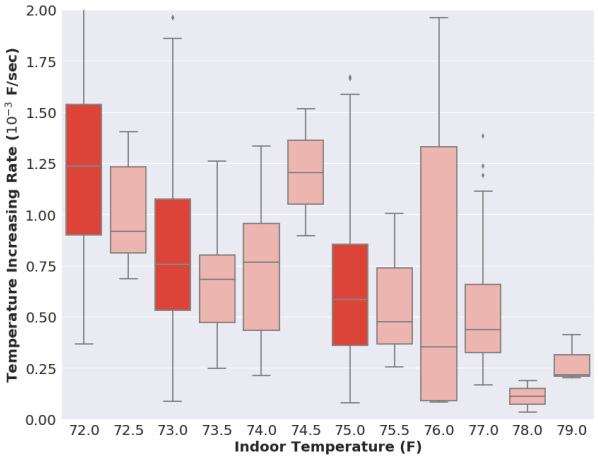
To study the correlation, boxplot is used to demonstrate how the temperature variation speeds changes with the variation of the feature. The relationships are plot in a 2-D coordinate, with X-axis represents possible values for the feature and Y-axis represents the temperature variation speed. The reason to use the boxplot is because the temperature variation speed value is a function of many other features as well, when consider only one feature, the value is better represented using a distribution. Therefore, at each possible value, a quartile box is used to show the distribution of temperature variation speed under this feature value. However, if the number of data points used to create the quartile box is small, the demonstrated distribution is not reliable (Not enough sample to conclude the statistics). As a result, for better illustration in the figures below, quartile boxes are shown for all feature values, but if the number of a specific feature value's data point is less than half the maximum number of feature value's data points, this quartile box is represented in diluted color, otherwise in a more vivid color, indicating the result shows by this quartile box is more reliable.

The correlations are illustrated from Figure 4-9 to Figure 4-14:

1) Indoor temperature vs. Temperature variation speed
   According to Figure 4-9, most data points are aggregated around three indoor temperature values, for both TDS and TIS cases. This is because the BEM system sets the building operation schedule which makes the cool set point of 72°F, 73°F and 75°F in different time of a day. So other values' quartile boxes are less reliable, but are presented below for reference. According to the trend depict by these three reliable quartile box, in general, with the increase of the indoor temperature, the TDS increases and the TIS decreases.



(a) TDS Cases

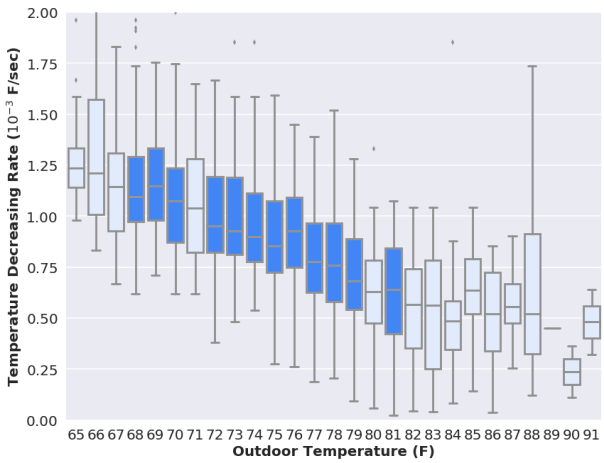


(b) TIS Cases

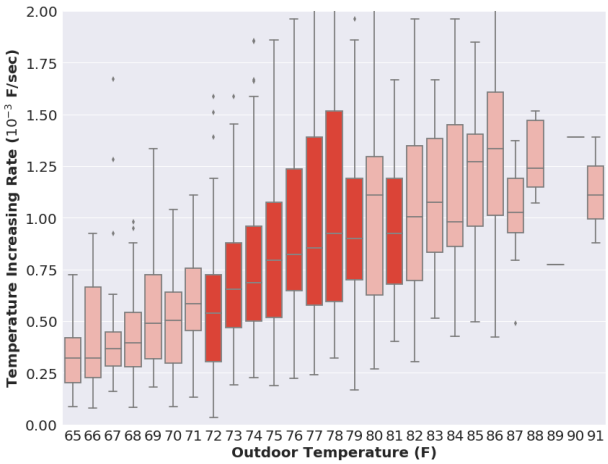
Figure 4-9 Indoor Temperature vs. Temperature Variation Speed

2) Outdoor temperature vs. Temperature variation speed

According to Figure 4-10, monotonically, with the increase of the outdoor temperature, the TDS decreases and the TIS increases. The reason of this is because the hotter outdoor environment, the higher the external heat gain, which makes the room harder to cool and easier to heat up.



(a) TDS Cases

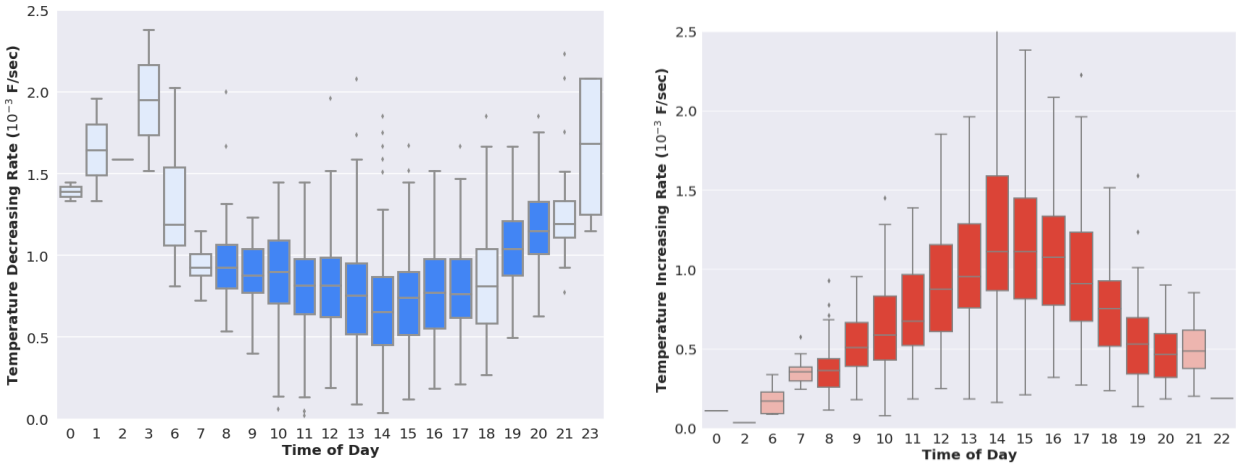


(b) TIS Cases

Figure 4-10 Outdoor Temperature vs. Temperature Variation Speed

3) Time of Day vs. Temperature variation speed

Figure 4-11 reveals that the TDS and TIS have a non-linear relationship between the time of day. In addition, the variance of TIS is much higher between 10:00 to 18:00 than other hours. This might because during business hour of this office, the randomness of the occupants' activity is high. For TDS case, because TDS is more dominated by the HVAC cooling capacity, the variance difference between business and non-business hours are not as significant.

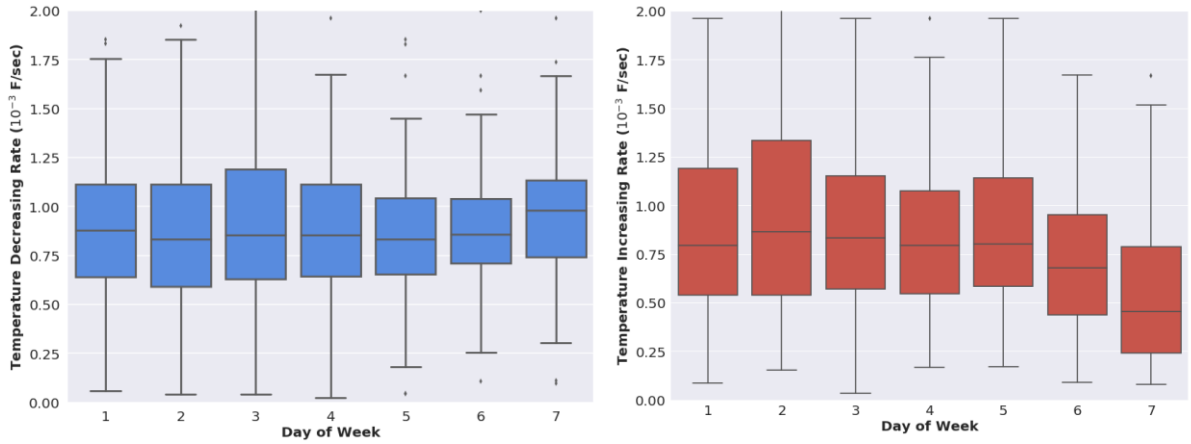


(a) TDS Cases (b) TIS Cases

Figure 4-11 Time of Day vs. Temperature Variation Speed

4) Day of week vs. Temperature variation speed

Figure 4-12 shows the influence of daily schedule on TDS and TIS: no significant difference among workdays, but the difference between workdays and weekend days are apparent. Because of the low internal heat gain result from a less occupancy level, this office can be cooled faster and takes longer to heat up during the weekends.

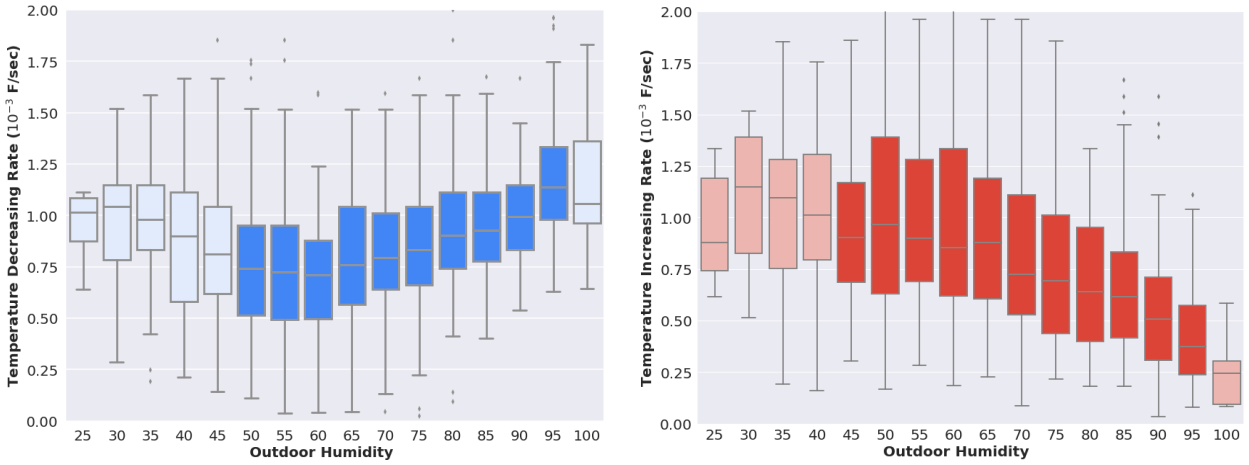


(a) TDS Cases (b) TIS Cases

Figure 4-12 Day of Week vs. Temperature Variation Speed

5) Outdoor humidity vs. Temperature variation speed

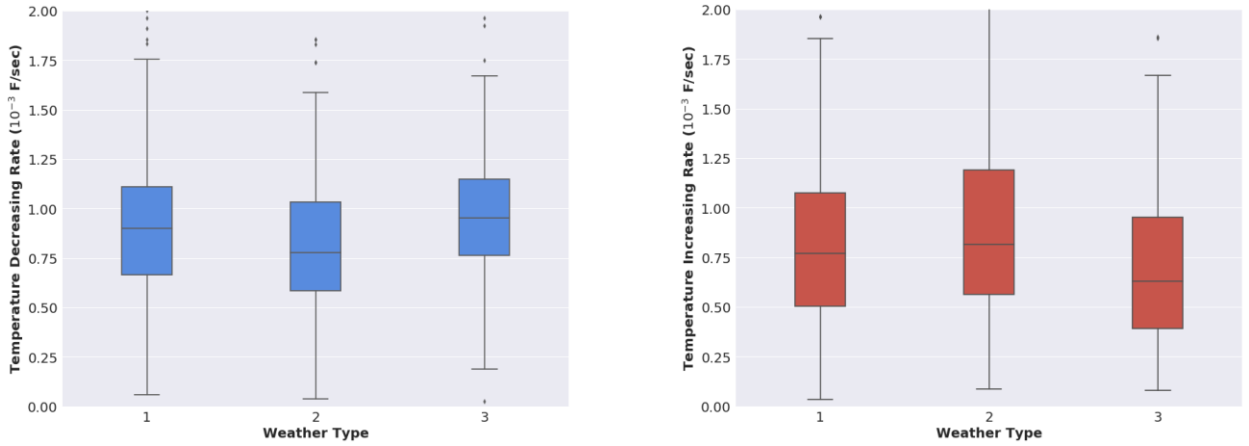
According to Figure 4-13, TDS increases and TIS decreases with the increase of the outdoor humidity.



(a) TDS Cases (b) TIS Cases

Figure 4-13 Outdoor Humidity vs. Temperature Variation Speed

6) Weather type vs. Temperature variation speed
Figure 4-14 shows how weather types (as shown in Table 4-2) will impact TDS and TIS. Apparently, during rainy days (class 3), TIS is lower and TDS is higher due to the lower heat gain. In addition, the figure shows that the external heat gain in a Class 1 weather day (clear sky) is lower than that of a Class 2 weather day: this is because of the additional solar heat gain caused by the irradiance reflection from the clouds in the cloudy (scattered or partly cloudy, not overcast) days.



(a) TDS Cases (b) TIS Cases

Figure 4-14 Weather Condition vs. Temperature Variation Speed

To sum up, the figures above show the influences of each factors, demonstrating how they impact the temperature variation speed. The shown correlations demonstrate the validity for using these features to infer the temperature variation speed.

In addition, feature importance is studied using random forest: a feature is more important if it is used as a splitting variable more frequently in the random forest estimator. The results are shown in Table 4-5: compared with other features, the day and weather play less roles in the temperature variation speed prediction, for the Suite 1 under investigation.

Table 4-5 Feature Importance for the Temperature Variation Speed Prediction Model

Feature	Mode	Indoor Temp	Outdoor Temp	Outdoor Hum	Time	Day	Weather
Feature Importance	TIS	19.35 %	30.71 %	17.74 %	20.21 %	9.09 %	2.90%
	TDS	17.13 %	40.37 %	13.44 %	18.56 %	7.61%	2.89%

4.6.2. Model Selection through Cross Validation

4.6.2.1 Model Hyper-parameters

Among all the five types of model proposed above, each of them can have different hyper-parameters. After initial testing, some possible hyper-parameters are found for each model, as shown in Table 4-6. As a result, there are plentiful ways for setting the hyper-parameters for a given model. To find out the best performer among many candidate models, the performance of these models will be evaluated using cross-validation.

Table 4-6 Choices of Hyper-parameters for Each Type of Model

Model	Hyper-parameters	Total Combination
PR	degree: {1,2,3}, α :{0.0001, 0.001, 0.01, 0.05}	12
SVR	kernel: {'rbf', 'poly'}, gamma: {1e-1, 1e-2, 1e-3, 1e-4}, C: {1, 10, 100}, ε : {0.05, 0.1, 0.2}	72
RF	n_estimator: {50, 100, 200}, max_features: {'auto', 'sqrt'}, min_samples_split: {4, 8, 12}	18
XGB	booster: {'gbtree', 'dart', 'gblinear'}, n_estimators: {50, 100, 200, 500}, max_depth: {2, 4, 6, 8}, subsample: {0.6, 0.9}	96
MLP	hidden_layer_sizes: {10, 8, (8,4), (6,3), (6,6,6)}, activation: {'relu', 'logistic'}, early_stopping: {True, False}, alpha: {1e-4, 1e-3, 1e-2}	60

4.6.2.2 Estimator Comparison in Cross Validation

Among all models with all possible hyper-parameters sets, the top five estimators for each training data set, namely the five models with the least average mean square error from the cross validation process, are listed in Table 4-7. For instance, five polynomial regression models with different hyper-parameters perform the best in the cross validation using training data from May. In addition, as pointed out in Section 4.3.3, the forecast of temperature increasing speed (TIS) and temperature decreasing cases (TDS) are learnt separately, and the results are shown separately in Table 4-7.

Table 4-7 Top 5 Estimators for Each Month and Both Temperature Increasing/Decreasing Cases

Training Month	TDS Forecast		TIS Forecast	
	Number of Samples	Top Performers	Number of Samples	Top Performers
May	166	P/P/P/P/P*	154	P/P/P/P/P
June	417	S/X/X/S/X	396	X/X/S/X/X
July	280	S/S/P/P/P	260	S/S/S/P/P
August	392	S/S/S/S/S	361	R/R/P/S/R

* P-Polynomial Regression, S-SVR, X-XGBoost, R-Random Forest

From Table 4-7, a few observations can be drawn:

a) For both TIS and TDS forecast in each month, the top five estimators' model types are very consistent: there is always one class of the model dominating the top five;

b) The best estimator for different month can be different: Polynomial regression models perform good in May, XGBoost models have the best result in June cross validation and SVR models make the best prediction in TDS forecast in August;
c) Polynomial regression models perform better when the size of the training data is comparatively smaller (as in May and July cross validation);
d) In general, SVR models have good performance in many cases, can be regarded as the most stable performing estimator among all 5 tested models.
e) Neural network models do not perform as good as other models, none of the top five performers is neural network.

Based on these observations and additional verification, there gives the following two conclusions:

Conclusion 1: *SVR is the best performer among many different models, and the RBF kernel is a better choice than the polynomials kernel.*

According to Table 4-7, SVR models appear the most times in the top five estimators in four months' cross validation and all of these top performing SVR models used the RBF kernel. This indicates the RBF kernel SVR can better capture the non-linear relationship between the features and the temperature variation speed. To further prove the capability of SVR on this problem, the following comparison is made: For each model type, five models, which differs in hyper-parameters among each other, with the least root mean square error (RMSE) in the cross validation are identified. Table 4-8 shows the average RMSE for the five best performing models of each kind. For each month, the least average RMSE are shown in highlighted blue box and the second least average RMSE are in highlighted green box.

Table 4-8 Average RMSE of the Top 5 Estimators of Each Model

Month	Prediction Type	Average RMSE of the top 5 estimators of each model (10^{-3})				
		SVR	RF	NN	XGB	PR
May	TIS	0.305261	0.300336	0.358987	0.303965	0.291227
	TDS	0.341709	0.36598	0.430454	0.361509	0.335342
June	TIS	0.273624	0.270667	0.302366	0.268249	0.274143
	TDS	0.220955	0.228032	0.247156	0.220279	0.224466
July	TIS	0.317055	0.336887	0.396083	0.33794	0.319847
	TDS	0.292022	0.295334	0.312906	0.298664	0.292453
August	TIS	0.592435	0.587878	0.634368	0.639223	0.597828
	TDS	0.313402	0.32506	0.33874	0.322438	0.317172

In Table 4-8, the SVR models have the Top 1 or Top 2 performance among most of the cases. According to this, SVR has the best and the most stable performance in this specific predicting problem. The reasons for this can be: 1. SVR has better capability in describing the underlying non-linear relationship; 2. it is easier to find the hyper-parameters for a SVR model that has the best performance in this problem.

Conclusion 2: For smaller training data set, polynomial regression models can generalize better on unseen data than other models.

According to both Table 4-7 and Table 4-8, the polynomial regression models also perform very well, especially in May and July. One possible reason for this is during these two months, the data set size is smaller than other months'. Because of its complexity, the SVR models are prone to overfitting using

smaller dataset; in contrast, the PR models are less likely for overfitting and thus they outcompete the SVR models.

Finally, the best performing estimators and their hyper-parameters identified in the cross validation are summarized in Table 4-9. These estimators will be evaluated using the testing data in the next section.

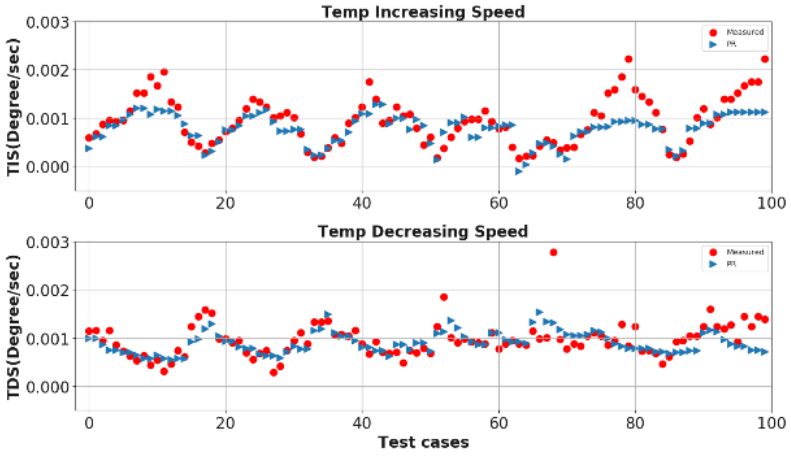
Table 4-9 Hyper-parameters for the Best TIS and TDS Estimators in Four Months

Training data month	TIS model parameters	TDS model parameters
May	"PR1": {"alpha": 0.01, "deg": 2, "interaction_included": false}	"PR1": {"alpha": 0.05, "deg": 1, "interaction_included": true}
June	"XGB1": {"booster": "dart", "max_depth": 2, "n_estimators": 100, "subsample": 0.6}	"SVR1": {"C": 100, "epsilon": 0.2, "gamma": 0.001, "kernel": "rbf"}
July	"SVR1": {"C": 1, "epsilon": 0.1, "gamma": 0.01, "kernel": "rbf"}	"SVR1": {"C": 100, "epsilon": 0.05, "gamma": 0.001, "kernel": "rbf"}
August	"RF1": {"bootstrap": true,"max_features": "sqrt", "min_samples_split": 4, "n_estimators": 100}	"SVR1": {"C": 10,"epsilon": 0.05, "gamma": 0.01,"kernel": "rbf"}

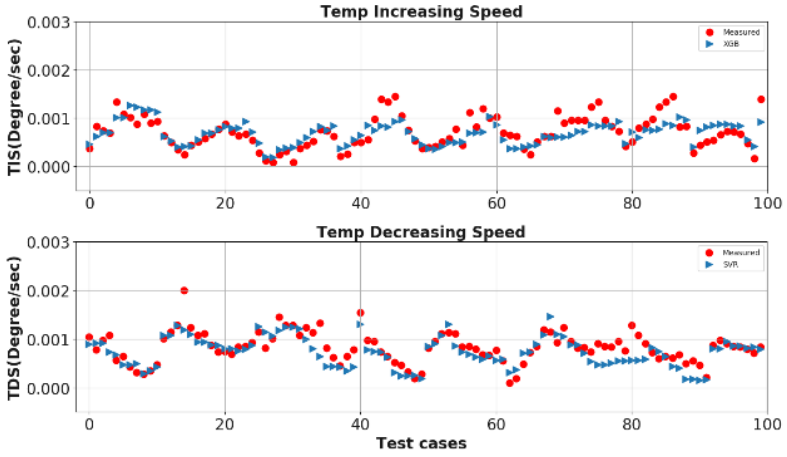
4.6.3. Prediction Error for the Best Predictors

In this section, the indoor temperature variation speed models will be evaluated: First, the best predictors (model and hyper-parameter pair) selected in the cross validation are trained using the training data; Then, the test data set will be used for evaluating the prediction error.

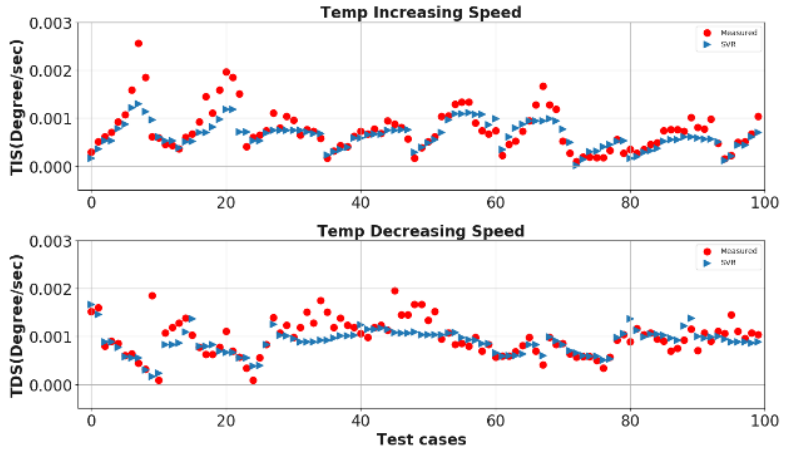
Figure 4-15 illustrates the temperature variation speed prediction results for four months. The y-axis of these figures is the temperature variation speed, either TIS or TDS in °F/second. The x-axis represents the first 100 instances in the testing dataset; it is worth noting that these instances are arranged in chronological order, but the time interval between consecutive instances are not equal. The red circles are the temperature variation speed ground truth calculated from the thermostat measured data and the blue triangles are the predicted values.



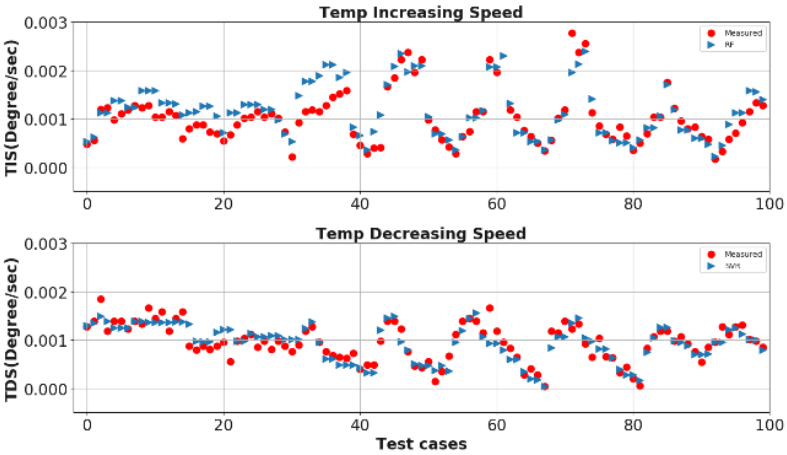
a) Testing the model trained using May data against data from early June



b) Test forecasting using data from early July



c) Test forecasting using data from early August



d) Test forecasting using data from early September

Figure 4-15 Prediction Results for Testing the Trained Model in Four Months

According to the instance-based prediction results in Figure 4-15, the proposed temperature variation speed forecast model can predict TIS and TDS with a good accuracy level in general. There are cases when the prediction is off from ground truth, such as a few instances in the TIS prediction in Figure 4-15 a). Figure 4-16 enlarges these high prediction error instances. Further investigation shows all these high error cases (as highlighted in the red shadow region) happen during specific time of day: from 12pm to 4pm in those two days. Considering the more accurate prediction results of the same period in previous days, it can be inferred that these high prediction error might result from abnormal daytime office activities. For instance, the door is constantly open during those hours, which introduces more heat gain and thus speeds up the increase of indoor room temperature.

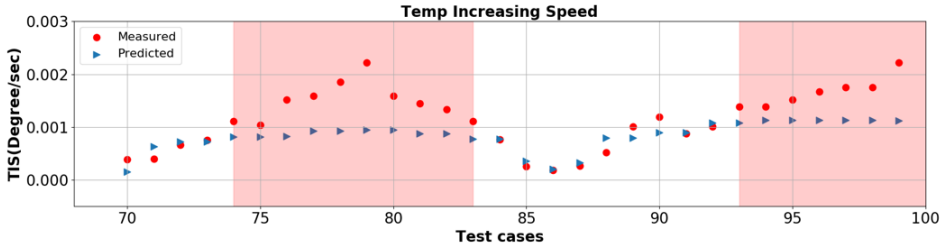


Figure 4-16 High Error Periods in June Testing Data Might Result from Abnormal Daytime Office Activity

Figure 4-17 plots the predicted temperature variation speed against those measured. The plots with red points are for the TIS prediction and those with blue points are for the TDS prediction.

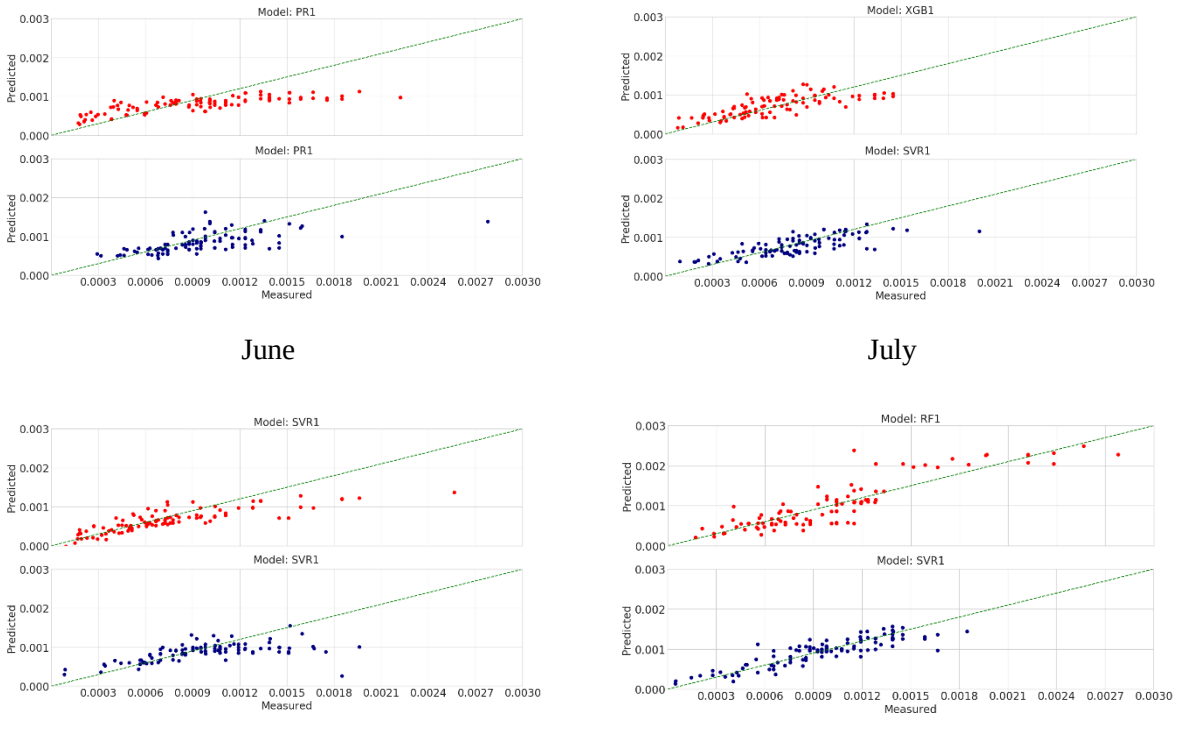


Figure 4-17 Predicted vs. Measured Temperature Change Rate for Four Months' Validation

To further quantify the accuracy level for the temperature variation speed prediction, two statistical indices are used: the variance explained and the mean absolute scaled error (MASE). The variance explained shows to what extent the proposed model accounts for the variation of the dataset, and the MASE shows the average error level. They are calculated using(4-17) and (4-18), where $y*$ is the predicted value and y is the actual one. The results are shown in Table 4-10.

$$explained\_variance(y, y*) = 1 - \frac{Var\{y - y*\}}{Var\{y\}} \tag{4-17}$$

$$MASE = \frac{Mean\ Absolute\ Error(y, y*)}{mean(y)} \tag{4-18}$$

Table 4-10 Variance Explained and MASE for Four Months' Testing

Month	Variance explained		MASE (%)	
	Best TIS Estimator	Best TDS Estimator	Best TIS Estimator	Best TDS Estimator
June	0.6111	0.3418	26.2427	20.3892
July	0.5024	0.5802	26.0111	20.0527
Aug	0.6442	0.3534	26.7047	20.5943
Sept	0.7743	0.7522	19.5635	15.0893

According to Table 4-10, the variance explained by the proposed model for the four months is between 34%~77%. When occupants' schedule and indoor activities are regular and constant over certain period, the proposed model can explain more variance; in contrast, in some months, more randomness in the indoor activities introduces more variance to the dataset which cannot explained by the model. The mean absolute scaled error of these estimators is around 20%.

4.6.4. Indoor Temperature Prediction: Algorithm and Validation

In this section, a step-by-step temperature prediction algorithm is proposed based on the temperature variation speed prediction model.

Table 4-11 Algorithm for Predicting Indoor Temperature Step-by-step

Indoor Temperature Prediction Algorithm

1: Divide the prediction horizon into N steps, each step represents a 5-minute interval
2 Initializing: get Temp(0) and HVAC unit operating schedule
3: **for** $i = 1 \to N$:
4: a. collect feature data from thermostats and online weather service (data in Table 4-3)
5: b. data preprocessing (transforming the data to the 10-dimension feature vector)
6: c. check HVAC unit operating schedule at Step i
if $status(i) = ON$:
predict temperature decreasing speed ($\Delta Temp$) using the TDS prediction model
elif $status(i) = OFF$:
predict temperature increasing speed ($\Delta Temp$) using the TIS prediction model
end if
7: d. predict temperature at Step i : $Temp(i) = Temp(i-1) + \Delta Temp \times 300$
8: **end for**
9: **return** Temp(i) for $i = 1 \to N$

Given the indoor temperature variation speed prediction model, the indoor temperature can be forecasted in a step-by-step manner using the algorithm shown in Table 4-11. Because the purpose of indoor temperature prediction is to evaluate the thermal comfort under a predefined HVAC unit control strategy, a proposed unit operating schedule is used as one input for the algorithm. The output of the algorithm is the indoor temperature profile during the prediction horizon.

Based on this algorithm, a few experiments of the indoor temperature prediction and their results are discussed in this section: first, experiments based on the historical data are conducted to evaluate the proposed model; second, two real-world field tests are conducted in the room where all these data were collected, to see how the model can be applied to real world.

4.6.4.1 Experiments based on historical data

In the historical data collected from the smart thermostat, the indoor temperature and status of the HVAC unit are logged. Therefore, it is possible to select a time and 'pretend' the future is unknown: based on the 'planned' unit schedule, the indoor temperature can be 'predicted'. Then, the 'predicted' indoor temperature profile and the recorded profile are compared.

To better describe the experiment procedure, a few terms used in the following discussion are presented:

- Status Change: The event when the status of the HVAC unit changes, either from ON to OFF or from OFF to ON;
- Session: A session is between two status changes, so there will be cooling sessions and heating-up sessions.
- Period: A duration of time consists of several consecutive sessions.

Based on these definitions, the procedure below is used for the evaluating the temperature prediction error:

1. Train the TIS and TDS models with the CV selected estimators using one month's data;
2. Choose the first nine days in the next month's data, which are collected between 6:00 and 21:00, for prediction experiments;
3. Specify the length for prediction period as in session number (e.g. length equals 4 sessions);
4. For each period, indoor temperature is predicted with 5-minute prediction intervals;
5. Error between the predicted and actual temperature at the end of prediction period is calculated for all periods in the experiment days;
6. Repeat the above steps for other months or other length for prediction.

Figure 4-18 demonstrates the period-end error for a four-session period.

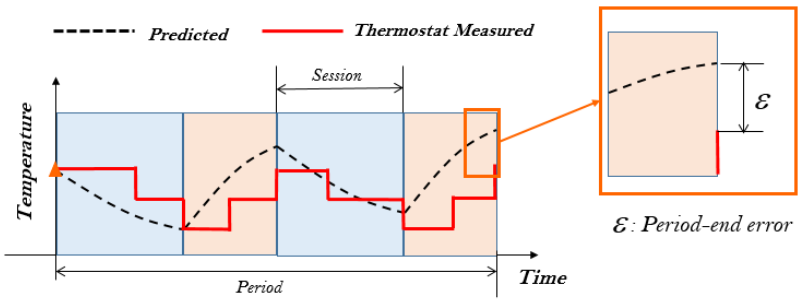
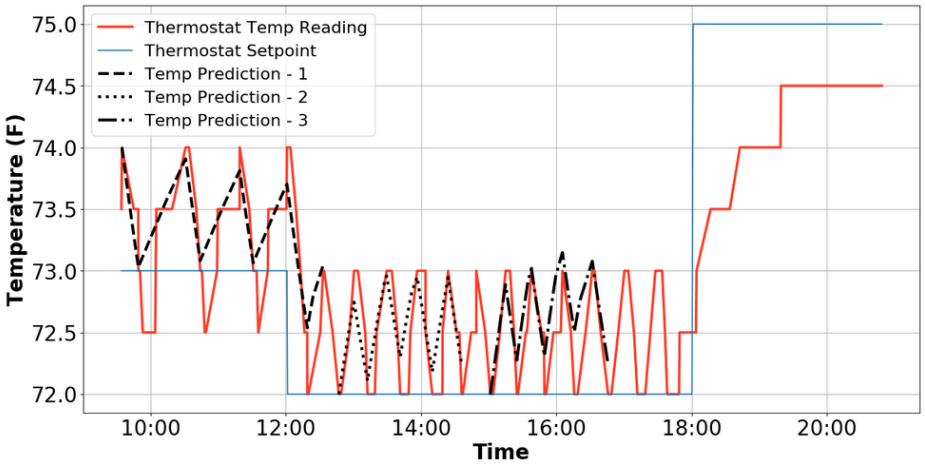


Figure 4-18 Error Evaluation on Time-series Prediction of Indoor Temperature

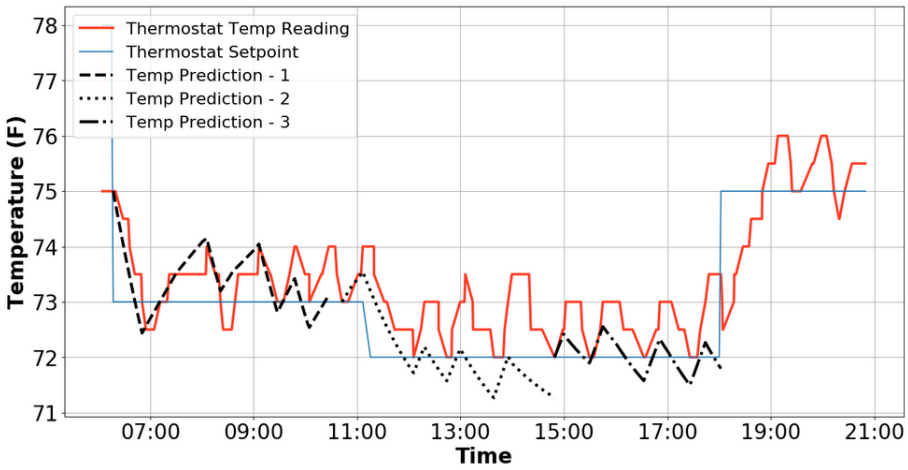
In this study, periods with session number of 1, 2, 5, 8 are investigated, to see how the indoor temperature prediction error will propagate through time. For each testing day, assuming N_{sess_all} is the number of sessions in the day, $L_{period} \in \{1,2,5,8\}$ is the number of session per period, then the number of testing period in a day N_{test} can be calculated using (4-19).

$$N_{test} = N_{sess\_all} - L_{period} + 1 \tag{4-19}$$

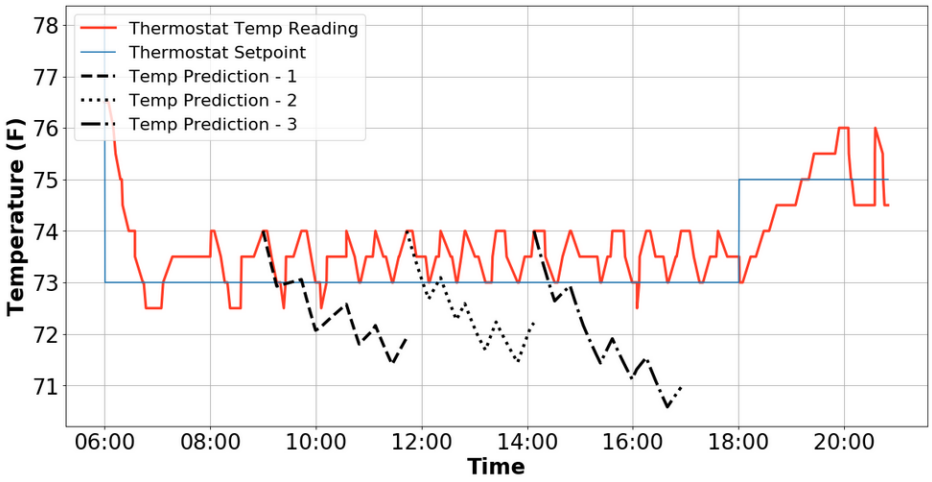
Figure 4-19 (a) shows three examples of 8-session prediction in a day. In this day, the temperature variation speed forecast models made very accurate prediction in every session, resulting in a precise indoor temperature profile prediction. Admittedly, there are other times when the prediction is not as accurate, Figure 4-19 (b) and (c) show cases where the prediction algorithm gives moderate and large forecast error. In Figure 4-19 (c), it can be seen that the TIS model predicts the temperature increase speed much smaller than the real one, and eventually causes the temperature at the end of the session deviates from the reality for more than 2 ℉.



(a) Three examples of 8-session prediction in a day (Accurate)



(b) Three examples of 8-session prediction in a day (Moderate)



(c) Three examples of 8-session prediction in a day (Inaccurate)

Figure 4-19 Examples of Predictions in Three Days at Three Accuracy Levels

To further analyze how the algorithm performs on indoor temperature prediction, the average error between the predicted and actual temperature at the end of prediction period collected during the experiments are quantified. The average error at the end of the prediction period for all four months is shown in Table 4-12 and visualized in Figure 4-20. Because in this experiment with historical data, the prediction period is defined in session length, average duration of the prediction in hours is calculated to better describe the time-related error.

Table 4-12 Average Error at the End of Prediction Period in Four Months

Testing Data Month	Session Length	Average Duration (Hour)	Average Error at Prediction Period End (°F)
June	1	0.4137	0.4863
	2	0.8257	0.7769
	5	1.86	1.4725
	8	2.8295	1.9458
July	1	0.5281	0.2991
	2	1.0688	0.3464
	5	2.529	0.6368
	8	3.8856	0.7507
August	1	0.4902	0.3677
	2	0.9705	0.4114
	5	2.274	0.8256
	8	3.5364	1.0862
September	1	0.4393	0.3481
	2	0.8885	0.522
	5	2.0424	0.9151
	8	3.1924	1.1009

From Table 4-12, it can be seen that the longer the prediction period, the larger the average error at the end of the prediction period. This is due to the propagation of the error: because in later steps the algorithm is making prediction based on a condition deviated from the reality, the prediction error thus accumulates over time. Another observation is that the error is more significant when testing using the data from June, in which case May data are used for training. This is due to the season transition from cool spring to much warmer summer happens around this time in the area. As shown in Figure 4-21, there lies a great difference

of the daily maximum temperature distribution between May and June. Such difference causes the weather and occupants' behavior of May and early June very different from each other; further it influences the accuracy of the indoor temperature prediction in June.

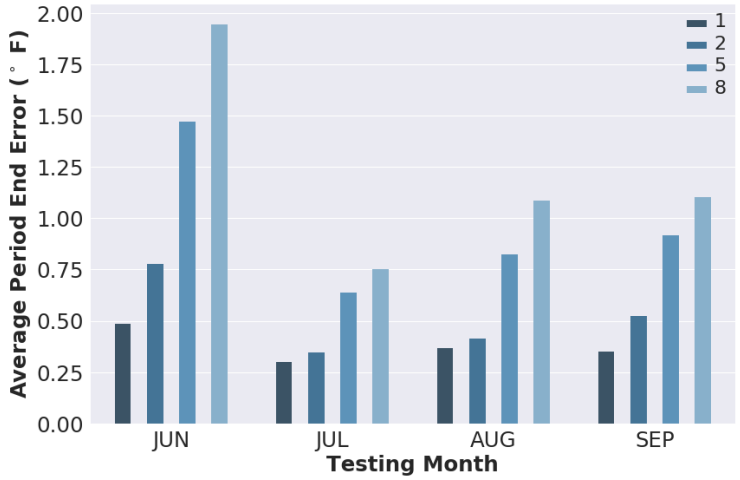


Figure 4-20 Average Period End Error in Four Months' Testing Data

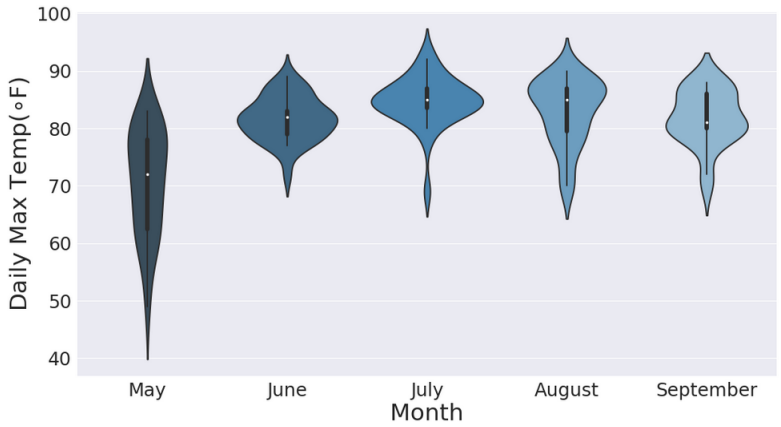


Figure 4-21 Distribution of Daily Maximum Temperature in Four Months

For other mid-summer months, the average period errors are much smaller. For a 1-session prediction, which lasts about 20 minutes to half an hour, the average error is around 0.3℉; for an 8-session prediction, which lasts about 3 to 4 hours, the average error is around 1.0℉. Because of such error level and considering a typical DR event usually lasts less than or around 3~4 hours and human beings are insensitive to 1.0℉ temperature difference, it is safe to say the proposed algorithm can produce comparatively reliable forecast results for the DR air-conditioner control.

4.6.4.2 Real building field test

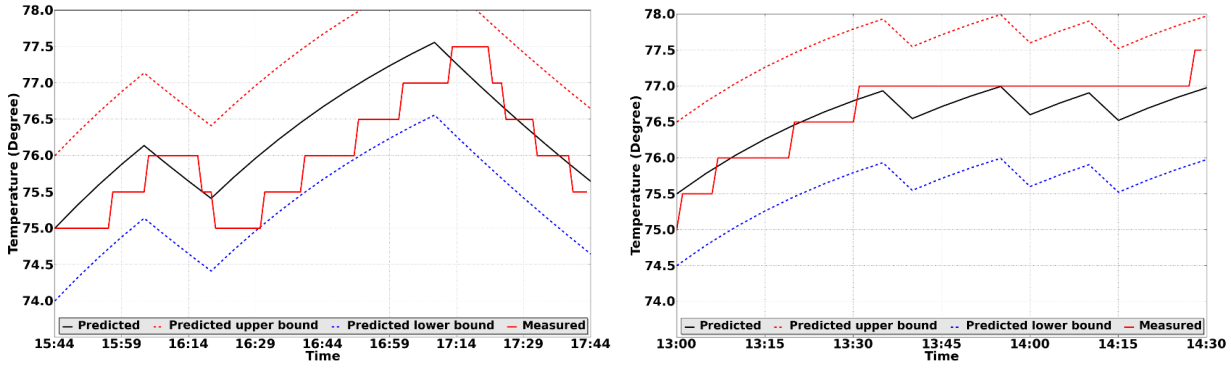
Besides the validation using historical data, experiments in real building are conducted as well. With the permission of the building engineer, two afternoons are randomly selected for validating the indoor temperature prediction algorithm. Detailed meteorological (forecast) and time information on the field test days are collected in advance, as listed in Table 4-13. Before testing, the AC unit control schedule is obtained; it shows how the unit will be controlled during the control horizon, at 5-minute interval. By

utilizing the aforementioned information, the indoor temperature profile can be predicted for the control horizon based on an estimated initial temperature. During the controlling period, the AC unit is turned ON/OFF as planned. After the control, actual indoor temperature profile logged by the thermostat is collected and compared with the predicted value.

Table 4-13 Two Testing Cases of Indoor Temperature Prediction in A Real Building

Case	Outdoor Temperature	Outdoor Humidity	Weather Category	Starting Time	Duration
a	80° F	55%	2	15:44	2.0 hours
b	84° F	57%	1	13:00	1.5 hours

Figure 4-22 shows the comparison between forecast temperature profile (as black solid line) and the actual readings (as red solid line). It demonstrates that temperature readings on the thermostat follow the predicted values closely and are constantly within ± 1°F error band.



(a) Case a (b) Case b

Figure 4-22 Comparison between the Predicted Temperature and Actual Temperature in Cases a and b

In summary, the results from these real building experiments demonstrate the proposed indoor temperature prediction model can be used to accurately predict the indoor temperature for the next a few hours.

4.7. Real World Application

In real world scenarios, the proposed building thermal property model can be learnt either locally on edge devices or on the cloud by the smart thermostat service providers (e.g., Nest and Honeywell).

If learning on edge devices, due to the limitation on the computation power, the TIS and TDS models can be re-trained once in a week using the past 30 days of data. Grid search can also be omitted and directly use SVR if the training data is enough (>300 instances) and use polynomial regression if otherwise. In contrast, for the smart thermostat service provider, these thermal models can be re-trained every night, by taking advantage of the enterprise level server. Grid search can also be conducted to identify the model that performs the best based on the data collected in the last 30 days.

5. AIR CONDITIONING (AC) UNIT POWER CONSUMPTION MODELING

AC units are modeled as fixed power load in previous studies on control strategy, however, their power consumption are time-variant variables in reality. Figure 5-1 shows the AC unit compressor states (1-ON/0-OFF), outdoor temperature (°F) and AC power consumption (kW) in 24 hours. As shown in the example, AC power changes according to various outdoor temperature, the variation is up to 2 kW in this example. Therefore, if the fixed power model is used, the error between the fixed value and the real value might aggregate among multiple AC units and cause the building failing at achieving the demand response goal. To avoid this, it is important to understand how much power each AC unit consumes in different environment. There are previous work on the performance study of AC system that can depict such feature of AC unit, however, units from different manufacturers have different feature, using the forward model will inevitably add configuration burden on the building managers. Therefore, an inverse model that learns such feature from historical data is much easier for the real world application.

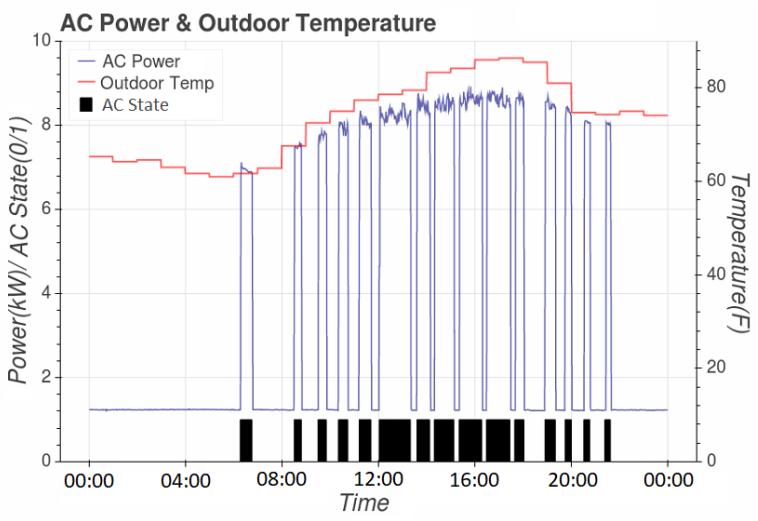


Figure 5-1 Power Consumption of One AC unit Change with Outdoor Temperature

To depict the relationship between a single unit's power consumption and the outdoor temperature, one straightforward approach is to install a power meter to measure the consumption for each unit. Nonetheless, considering that each power meter usually costs several hundreds of dollars, this approach is prohibitively expensive. To tackle this problem, a power disaggregation algorithm is proposed so that multiple AC units' power consumption models can be obtained using data collected from a single power meter; by doing this, the number of power meters that the building owners need to install is largely decreased to one and thus reduce the hardware investment drastically.

The work in this chapter was originally published in [154].

5.1. Single AC Unit Power Consumption Model

To start with, the power consumption model of a single unit is investigated. According to [52], the power of a AC unit can be expressed as a linear function:

$$P = k_1 \cdot \frac{Temp_{out} - Temp_{in}}{R_{air}} + k_2 \tag{5-1}$$

R_{air} is the heat resistance between building air and outdoor air. For simplicity, k_1 and k_2 can be considered as constants here. Comparing with the outdoor temperature $Temp_{out}$, indoor temperature $Temp_{in}$ is usually maintained at a comparatively constant level. Thus, the AC power consumption is linearly related to the outdoor temperature. To validate this linear relationship, a power meter is installed in a building on Virginia Tech campus to measure the power consumption of one AC unit. The power measurement is implemented using a BACnet power meter, sampling interval is one minute. By integrating the outdoor temperature data from the online weather resource, the relationship between outdoor temperature and the AC power consumption is exemplified in Figure 5-2. The Pearson coefficient between outdoor temperature and AC power consumption is 0.9185, which indicating a strong linear relationship.

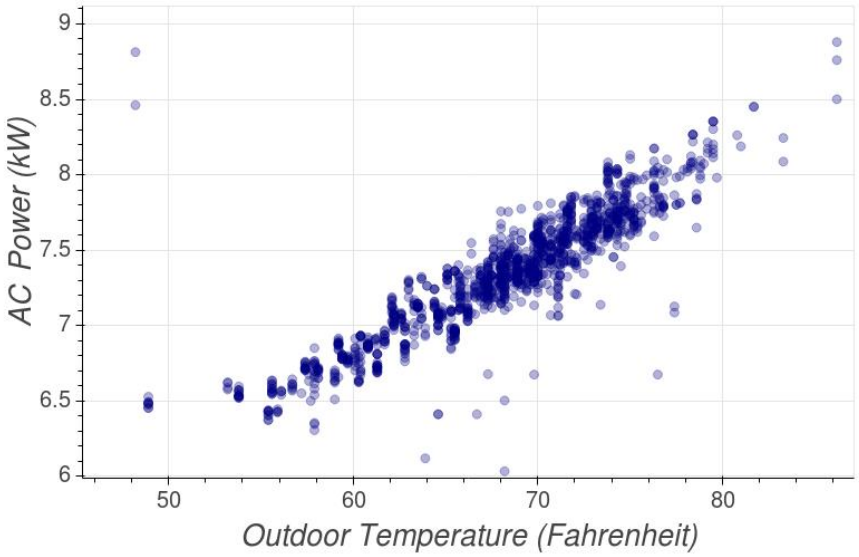


Figure 5-2 Single AC Power Consumption vs. Outdoor Temperature

Outdoor humidity is also considered in this study, however, the linear correlation observed from the data collected is not as strong, with a Pearson coefficient of -0.4709. To further study whether humidity should be included as a predicting feature, the feature importance of outdoor temperature and humidity are investigated. Random forest, a decision tree based bootstrapping aggregated model, provides a convenient way for evaluating feature importance. The importance of a feature is represented by the percentage of time this feature is used as a splitting feature in the decision trees. As a result, a 200-estimator random forest regression model is used for feature importance evaluation, the outcome is listed in Table 5-1.

Table 5-1 Feature importance of outdoor temperature and humidity

Features	Temperature	Humidity
Feature Importance	0.9326	0.0674

In conclusion, the outdoor temperature is the dominating feature that will strongly influence the AC power consumption. Thus, power consumption of an AC unit is a linear function with respect to outdoor temperature. Except for the cooling unit/compressor power, there is also power needed for driving an air-circulating fan, which consumes constant power.

Hence, the total power consumption of an AC unit can be expressed as (5-2):

$$P = S(\omega \cdot Temp_{out} + b) + S_f \cdot f \tag{5-2}$$

Where, P is the power consumption of an AC unit; $Temp_{out}$ is the outdoor temperature; $S \in \{0,1\}$ is the status of the AC compressor; and $S_f \in \{0,1\}$ is the status of the AC fan. These variables are recorded from the building under study. The rest of the variables, ω, b and f are determinants of the power-temperature model; the meaning of f is the power consumption of an HVAC unit's fan.

In Figure 5-1, it also shows there are two components of HVAC power consumption: that of the compressor and the fan. The fan and compressor work together according to some rules, and mostly the fan runs more than the compressor. According to Figure 5-1, when the HVAC cooling unit is OFF, the fan is still on for ventilation purpose and constantly consumes around 1.4 kW. In practice, there are three possible conditional probabilities between the status of the compressor and the ventilation fan: $\Pr(S_f = 1 \mid S = 1) = 1$, $\Pr(S_f = 0 \mid S = 1) = 0$ and $\Pr(S = 0 \mid S_f = 0) = 1$.

The relationship described in (5-2) is also observed based on the data collected from other AC units. This further validates the proposed single AC power model. However, it is worthwhile to note that this model only suits for single-stage AC units, but not for multi-stage units.

5.2. Multiple HVAC Units Power Consumption Disaggregation

To reduce the hardware investment on power meters, which are expensive, a power disaggregation algorithm is investigated in this section. This algorithm can be used to identify the power-temperature model of multiple AC units, using the aggregated power data measured by a single power meter. Thus, this introduces a more applicable and affordable approach.

When a single power meter is measuring multiple AC units' power, and each of the unit has the power consumption feature as (5-2) shows, identifying parameters (ω and b) will help find the individual model of each unit. Assuming the power consumption model of AC Unit i is:

$$P_i = S_i(\omega_i \cdot Temp_{out} + b_i) + S_{fi} \cdot f_i \tag{5-3}$$

The aggregated AC power consumption measured at time t (P^t) is equal to the sum of the power of all individual units (P_i^t) plus an offset (μ), as expressed in (5-4):

$$\begin{aligned} P^t &= \sum_{i=1}^{K} P_i^t + \mu \\ &= \sum_{i=1}^{K} \left[S_i^t(\omega_i \cdot Temp_{out}^t + b_i) + S_{fi}^t \cdot f_i \right] + \mu \\ &= \sum_{i=1}^{K} \left(S_i^t \cdot \omega_i \cdot Temp_{out}^t + S_i^t \cdot b_i + S_{fi}^t \cdot f_i \right) + 1 \cdot \mu \end{aligned} \tag{5-4}$$

where the status of each AC unit's compressor at time *t* is represented by S_i^t and S_{fi}^t stands for the fan status. The offset μ represents the constant noise measured by the power meter or even the power of some always-on and fixed power devices. K is the number of units being measured.

Out of the seven variables in (5-4), three of them ($S_i^t, Temp_{out}^t, S_{fi}^t$) are measurable: the status of compressor and fan are recorded by the smart thermostats and the outdoor temperature can be retrieved from online weather services; the rest four types (ω_i, b_i, f_i, μ) are the parameters to be learnt.

Put all measurable variables in a vector, for time step t:

$$\mathrm{x}^t = [S_1^t \cdot Temp_{out}^t, S_1^t, S_{f1}^t, \cdots S_K^t \cdot Temp_{out}^t, S_K^t, S_{fK}^t, 1]^T \tag{5-5}$$

After collecting data over a period of time, assuming there are total U time steps, measurable variables of all time steps can be put into a matrix:

$$\mathbf{X} = [\mathrm{x}^1, \mathrm{x}^2, \cdots \mathrm{x}^U]^T \tag{5-6}$$

Correspondingly, the aggregated power measurement for these time steps are:

$$\mathbf{y} = [P^1, P^2, \cdots P^U]^T \tag{5-7}$$

In addition, the parameters to be learnt are shown as below, including the ω_i, b_i, f_i for all units and the offset power μ:

$$\mathbf{w} = [\omega_1, b_1, f_1, \cdots, \omega_K, b_K, f_K, \mu]^T \tag{5-8}$$

To sum up, $\mathbf{X}$ is a matrix with dimension of $U \times (3K+1)$. $\mathbf{y}$ and $\mathbf{w}$ are vectors of dimension $U \times 1$ and $(3K+1) \times 1$ respectively. Because there is $\mathbf{y} = \mathbf{Xw}$, as a result, the parameter identification problem become a multi-variate regression problem (i.e., minimizing residual $\mathbf{y} - \mathbf{Xw}$). It can be solved using normal equation analytically:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \tag{5-9}$$

5.3. System Configuration and Validating Procedure

5.3.1. System Configuration

In order to validate the above-mentioned power consumption disaggregate algorithm, power measurement data of five AC units from a building on Virginia Tech campus are used.

The building being studied has the BEMOSS (Section 3.1.2) installed, and BEMOSS serves as a platform for communicating with and archiving data from smart thermostats and smart meters in the building. Five AC units are responsible for the climate control of five thermal zones inside the building. Each HVAC unit is controlled by a smart thermostat located in the corresponding thermal zone. Based on the measured data, the approximated power consumption of each HVAC unit when outdoor temperature is 70℉ is shown in Table 5-2.

Table 5-2 Power of Each HVAC Unit when Outdoor Temperature is 70℉

HVAC#	1	2	3	4	5
Power (kW)	7.5	6.3	6.3	12.4	3.7

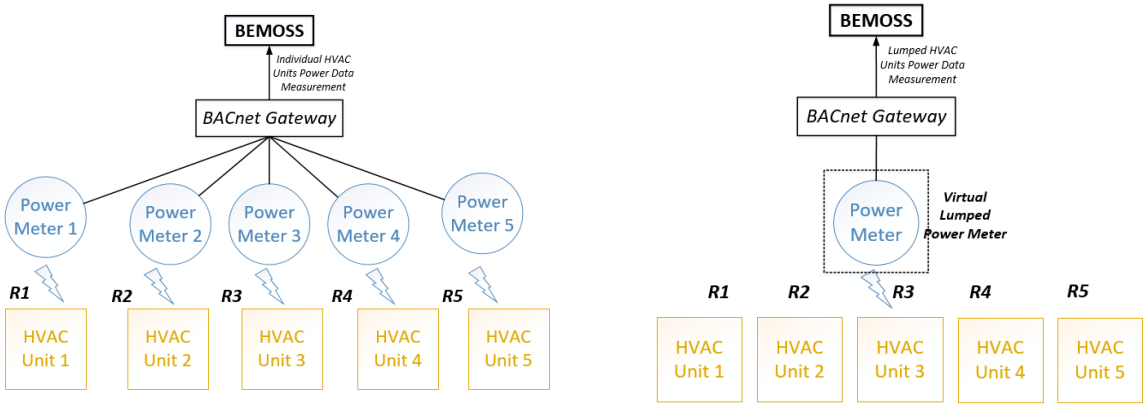


Figure 5-3 Power Meter Placement for This Study

For validation purpose, to provide the actual temperature-power model, each HVAC unit has one power meter installed to measure the actual unit power consumption, as shown in Figure 5-3 (a). By analyzing these individually measured power meter data, the power consumption feature of each unit can be generated as stated in Section 5.1, they will be used as the ground truth in the following study. However, no power meter is installed to measure the aggregated power consumption of all AC units. In this case, the power measurement data from all power meters are summed as if they were measured by a single power meter, i.e., the virtual lumped power meter in Figure 5-3 (b). These data are to be used by the power disaggregation algorithm proposed in Section 5.2 to generate the temperature-power model of each AC unit. Finally, the result will be compared with the ground truth.

5.3.2. Validation Procedure

The validation procedure is explained as follows:

1) Data collection: historical data of all five thermostats and five power meters were downloaded from the BEMOSS platform.

2) Data fusion: the AC units' power consumption data, their ON/OFF status and weather data were merged according to the data timestamp. A column in the data frame was created as the sum of all power meters' readings, representing the aggregated AC power consumption.

3) Data cleaning: Bad data occurs in several occasions. For instance, AC power readings can be much higher when the transient power is recorded during the start of a unit; in addition, it is possible to have data gaps when meters fail to record the data. In this study, the data were grouped by states and outliers of each state were detected and deleted.

4) Training/Testing data splitting: The clean dataset was first randomly shuffled and then split into two groups: training and testing datasets. The training dataset was used for power disaggregation to determine the individual AC power consumption model; the testing dataset was used to verify the result of power disaggregation.

5) AC power disaggregation: power disaggregation was conducted using the algorithm proposed in Section 5.2. Parameters of each AC unit (ω_i, b_i, f_i, μ) were determined.

6) Model validation: Firstly, the disaggregated AC power-temperature models obtained from the proposed approach were compared with those learnt from individually measured AC power consumption data. Secondly, with these power-temperature models, the power consumed by all AC units under different statuses were calculated using (5-4) and were compared with the measured data in the testing dataset.

5.4. Experiment Results and Discussion

Following the validation procedure above, the data in one-minute intervals were collected during a three-month summer period in a Virginia Tech building. After the process of data merging and cleaning, 25% was randomly chosen as the test dataset, while the remaining 75% became the training dataset. After disaggregating the power consumption of individual AC units using the proposed algorithm, the following validations were conducted:

a) Validation on individual AC unit power-temperature models
b) Validation on estimating the total AC power consumption using disaggregated models

5.4.1. Validation on Individual AC Power-Temperature Models

In this section, the power-temperature models generated by the power disaggregation algorithm are compared with those fitted using data from individual meters. If the models are matching, the validity of the algorithm is verified. By solving (5-9), the parameters ω , b and f characterizing the power-temperature model of individual HVAC units in all five zones of the building were determined and shown in Table 5-3. In addition, the offset power μ was determined as 1.1176 kW, representing the constant power measured when none of the HVAC units and their fans is ON.

Table 5-3 Parameters Characterizing Individual HVAC Power Consumption

AC Unit	ω (10^{-2})	b	f
1	6.7286	1.5192	1.0527
2	6.3229	1.1254	0.7479
3	2.5024	3.3280	1.2523
4	6.0426	6.7293	1.1074
5	2.3944	1.5707	0.0003

For each unit, the power being measured by the individual power meter should have the following form:

$$P_i = S_i(\omega_i \cdot Temp_{out} + b_i) + S_{fi} \cdot f_i + \mu_i \qquad (5\text{-}10)$$

Table 5-4 Offset Power (kW) of Five Individual Power Meters

μ_1	μ_2	μ_3	μ_4	μ_5
0.1796	0.0717	0.0474	0.2099	0.6128

Similar to μ in (5-4), the offset power of an individual meter *i* (μ_i) represents either noise or the power consumption of always-on devices.

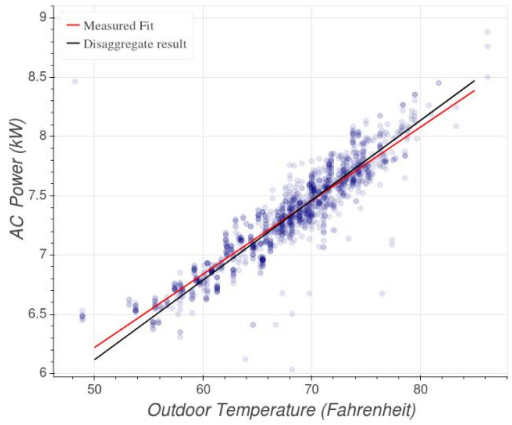
Table 5-4 shows the offset power of each individual power meter, which is derived as the average power readings when S_i^t and S_{fi}^t are both 0.

Please note that (5-10) is only for validating the disaggregation algorithm. In practice, when only one power meter is installed, it is not possible and not necessary to find out the μ_i.

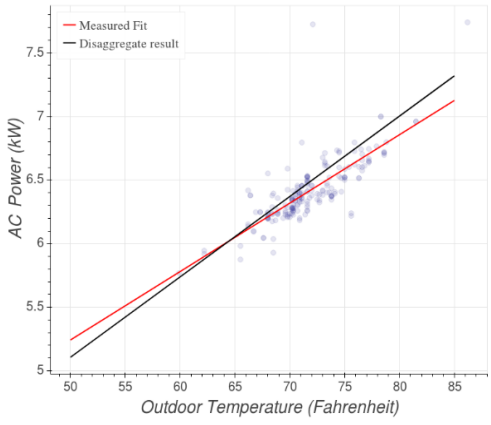
The sum of the offset power of individual power meters ($\sum_{i=1}^{5}\mu_i = 1.1214$) is consistent with the offset disaggregated from the total AC power consumption (μ=1.1176). It is worth noting the values of f_5 and μ_5: f_5 is extremely low when compared with other fan power, and μ_5 is very high with respect to other offset power. The reason of this might be the following: due to certain ventilation requirement, the fan in the thermal zone 5 is constantly in the ON state and thus it is 'categorized' in the always-ON load by the disaggregation algorithm. As a result, it appears in the offset power. As long as this fan continues to act as an always ON load, the disaggregated model is valid; otherwise, new set of data should be collected and new models need to be learnt.

Using the information in Table 5-3 and

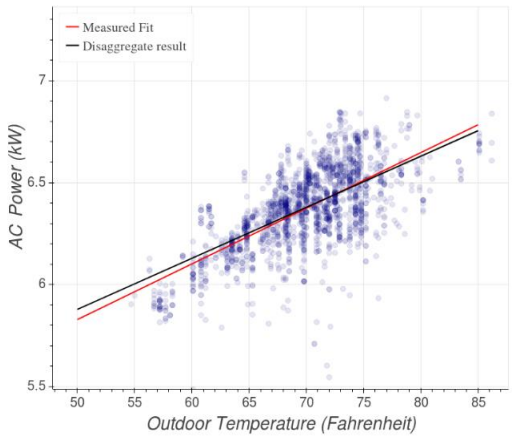
Table 5-4, the disaggregated power-temperature models are determined by (5-10). These models are plotted as black lines in Figure 5-4. The blue points represent the individually measured AC power consumption when both the compressor and fan are ON (S_i =1).The red lines are directly fitted from the individually measured data from five power meters.



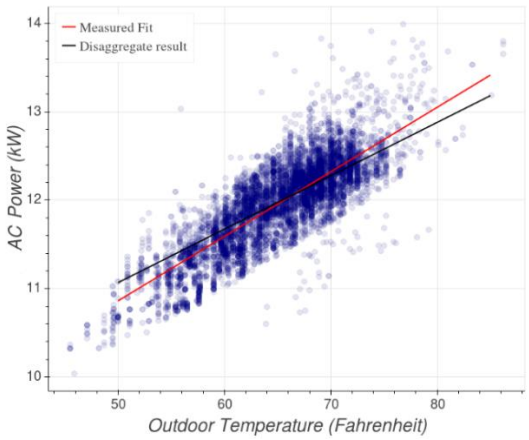
a) HVAC#1



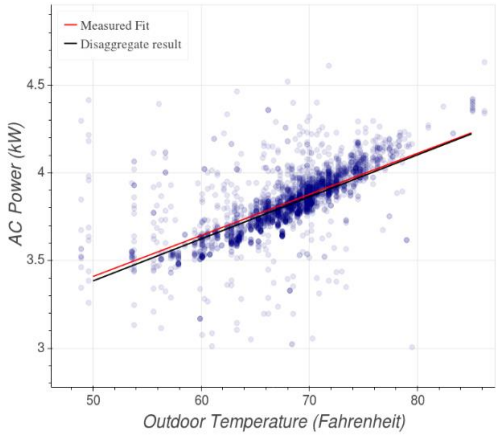
b) HVAC#2



c) HVAC#3



d) HVAC#4



e) HVAC#5

Figure 5-4 Single AC Unit Power based on the Power-Temperature Models vs the Measured Consumption.

The results above show that by disaggregating the total power consumption data of multiple AC units, power consumption characteristics of each AC unit can be accurately modeled; and the disaggregated models are close to those fitted by the individually measured data. To compare these two models, a 'model-to-model' error is shown in Table 5-5 to see how different are the disaggregated model from the individually measured model. The model-to-model error here is defined as the absolute difference between the power consumption values predicted by the two models. The errors are small compared with the AC power consumption, this means the disaggregated models are very close to those learnt from the individually measured data.

Table 5-5 Model-to-model Error between Disaggregated and Individually Measured Model

AC Unit	Model-to-model Error (kW)	AC Power at 75 °F Using Individually Measured Model (kW)
1	0.0358	7.7683
2	0.0997	6.5875
3	0.0126	6.5106
4	0.1095	12.6858
5	0.0141	3.9937

These models also reflect the normal electric behavior of AC units; they can be used for fault detection by detecting abnormal change of AC behavior.

5.4.2. Validation on Estimating the Total AC Power using Disaggregated Models

The purpose of learning the power-temperature model is for more accurate estimation of the aggregated multiple AC units' power consumption, especially during the DR control. Therefore, the total power consumption of five AC units are estimated given the outdoor temperature and compressor/fan status from the testing data and the power consumption models learnt from training data. Then the estimations are compared with the ground truth values in the testing data.

To visualize the comparison, 50 data points from the testing data set are randomly selected, and the estimated power consumption of each unit are stacked in the bar chart to represent the total AC power consumption; measured ground truth data are marked as circle in Figure 5-5. It can be observed that the estimated total AC power consumption and the measured data are very close.

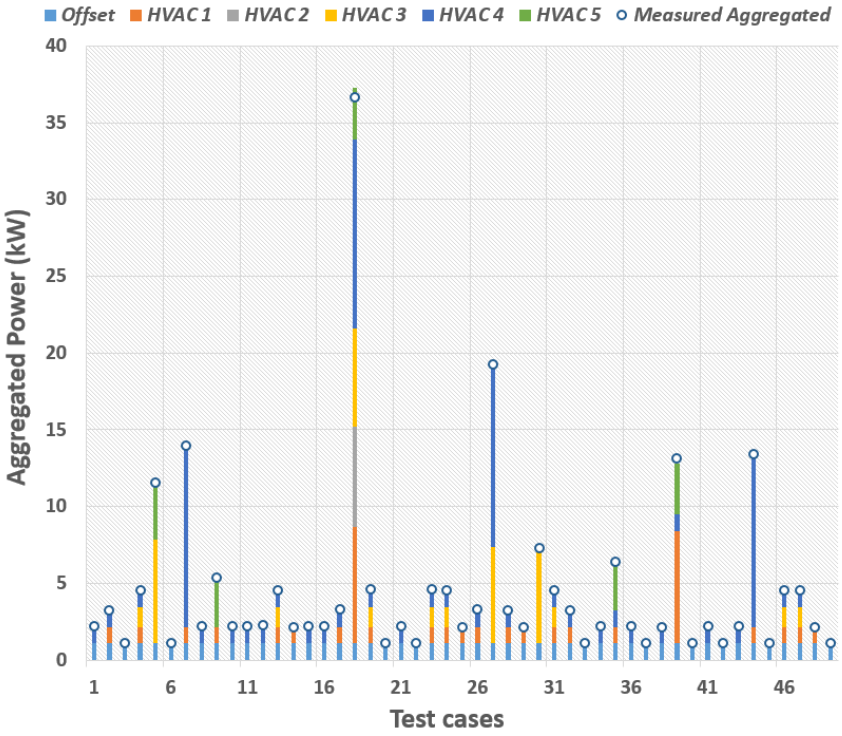


Figure 5-5 Estimated Total AC Power Consumption vs. the Measured Total Power Consumption from the Lumped Virtual Power Meter (50 Data Points)

In addition to this, the whole testing dataset is used for validating the aggregated AC power consumption based on individual power-temperature models. The result is shown in Figure 5-6, where the x-axis represents the measured power consumption from the single power meter, and the y-axis represents the aggregated AC power consumption estimated using (5-4). The distribution of the data points show a reliable $y = x$ relationship with R^2=0.9994, demonstrating that the power can be precisely estimated.

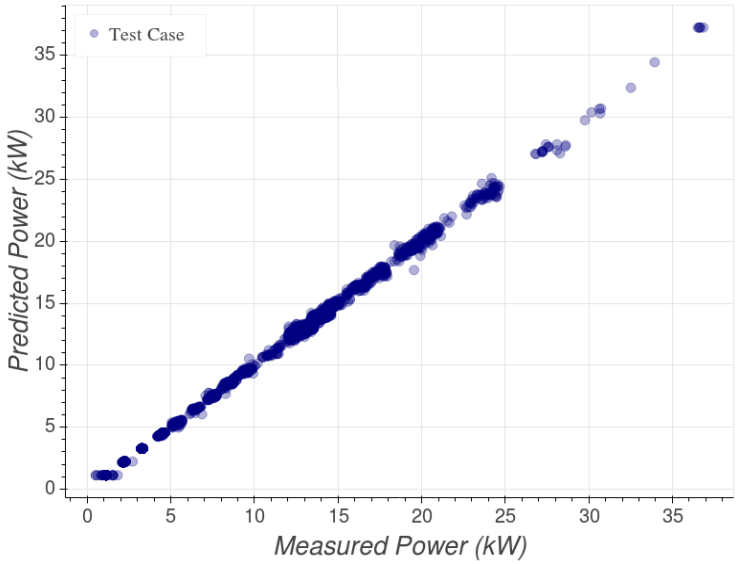


Figure 5-6 Aggregated HVAC Power Consumption based on Power-Temperature Models vs. Total HVAC Power Consumption from the Lumped Power Meter

Therefore, the power disaggregation algorithm has demonstrated a strong capability for precisely estimating multiple AC unit power consumption when the control strategy is given (which units to be ON/OFF). This accurate estimation enables a precise power control of AC units to maintain the total power consumption under a certain demand limit during DR events.

5.5. Discussion

One question arises when it comes to the real world application: for how long should the data collection process be in order to have a dataset large enough to learn the disaggregated models accurately? The answer is it depends. First, the number of AC units being measured matters. With more units to be modeled, more parameters in (5-9) need to be learnt and thus more data are needed. Second, it depends on the variety of the data points: giving an extreme case, if all the units are turned ON and OFF exactly at the same time, it will be impossible to disaggregate the individual unit models no matter how much data are provided. As a result, the BEM system can be designed so that it will intentionally control multiple AC units in different pattern to reach to the data variety level sooner. The variety in the outdoor temperature should also be guaranteed during the data collection period.

Theoretically, the power disaggregation algorithm will work with arbitrary number of HVAC units being measured, as long as the data diversity presents. However, in the future, more tests are needed to investigate if there is an upper boundary for the number of units being measured.

6. OPTIMIZATION BASED AC UNITS COORDINATION CONTROL UNDER DEMAND RESPONSE

In Chapter 4 and Chapter 5, a building thermal property model along with an AC power consumption model are proposed. These two models are the building blocks for considering indoor thermal comfort and AC real time power consumption, which relate to the thermal comfort constraint and DR power limit respectively. In this chapter and the next one, algorithms for automated optimal AC coordination control during DR events will be proposed.

According to Section 3.2.3, the problem of AC coordination during DR is similar to the power system unit commitment (UC) problem: both of them are solving a scheduling problem with an optimization target and several constraints. Therefore, in this chapter, an algorithm based on the mixed integer linear programing (MILP), which is very popular in solving the UC problem, is proposed.

The work in this chapter has been originally published in *Applied Energy*, please see [140] for detail.

6.1. Optimization Model

According to Section 3.2.2, the goal of the AC coordination algorithm is to reduce peak load of all AC units to a predefined limit while minimize the cost from occupants' thermal discomfort and energy consumption (if considered). Since the utility companies are yearning for a fast responsive load reduction, according to Section 2.2.2, it will be appealing if the algorithm can be computationally efficient, and be able to generate the control strategy within short time of notice (ideally within minute level). Therefore, to achieve this efficiency level, a linear programming (LP) is applied.

6.1.1. Building Thermal Property Model Linearization

In order to formulate a LP problem, the building thermal property models proposed in Chapter 4 should be linearized. Given that in a short period of a DR event, five out of six variables of the indoor temperature prediction model typically do not change drastically, namely outdoor temperature, outdoor humidity, day of week (not change at all), hour (minor change during short time) and weather condition (usually clear or cloudy hot days when DR happens). This means the indoor temperature variation speed, either increasing or decreasing, can be expressed as a function of indoor temperature only after all other values are given (weather forecast can help determine the value of $\overline{T_{out}}, \overline{H_{out}}$ and $\overline{w}$ for the incoming DR event).

$$\frac{dT_i}{dt} = f(T_i, \overline{T_{out}}, \overline{time}, \overline{dow}, \overline{H_{out}}, \overline{w}) = g(T_i) \tag{6-1}$$

In addition, according to (4-2) and (4-3), the temperature variation speed is linearly related to the indoor temperature; thus the temperature decreasing speed and temperature increasing speed can be expressed as (6-2) and (6-3). μ_d , μ_r , ν_d and ν_r are the parameters for the linear models.

$$TDS = \mu_d T_i + \nu_d \tag{6-2}$$

$$TIS = \mu_r T_i + \nu_r \tag{6-3}$$

If the building thermal models are trained using supervised learning models other than the polynomial regression, a process of linear fitting is needed to obtain the explicit form of these linear thermal models.

6.1.2. Mixed Integer Linear Programming Model

For better describing the problem, Table 6-1 shows the definition of relevant variables, among which the status of AC units at different time slots (S_t^h) are the control variables.

Table 6-1 Variables in AC Units' Coordination Control Optimization Problem

Variable	Definition	Variable	Definition
D	Total cost for occupant thermal discomfort and energy expenses.	T_t^h	Indoor temperature of Zone h in Time slot t (°F)
ω	Cost of electricity per kWh, in currency unit. (ω =0 if do not consider energy conservation)	P_t^{total}	Total AC power consumption in Time slot t (kW)
H	Total number of thermal zones	T	Total number of time slots
$D_{HVAC}(T_t^h)$	Cost for occupants' thermal discomfort under the indoor temperature of T_t^h	P_{AC-DR}^h	Power of AC Unit h (kW) under typical DR weather
S_t^h	Status of AC Unit h in Time slot t (0,1 stands for OFF/ON)	T_{max}^h	Maximum tolerable temperature in Zone h (°F)
α_t^h	Monetary productivity loss caused by thermal discomfort of Zone h in Time slot t , reflecting zone priority.	T_{min}^h	Minimum tolerable temperature in Zone h (°F)
P_t^{DR}	Demand response AC power limit (kW)	Δt	Length of time slots (e.g., 5 mins or 15 mins)
β_t^h	New variable introduced to linearizing the problem	M	Big constant for solving the optimization problem

The objective of the multiple AC coordination control is to minimize total cost derived from the occupants' thermal discomfort as well as energy consumption (optional) during DR events. Therefore, an objective function consists of two components is proposed, as shown in (6-4).

$$\text{Minimize}: D = \sum_{h=1}^{H}\sum_{t=1}^{T} D_{HVAC}(T_t^h) + \omega \cdot \sum_{h=1}^{H}\sum_{t=1}^{T} P_{AC-DR}^h \cdot S_t^h \cdot \frac{\Delta t}{60} \tag{6-4}$$

The first component is the cost for the thermal discomfort productivity loss, which is defined as a hinge function as shown in (6-5); the second component is the energy cost for AC operation during the DR event.

$$D_{HVAC}(T_t^h) = \begin{cases} 0 & if\ T_t^h \le T_{max}^h \\ \alpha_t^h (T_t^h - T_{max}^h) & if\ T_t^h > T_{max}^h \end{cases} = \max[0, \alpha_t^h (T_t^h - T_{max}^h)] \tag{6-5}$$

The energy cost component is optional, depending on whether the specific DR program the building has. If energy cost is not considered, $\omega = 0$ is used; otherwise ω is used to represents the unit cost for the electricity. P_{AC-DR}^h is the power consumption of a AC unit, calculated using the model proposed in Chapter 5, using the forecasted outdoor temperature. Because the cost of energy consumption is monetary, as a result, this study also monetarize the cost of thermal discomfort to unify these two aspects. A study has shown that thermal discomfort will cause building occupants productivity loss [155]. To generalize, the study first assumes that a building manager can quantify an equivalent economic loss caused by the

productivity loss based on their understanding to the building's business. α_t^h in (6-5) is given as such indicators.

With both α_t^h and ω, the optimization model is trying to find the optimal tradeoff between occupants comfort and total energy consumption.

In addition, the reasons using this hinge function and a threshold $T_{\max}^h$ in (6-5) are:

1. The occupants are aware of the fact that the building is in an energy saving mode, as long as the temperature is below $T_{\max}^h$, they are psychologically comfortable with that.
2. Once the limit has been breached, the occupants will suffer both physically and psychologically, and there will be productivity loss (if it is an office building or school). The loss is assumed to be linear in this model, similar to the findings in [136].

Finally, it is worth noting that the unit cost for thermal discomfort productivity loss, the variable α_t^h, can be set differently to prioritize different thermal zones. With higher α_t^h, the more priority the room has and thermal discomfort is less tolerable in it. An approach to facilitate the building managers to determine this value is presented in later discussion.

This objective function is subject to the following constraints considering the building operation during DR programs:

Inequality Constraints:

1. Room temperature should not be lower than a certain threshold at any time:

$$T_t^h \geq T_{min}^h \quad (\forall h, \forall t) \tag{6-6}$$

2. Total power consumption of multiple AC units should be under the DR AC power limit P_t^{DR} at any time during a DR event:

$$P_t^{total} = \sum_{h=1}^{H} S_t^h \cdot P_{AC-DR}^h \leq P_t^{DR} \quad (\forall t) \tag{6-7}$$

The DR AC power limit is determined by subtracting the total DR power limit by the amount of the base critical load predefined by the building manager, as shown in (3-1).

Equality Constraint:

Room temperature prediction at time t given the room temperature and AC unit status at time $t-1$:

$$T_t^h = f(T_{t-1}^h, S_{t-1}^h) = T_{t-1}^h + \Delta t \cdot TDS_{t-1}^h \cdot S_{t-1}^h + \Delta t \cdot TIS_{t-1}^h \cdot (1 - S_{t-1}^h) \quad (\forall h, \forall t > 1) \tag{6-8}$$

Typically, there are $TDS_t^h < 0$ and $TIS_t^h > 0$. Considering (6-2) and (6-3), (6-8) becomes:

$$\begin{aligned} T_t^h = {} & T_{t-1}^h + \Delta t \cdot (\mu_d^h - \mu_r^h) \cdot S_{t-1}^h \cdot T_{t-1}^h \\ & + \Delta t \cdot (\nu_d^h - \nu_r^h) \cdot S_{t-1}^h + \Delta t \cdot \mu_r^h T_{t-1}^h + \nu_r^h \cdot \Delta t \quad (\forall h, \forall t > 1) \end{aligned} \tag{6-9}$$

Apparently, a quadratic term between control variable and state variable ($S_{t-1}^h \cdot T_{t-1}^h$) is introduced in (6-9). Because the control variable is binary, a new variable and some inequality constraints can be considered to linearize this problem:

$$\beta_{t-1}^h = S_{t-1}^h \cdot T_{t-1}^h \tag{6-10}$$

Inequality constraints added are:

$$-M(1-S_{t-1}^{h}) \le \beta_{t-1}^{h} - T_{t-1}^{h} \le M(1-S_{t-1}^{h}) \tag{6-11}$$

$$-M \cdot S_{t-1}^{h} \le \beta_{t-1}^{h} \le M \cdot S_{t-1}^{h} \tag{6-12}$$

where M is a constant. Since the indoor temperature is bound by M according to (6-12), $M = 100$ is sufficient and is used in this study.

Although the optimization model above is formulated according to summer DR events, it can be easily modified ((6-5) and (6-8)) to be applicable for winter DR events, when HVAC system is in heating mode.

By solving this MILP problem, the AC units control schedule is obtained as the collection of control variables of S_t^h.

6.2. Algorithm Implementation

6.2.1. System Design

To implement the algorithm, an application for DR AC coordination is developed as shown in Figure 6-1, which acts like an add-on application for the BEM, BEMOSS in this study. The application consists of a learning process for building thermal property learning (Chapter 4) and an optimization process for AC units control schedule generation. These two processes are fulfilled by a learning agent and an optimization agent, respectively.

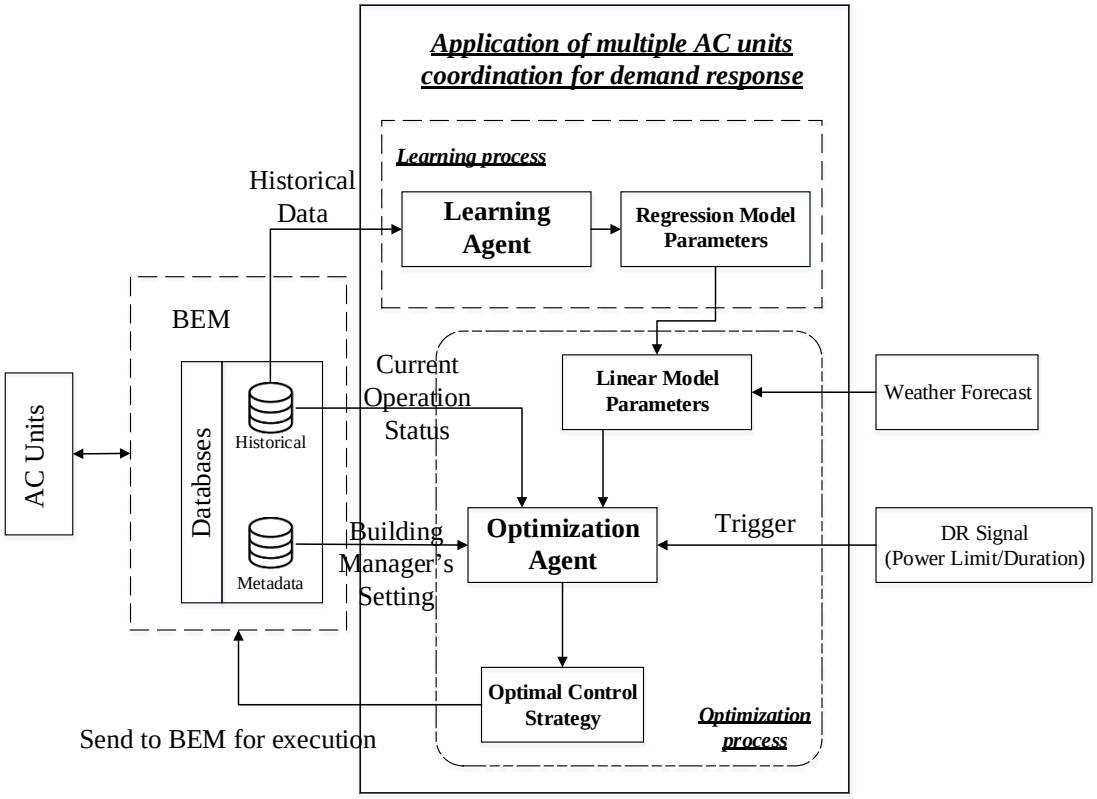


Figure 6-1 The Integration of the Proposed System and BEM

Three inputs will be passed from the BEM to the application:

1. Thermostat historical data:
   Historical data including the indoor temperature profile collected by the thermostat and outdoor weather information of the past 30 days will be used by the learning agent to train the building thermal property model.
2. Building current operation status:
   Current status such as real time indoor temperature are needed by the optimization agent for formulating the MILP problem.
3. Miscellaneous settings from the building managers
   Building managers have some settings such as the cost for productivity loss and electricity price, these will be needed by the optimization agent.

The interface between BEM and the proposed algorithm is bilateral. That is, historical data, current status and miscellaneous settings from the BEM system must be provided to the algorithm to allow learning of zones' thermal properties. In this study, the BEM used is BEMOSS; it saves historical data in a Cassandra database[156], and current status and settings are saved in the PostgreSQL database. The system has granted open and secure access of both databases to the algorithm application. On the other hand, the control strategy generated by the algorithm should be sent for execution by the BEM. When the BEM received the control strategy, thermostat agents in BEMOSS will execute corresponding commands.

6.2.2. Operation Process

Step 1: Building thermal model learning

Every night in the DR season, the learning agent uses the past 30 days of historical data, to update the building thermal property model based on the learning algorithms proposed in Chapter 4. The reason for updating the model is because some hidden influence factors such as the solar angle, are assumed constant in short period (30~40 days) and not considered in the model. Based on this assumption, prediction error increases when using a model trained by data collected in May to make prediction in August. Therefore, the building thermal property model is re-trained frequently. After the training, the model parameters are saved. This property learning process is conducted offline during late night.

Step 2: Optimization activated

On the DR event day, when the BEM received the DR signal from the utility company, via OpenADR protocol for instance, the optimization process is activated. The optimization agent first gathers current system status from the database, then linearizes the thermal property model obtained from Step 1 with consideration of current weather, formulates the MILP problem and finally uses a solver to generate the optimal AC control schedule. The MILP model will guarantee the computational efficiency so that the building can implement DR as soon as possible. Since an immediate result is needed in this process, it is called online optimization, when compared with offline thermal model learning.

Step 3: Execution

The control schedule, which consists of series of control commands, are sent to the BEM. Agents controlling corresponding thermostats/AC units will follow the instructions and execute the commands sequentially over the DR period.

6.3. Case Study

6.3.1. Comparison Cases Design

Comparisons between the proposed optimization based AC coordination algorithm and the most common practice of AC control in small- and medium-sized building during DR events, namely increasing thermostat set point, are made. Detail of these control are shown in Table 6-2.

Table 6-2 Control Methods for Demand Response

Control Methods	Details
Dead-band based control (DBBC)	• ***Dead-band based control (DBBC)***: Most commonly, the control of AC units rely on thermostats' dead-band control. Usually the temperature set point will be raised to make AC units run less frequently during DR events. • ***Dead-band based control with power limit (DBBC-PL***): Similar to the DBBC, but a AC unit will not start if its start will cause the total power consumption exceed the predefined power limit, even though the zone temperature is over the upper bound of the dead-band. • ***Dead-band based control with priority (DBBC-Pri)***: Based on the DBBC-PL, however, some AC units have higher priority to be operated even when power limit is reached. In such cases, running AC units with low priority will be shut down to limit the total power.
Proposed coordination control	As stated in this Chapter.

6.3.2. Simulations and Results

Building information and operation data from four suites (including Suite 1 studied in Chapter 4) in a Virginia Tech building in Blacksburg, VA, USA are used as prototype to showcase the proposed algorithm. These four suites mainly consist of offices and laboratories. Each of them can be considered as a thermal zone and has its own thermostat and AC unit. The electric power consumption from four AC units under typical DR weather are listed in Table 6-3, with the total AC power of 32 kW.

Table 6-3 Information of Testing Thermal Zones

Suite	1	2	3	4
Main Usage	Offices	Laboratory	Laboratory	Offices
AC Unit Power (kW)	8.5	7.0	12.0	4.5
Approximate Area (square feet)	3600	3150	5800	2400
Approximate Occupants Number	9	10	20	5

The proposed algorithm is designed for AC units coordination during a DR event, which typically lasts for a few hours. For example, the length of DR in STOR, UK, can be as short as 2 hours [84]. In addition, as authors in [40] point out, with a 'temporal' aggregation, the length of each end user's DR will be shorter. In this section, it is reasonable to simulate a 90-minute DR event, which happens in a Class 2 weather condition day during 13:00~14:30, with an outdoor temperature of 85 °F and humidity of 49%. Initial temperature in Suite 1~4 before the DR event starts are 74.0 °F, 73.0 °F, 76.5 °F and 76.0 °F respectively.

Two parameters need to be set before the operation of the control system: α_t^h and ω. In these simulations, α_t^h is set to be 1.25 reflecting the monetary productivity loss of 1.25 Dollars for every time slot (5 minutes) and every 1 °F increase above $T_{\max}$. ω is 0 when energy consumption is not considered or otherwise it is

set to 1, representing the electricity price during a DR event is 1 Dollar per kWh. Other values of α_t^h and ω will be discussed later.

Eight scenarios with different requirements and control methods during the DR event are studied:

Scenario 1: Control using DBBC. Increasing the set point of all thermostats to 76 °F (Considering the dead-band of 1 °F, the maximum temperature in each zone will be 77 °F).

Scenario 2: Control using DBBC-PL. Increasing the set point of all thermostats to 76 °F, meanwhile limiting the total power consumption under 13 kW.

Scenario 3: Control using DBBC-Pri. Increasing the set point of all thermostats to 76 °F. Starting from 13:30, change the set point of Suite 3 to 75.5 °F so that the temperature will be around 76 °F during a meeting from 13:45-14:30 in Suite 3. Total power consumption limited under 20 kW.

Scenario 4: Control using proposed coordinated control. Set occupants $T_{\max}$ as 77 °F and DR AC power limit as 20 kW. Energy consumption is not considered, with $\omega = 0$.

Scenario 5: Control using proposed coordinated control. Set occupants $T_{\max}$ as 77 °F and DR AC power limit as 13 kW. Energy consumption is not considered, with $\omega = 0$.

Scenario 6: Control using proposed coordinated control. Set occupants $T_{\max}$ as 77 °F and DR AC power limit as 20 kW, total energy consumption is considered, with $\omega = 1$.

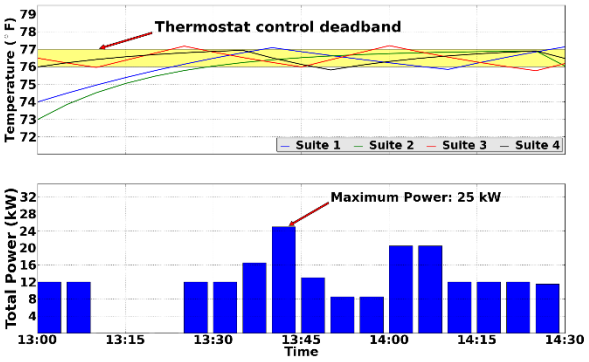
Scenario 7: Control using proposed coordinated control. Set occupants $T_{\max}$ as 77 °F and DR AC power limit as 20 kW. Temperature of Suite 3 should be around 76 °F same as in Scenario 3. Total energy consumption is considered, with $\omega = 1$.

Scenario 8: Control using proposed coordinated control. Set occupants $T_{\max}$ as 76.5 °F and DR AC power limit as 13 kW. Energy consumption is not considered, with $\omega = 0$.

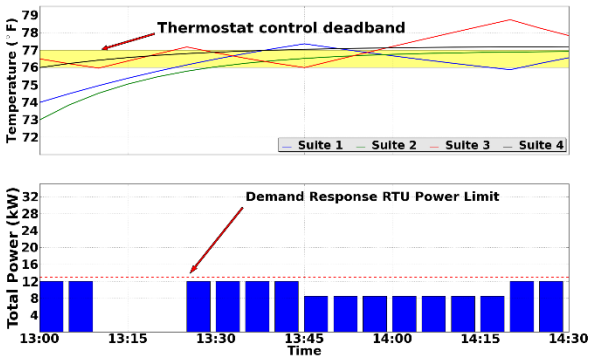
The simulation results on these eight scenarios are shown below: the productivity loss and total electricity consumed are summarized in Table 6-4; the temperature profiles and the total AC power profiles are shown in Figure 6-2; and two types of average temperature of all suites under 8 scenarios are shown in Figure 6-3.

Table 6-4 Occupants Discomfort, Energy Consumption and Maximum Power of Eight Scenarios

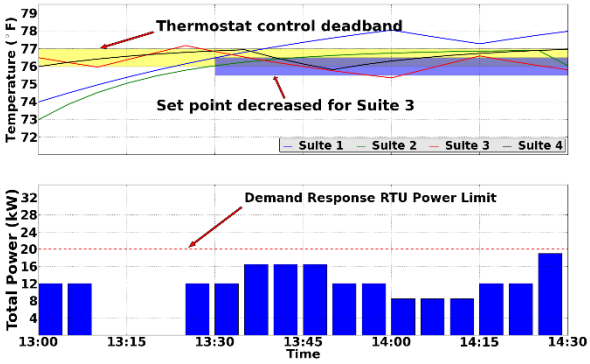
Scenario	Total Monetary Productivity Loss from Occupants Discomfort ($)	Energy Consumed (kWh)	Maximum Power (kW)
1	0.80	17.33	25
2	11.58	12.96	12
3	8.85	15.83	19
4	0	19.04	19
5	0	16.08	13
6	0.68	14	16.5
7	1.73	16	16.5
8	5.08	18.58	13



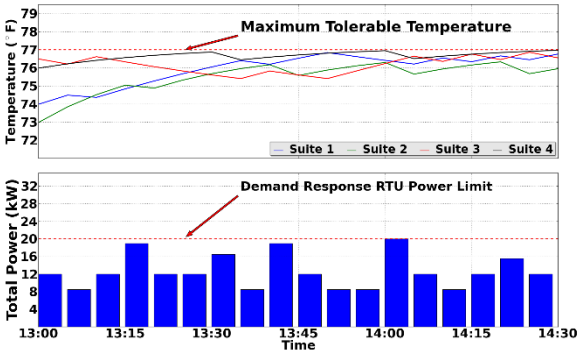
(a) Scenario 1



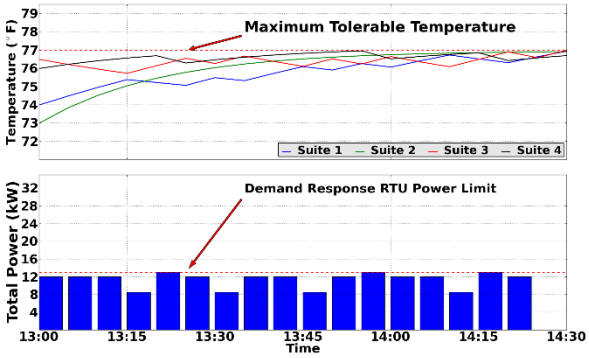
(b) Scenario 2



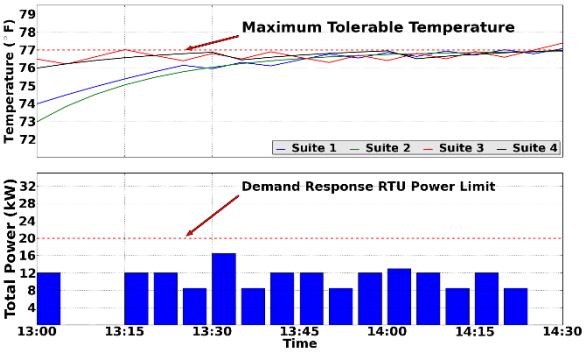
(c) Scenario 3



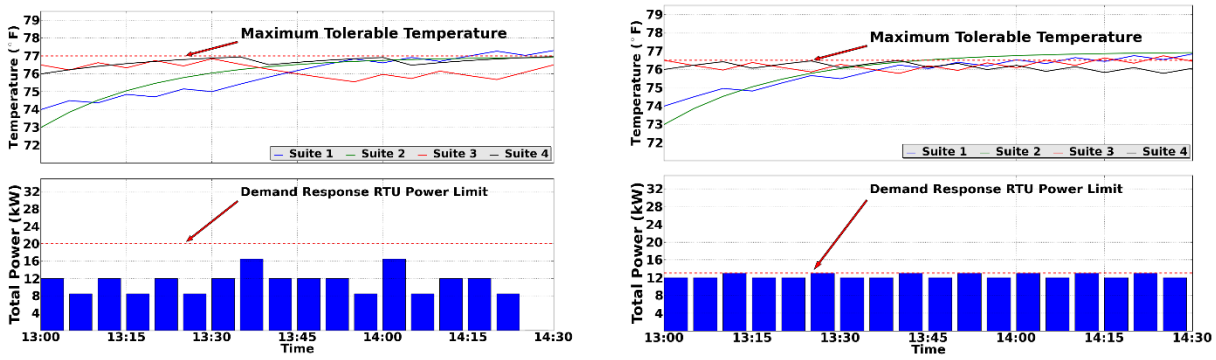
(d) Scenario 4



(e) Scenario 5



(f) Scenario 6



(g) Scenario 7 (h) Scenario 8

Figure 6-2 Temperature and Total Power Profile of Eight Demand Response Scenarios

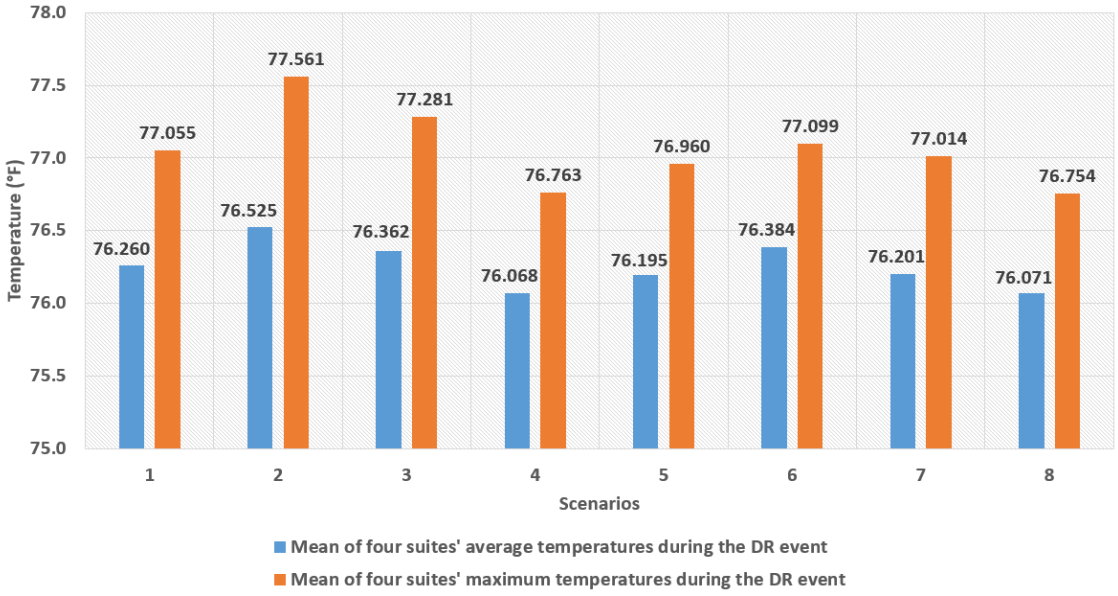


Figure 6-3 Mean of Four Suites' Temperatures under Eight Scenarios

According to the results, the proposed AC coordination control algorithm demonstrates three advantages over the DBBC control series, on the following aspects:

A. Peak load shaving effect. (Comparison between Scenario 1, 4 and 5)

Giving the same T_{max}, the maximum power consumption in Scenario 1 is 25 kW; while under the proposed control approach, consumptions in Scenario 4 and 5 are strictly limited under 20 kW and 13 kW respectively, both with zero occupants' discomfort. By reducing the maximum power, the building owners can reduce their capacity reserve charge during DR events.

B. Indoor temperature control (Comparison between Scenarios 2 and 5)

Both with 13 kW power limit, in Scenario 2, the DBBC-PL method causes occupants discomfort which is equivalent to $11.58 productivity loss while in Scenario 5, no occupants discomfort is observed.

C. Zone priority management (Comparison between Scenarios 3 and 7)

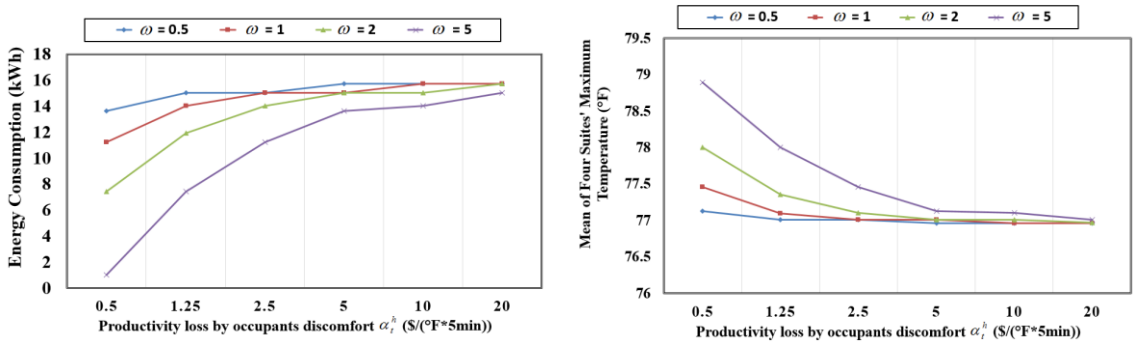
Given the same power limit and temperature request in Suite 3, the coordinated control can save up to 80% of occupants' discomfort with similar energy consumption, compared with DBBC-Pri. In addition, discomfort is distributed among different suites in Scenario 7 while in Scenario 3 it is originated from a single suite's suffering.

To sum up, these advantages can be attributed to the load shifting feature of the coordinated control. Due to the lack of coordination, the ON/OFF status of each AC unit is a random process when using DBBC. On the other hand, the coordination eliminates the cases when multiple AC units operating at the same time and causing undesirable high demand, as shown in Figure 6-2 (a) 13:40-13:45. Moreover, the coordination over a period of time will enable some AC units to pre-cool when resource, namely the power capacity, is available. Thus, spread the demand over the temporal range. In all, the coordinated control results in a higher capacity factor, which enables taking most advantage of the power limit; and from the electric utility perspective, an increased load predictability is highly welcomed during DR events.

Besides the advantages mentioned above, the coordinated control also provides the following flexibilities.

1) Jointly consider occupants' discomfort and energy consumption, suitable for different DR programs.

The values of α_t^h and ω are determined by the building manager, according to the building's business type and the DR program they participate in. Those two values will influence the balance point of the tradeoff between user discomfort and total energy consumption, therefore, different settings of α_t^h and ω are provided and compared, as presented in Figure 6-4.



(a) (b)

Figure 6-4 Comparison of Energy and Thermal Conditions under Different Monetary Settings

According to Figure 6-4, two general rules can be concluded:

a) For the same value of α_t^h, more or equal amount of electricity will be consumed if the price is cheaper, meanwhile delivers a lower maximum average temperature (more comfortable).
b) For the same ω, usually more electricity will be consumed if the productivity quality is more valuable (larger α_t^h), also results in less occupants' discomfort.

In general, the building manager will set ω as the electricity price for their DR program if they want to consider energy consumption and set α_t^h according to their evaluation of occupants' productivity. Similar

simulations as Figure 6-4 can be run prior to the system configuration to give the building manager a better sense about how to set the value of α_t^h .

2) Flexible DR settings

A building manager can also determine the DR settings such as occupants $T_{\max}$ and power limit flexibly. From the scenarios above, there are two points worth noting:

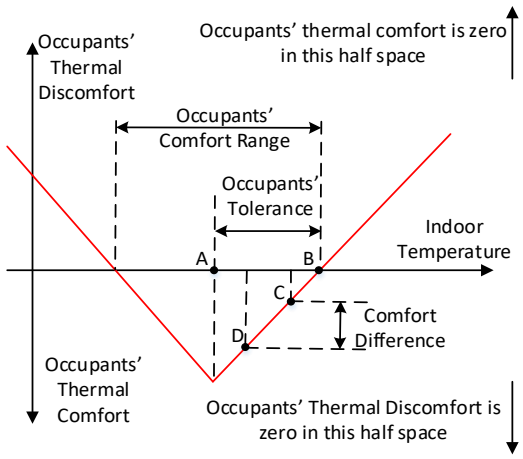


Figure 6-5 Schematic Diagram for Occupants' Thermal Comfort/Discomfort

(A = the temperature set point during normal operation (without DR); B = the occupants' $T_{\max}$; C and D = zero discomfort points)

a. In Scenario 4 and 5, the power limits are 20kW and 13kW respectively. Under both scenarios, the occupants do not suffer from thermal discomfort, however, if the power limit is 20 kW, the building manager needs to pay higher capacity reserve charge ($/kW) in the monthly bill. Thus, power limit of 13 kW is a better choice since it yields more monetary savings but does not exacerbate occupants' discomfort. In addition, the 20 kW power limit in Scenario 4 may also cause higher energy consumption than the 13 kW limit in Scenario 5. As shown in Table 6-4, an extra 2.96 kWh of electricity is consumed in Scenario 4. This can be explained using Figure 6-3 and Figure 6-5.

In Figure 6-5, since D has a smaller temperature difference from the normal set point than C, the operation point D thus yields higher comfort level. According to Figure 11, since indoor temperature under Scenario 4 is lower than that of Scenario 5, it is reasonable to use C and D in Figure 6-5 to represent Scenarios 5 and 4, respectively, and the comfort difference in the figure can explain the 2.96 kWh of extra electricity consumed in Scenario 4. Since energy consumption is not considered in these scenarios, the solver will provide an optimal solution among many, and this solution does not necessarily use less energy. In fact, the optimal control schedule might control the AC units to use more energy to make occupants more comfortable.

In all, the building manager can determine the optimal power limit with the consideration of capacity reserve charge and the expected occupants' comfort level.

b. A building manager should set $T_{\max}$ and power limit correspondingly. To be specific, a low power limit will not allow AC units running frequently and as a result, might not be able to satisfy a low $T_{\max}$. For

instance, in Scenario 8 above, the power capacity factor is nearly 1 yet the temperature in each room can hardly be controlled below T_{max} of 76.5 °F. This means T_{max} of 76.5 °F is not very reasonable under the power limit of 13 kW.

To find out an optimal power limit and a reasonable T_{max} and power limit pair, simulations can be run under some typical DR conditions, and based on simulation results, a building manager can decide the power limit and T_{max} to be implemented for each building.

6.3.3. Commercial Building Control and Validation

To validate the feasibility of implementing the proposed system in a real-world environment, a building control experiment is conducted in an afternoon during 16:30 to 18:00, with Class 2 weather category. Four thermal zones in the building are the prototype for the simulation study in Section 6.3.2 and the suite information is provided in Table 6-3. On-site system set-up is illustrated in Figure 6-6: with four smart thermostats installed and the BEMOSS running as the IoT-based BEM. The forecasted outdoor temperature and humidity during a DR event are 82 °F and 52%, respectively. The DR power limit and T_{max} are set to be 18 kW and 78 °F, respectively. Energy savings during DR is also considered in the optimization process with the electricity price of 1 Dollar per kWh ($\omega = 1$).

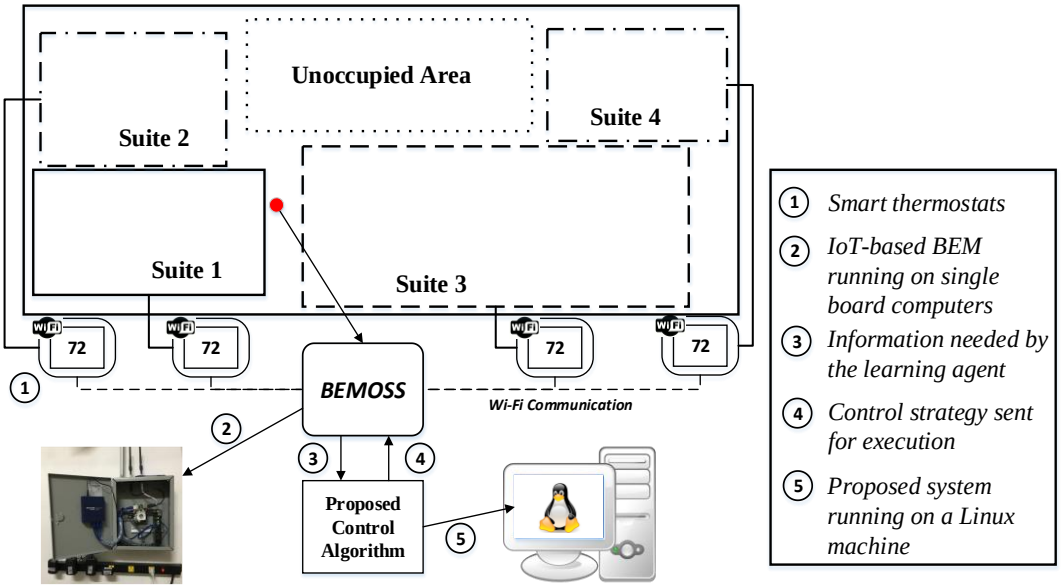


Figure 6-6 Schematic Diagram of On-site System Set-up in the Building in Blacksburg, VA

The initial temperature T_0 used by the optimization model is the temperature reading from thermostats with some adjustment, which complied with the following rule:

Condition A: At the time of computation, the thermal zone's AC unit is in off (not cooling) state.
Condition B: At the time of computation, the thermal zone's AC unit is in cooling state.

$$T_0 = \begin{cases} T_{dev} + \Delta T_{acc} & (if\ condition\ A) \\ T_{dev} & (if\ condition\ B) \end{cases} \quad (6\text{-}13)$$

In (6-13), T_{dev} is the temperature reading from thermostat at that time; ΔT_{acc} is the accuracy granularity unit of the thermostat; adding this value to T_{dev} under Condition A is a conservative measure.

Figure 6-7 shows the control results, with the black line represents the predicted temperature variation under optimal control strategy generated from the proposed algorithm; and the red solid line shows the actual temperature reading later acquired from the smart thermostats.

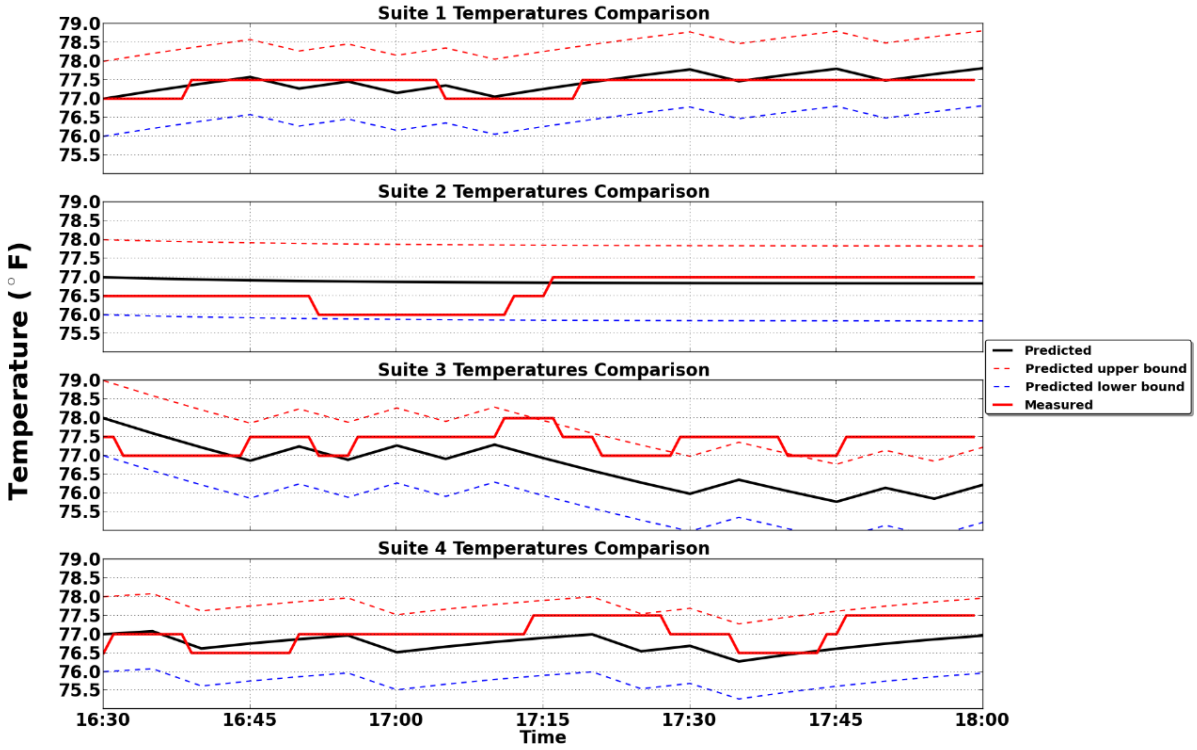


Figure 6-7 Temperature Change in Four Suites during A DR Event

The temperature profiles in all suites demonstrates a desirable control result. The temperature profiles in Suites 1, 2 and 4 stay closely with the predicted temperature (within ±1 °F error band). For Suite 3, around two third of the time, the temperature is within ±1 °F error band while the rest is slightly above this range. There are two reasons for the deviation: First, due to communication reason, the smart thermostat in Suite 3 missed the signal to turn on its AC unit at 17:10, and thus didn't cool the room as expected. Second, Suite 3 is a research lab, with some large heat-generating pumps running randomly according to experiments schedule. Such a stochastic change in internal heat gain makes the thermal model less reliable and thus causes such control errors.

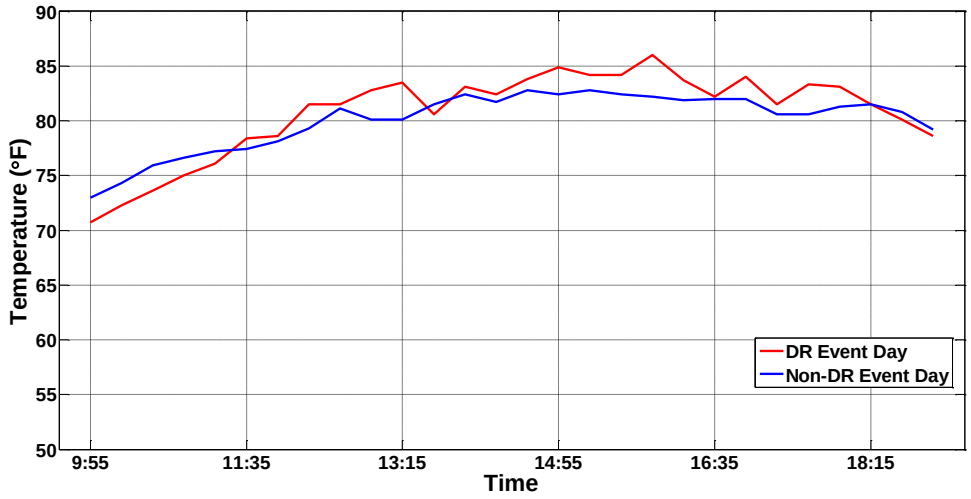


Figure 6-8 Weather Profile of DR Event and Non-DR Event Day

To study the peak load reduction effect, another non-DR event day with similar weather profile is found in historical data as a control group. The daytime outdoor temperature profiles of both days are shown in Figure 6-8, implying that without a DR event, the DR event day should consume similar level of power of the non-DR event day.

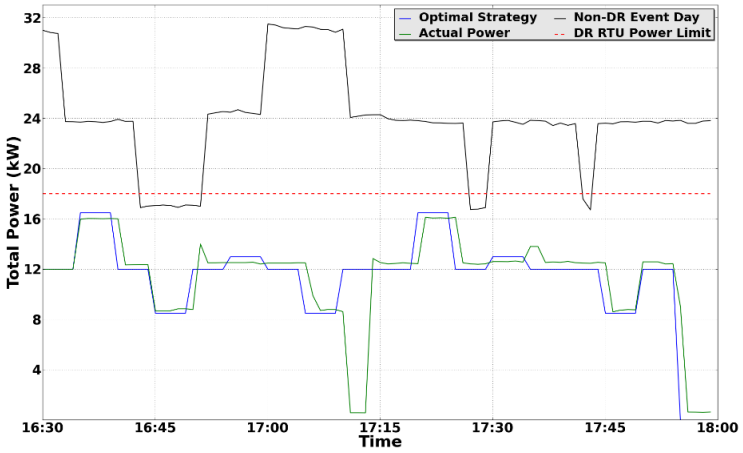


Figure 6-9 Power Consumption During the DR Event and Non-DR Event Day

As part of the research setting, some BACnet power meters are used to collect the power consumption data., the total AC power consumption of the test day and the control group is shown in Figure 6-9. The maximum power consumption decreases by almost 50% during the DR event (as compared with the non-DR event) in exchange with slight occupant discomfort according to Table 6-5.

Table 6-5 Indoor Temperature of Four Suites During the DR Event

Suite	1	2	3	4
Non-DR event day set point (°F)	72	73	75	72
DR event day Suites temperature (°F)	Around 77. See Figure 6-7			

The temperature in four suites during the DR event, which does not exceed T_{max} (78°F), is the indicator that the occupant discomfort has been minimized by the algorithm proposed.

According to Figure 6-9, the smart thermostat's failure of turning on its AC unit at 17:10 also caused an actual power ditch, compared with the expected power. This problem can be avoided by providing a strong and reliable wireless network.

6.4. Algorithm Efficiency

6.4.1. Improve Algorithm Efficiency for Edge Computing

Some fast DR programs require response within minutes to provide operating reserves to power system[86]. Therefore, the proposed algorithm is designed to be able to respond quickly to meet such a requirement. Since the thermal model is learnt offline and on weekly basis, only the optimization problem is solved as DR event is about to start, therefore, the efficiency of the optimization process is discussed in this section.

Generally speaking, the lower the DR power limit and the lower T_{max} are, the longer it takes to reach the optimal solution. Below lists two suggestions to ensure the efficiency.

1. Offline simulation – A few simulation should be conducted given some typical DR event settings (e.g., temperature, time and weather), there are two purposes: First, it helps to determine the reasonable DR AC power limit and T_{max} ; Second, these reasonable setting will allow the optimization problem to be solved efficiently;
2. Timeout option for the optimizer – when the computation time is more than five minutes and the precision is under a preset level, further computation will be terminated. This prevents the solver from spending unnecessary time searching for the absolute optimum.

Table 6-6 shows the computation time to obtain an optimal solution in Scenarios 4 to 8. The computation platform is a 2GB RAM Linux virtual machine with quad-core CPU, which emulates the specifications of some embedded systems. The result implies the proposed algorithm can be solved quickly enough for real-time implementation:

Table 6-6 Computation Time for Scenario 4 to 8 in Section 6.3.2

Scenario	**4**	**5**	**6**	**7**	**8**
Time (sec)	0.09	0.08	1.95	2.43	2.24

6.4.2. Impact of the Number of Thermal Zones on Algorithm Efficiency

To further study the impact of the number of thermal zones has on algorithm efficiency, additional scenarios are evaluated. For simplification, the settings from Suites 1-4 are doubled and tripled to create a group of eight and twelve thermal zones. The time it takes to solve the optimization problem is shown in Table 6-7 and

Table 6-8 for eight and twelve thermal zones, respectively. The timeout setting for the optimizer is 300 seconds in the study.

According to [4], around 72.1% of total commercial buildings in the U.S. have area less than 10,000 square feet. Assuming each thermal zone has 1,000 square feet on average, total area of twelve thermal zones is up to 12,000 square feet, and thus the testing of up to twelve thermal zones is reasonable.

Table 6-7 Algorithm Efficiency Test for 8 Thermal Zones Optimization (Total Load: 64 kW)

T_{max}	DR AC Power Limit (kW)	$\omega=1$		$\omega=0$	
		Time (sec)	GAP	Time(sec)	GAP
79	26	0.23	Optimal	0.1	Optimal
	30	0.22	Optimal	0.09	Optimal
	34	0.22	Optimal	0.09	Optimal
	38	0.22	Optimal	0.07	Optimal
	42	0.2	Optimal	0.07	Optimal
78	26	300	0.52%	0.1	Optimal
	30	300	0.72%	0.12	Optimal
	34	300	0.15%	0.09	Optimal
	38	255.24	0.03%	0.09	Optimal
	42	300	0.39%	0.09	Optimal
77	26	300	1.77%	0.61	Optimal
	30	300	1.72%	0.17	Optimal
	34	300	1.46%	0.16	Optimal
	38	300	1.53%	0.16	Optimal
	42	300	1.48%	0.14	Optimal
76	26	300	3.49%	300	4.99%
	30	300	3.03%	300	7.69%
	34	300	2.95%	300	28.02%
	38	300	2.14%	300	2.95%
	42	300	1.55%	1.02	Optimal

Table 6-8 Algorithm Efficiency Test for 12 Thermal Zones Optimization (Total Load: 96 kW)

T_{max}	DR AC Power Limit (kW)	$\omega=1$		$\omega=0$	
		Time (sec)	GAP	Time(sec)	GAP
79	39	0.88	Optimal	0.13	Optimal
	45	0.86	Optimal	0.12	Optimal
	51	0.87	Optimal	0.1	Optimal
	57	0.88	Optimal	0.08	Optimal
	63	0.89	Optimal	0.1	Optimal
78	39	300	2.00%	0.15	Optimal
	45	300	2.16%	0.13	Optimal
	51	300	1.89%	0.12	Optimal
	57	300	1.85%	0.12	Optimal
	63	300	1.66%	0.12	Optimal
77	39	300	2.91%	0.78	Optimal
	45	300	2.71%	0.33	Optimal
	51	300	1.89%	0.24	Optimal
	57	300	2.00%	0.25	Optimal
	63	300	2.03%	0.2	Optimal
76	39	300	7.65%	300	13.05%
	45	300	6.54%	300	15.60%
	51	300	5.60%	300	81.20%
	57	300	2.83%	300	13.11%
	63	300	1.89%	0.67	Optimal

In Table 6-7 and Table 6-8, the testing cases which eventually reach to optimal solutions are marked in green background, and those reached 5% GAP are marked in light blue background. The GAP used in the tables is relative MIP gap, which is the relative relationship between the best integer solution and the best

bound. For instance, 5% GAP value means the current solution is a feasible integer solution proved to be within 5% of the optimal. According to the testing results, three observations can be made:

1. The problem can be solved faster when energy-saving is not considered.
2. The problem can be solved faster if DR power limit and T_{max} are higher.
3. Most of the test cases reached an optimum or suboptimal solution in 5 minutes' computation.

6.4.3. Further Reducing Computation Time using Cloud Computing

It is worth noting that the experiments above are conducted on a virtual machine emulating single board computer's computation power, this is called edge computing. If the computation time needs to be further reduced, more powerful computer can be used. One example is the cloud computing, especially for a web service called serverless computing. The user for such service can use a powerful computer to run a small code snippet occasionally at very low cost. Such service is very popular now for the IoT industry, examples are Google Cloud Functions and Amazon AWS Lambda Function. Figure 6-10 demonstrates how the edge device, where the BEM system resides, can utilize the serverless computing service, using AWS as an example.

Six steps are explained below:

1. Upon receiving the DR signal, the BEM on edge device starts to collect information needed for the optimization problem and then write them into the Input Table on AWS Dynamo Database;
2. After all inputs have been properly uploaded, the edge device will publish an event to invoke the Lambda Function;
3. Triggered by the event source, the Lambda Function will gather all information needed for the computing from Dynamo Database;
4. Lambda Function starts the computing using the cloud server infrastructure, to solve the MILP problem;
5. Lambda Function writes the solution to an Output Table in the database;
6. BEM system on the edge device will retrieve the optimal control schedule from the database and start to execute the control command.

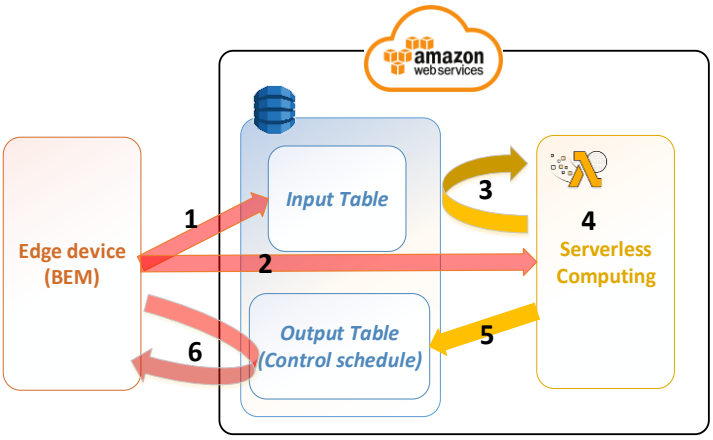


Figure 6-10 Illustration for Using Cloud Computing to Accelerate Online Computing

As of now, there is no cost for using the Lambda Function service in this case: AWS provides free tier service of 400,000 GB-Second every month to all users. That means if configured using the most powerful

computation setting (3008 MB memory and proportional CPU power), it will be free for the first 136,170 seconds of computation. Assuming for each call, the MILP can be solved in 10 minutes, i.e. 600 seconds, this gives more than 200 times of Lambda calls in a month. As a result, this solution will be totally free for most of the SMCB use cases. The estimation above are based on the Lambda Function pricing policy as of the end of 2018[157].

To sum up, in most of the use case of this algorithm, the problems can be solved (either reached optimal or at satisfactory sub-optimal) efficiently, with proper DR power limit and T_{max} settings.

6.5. Real World Application.

Figure 6-11 shows the timeline representation for using the optimization based algorithm in real world DR application. The major steps are elaborate as follow:

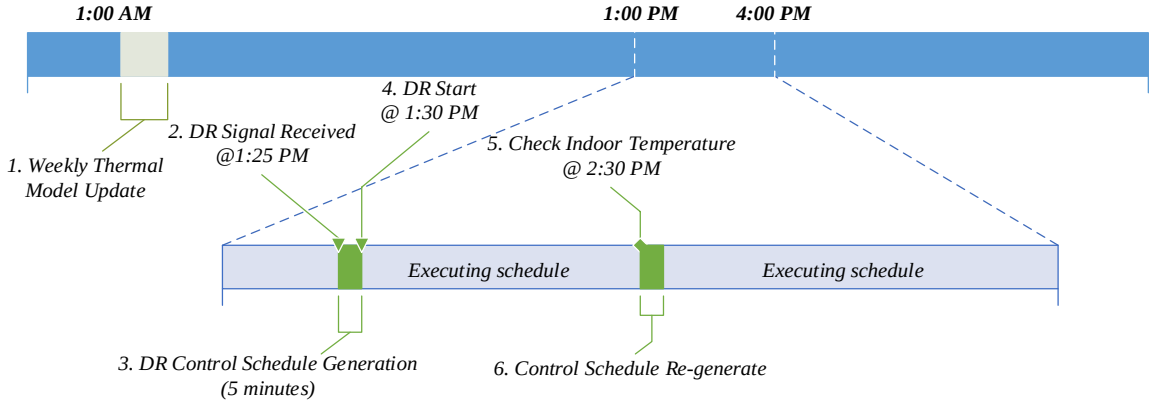


Figure 6-11 Timeline Representation for the Optimization-based Algorithm for DR Control

1. Weekly thermal model update: due to certain extent of occupants' schedule change and weather variation, the thermal model proposed in Chapter 4 should be updated every week or even more frequent. This to avoid the indoor temperature prediction error due to model drift. This model update can be scheduled in weekend late night.
2. Receiving DR signals via OpenADR or other protocols, with detailed information on the scheduled DR event (e.g. start as soon as possible). The communication agent receiving this signal should trigger the optimization process.
3. Optimization process starts with collecting the most recent weather data and linearizing some constraints. After that, an optimization problem is formulated and is sent to the solver to find the optimal solution.
4. When the DR event starts, the AC units' control schedule is ready, during the next hour, the BEM system will coordinate the status of the AC units following the generated schedule.
5. An hour after the DR starts, the system should check the indoor temperature to see if the actual indoor temperature in all zones are as expected.
6. If the actual indoor temperature differ a lot from the planned ones, this means the indoor temperature prediction is not reliable at this moment. Continuing the control with the previously generated schedule might jeopardize the indoor thermal comfort and thus a new control schedule should be re-generated.

7. REINFORCEMENT LEARNING BASED HVAC UNITS COORDINATION CONTROL UNDER DEMAND RESPONSE

For the same AC units' coordination problem, this chapter will tackle it using another optimal decision making tool: the reinforcement learning (RL). RL is a general-purpose framework for artificial intelligence, and is able to be used to model a variety of control problems. It is a learning process based on experience, which helps an agent (usually a software agent) to make sequential optimal decisions at different states about what actions to take to interact with the environment; these optimal decisions, in a form of best policy, will help it gain a maximum cumulative reward in the future. Moreover, it is capable to consider delayed reward due to earlier actions; the capability of understanding such relationship helps the agent make better overall decisions. Because of its fit to the context of the AC units' coordination, this chapter will discuss how to use RL to help determine the operation of the HVAC unit fleet.

7.1. Reinforcement Learning Formulation

In this section, general knowledge of reinforcement learning is reviewed and the formation of the AC unit coordination problem into a RL problem is discussed in detail.

7.1.1. Reinforcement Learning Model

To begin with, some general terms regarding the RL used in this study is summarized.

7.1.1.1 Markov Decision Process (MDP) and RL

In a control problem, the reinforcement learning agent is to make sequential decisions about what action to take at each state to interact with the environment, in order to achieve a maximum future cumulative reward. This can be appropriately modelled with the Markov decision process (MDP). A typical MDP consists of the following elements:

1. **Agent**: The decision maker.
2. **States**: different states/situations agent can be in.
3. **Actions**: different actions agent can take.
4. **Environment**: the system agent is interacting with.
5. **Reward**: gain or loss determined by the environment depend on the agent's state and action, is in scalar form.

Besides, other related terms are shown below:

Policy: A policy determines what actions to take at different states. It is usually represented by a function mapping a state to an action. As shown in (7-1), by following the policy π, the action to be taken at state s is a.

$$a = \pi(s) \tag{7-1}$$

Value function: Value function maps specific state or state-action pairs to the expected future cumulative discounted reward starting from current state. The state-value function and action-value function are shown below:

$$Q^{\pi}(s,a) = E_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s,a] \quad (7\text{-}2)$$

$$V^{\pi}(s) = E_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s] \quad (7\text{-}3)$$

r_{t+i} are the reward at the following steps, and γ is the discount value.

Optimal Policy: An optimal policy is a policy, if follows by an agent, that yields the maximum possible future reward. The comparison between two policies is defined as below:

$$\pi \text{ is better than } \pi' \text{ if } V_{\pi}(s) \geq V_{\pi'}(s), \forall s$$

7.1.1.2 RL Model Training

The training of the RL agent is essentially the process of finding the optimal policy through agent's interaction with the environment. For value-based RL, after agent training, the optimal policy can accurately evaluate (7-2) and (7-3) for all states and state-action pairs. With the optimal policy, the agent then is able to follow it to interact with the environment to achieve maximum reward.

Mathematically, the training of a reinforcement learning model is based on the Bellman Expectation Equation, as shown in (7-4) and (7-5).

$$V^{\pi}(s) = E_{\pi}[r_{t+1} + \gamma V^{\pi}(S_{t+1}) | S_t = s] \quad (7\text{-}4)$$

$$Q^{\pi}(s,a) = E_{\pi}[r_{t+1} + \gamma Q^{\pi}(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \quad (7\text{-}5)$$

To find the expected value of these value function according to the agent learning experience, one popular algorithm is the temporal-difference (TD) learning algorithm. The idea is to use a guess ($V^{\pi}(S_{t+1})$ and $Q^{\pi}(S_{t+1}, A_{t+1})$) to update another guess ($V^{\pi}(s)$ and $Q^{\pi}(s,a)$), which is called bootstrapping. Eventually, the value will converge to the estimated future reward. Detailed explanation can be found in [158].

Q-learning is an off-policy TD learning algorithm, it updates the Q-value iteratively until it converges, and this is very similar to the gradient descent update:

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha(r_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, a_t)) \quad (7\text{-}6)$$

$r_{t+1} + \gamma \max_a Q(S_{t+1}, a)$ is called the TD target and $r_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, a)$ is the TD error. α is the learning rate.

There are two major types of Q-learning implementations: table-based or function approximated. The table-based Q-values representation, as exemplified in Table 7-1, uses a table to store the Q-values of each action-state pairs. During the control, the agent finds out the maximum Q-value of the state it is in, and take the corresponding action.

Table 7-1 Table-based Q-values Representation

	State 1	State 2	…	State X
Action 1	Q11	Q12	…	Q1X
Action 2	Q21	Q22	…	Q2X
…	…	…	…	…
Action k	Qk1	Qk2	…	QkX

But the table-based implementation only suitable for the problem with limited number of states and actions. In many other problems, depending on the definition of state, the number of states can be extraordinarily high or even infinite. Under these circumstances, it would be inefficient or even impractical to save the Q-values on a table; and thus, an alternative know as *function approximation* is used. The idea is using a supervised learning method to approximate the Q-value. In early stage, people use some hand-crafted features obtained from states as input of the approximate function, and in some problems the approximation has very good performance. But finding those features are not easy, and also time consuming. Researchers start to use states directly as an input for the approximation method.

When using deep neural network for the supervised learning, a deep Q-network is formed. The deep Q-network (DQN) uses a neural network to approximate the Q-value for each state-action pair. According to Figure 7-1, the input vector is the state, and the output vector is the Q-values of all possible actions.

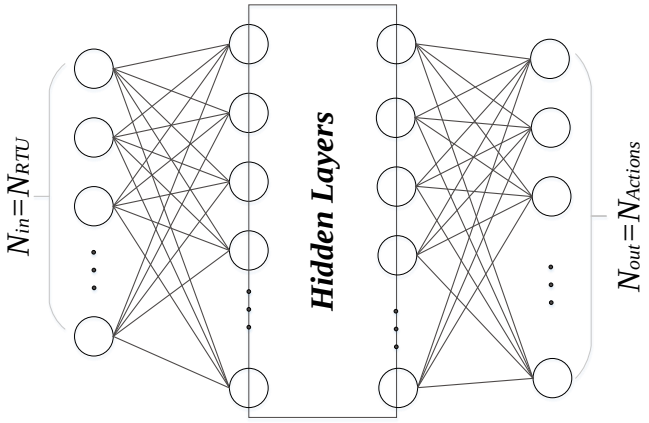


Figure 7-1 DQN Architecture

However, previous study shows that when Q-value is approximated using non-linear functions, the convergence of the Q-function cannot guaranteed. To tackle this, in [159], the authors proposed two techniques that facilitate the convergence of the algorithm; they are: experience replay and fixed Q-targets.

The experience replay helps de-correlate the past experience samples before putting them into the model training. The agent's experience at each time-step are saved in a replay memory buffer. During Q-learning update, sample of experience will be drawn at random from the buffer; the randomness will make sure the decorrelation between adjacent inputs. Several advantages of this technique are summarized in the paper: such as it improves the data efficiency, the decorrelation of samples can reduce the variance of the updates and it reduces the chance of stuck in local minimum or even divergence.

The fixed Q-function target is used to avoid oscillations. Similar to the supervised learning, the target of Q-value is provided by this fixed Q-network. A loss function presented as mean square error between Q-network and Q-learning target will be minimized. Meanwhile, the online Q-network parameters are copied to the fixed Q-network periodically.

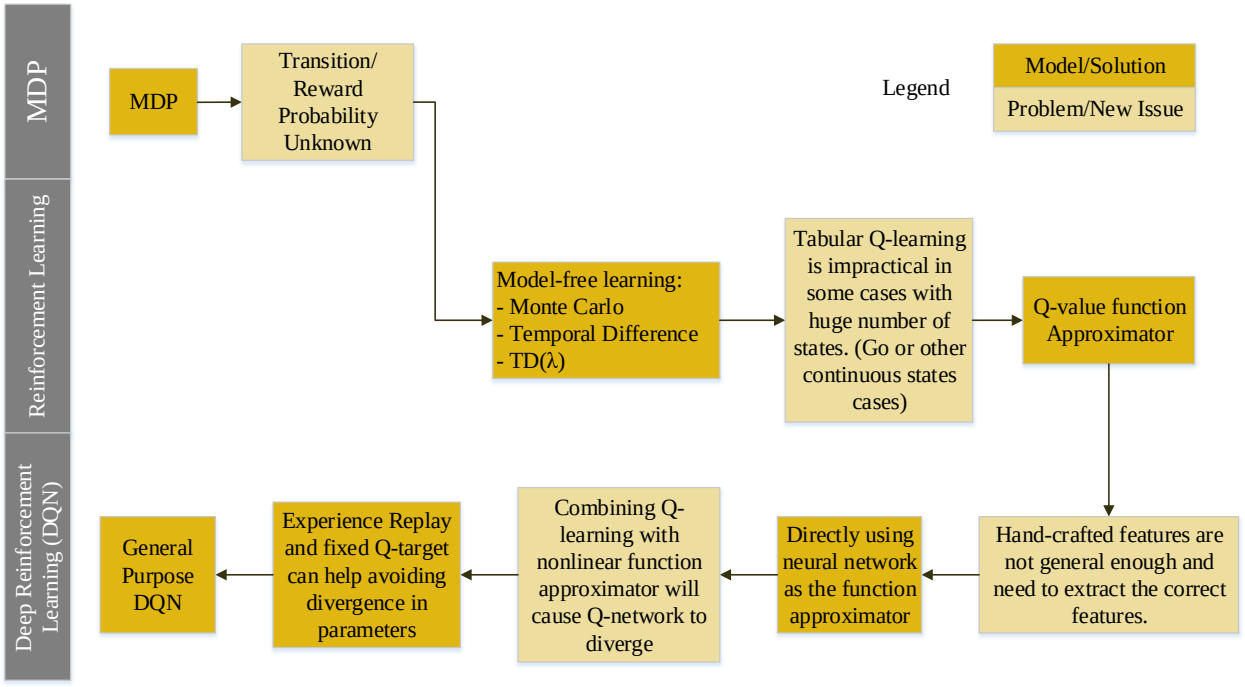


Figure 7-2 Evolution of DQN from MDP

To sum up, Figure 7-2 shows the development path of the DQN model from the original Markov decision process. In the rest of this section, the DQN model is used to train and obtain the optimal policy for the AC units' coordination problem.

7.1.2. Problem Formulation & Reinforcement Learning Model

Many control problems can be modeled as RL problems, including the robotic control, playing Go or Chess, etc. Similarly, the AC units' coordination problem can also be modeled as a RL problem for the following reasons:

1. The control system is a decision making agent that makes step-wise decision.
2. The thermal status of different rooms in the building forms the state.
3. At each time interval, the agent has to decide the status of each AC unit and take actions to control them.
4. Overall thermal dynamic system including outdoor environment and indoor occupancy condition is the environment.
5. Occupants' thermal comfort and energy usage can be modelled as reward and penalty.

From this perspective, the AC unit coordination problem can be well-modeled using the reinforcement learning model: Over the period of a DR event, the ON/OFF status combination of multiple AC units is a problem of sequential decision making. By taking different actions, there will be corresponding thermal discomfort and energy consumption, the goal is to train the agent for the optimal control policy, which will minimize the discomfort level as well as the energy cost.

Figure 7-3 shows the basic components of this problem. In addition, to better illustrate how to solve the AC units' coordination problem using RL, Table 7-2 presents a comparison between the problem formulation of this AC units coordination and that of solving the Atari game using RL [159]. The purpose of this

comparison is to show that the AC units' coordination problem can also be aptly modelled as a Markov decision process, same as the Atari game example, which has been widely recognized can be solved by RL.

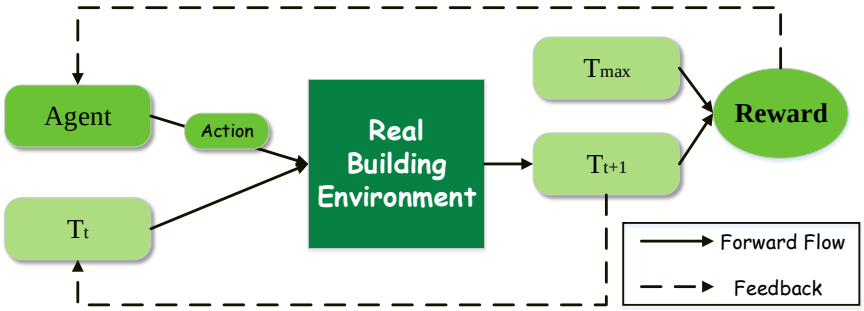


Figure 7-3 Markov Decision Process in the AC Units Coordination Problem

Table 7-2 Reinforcement Learning Problem Formulation

Reinforcement Learning	AC Units Coordination Problem	Atari Game Playing (Pong game)
Agent	Smart building controller	Computer program, the game player
States	The indoor temperature of different thermal zones	Position and velocity of the ball, the location of the player itself
Actions	Selectively turn ON/OFF each AC units	Move the board up/down
Reward/Penalty	A measurement for the thermal comfort and energy consumption	Not missed: reward; Missed: infinite penalty (game over)
Optimal Policy	A series of decisions that maximize the reward over a demand response period	A series of decisions that maximizing the reward by not missing the incoming ball
Environment	All factors influencing the indoor temperature, represented by the building thermal model	The opponent, the angle and velocity of the ball
Horizon	Infinite (until the end of training).	Up to infinite (Until missing the ball)

As a result, an environment is needed, so that the agent can be trained against. The environment can be a real building or a simulator. In this study, a simulator will be developed, and the reasons are discussed in the next section.

7.2. Building Thermal Simulator

Unlike the case with Atari game, in this study, it is not preferable to directly let the agent train with the real world environment, namely the building, for two reasons:

1. In the initial learning phase, the actions taken by agent might cause operational issues in real buildings;
2. DR typically happens infrequently, mostly on extreme weather conditions, using real DR control is slow on data collection and inefficient for the overall learning process.

As a result, in order to collect enough reliable data for training the DQN, a building thermal simulator is need to emulate the environment. The design and development of this simulator is crucial, because after trained with the simulator, the optimal policy should be able to be directly used in a real building control. Therefore, to meet this goal, there are two requirements for the building thermal simulator:

1. Reflecting the truth of building thermal properties;
2. Providing meaningful reward to agent to learn the appropriate control policy.

Requirement 1 is indispensable, meaning the simulator should be acting as close as possible to the real building, so that the RL model trained with this simulator can be directly applied to real building control. Fulfilling Requirement 2 is also vital, it lays the foundation for the agent to implement optimal control.

7.2.1. Thermal Model

Previous chapter has shown that the indoor temperature changes linearly at certain rate, as shown in (4-3) and Figure 4-1; By applying the building thermal model proposed in Chapter 4, it is appropriate to use TIS_i^t and TDS_i^t to represent the temperature variation speed due to AC unit not running (temperature increasing) and running (temperature dropping).

$$TIS_i^t = f_i^r(T_i^t, T_{out}, time, dow, H_{out}, w) \tag{7-7}$$

$$TDS_i^t = f_i^d(T_i^t, T_{out}, time, dow, H_{out}, w) \tag{7-8}$$

To simplify the problem, typical values of time, day, outdoor temperature/humidity and weather type of a demand response event is used. By doing this, the temperature variation rate in a typical DR event day is a function of only the indoor temperature.

Therefore, the theoretical value of the temperature at next time step can be expressed as:

$$\overline{T}_i^{t+1} = T_i^t + (1 - s_i^t) \cdot TIS_i^t + s_i^t \cdot TDS_i^t \tag{7-9}$$

s_i^t is a binary value representing the status of the AC unit (1 for ON and 0 for OFF).

However, the actual temperature might diverge from the theoretical calculated value because of some stochastic reasons (different occupancy level, etc.). A Gaussian distribution is used to represent the distribution of the actual next step temperature, and a value will be sampled by the simulator.

$$T_i^{t+1} \sim N(\overline{T}_i^{t+1}, \sigma^2) \tag{7-10}$$

The value of σ can be determined by analyzing the indoor temperature prediction performance, as discussed in Chapter 4.

From the perspective of reinforcement learning, the building thermal model will decide which state the agent will go to given current state and action taken; i.e., the simulator is acting as the environment.

7.2.2. Reward System Design

Reward system design is a key part of this study. Since the agent in the reinforcement learning model learns what action to take to maximize future reward, the design of reward system will ultimately influence the behavior of the agent and the decisions made. As a result, the reward system should be designed such that it will lead the agent to take actions that fulfill the following requirements:

1. Keeping temperature in all rooms under the limit if possible; also in general close to each other's (so that it will not be a case that one room is much hotter than another.)
2. If the temperature in one room exceeded the limit and start causing severe thermal discomfort to the tenants, the agent should be able to take immediate actions so that the temperature in this room will decrease, ameliorating the thermal condition in this room.

3. When one room's temperature is higher than another room's and is about to exceed the limit, it should have higher priority to be cooled than the other room.
4. If energy saving is considered, the agent should be able to maintain the thermal comfort using as less electricity as possible.

In this section, a reward system consists of the thermal comfort margin increment, violation penalty and energy factor is introduced. Consider all these three factors, all the four requirements above will be satisfied.

A. Thermal comfort margin reward

As proposed in Chapter 6, a temperature limit named maximum tolerable temperature ($T_{\max}$), is specified. In general, it is ideal when the temperature in each room is below $T_{\max}$. When the indoor temperature is close to $T_{\max}$, the thermal comfort threshold might be easily breached. As a result, thermal comfort margin is defined here to measure the distance between indoor temperature and $T_{\max}$. During a DR event, the total thermal comfort margin of all rooms should be maximized.

One simple definition of the thermal comfort margin is the linear distance between indoor temperature and $T_{\max}$. In Figure 7-4, assuming the arrow direction is the one represent temperature increasing, then Temp1 and Temp2 is higher than $T_{\max}$. M1 and M2 are the margins for indoor temperature Temp1 and Temp2, as defined in (7-11), they have negative values. The process of Cooling1 brings the temperature down from Temp2 to Temp1, meaning an increase in the thermal comfort margin (margin increased, though still negative). The reward for cooling can be defined as the increment of thermal comfort margin; it becomes penalty when the value is less than 0.

$$Margin = T_{\max} - Temp \tag{7-11}$$

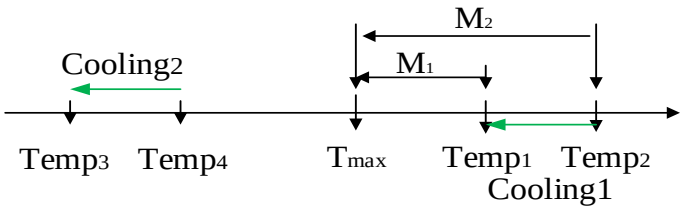


Figure 7-4 Linear Thermal Comfort Margin

Though this definition of thermal comfort margin is straightforward, it fails to help the agent to establish a sense of priority. For instance, in Figure 7-4, the process of Cooling2 brings the indoor temperature from Temp4 to Temp3; assuming the following relationship exists:

$$Temp_4 - Temp_3 = Temp_2 - Temp_1 + \varepsilon \tag{7-12}$$

ε is a close to 0 positive number, then the reward of Cooling1 will be slightly smaller than that of Cooling2 and drive the agent to make decision to cool the room with indoor temperature of Temp4 instead of that with Temp2. Apparently, this decision is in violation of the Requirement 2 mentioned at the beginning of this section, because the room with indoor temperature of Temp4 should have higher priority than the one with Temp4.

To tackle this problem, an alternative non-linear thermal comfort margin is introduced, with definition shown as (7-14).

$$x = T_{\max} - Temp \tag{7-13}$$

$$Margin(x) = \xi \cdot \int_0^x f(z)dz \tag{7-14}$$

$f(x)$ is a decreasing exponential function, that means the reward for decreasing one degree of indoor temperature will increase exponentially with the increase of indoor temperature, as shown in Figure 7-5. In other words, agent now will gain much higher reward to cool a hotter room instead of a cooler room, and thus satisfy the aforementioned Requirement 2. ξ is used to prioritize different thermal zone.

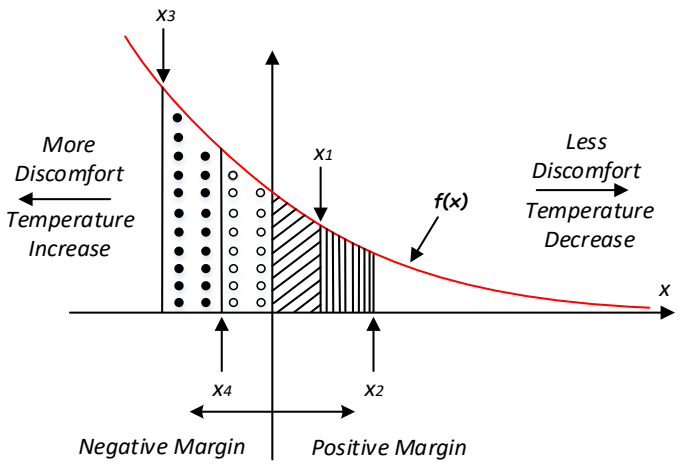


Figure 7-5 Non-linear Thermal Comfort Margin

(Margin(x2)-Margin(x1) < Margin(x4)-Margin(x3), given x4-x3=x2-x1)

In all, the reward related to thermal comfort margin of Zone i at time t can be expressed by (7-15):

$$m_t^i = \xi \cdot \int_0^{T_{\max} - Temp_{t+1}^i} f(x)dx - \xi \cdot \int_0^{T_{\max} - Temp_t^i} f(x)dx \tag{7-15}$$

B. Violation penalty

Considering Requirement 3, the agent should be capable to distinguish which rooms have the tendency to exceed $T_{\max}$ and should cool this room before it is too late. Therefore, a violation penalty is added to the reward if the temperature just exceeded $T_{\max}$, because the agent did not cool the room in time. The violation penalty of Zone i at time t is defined as (7-16).

$$v_t^i = \begin{cases} C & (Temp_t^i < T_{\max} \ \& \ Temp_{t+1}^i > T_{\max}) \\ 0 & (else) \end{cases} \tag{7-16}$$

C is a constant penalty, which applies per violation per room.

C. Energy usage penalty

When energy saving is considered, an energy usage penalty should be added. This is useful for the case that when a room's temperature is already low, the additional cooling become unnecessary since it consumes energy yet cannot bring significant thermal comfort improvement. As a result, the penalty will discourage agent to do such unnecessary cooling.

To bridge the tradeoff between thermal comfort and energy consumption, a weight λ^i is introduced to map the energy consumption to a penalty value, as shown below. In (7-17), E_t^i is the energy used, measured in kWh, P^i is the power consumption of the HVAC unit and E_t^i is its status at time t.

$$e_t^i = \lambda^i \cdot P^i \cdot s_t^i \cdot \frac{300}{3600} = \lambda^i \cdot E_t^i \tag{7-17}$$

The value of λ^i is dependent on the building engineer's expectation of the thermal comfort level of Zone i. To start with, first define an economic temperature $Temp_{eco}$ as the minimum indoor temperature the building engineer is willing paying for. For instance, $Temp_{eco} = 76$ means the building engineer thinks the energy it cost to cool from 77℉ to 76℉ is worthwhile and consider temperature below 76℉ is unnecessarily cool. Then λ^i is determined by the ratio below.

$$\lambda^i = \frac{Margin(Temp_{eco}) - Margin(Temp_{eco}+1)}{E_{Temp_{eco}+1 \to Temp_{eco}}} \tag{7-18}$$

Thermal comfort margin is calculated using the aforementioned non-linear function. Figure 7-6 exemplifies the margin increment by cooling from 77℉ to 76℉ (assuming $T_{\max} = 78$).

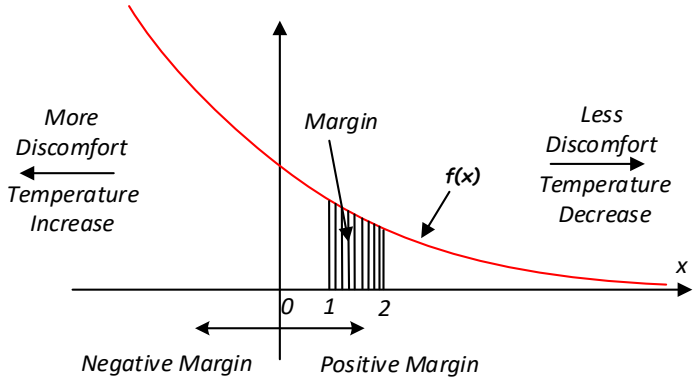


Figure 7-6 Use Margin Increment to Determine λ^i

The amount of energy used for cooling from $Temp_{eco}+1$ to $Temp_{eco}$ is determined by (7-19). $t_{Temp_{eco}+1 \to Temp_{eco}}$ is the time (hour) it takes for the cooling, can be calculated using the thermal model.

$$E_{Temp_{eco}+1 \to Temp_{eco}} = P \cdot t_{Temp_{eco}+1 \to Temp_{eco}} \tag{7-19}$$

The rationale behind this definition is following:

1. When indoor temperature is higher, the room will be cooled faster, plus the non-linear function, it results in a higher comfort margin reward. For $Temp > Temp_{eco}+1$, $m_t - e_t > 0$ will encourage the agent to cool the room;
2. In contrast to 1, for $Temp \le Temp_{eco}+1$, there will be $m_t - e_t \le 0$, meaning the thermal comfort reward is not worthwhile the energy cost and further discourage the agent for cooling this room;
3. Different rooms can have different economic temperature setting, thus, setting a lower $Temp_{eco}$ will prioritize a room.

Finally, the single step reward at time t is defined as:

$$r_t = \sum_i^N m_t^i - v_t^i - e_t^i \tag{7-20}$$

This reward r_t will be sent back to the agent from the building thermal simulator, and later be used during DQN training.

7.3. RL Model Training

In general, during the DQN model training, the agent will interact with environment, and continuously learn the best control policy according to its experience. The Q-value of each state-action pair will be learnt using deep Q-learning. The DQN and learning approach for the HVAC units' coordination problem are discussed below.

7.3.1. DQN in This Study

Recall in earlier this chapter, the structure of the Deep Q-Network (DQN) is discussed. It is a neural network mapping the state to the Q-values of different actions. In this section, the states and actions in the AC units' coordination problem are investigated.

7.3.1.1 States

The state is defined as $[Temp_t^1, Temp_t^2, \cdots, Temp_t^N]$, the collection of indoor temperature of different rooms.

7.3.1.2 Legal Actions

The action is defined as $[s_t^1, s_t^2, \cdots, s_t^N]$, the combination of AC status at time step t. Considering that $s_t^i \in \{0,1\}$, the number of the action grows exponentially with the number of the HVAC units to be controlled. However, not all combinations are suitable during a DR event, because the total power consumption under certain combinations is larger than the DR power limit. As a result, only the combinations resulting in total power consumption less than DR limit is are considered. These combinations are called *legal actions*.

According to the conclusions of Chapter 5, the power consumption of an AC unit is dependent on outdoor temperature. As a result, when provide the power consumption information to the reinforcement learning model, a typical average outdoor temperature during DR events should be used to calculate them.

$$P^i = \omega_i \cdot Temp_{out} + b_i + f_i \tag{7-21}$$

Before the training, the power consumption of all the N AC units are provided, i.e. $[P^1, P^2, \cdots, P^N]$.

In addition, a DR power limit of all AC units P_{AC} (in kW) is given according to an agreement between the building and the utility company. The set of all legal actions is expressed by (7-22).

$$A = \{\mathbf{a} \mid \mathbf{P} \cdot \mathbf{a} = \sum_i^N P^i \cdot s_t^i < P_{AC}, s_t^i \in \{0,1\}\} \tag{7-22}$$

All actions in A are legal, meaning controlling AC units according to any action in A will not result in a total power consumption larger than the DR power limit. In contrast, for instance [1, 1, … , 1] usually are not legal action since when turn on all AC units, the power consumed is larger than DR limit.

In all, the DQN input has a dimension of N and the output has a dimension equals to the number of the legal actions. In contrast to training a DQN for playing Atari game using raw pixels input, the DQN in this paper has comparatively smaller size of input and output, and simpler input. In practice, the network can be shallower.

7.3.2. Training Paradigm

The same training paradigm in [159] is applied in this study. Two key implementations, experience replay and fixed Q-targets, are used to facilitate the convergence of the algorithm. By interacting with the environment, the agent will save its experience, namely current state, action taken, reward/penalty obtained and next state, to a buffer. During the training, the experience will be randomly sampled from the buffer to train the DQN. The randomness here will help de-correlate the sample used in training. Two sets of DQN is used: fixed target Q-value and online learning Q-value. Fixed Q-target is used to improve the stability of the algorithm, and will copy the online DQN at certain interval.

7.3.3. Exploration and Exploitation

Because in reinforcement learning, the agent learns from experience; this means the agent need to explore different actions at various states, and finally according to the reward received to decide which the best actions at different states are. After exploring enough, the agent will use the actions to the best of its knowledge to interact with the environment, which further makes the DQN converge. Therefore, in the early stage of the model training, the agent should explore more while in later stage exploit more is a better option.

In practice, a probabilistic action selection approach called ε -greedy policy. Based on this policy, an action is selected using the following relationship:

$$a_t = \begin{cases} random\ action\ (with\ probability\ \varepsilon) \\ \max_a Q(s_t, a, \theta)\ (otherwise) \end{cases}$$

The probability ε is not constant, according to [56], the following form is used to determine ε , given the desired range of $(\varepsilon_{\min}, \varepsilon_{\max})$:

$$\varepsilon = \max(\varepsilon_{\min}, \varepsilon_{\max} - (\varepsilon_{\max} - \varepsilon_{\min}) \times \frac{step}{\varepsilon - decay}) \tag{7-23}$$

7.3.4. Control Policy

When the training is finished, the parameters for DQN as well as the optimal control policy have been determined. Soon as the DR starts, the controller first get the temperature readings from all zones and other real time information, then feed this combined information as state to the neural network, feed-forward calculate the Q-value for each of the legal actions. Choose the action with the highest Q-value and control the corresponding AC units.

7.4. Case Study

Similar to the case study in Chapter 6, in this case study, four AC units in a building will be coordinated controlled during a DR event, with an AC power limit of 13 kW and $T_{\max}$ equals to 78℉.

7.4.1. Simulator Development and Training Settings

7.4.1.1 Simulator Development

A building thermal simulator is developed in Python, integrated with the building thermal model (Chapter 4) and HVAC power model (Chapter 5). To be specific, the simulator is implemented in a Python class, which has the functions and their inputs/outputs as shown in Table 7-3.

Table 7-3 Python Functions in the Simulator Class

Function	Input	Output
state_random_initialize	Number of thermal zones, indoor temperature distribution	A vector with dimension of thermal zones' number, each element is the initial indoor temperature of the corresponding thermal zone.
get_legal_actions	Number of thermal zones, HAVC power model, typical outdoor weather condition. HVAC DR power limit.	An array, each element in the list is a valid HVAC control operation, with the total power consumption less than the power limit.
_margin	Current indoor temperature	Non-linear thermal comfort margin
step	Current state, action taken	Next state and a scaler reward for taking the action
_get_tds	Current state, building thermal model	Temperature decreasing speed for all thermal zones under their current temperature
_get_tis	Current state, building thermal model	Temperature increasing speed for all thermal zones under their current temperature.

Some of these functions are private, which means they will only be called internally from the class by other functions. For instance, those functions for getting thermal comfort margin of current states and temperature variation speed are private. Other functions, i.e., get_legal_actions, state_random_initialize and step, are interfaces to the DQN training framework. For example, the function 'get_legal_actions' will return all legal actions and provide them to agent as all possible choices; the function 'step' will take current state and the action chose by the agent and determined the next state (what temperature each thermal zone will have at the next time step) and the reward for taking this action.

To get all legal actions in this case study, assuming under the typical DR weather, using the model developed in Chapter 5, the power consumption of four AC units are:

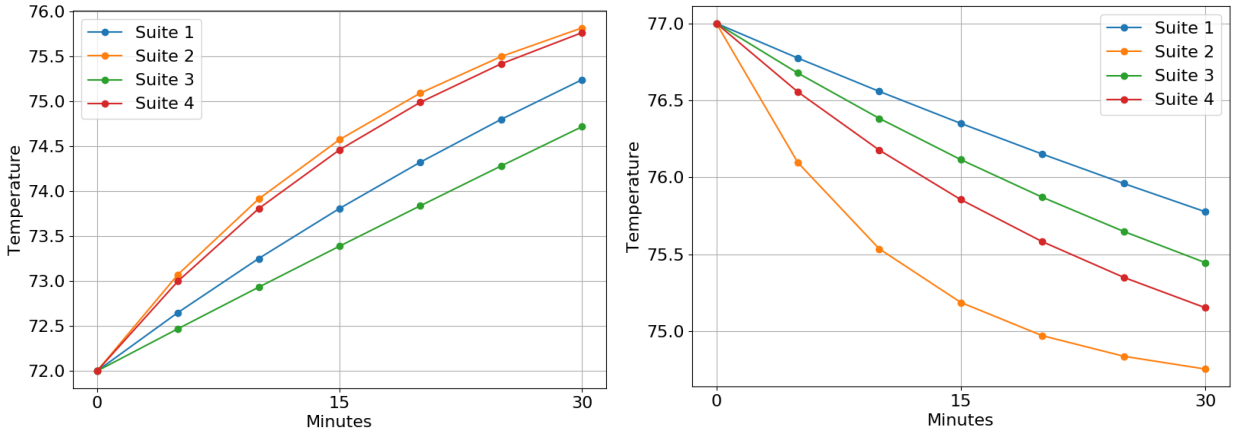
$$[P^1, P^2, P^3, P^4] = [8.5, 7.0, 12.0, 4.5] \tag{7-24}$$

Considering the agreement between the building owner and the utility company specifies that during DR period, these AC units' total power consumption should be lower than or equal to 13 kW at any time ($P_{AC} = 13$). As a result, there are seven legal actions available to the agent:

$$A = \{\mathbf{a} \mid \mathbf{P} \cdot \mathbf{a} = \sum_{i}^{N} P^i \cdot s_t^i < P^{DR}, s_t^i \in \{0,1\}\} = \{0000, 0001, 0010, 0100, 0101, 1000, 1001\} \quad (7\text{-}25)$$

The four-digit numbers in (7-25) are binary representations of the operation condition for four HVAC units. For instance, '0000' means all HVAC units are not cooling and '0101' indicates the second and fourth units are cooling while the first and third are not. Other combinations, such as '1101', cannot be chosen by the agent because they will results in a breach on the power limit. It is worth noting that the action number grows exponentially (base 2) with the number of AC units; however, the power limit helps reduce it to a large extent.

The thermal model used in the simulator is learnt from historical thermostats' data, as described in Chapter 4. Figure 7-7 demonstrates the four thermal zones' temperature change when the AC units are turned ON and OFF for half an hour. It shows that they have different behavior: different temperature profile from the same start point.



(a) AC units turned OFF for 30 minutes (b) AC units turned ON for 30 minutes

Figure 7-7 Four Thermal Zones Have Different Thermal Property

As indicated by (7-10), certain level of randomness is added in the simulator to emulate the inaccuracy of indoor temperature forecasting model. In the following study, two choices of the standard deviation σ are discussed:

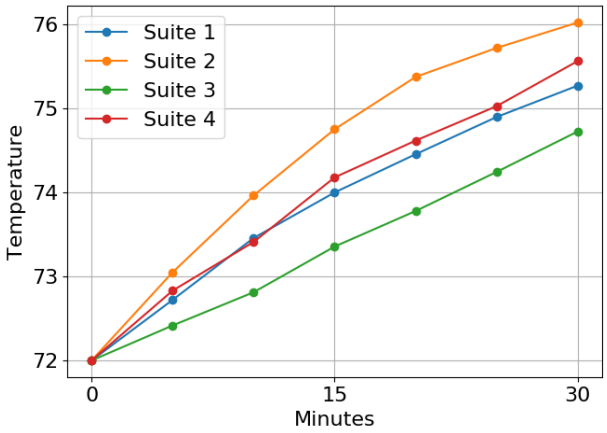
I. σ is assumed to be 20% of the temperature variation in this 5 minutes (300 seconds). For instance, the TIS error standard deviation is assumed as below:

$$\sigma_{TIS} = TIS \times 300 \times 20\% \quad (7\text{-}26)$$

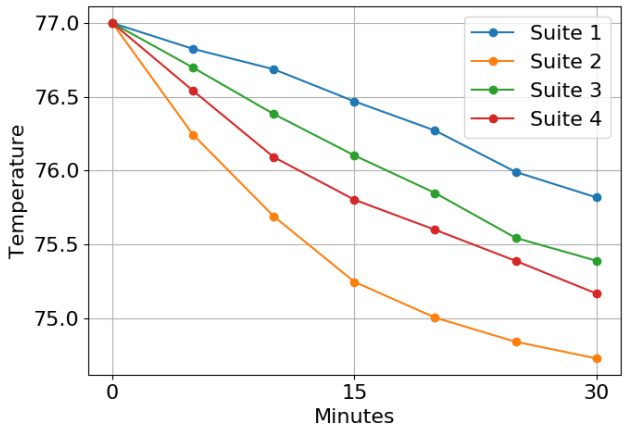
Similar for the TDS error.

II. σ is assumed to be 0.2 ℉ arbitrarily for all cases.

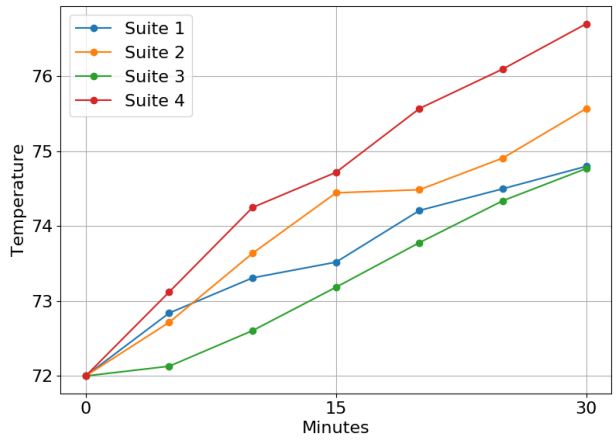
In general, error using Assumption I standard deviation is smaller than that uses Assumption II standard deviation. Figure 7-8 shows the examples of considering randomness in the building thermal models using both Assumption I and Assumption II.



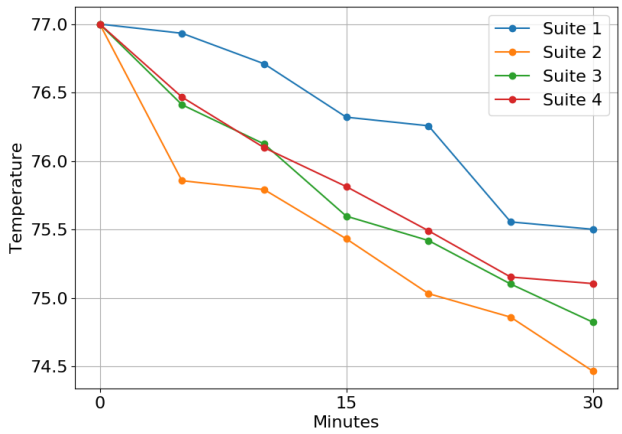
a) AC units turned OFF for 30 minutes (Assumption I)



b) AC units turned ON for 30 minutes(Assumption I)



c) AC units turned OFF for 30 minutes (Assumption II)



d) AC units turned ON for 30 minutes(Assumption II)

Figure 7-8 Four Thermal Zones Have Different Thermal Property (Considering Temperature Prediction Error)

The reason for considering two levels of error in this study is to find out how the controller will react if the indoor temperature prediction model has higher error than expected. In later study, the simulator will consider error level with Assumption II during the training process, but will use both Assumption I and Assumption II in the testing cases.

The simulator in this study uses the thermal comfort margin function as shown in (7-27), and the thermal margin gain by cooling 1℉ from different temperature are exemplified in Table 7-4.

$$Margin(x) = 2 \cdot \int_0^{T_{\max} - Temp} 2^{-x} dx \tag{7-27}$$

The constant penalty in (7-16) is 10 in this study. This value is chosen according to the following two principles:

1. This penalty should be much higher than the thermal margin gain around $T_{\max}$, so that it appears to be a big penalty and strongly discourage the breach of $T_{\max}$;

2. This penalty shouldn't be too high, because when indoor temperature is already over T_{max}, the agent will not be willing to bring it back below T_{max} since it might cause a huge penalty if T_{max} breach happens again.

Table 7-4 Thermal Margin Gain by Cooling 1 ℉ at Different Temperature

Temperature reduction (1 ℉)	Reward	Temperature reduction (1 ℉)	Reward
85→84	184.66	79→78	2.89
84→83	92.33	78→77	1.44
83→82	46.17	77→76	0.72
82→81	23.08	76→75	0.36
81→80	11.54	75→74	0.18
80→79	5.77	74→73	0.09

Therefore, in the following study, the constant penalty is 10.

7.4.1.2 Training Setting

Considering the input and output size of this problem (4 inputs & 7 outputs) is pretty small, the structure of the neural network is simple: a two-layer neural network with 20 neurons on each layer is used in the following study. ReLU is used as the activation function in the network.

```
NUM_OF_NEURONS_LAYER_1 = 20
NUM_OF_NEURONS_LAYER_2 = 20

def q_network(X_state, name):
  with tf.variable_scope(name) as scope:
    hidden1 = tf.layers.dense(X_state, NUM_OF_NEURONS_LAYER_1, name='hidden1',
                              activation=tf.nn.relu,
                              kernel_initializer=initializer)
    hidden2 = tf.layers.dense(hidden1, NUM_OF_NEURONS_LAYER_1, name='hidden2',
                              activation=tf.nn.relu,
                              kernel_initializer=initializer)
    outputs = tf.layers.dense(hidden2, n_legal_action,
                              kernel_initializer=initializer)
  trainable_vars = tf.get_collection(tf.GraphKeys.TRAINABLE_VARIABLES, scope=scope.name)
  trainable_vars_by_name = {var.name[len(scope.name):]: var for var in trainable_vars}
  return outputs, trainable_vars_by_name
```

Figure 7-9 Code Snippet of DQN used in the study

For training the DQN, optimizer using momentum is utilized, with momentum set as 0.95. Other parameters used in the study are shown in Table 7-5.

Table 7-5 Parameters Used in Model Training

Total training step	4,000,000	Training start	After 10,000 iterations
Replay buffer size	50,000	ε -decay factor	2,000,000
ε_{max}	1.0	ε_{min}	0.1
Batch size	64	Copy step	10,000

Figure 7-10 shows the change of the ε probability, calculated using (7-23). It shows the probability that agent will explore is high at the beginning and linearly decreases until reaching ε_{min}.

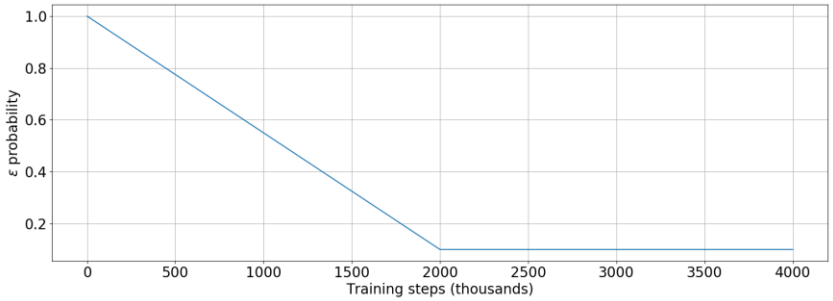


Figure 7-10 Agent's Exploration Probability Changes with Training Steps

Finally, there are three options for choosing the control horizon:

1. Fixed length: step number equals to the DR period length divided by control interval
2. Indefinite length: step number is undefined, the session stops whenever any room's temperature exceeds $T_{\max}$ (a game over sign)
3. Infinite length: the agent interacts with the simulator until the end of training

For Option 1, the trained policy will only be optimal for a fixed length of DR event, meaning if the model is trained on a 2-hour DR event, it might not be optimal if in reality the DR actually lasts for 4 hours. For Option 2, since the session ends every time when $T_{\max}$ limit is breached, then if in reality, during a real time control, the temperature of one room is over this threshold, the optimal policy cannot tell the agent what to do to bring the temperature back to normal because it has never been trained for this condition. Therefore, Option 3 is used in this study.

Finally, using the abovementioned configuration, the agent interacts with the simulator and train the DQN using the experience obtained. As shown in later section, two cases with and without energy saving consideration are trained and tested separately.

7.4.2. Results of DR Event Control I (w/o energy saving consideration)

After the DQN is trained, the learnt optimal control policy is tested in an emulated demand response event, which happens during 12:00-15:00. Since energy saving is not considered in this case, $\lambda^i = 0$ in (7-17) for all thermal zones. That is to say the controller is controlling AC units to achieve maximum thermal comfort regardless of energy cost.

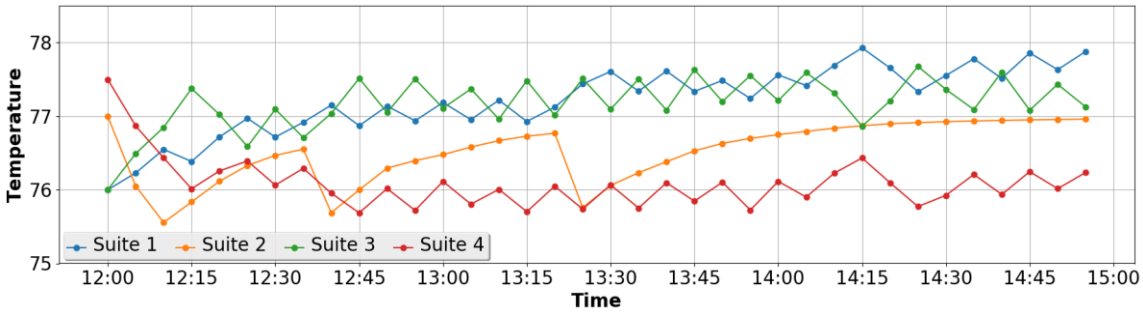


Figure 7-11 Temperature in Four Rooms During a 3-hour DR Period (w/o Energy Saving, Assumption I Error)

Figure 7-11 shows the result of the AC units' coordinated control. At the start of the DR period, the temperature in four rooms are [76.0, 77.0, 76.0, 77.5], during the 3-hour period, the smart building controller is able to maintain the temperature under a maximum tolerable temperature of 78℉.

To visualize the decision-making process, Figure 7-12 shows the Q-values of all the state-action pairs (36 steps/states and 7 actions: 36*7=252). From the figure, apparently, the Q-values of all state-action pairs with action '0000' are the lowest (with the darkest color). This is because when no AC unit is turned on, temperature in all rooms are increasing, which damage the thermal comfort margin and thus always causing penalty instead of reward. Similarly, actions '0001', '0100' and '1000' also yield low Q-value among various states. This is because instead of cooling just one room, cooling two instead always generates more reward at 'no cost' (because energy cost is considered). Therefore, choosing action '0101' is better than action '0001' and '0100' alone; Action '1001' is better than action '0001' and '1000' alone. This also explains the low temperature in Suite 4 in Figure 7-11: whenever Suite 1 or Suite 2 need to be cooled, Suite 4 will be cooled as well even though the indoor temperature of it is already lower.

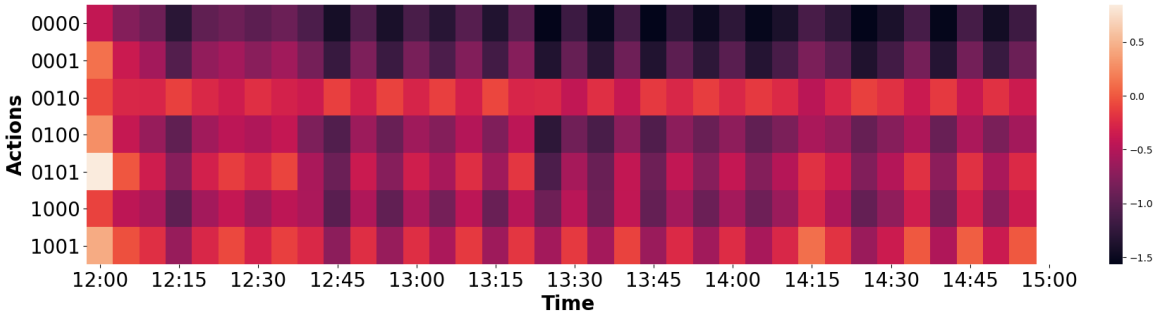
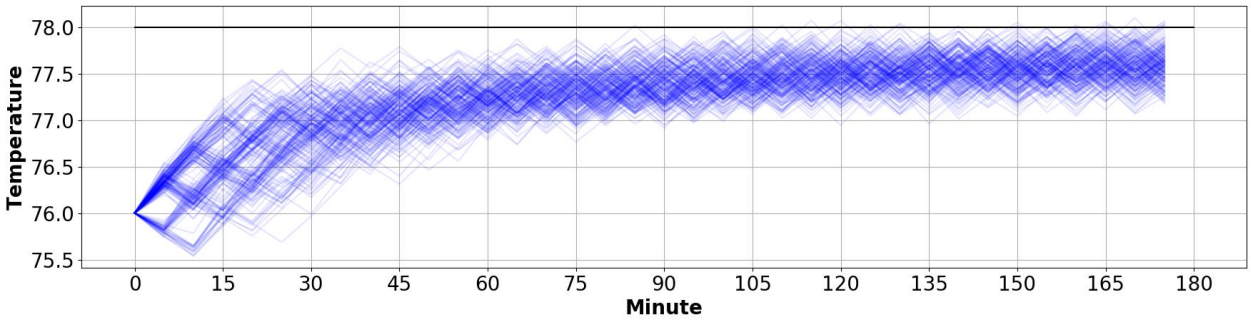


Figure 7-12 Heat Map of Q-values of State-Action Pairs in the DR Period (w/o Energy Saving, Assumption I Error)

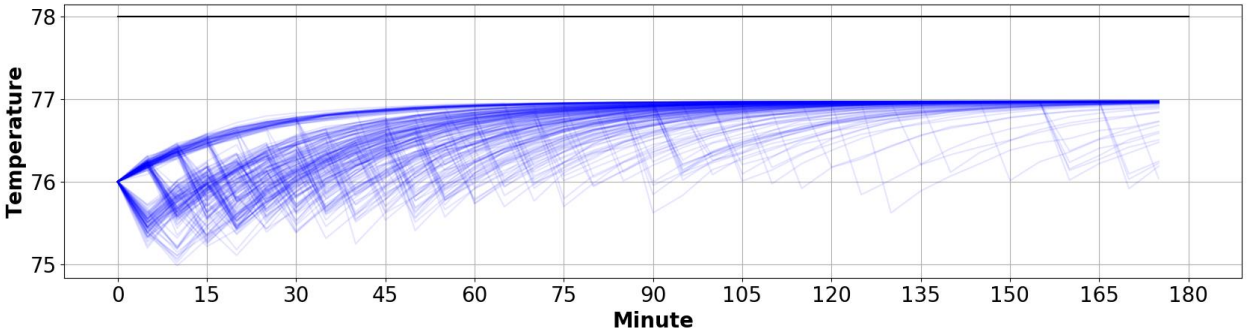
To further investigate the effectiveness of this algorithm, a series of Monte Carlo simulation is conducted according to the following described procedure:

1. Fix the initial temperature of the thermal zone of interest, for instance 76℉.
2. Sample the initial temperature of the other thermal zones according to a distribution, e.g., uniform distribution from 75℉ to 78℉.
3. Start a three-hour coordinated control simulation and obtain the indoor temperature profile of the thermal zone under investigation.
4. Repeat the above steps for multiple times, e.g., 200 times. Plot the indoor temperature profiles of all 200 times' simulation.
5. Repeat the above steps for all thermal zones.

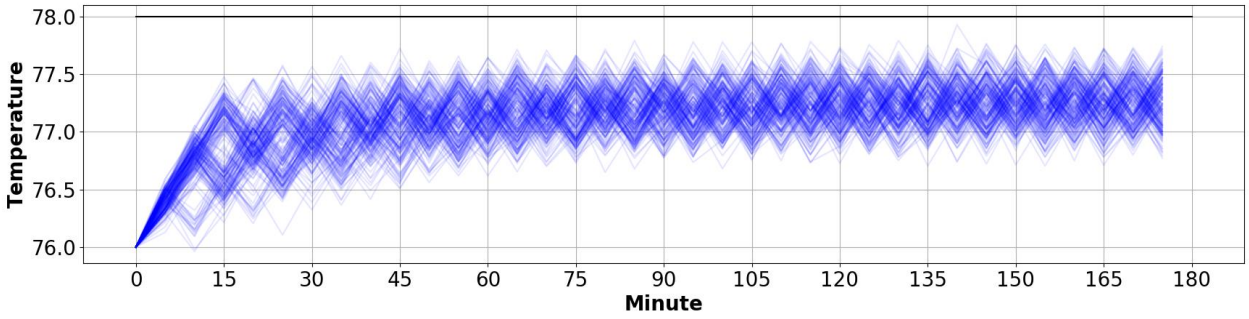
The results of the simulations with a smaller temperature prediction error (Assumption I in Section 7.4.1.1) are shown in Figure 7-13.



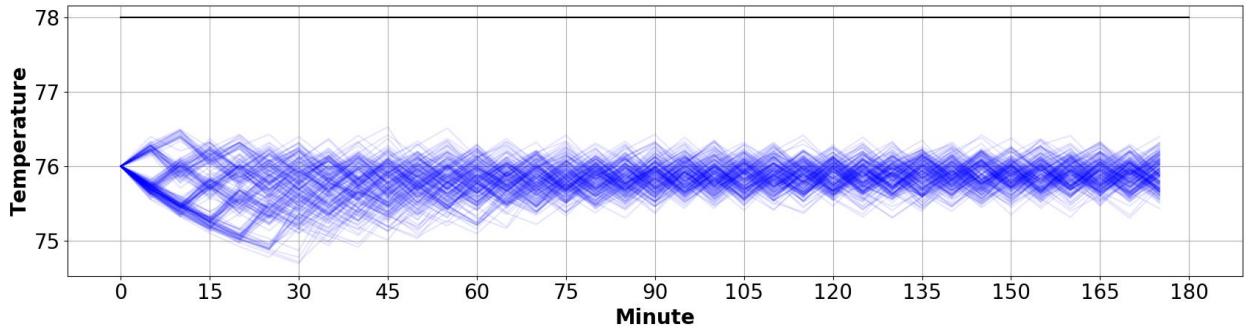
(a) Temperature profiles of Suite 1 in 200 simulations



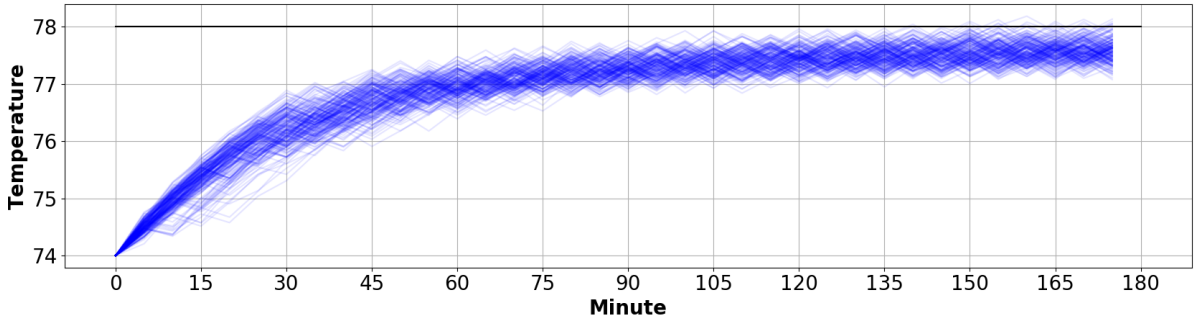
(b) Temperature profiles of Suite 2 in 200 simulations



(c) Temperature profiles of Suite 3 in 200 simulations



(d) Temperature profiles of Suite 4 in 200 simulations



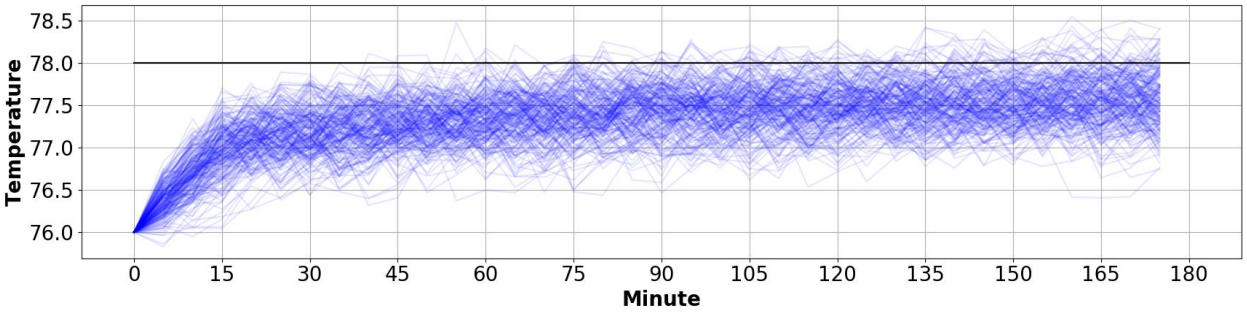
(e) Temperature profiles of Suite 1 in 200 simulations with initial temperature of 74℉

Figure 7-13 Thermal Zones' Temperature Profile in Monte Carlo Simulations (w/o Energy Saving, Assumption I)

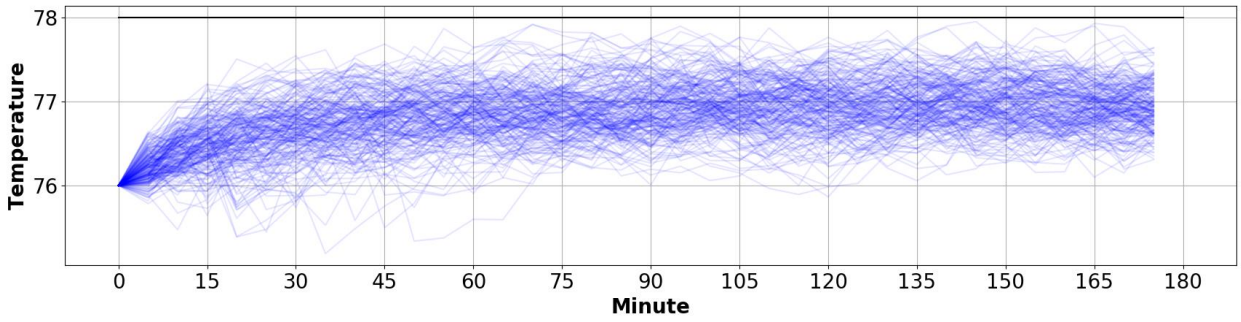
According to Figure 7-13, most of the time in all thermal zones, the indoor temperature can be maintained under the maximum tolerable temperature T_{max}, as indicated in the black horizontal line. There are times when the indoor temperature exceeds T_{max}, but most likely, it will reduce below this threshold due to the high reward for cooling this thermal zone.

Figure 7-13 (e) shows the temperature profiles of Suite 1 when the initial temperature is 74℉. Compared with Figure 7-13 (a), though the temperature is initially lower, it eventually stabilized at the same temperature range (77℉-78℉). This is because at the beginning, when the indoor temperature of this thermal zone is low, the temperature in other zones are higher, so the controller decides to cool other zones due to the higher reward. This situation remains until the temperature of Suite 1 also reach to a higher level when the cooling of Suite 1 results in comparative reward as cooling other zones.

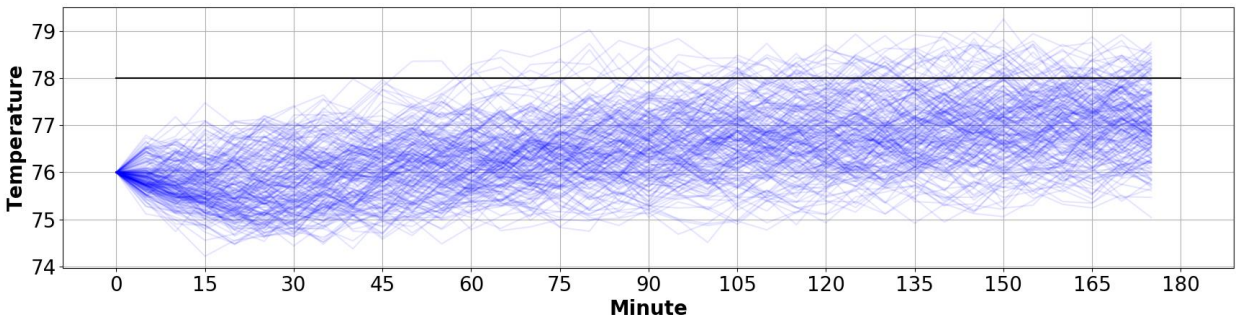
Similarly, the Monte Carlo experiments using a larger error level (Assumption II error in Section 7.4.1.1) are also conducted, results are shown in Figure 7-14.



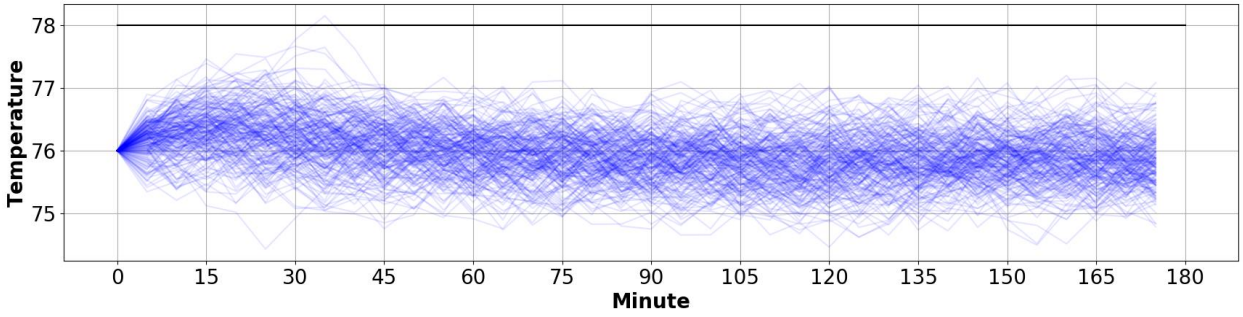
(a) Temperature profiles of Suite 1 in 200 simulations



(b) Temperature profiles of Suite 2 in 200 simulations



(c) Temperature profiles of Suite 3 in 200 simulations



(d) Temperature profiles of Suite 4 in 200 simulations

Figure 7-14 Thermal Zones' Temperature Profile in Monte Carlo Simulations (w/o Energy Saving, Assumption II)

According to the results, the temperature profile becomes more erratic due to the higher level of prediction error. However, because the agent take action at each step according to the specific state it is in, a step-wise correction effect can help to keep all room's temperature under the limit. This shows the robustness of the proposed algorithm.

7.4.3. Results of DR Event Control II (w/ energy saving consideration)

In cases where energy saving is also considered during a DR event, the energy-margin weight λ^i for each zone is needed. For the building engineers, they only need to specify what the economic temperature $Temp_{eco}$ is, then the system is able to calculate λ^i using (7-18). The importance of each thermal zone can be differentiate by the value of $Temp_{eco}$, the zone has a priority to be cooled if the value is lower.

Figure 7-15 and Table 7-6 give an example of calculating λ^i for each thermal zone: assuming the economic temperature is set to be the same for all zones (76℉). The time it takes to cool each zone from $Temp_{eco}+1$ to $Temp_{eco}$ is determined by the thermal models, as shown in Figure 7-15.

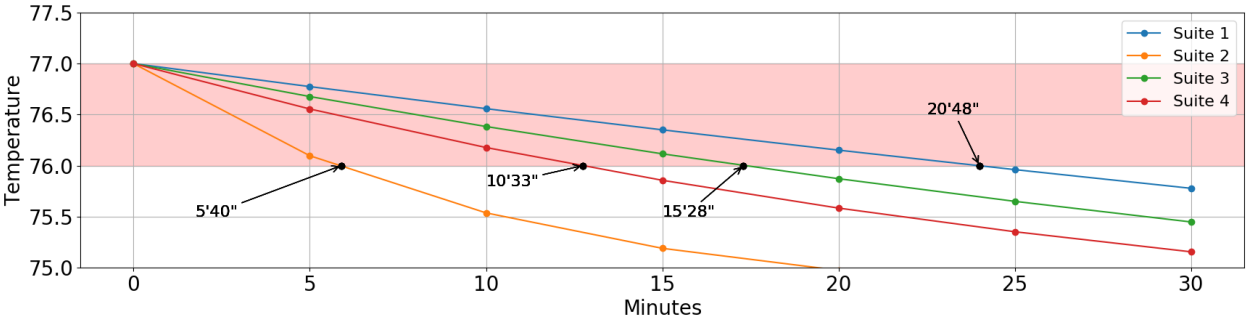


Figure 7-15 Time for Cooling the Zones from 77.0℉ to 76.0℉

Table 7-6 Energy-Margin Weights for Each Zone

Margin Increment	Zone	Energy-Margin Weights	HVAC Unit Power at Typical DR Day (kW)	Cooling Time (seconds)
0.72135	1	**0.2448**	8.5	1248
	2	**1.0911**	7.0	340
	3	**0.2331**	12.0	928
	4	**0.9116**	4.5	633

The energy-margin weight of each room in Table 7-6 is calculated using (7-18).

In this case, $\lambda = [0.2448, 1.0911, 0.2331, 0.9116]$ is plugged in to the simulator. The control result during a DR event and with the same initial temperature is presented in Figure 7-16.

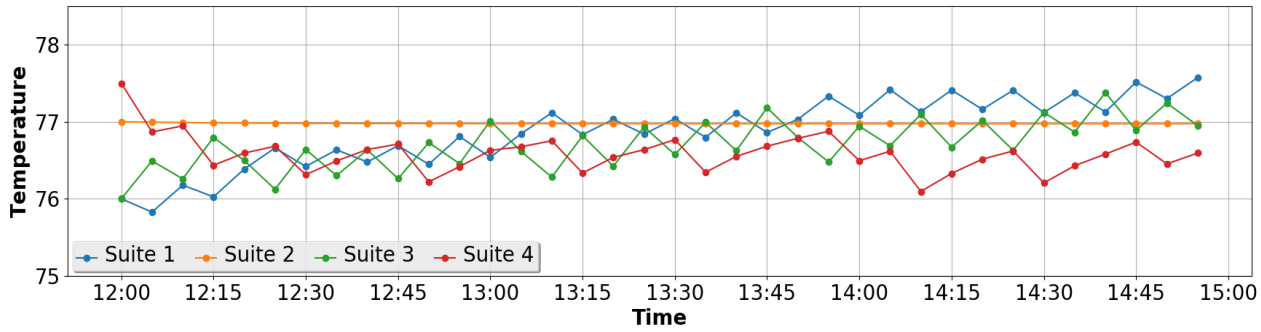


Figure 7-16 Temperature in Four Rooms During a 3-hour DR Period (w/ Energy Saving)

First, according to the temperature profile, the smart controller again make the indoor temperature of all zones below 78℉ for the whole period of demand response. Second, in contrast with the previous scenario, the AC unit of the Suite 4 does not show any sign of unnecessary cooling: at 14:10, the indoor temperature of Suite 4 is approaching $Temp_{eco}$, and the controller stopped cooling this suite in advance.

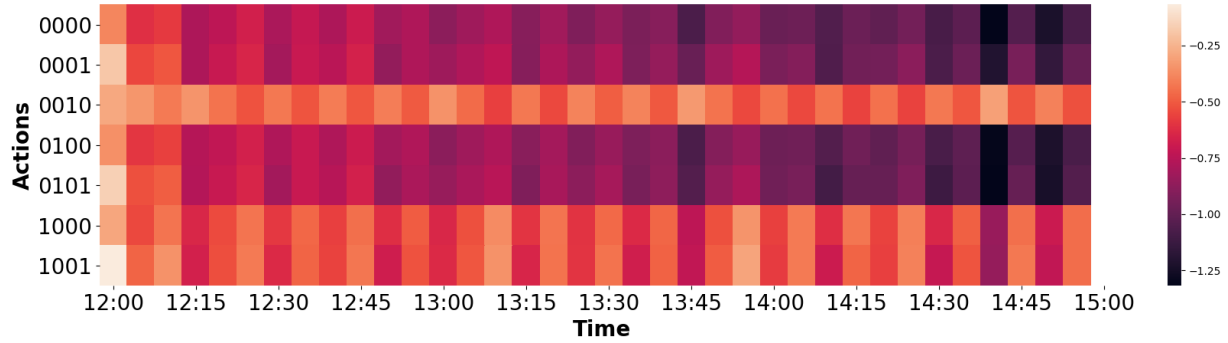


Figure 7-17 Heat Map of Q-values of State-Action Pairs in the DR Period (With Energy Saving)

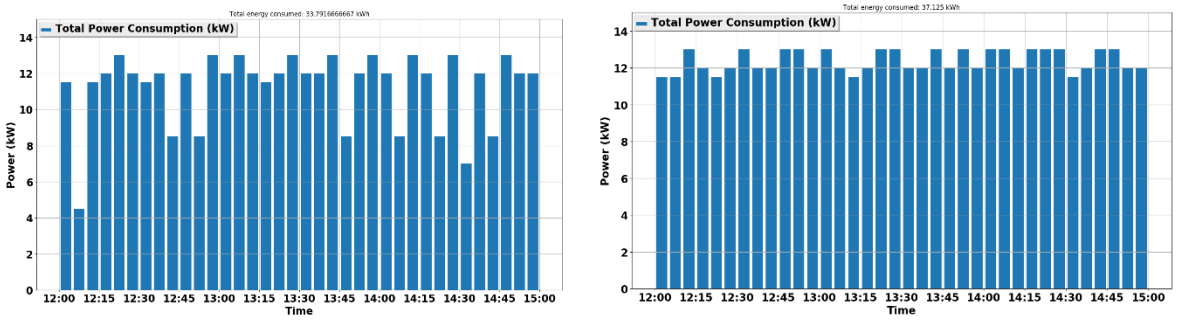
Though taking Action '0000', namely turning off all the AC units at some states will make temperature increase in all rooms and thus yield penalty, it might be better than other inappropriate actions; as a result, Action 0's Q-values are not necessarily the lowest when energy saving is considered, according to Figure 7-17. In contrast, as shown in Figure 7-12, Action '0000' always has the lowest Q-value at all time.

The energy consumption for both scenarios are plotted in Figure 7-18: When energy consumption is considered as a penalty, during this 3-hour DR period, 33.792 kWh of electricity is used; in contrast, 37.125 kWh of energy is used if do not take energy consumption into consideration.

Figure 7-19 shows the energy usage distribution by conducting 200 simulations (3-hour DR events) in both cases. The following observations can be drawn:

1. On average, 2.5 kWh of energy can be saved in this 3-hour DR event, considering the median of both cases as shown in Figure 7-19.

2. Energy saving is not significant: less than 10% according to 1. This is because in this case, the DR limit is already low, even if energy saving is considered, still a big portion of power consumption is needed.
3. The variation of energy consumption is higher when energy-saving is considered. This is because in the case without energy-saving, the controller will always use as much energy as possible, which is already around the limit. In contrast, when energy-saving is considered, the actual energy consumption changes according to different initial states: if the initial indoor temperature is low, less energy is needed overall.



(a) w/ energy saving *(b) w/o energy saving*

Figure 7-18 Energy Usage in kWh in Both Scenarios

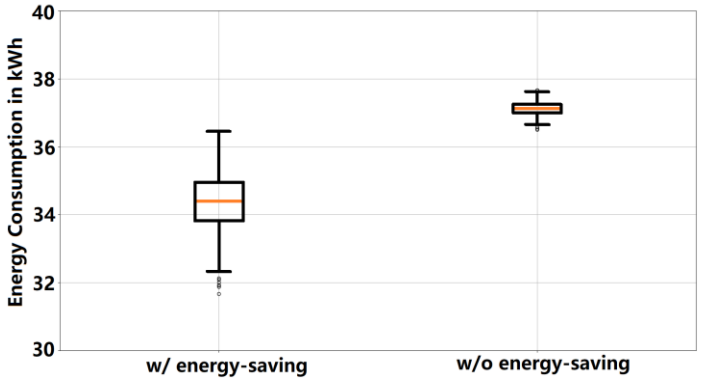
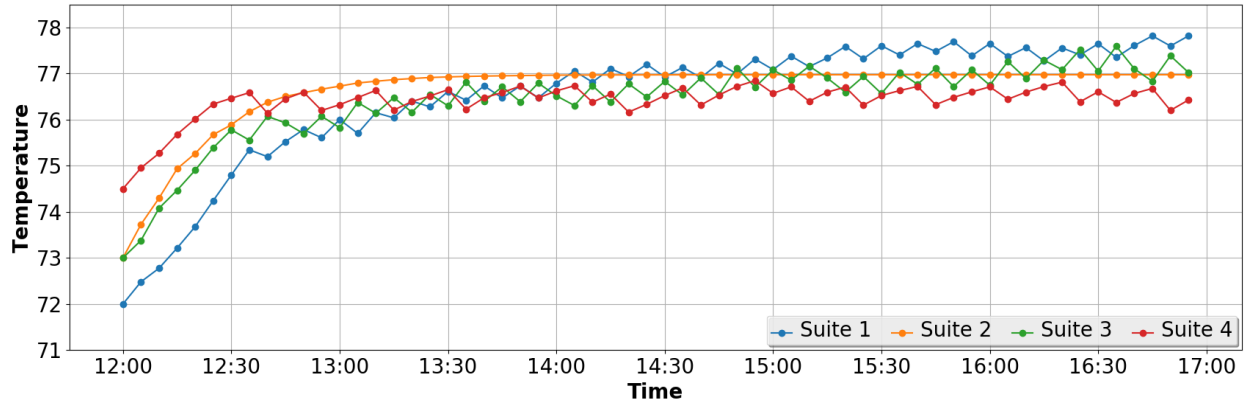
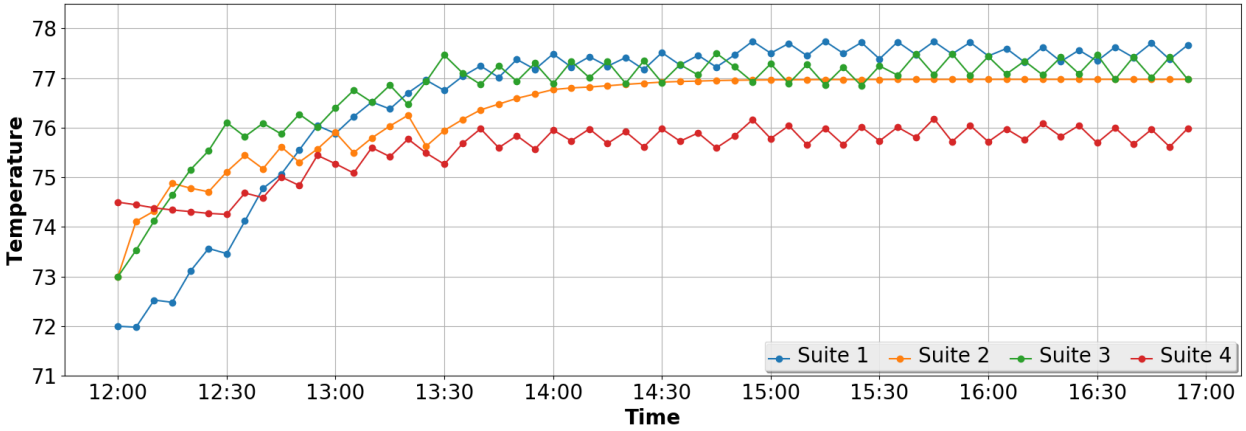


Figure 7-19 Energy Usage Distribution in kWh in Both Scenarios

In addition, it is worth noting that if energy saving is considered, the room will not be cooled as long as the indoor temperature is below $Temp_{eco}$, as shown in Figure 7-20 (a). The reason is obvious, any cooling below $Temp_{eco}$ incurs higher energy penalty than the gained thermal margin. In contrast, in Figure 7-20 (b), it shows the room with temperature already lower will still be cooled, which might shift some cooling load and 'make it easier' for later operation.



(a) w/ energy saving



(b) w/o energy saving

Figure 7-20 Indoor Temperature Profile when DR Starts From Lower Temperature

Figure 7-20 also shows the effectiveness of the algorithm for using the same optimal control policy in the situation when DR event last longer (5 hours in the figure). This is one advantage as discussed in Section 7.4.1.2.

It is also worth noting that the agent has learnt the building thermal property from its experience: it knows that even without AC cooling, the Suite 2 will reach a thermal equilibrium with the outdoor environment at around 77℉. As a result, no need to cool at all when energy saving is considered, as shown in Figure 7-21.

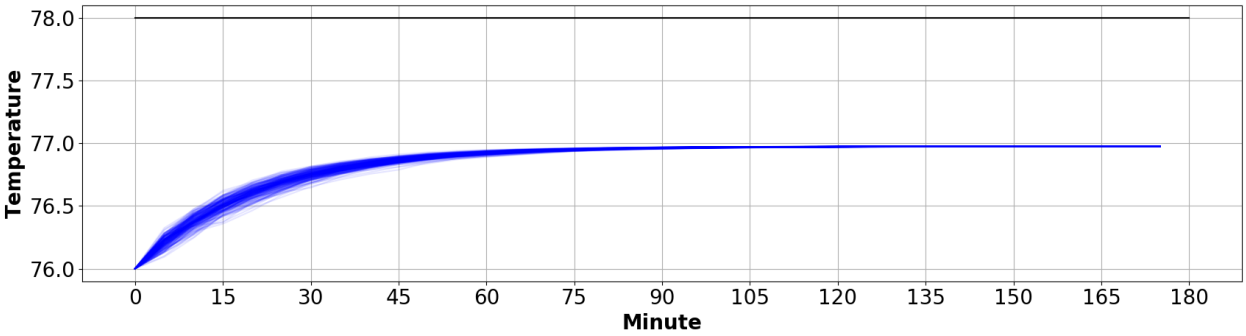


Figure 7-21 Indoor Temperature Profile of Suite 2 when Energy Saving is Considered

7.4.4. Model Training Efficiency

Figure 7-22 shows the learning curve for both controls with and without energy saving consideration. In both cases, the mean squared error of a batch of instance are decreasing with the training steps, demonstrating an effective training.

The total time for these four million training steps, however, is around 14 hours and 30 minutes for both cases. The training process runs on an Amazon Web Service r5.large instance without GPU acceleration. In contrast to the optimization-based control algorithm, whose computation starts upon receiving the DR signal, the RL-based control algorithm trains the model offline, which has a more lenient computation efficiency requirement. Admittedly, the computation efficiency can be improved by utilizing GPU acceleration and other distributed RL algorithms, which can possibly reduce the wall-time by an order of magnitude according to [160]. Since this dissertation mainly propose this prototype and validate its efficacy, the performance improvement will be investigated in the future work, as discussed with more details in Section 8.3.

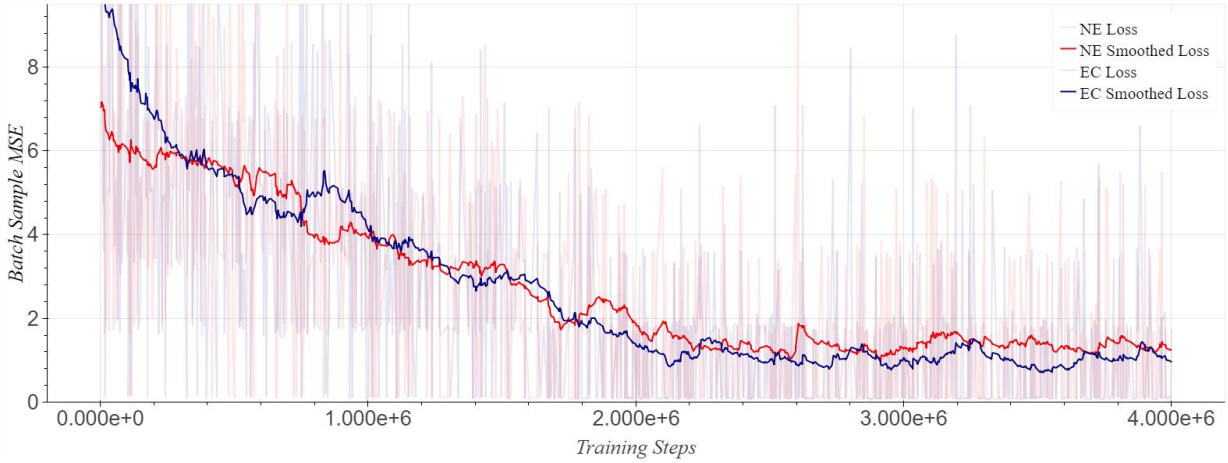


Figure 7-22 MSE Loss Over Training Steps

(EC stands for energy-conservation and NE stands for non-energy-consideration)

Admittedly, compared with the optimization-based algorithm, this one requires a lot of computation power and thus makes it only suitable for AC control cloud service provider such as Nest and Honeywell.

7.5. Real World Application

Figure 7-23 shows a timeline representation for using the RL based algorithm for DR control.

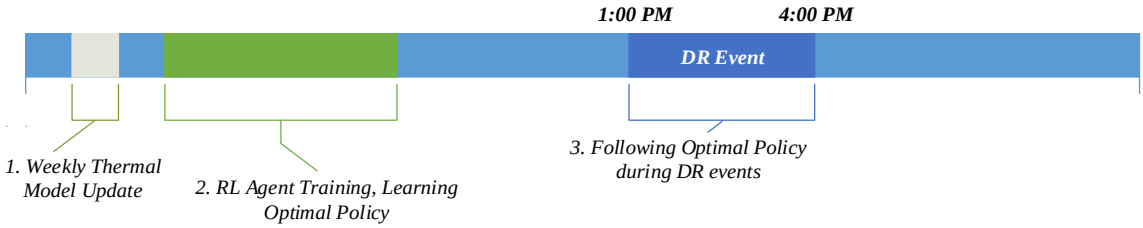


Figure 7-23 Timeline Representation for the RL-based Algorithm for DR Control

The steps are described below:

1. Weekly thermal model update: updating the thermal model using the most up to date training data to avoid the model drift over time;
2. RL agent training: let the agent interact with the simulator with the updated thermal model and use the updated experience to train the optimal control policy;
3. When DR event starts, no online computation is needed, the BEM system can control the AC units according to the optimal policy. In addition, since the agent will determine the action according to specific state for every steps, it is actually doing a step-wise correction. Therefore, no need to check indoor temperature every one hour as the optimization based algorithm requires.

8. SUMMARY, CONCLUSIONS AND FUTURE WORK

8.1. Summary

The objective of this study is to propose a cost-effective, plug-and-play and intelligent control framework for coordinating air-conditioning (AC) units in small- and medium-sized commercial buildings (SMCB) during demand response events. To achieve this goal, a thermal model based on supervised learning is proposed: by learning from temperature readings collected from the smart thermostats, the dynamics of indoor temperature variation can be predicted and used for temperature profile prediction. Then such thermal model is plugged in control algorithms to generate AC units' control schedule, which will be execute by the BEMs during a DR event.

By reviewing literature on existing thermal models, it can be found that the majority of the work focuses on the physical forward model, which requires detailed model description. Such models, however, might be impractical to use in real life because too much domain knowledge as well as a sophisticated sensor network are required. As a result, an easy-to-use thermal model with minimum amount of hardware investment and configuration is studied in this dissertation to fill the knowledge gap and provide a feasible engineering solution. Besides, the study on intelligent DR control algorithms targeting at the SMCB is scarce. With the Internet-of-things (IoT) devices gaining more popularity, this dissertation focuses on the AC control algorithms based on an IoT-based BEM system that is tailored for the SMCB.

Three major work are conducted in this dissertation: First, a self-learning thermal model is proposed; it describes the building thermal property and can be learning merely on smart thermostat coarse-grained temperature data. The efficacy of this proposed model drastically decreases the difficulty for configuring the system and thus removes one major road blocker for popularizing automated AC control during DR events. Second, a power disaggregation method is investigated, which models the power-outdoor temperature relationship for multiple AC units using data from a single power meter. This algorithm enables a more precise power control on AC units and requires only minimum amount of hardware investment. Finally, two AC units coordination algorithms based on traditional optimization technique and reinforcement learning are proposed and studied. These online and offline optimizing control tools are validated in simulations and real buildings, showing that within short notice (as short as 5 minutes), an optimal AC coordination control schedule can be generated for execution.

8.2. Conclusions

8.2.1. The proposed building thermal model can reliably predict indoor temperature

In Chapter 4, a supervised learning-based building thermal property model is proposed and validated. According to the experiment results, the following conclusions can be made:

a. The proposed model is cost-effective:
   This model relies only on thermostats' data, and the training process has been carefully devised so that it can learn from the coarse-grained thermostat reading. Because no other hardware investment is needed, this model provides a cost-efficient solution to SMCB;
b. Feature approximation used in the model is valid:
   To reduce the cost, some sensors such as occupancy and irradiance sensors are not installed and the corresponding features (occupancy level and solar irradiance) are approximated using other easily

observable features. Though this approximation, inevitably, will bring error to the prediction; in fact as proved in this dissertation, the error is acceptable for buildings with constant schedule and shows the validity of this approximation.

c. The proposed model is extensible:
   In case where the buildings have the budget to install occupancy sensor network, this model is flexible to seamlessly add other sensor data to the model to improve model accuracy;
d. Entirely self-learning without complicated configuration:
   Due to the data-driven feature, this algorithm learns the thermal model entirely from the building operating data. Thus, complicated configuration is avoided, enabling the model to work in a plug-and-play manner;
e. The indoor temperature prediction by the model has provided a good accuracy level for AC control:
   According to real building experiments and the tests with the historical data, the temperature prediction for up to four hours only causes an error level of 1℉ on average. Considering the limited information used to build the model and human's low sensitivity to temperature, this prediction accuracy is good enough for AC control.

By and large, a practical thermal model that can predict indoor temperature is proposed and tested in this dissertation. Due to its features of easy-to-use and cost-efficiency, it is a feasible and appropriate model to be used in the SMCB. The efficacy of the model is validated by case study and thus the model provides the basis for the following study of AC control.

8.2.2. The studied power disaggregation algorithm can help accurately predict the total power consumption of all AC units under different weather and control signals.

To accurately predict the aggregated power consumption of multiple AC units during DR control, the influence of the outdoor temperature on the AC unit power consumption is modeled instead of using a fixed power. First, a linear relationship between the outdoor temperature and the power consumption of a single AC unit is verified in a field test, using data collected from four units in a real building. Second, a power disaggregation algorithm is proposed: given the outdoor temperature from online weather service, status of each AC unit recorded by thermostats and the aggregated power of all units from a power meter, the parameters of the power-temperature linear models can be identified. The efficacy of this proposed algorithm is validated using real building operating data: utilizing a separate testing data set, it shows that the aggregated power consumption estimated using the learnt power-temperature models is equal to the actual measured aggregated power. Thus, the proposed algorithm is useful to precisely predict the total power consumption of all HVAC units under different control signals.

8.2.3. The optimization based AC units' coordination algorithm is suitable for the edge controlling scenarios.

A mixed integer linear programming problem is formulated for the AC units' coordination: with the cost of user thermal discomfort and energy consumption as the objective function, and the total power consumption limit and the thermal model as constraints, the optimal solution of such problem is the best controlling schedule for all AC units during the DR period. This algorithm is carefully designed so that it can provide enough flexibility to building engineers: first, the building engineers can choose whether energy saving should be considered during the control or not, according to the DR plan they opt-in. In addition, priority can be set based on either zone or time, such types of priority will provide enough flexibility according to the specific building operating condition. The effectiveness of the algorithm is tested in simulation, where the performance of the proposed algorithm is better than other commonly used DR

control, such as raise set point and arbitrarily limit power consumption. In addition to simulation, real building validation is conducted where a 90-minute DR event is scheduled in a campus building. In retrospect, the control schedule generated by the proposed algorithm successfully reduced the total power consumption under certain limit and the indoor temperature of each thermal zone is guaranteed. The experiment results on the algorithm efficiency shows that the proposed algorithm can enable the building to participate in a fast DR: In most case, the optimal/sub-optimal control schedule can be generated within 5 minutes after receiving the DR signal. The experiment is conducted on a machine with merely 2GB RAM, and thus proves that this algorithm is suitable for edge controlling scenarios. Finally, the proposed algorithm can also be used to evaluate the DR potential of a building: by running this algorithm in simulations, the building engineers can find the maximum amount of load reduction given the maximum tolerable temperature during a DR event.

8.2.4. The RL based AC units' coordination algorithm is appropriate for cloud controlling services.

For the same AC units' coordination problem, this dissertation proposes another solution based on reinforcement learning, a powerful tool to solve the step-wise optimal control problems. In this approach, a controlling agent is designed to interact with a building thermal simulator, which reflects the thermal dynamics of the building under study; then the agent will learn from its experience and eventually with enough training, it is able to take optimal action/control signal at each step. In the AC controlling scenarios, for each step, the agent will determine which units to be ON/OFF according to the temperature in different thermal zones. Because the agent always check the temperature distribution before taking actions, this essentially is a step-wise correction for any error made by indoor temperature prediction model. As a result, this algorithm relaxes the requirement for the thermal model accuracy, and make it suitable for those buildings with irregular thermal behavior. Affected by a non-linear thermal margin function, the agent is trained to give more priority to cooling the zones with higher temperature. In addition, the tradeoff between the thermal discomfort and energy consumption can be aptly tuned by setting the economic temperature in each zone. However, due to the comparatively heavier computation for training such agent, this algorithm is more suitable for cloud controlling services, where cloud computing can help training the agent more efficiently.

8.3. Future Work

1) Further testing the thermal model in other types of buildings

In this dissertation, the proposed thermal model uses some observable features to represent some features that cannot be easily observed without the proper hardware: the occupancy level is represented by the time of day and day of week, relying on an assumption that the occupancy level is strongly related to these two observable features. Though this approximation has been validated in the study based on the simulation and actual control in an AC controlled thermal zone, it is worth noting that these experiments are based on an office environment. Therefore, other commercial building types, such as grocery stores and banks where the occupancy level is more erratic, should also be tested to see if this model is still suitable for those environments.

2) RL-based AC coordination algorithm: improving training efficiency.

This dissertation shows an effective control using an agent trained by a customized reinforcement learning framework. To boost the performance of this prototype, improving algorithm efficiency is of great

importance: according to Chapter 7, training the agent to be fully learnt is time-consuming; after all, only with enough experience can the agent act optimally. However, there are some measures to speed up the training:

a. Taking advantage of GPU parallelism;
b. Using other distributed architecture for deep reinforcement learning;
c. Investigating the potential of transfer learning: assuming an optimal control policy is trained for a building, if another building has very similar thermal behavior and the same number of AC units to be controlled, is it possible to train the agent for the new building starting from the optimal control policy of the old one and largely reduce the training time?

3) RL-based AC coordination algorithm: model extension.

In the proposed algorithm, an assumption is made that all DR events happens under similar weather condition, and thus a typical DR day weather is used to fix the thermal model. This means the model is trained under a single configuration of weather condition. To improve the generality of the model, outdoor environmental variables should also be considered in the state, which extends the input of the DQN as shown in Table 8-1. In this case, the model is trained to deal with different weather condition instead of only one typical weather condition.

Table 8-1 Comparison between the Model Used in This Study and the Improved One Recommended

Model	Inputs	Number of Inputs
DQN in this study	Indoor temperature of every thermal zones	Number of thermal zones
A more generalized model	Indoor temperature of every thermal zones, environmental variables used in the thermal model: outdoor temperature, outdoor humidity, day of week, time of day, weather	Number of thermal zones + 9 (number of environmental variables after preprocessing)

4) Extend the work to consider multi-stage cooling

In this study, only single-stage cooling is considered due to the data used in this study are collected from single-stage AC units. But the model and algorithms proposed in this dissertation can be easily extend to the multi-stage cooling. In multi-stage AC system, second stage cooling usually kicks in when the first stage cooling cannot effectively cool the room, namely when the difference between indoor temperature and set point is over certain threshold. Of course, second stage cooling is more powerful than first stage and thus the TDS and the power consumption should be different. Therefore, the modification are listed below:

a. For the thermal model:
   Temperature set point $Temp_{target}$ should be considered in the thermal model, together with $Temp_{room}$, they can indicate which stage of cooling the unit is doing, and thus influence the TDS.

$$\frac{dTemp_{room}}{dt} = f(Temp_{room}, Temp_{target}, Temp_{out}, time, dow, H_{out}, w) \tag{8-1}$$

b. For the power-temperature model:
   Second stage cooling consumes more power than the first stage cooling, therefore, for power consumption, they can be considered as two different processes. As a result, the power of the single

unit of multi-stage cooling can be represented using the following model, with different parameters for the first stage cooling and second stage cooling. This model can be plugged in the same power disaggregation algorithm for parameter identification.

$$P_i = S_i^{1st}(\omega_i^{1st} \cdot Temp_{out} + b_i^{1st}) + S_i^{2nd}(\omega_i^{2nd} \cdot Temp_{out} + b_i^{2nd}) + S_{fi} \cdot f_i \quad (8\text{-}2)$$

$$S_i^{1st} + S_i^{2nd} = 1, \quad S_i^{1st} \cdot S_i^{2nd} = 0, \quad S_i^{1st} \in \{0,1\}, \quad S_i^{2nd} \in \{0,1\} \quad (8\text{-}3)$$

c. Extend the action space for control
With multi-stage cooling, during a DR event, the controlling agent not only need to consider which AC unit to be ON/OFF; for those to be turned on, the agent also need to choose which stage of cooling to achieve an overall optimal control.

REFERENCES

[1] IEEE Approved & Proposed Standards Related to Smart Grid. IEEE n.d. https://smartgrid.ieee.org/resources/standards/ieee-approved-proposed-standards-related-to-smart-grid (accessed January 1, 2018).

[2] Zhang X, Liu F, Yao R, Zhang X, Mei S. Identification of Key Transmission Lines in Power Grid Using Modified K -core Decomposition. Electr. Power Energy Convers. Syst. (EPECS), 2013 3rd Int. Conf., IEEE; 2013, p. 0–5. doi:10.1109/EPECS.2013.6713013.

[3] Somasundaram S, Pratt R, Akyol B, Fernandez N, Foster N, Katipamula S, et al. Reference Guide for a Transaction- Based Building Controls Framework. Pacific Northwest Natl Lab 2014. https://energy.gov/sites/prod/files/2014/06/f16/PNNL-23302_draft.pdf.

[4] Commercial Building Energy Consumption Survey 2012, Table B6. Building size, number of buildings. US Energy Inf Adm 2016. https://www.eia.gov/consumption/commercial/data/2012/bc/cfm/b6.php.

[5] Lucero S. IoT platforms: enabling the Internet of Things. 2016.

[6] U.S. Congress. Energy independence and security act of 2007. Public Law 2007:1–311. doi:papers2://publication/uuid/364DB882-E966-450B-959F-AEAD6E702F42.

[7] Tuballa ML, Abundo ML. A review of the development of Smart Grid technologies. Renew Sustain Energy Rev 2016;59:710–25. doi:10.1016/j.rser.2016.01.011.

[8] Peng L, Yan GS. Clean energy grid-connected technology based on smart grid. Energy Procedia 2011;12:213–8. doi:10.1016/j.egypro.2011.10.030.

[9] Feltes JW, Gemmell BD, Retzmann D. From smart grid to super grid: Solutions with HVDC and FACTS for grid access of renewable energy sources. IEEE Power Energy Soc Gen Meet 2011:1–6. doi:10.1109/PES.2011.6039346.

[10] Güngör VC, Sahin D, Kocak T, Ergüt S, Buccella C, Cecati C, et al. Smart grid technologies: Communication technologies and standards. IEEE Trans Ind Informatics 2011;7:529–39. doi:10.1109/TII.2011.2166794.

[11] Bian D, Kuzlu M, Pipattanasomporn M, Rahman S. Assessment of communication technologies for a home energy management system. 2014 IEEE PES Innov Smart Grid Technol Conf ISGT 2014 2014:1–5. doi:10.1109/ISGT.2014.6816449.

[12] Bian D, Kuzlu M, Pipattanasomporn M, Rahman S, Wu Y. Real-time co-simulation platform using OPAL-RT and OPNET for analyzing smart grid performance. IEEE Power Energy Soc Gen Meet 2015;2015–September:1–5. doi:10.1109/PESGM.2015.7286238.

[13] Aminifar F, Fotuhi-Firuzabad M, Safdarian A, Davoudi A, Shahidehpour M. Synchrophasor Measurement Technology in Power Systems: Panorama and State-of-the-Art. IEEE Access 2014;2:1607–28. doi:10.1109/ACCESS.2015.2389659.

[14] Bian D, Kuzlu M, Pipattanasomporn M, Rahman S. Analysis of communication schemes for Advanced Metering Infrastructure (AMI). IEEE Power Energy Soc Gen Meet 2014;2014–October:1–5. doi:10.1109/PESGM.2014.6939562.

[15] Huang P, Kalagnanam J, Natarajan R, Hammerstrom D, Melton R, Sharma M, et al. Analytics and Transactive Control Design for the Pacific Northwest Smart Grid Demonstration Project. 2010 First

IEEE Int Conf Smart Grid Commun 2010:449–54. doi:10.1109/SMARTGRID.2010.5622083.

[16] Jian L, Zheng Y, Xiao X, Chan CC. Optimal scheduling for vehicle-to-grid operation with stochastic connection of plug-in electric vehicles to smart grid. Appl Energy 2015;146:150–61. doi:10.1016/j.apenergy.2015.02.030.

[17] Honarmand M, Zakariazadeh A, Jadid S. Optimal scheduling of electric vehicles in an intelligent parking lot considering vehicle-to-grid concept and battery condition. Energy 2014;65:572–9. doi:10.1016/j.energy.2013.11.045.

[18] Liu L, Kong F, Liu X, Peng Y, Wang Q. A review on electric vehicles interacting with renewable energy in smart grid. Renew Sustain Energy Rev 2015;51:648–61. doi:10.1016/j.rser.2015.06.036.

[19] Galus MD, Zima M, Andersson G. On integration of plug-in hybrid electric vehicles into existing power system structures. Energy Policy 2010;38:6736–45. doi:10.1016/j.enpol.2010.06.043.

[20] Aloul F, Al-ali AR, Al-dalky R, Al-mardini M. Smart Grid Security : Threats , Vulnerabilities and Solutions. Smart Grid Clean Energy Smart 2012:1–6.

[21] MHI-VESTAS Offshore Wind: The V164-9.5 MW Turbine 2018. http://www.mhivestasoffshore.com/innovations-old/.

[22] Tesla Energy-Solar Roof n.d. https://www.tesla.com/solarroof (accessed January 1, 2018).

[23] Liserre M, Sauter T, Hung JY. Future energy systems: Inegrating renewable energy into the smart power grid through industrial electronics. IEEE Ind Electron Mag 2010;4:18–37. doi:10.1109/MIE.2010.935861.

[24] Wang J, Huang a, Sung W, Liu Y, Baliga B. Development of 15-kV SiC IGBTs and Their Impact on Utility Applications. IEEE Ind Electron Mag 2009.

[25] Jiayi H, Chuanwen J, Rong X. A review on distributed energy resources and MicroGrid. Renew Sustain Energy Rev 2008;12:2465–76. doi:10.1016/j.rser.2007.06.004.

[26] Yuan C, Illindala MS, Haj-Ahmed MA, Khalsa AS. Distributed energy resource planning for microgrids in the United States. IEEE Ind Appl Soc - 51st Annu Meet IAS 2015, Conf Rec 2015:1–9. doi:10.1109/IAS.2015.7356784.

[27] Yuan C. Resilient distribution systems with community microgrids. Ohio State University, 2016.

[28] Yuan C, Illindala MS, Khalsa AS. Co-Optimization Scheme for Distributed Energy Resource Planning in Community Microgrids. IEEE Trans Sustain Energy 2017;8:1351–60. doi:10.1109/TSTE.2017.2681111.

[29] Zamora R, Srivastava AK. Controls for microgrids with storage: Review, challenges, and research needs. Renew Sustain Energy Rev 2010;14:2009–18. doi:10.1016/j.rser.2010.03.019.

[30] Basak P, Chowdhury S, Halder Nee Dey S, Chowdhury SP. A literature review on integration of distributed energy resources in the perspective of control, protection and stability of microgrid. Renew Sustain Energy Rev 2012;16:5545–56. doi:10.1016/j.rser.2012.05.043.

[31] Wang Y, Pulgar-Painemal H, Sun K. Online analysis of voltage security in a microgrid using convolutional neural networks. IEEE Power Energy Soc Gen Meet 2018;2018–January:1–5. doi:10.1109/PESGM.2017.8274200.

[32] Xu H, Domínguez-García AD, Sauer PW. Data-driven Coordination of Distributed Energy Resources for Active Power Provision 2018:1–11.

[33] Reports on Demand Response & Advanced Metering n.d. https://www.ferc.gov/industries/electric/indus-act/demand-response/dem-res-adv-metering.asp (accessed January 1, 2018).

[34] U S Department of Energy. Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them. US Dep Energy 2006:122. doi:citeulike-article-id:10043893.

[35] Albadi MH, El-Saadany EF. A summary of demand response in electricity markets. Electr Power Syst Res 2008;78:1989–96. doi:10.1016/j.epsr.2008.04.002.

[36] SD G&E Time of Use Plus Whenergy Plan. San Diego Gas Electr Co 2016. https://www.sdge.com/business/demand-response/cpp.

[37] SDG&E. Understanding Capacity Reservation for Medium Sized Businesses. 2016.

[38] EnerNOC Demand Response Program. EnerNOC 2016. https://www.enernoc.com/products/businesses/capabilities/demand-response.

[39] PG&E Peak Day Pricing. Pacific Gas Electr Co 2016. https://www.pge.com/en_US/business/rate-plans/rate-plans/peak-day-pricing/peak-day-pricing.page (accessed December 14, 2017).

[40] Grünewald P, Torriti J. Demand response from the non-domestic sector: Early UK experiences and future opportunities. Energy Policy 2013;61:423–9. doi:10.1016/j.enpol.2013.06.051.

[41] San Diego G&E Demand Response Overview 2018. https://www.sdge.com/business/demand-response-overview (accessed January 1, 2018).

[42] SDG&E. San Diego Gas & Electric - Capacity Bidding Program 2016. http://regarchive.sdge.com/tm2/pdf/ELEC_ELEC-SCHEDS_CBP.pdf (accessed January 1, 2018).

[43] Southern California Edison Demand Response Programs n.d. www.sce.com/drp (accessed January 1, 2018).

[44] Electric Power Sales, Revenue, and Energy Efficiency Form EIA-861 Detailed Data Files 2017. https://www.eia.gov/electricity/data/eia861/ (accessed January 1, 2018).

[45] Rawal G, Raschke S. Costs, Savings, and ROI for Smart Building Implementation. Intel Blog 2016. http://blogs.intel.com/iot/2016/06/20/costs-savings-roi-smart-building-implementation/ (accessed January 1, 2017).

[46] Whitmore A, Agarwal A, Da Xu L. The Internet of Things—A survey of topics and trends. Inf Syst Front 2015;17:261–74. doi:10.1007/s10796-014-9489-2.

[47] Pan J, Jain R, Paul S, Vu T, Saifullah A, Sha M. An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments. IEEE Internet Things J 2015;2:527–37. doi:10.1109/JIOT.2015.2413397.

[48] Minoli D, Sohraby K, Occhiogrosso B. IoT Considerations, Requirements, and Architectures for Smart Buildings – Energy Optimization and Next Generation Building Management Systems. IEEE Internet Things J 2017;4:1–1. doi:10.1109/JIOT.2017.2647881.

[49] Moreno MV, ??beda B, Skarmeta AF, Zamora MA. How can we tackle energy efficiency in iot based smart buildings? Sensors (Switzerland) 2014;14:9582–614. doi:10.3390/s140609582.

[50] Khamphanchai W, Saha A, Rathinavel K, Kuzlu M, Pipattanasomporn M, Rahman S, et al. Conceptual architecture of building energy management open source software (BEMOSS). IEEE

PES Innov Smart Grid Technol Eur Istanbul 2014:1–6. doi:10.1109/ISGTEurope.2014.7028784.

[51] Zhang X, Adhikari R, Pipattanasomporn M, Kuzlu M, Rahman S. Deploying IoT devices to make buildings smart: Performance evaluation and deployment experience. 2016 IEEE 3rd World Forum Internet Things, Reston, VA, USA: 2016, p. 530–5. doi:10.1109/WF-IoT.2016.7845464.

[52] Adhikari R, Zhang X, Pipattanasomporn M. A data-driven approach for quantifiying energy savings in a smart building. Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT), 2017 IEEE, 2017. doi:10.1109/ISGT.2017.8085994.

[53] Lee SH, Hong T, Piette MA, Sawaya G, Chen Y, Taylor-Lange SC. Accelerating the energy retrofit of commercial buildings using a database of energy efficiency performance. Energy 2015;90:738–47. doi:10.1016/j.energy.2015.07.107.

[54] Nandivada S. Energy-Efficiency Retrofits in the Commercial Sector: An Analysis of PACE Financing, On-Bill Repayment, and Energy Savings Performance Contracts. J Environ Law Litig 2014:363–88. doi:10.3366/ajicl.2011.0005.

[55] Sweatman P, Managan K. Financing Energy Efficiency Building Retrofits International Policy and Business Model Review and Regulatory Alternatives for Spain. Int Policy Bus Model Rev Regul Altern Spain 2010:1–68.

[56] Géron A. Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc.; 2017.

[57] Zhou K, Fu C, Yang S. Big data driven smart energy management: From big data to big insights. Renew Sustain Energy Rev 2016;56:215–25. doi:10.1016/j.rser.2015.11.050.

[58] Yildiz B, Bilbao JI, Sproul AB. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renew Sustain Energy Rev 2017;73:1104–22. doi:10.1016/j.rser.2017.02.023.

[59] Jurado S, Nebot À, Mugica F, Avellana N. Hybrid methodologies for electricity load forecasting: Entropy-based feature selection with machine learning and soft computing techniques. Energy 2015;86:276–91. doi:10.1016/j.energy.2015.04.039.

[60] Kavousi-Fard A, Samet H, Marzbani F. A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting. Expert Syst Appl 2014;41:6047–56. doi:10.1016/j.eswa.2014.03.053.

[61] Chen Y, Xu P, Chu Y, Li W, Wu Y, Ni L, et al. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. Appl Energy 2017;195:659–70. doi:10.1016/j.apenergy.2017.03.034.

[62] Deb C, Zhang F, Yang J, Lee SE, Shah KW. A review on time series forecasting techniques for building energy consumption. Renew Sustain Energy Rev 2017;74:902–24. doi:10.1016/j.rser.2017.02.085.

[63] Cai M, Pipattanasomporn M, Rahman S. Day-ahead Building-level Load Forecasts using Deep Learning vs. Traditional Time-series Techniques. Appl Energy 2019.

[64] Xu H, Member S, Dom AD, Sauer PW, Fellow L. Optimal Tap Setting of Voltage Regulation Transformers Using Batch Reinforcement Learning. arXiv 2018:1–8.

[65] Dalamagkidis K, Kolokotsa D, Kalaitzakis K, Stavrakakis GS. Reinforcement learning for energy conservation and comfort in buildings. Build Environ 2007;42:2686–98.

doi:10.1016/j.buildenv.2006.07.010.

[66] Wen Z, O'Neill D, Maei HR. Optimal Demand Response Using Device Based Reinforcement Learning. Smart Grid, IEEE Trans 2014;6:1–23. doi:10.1109/TSG.2015.2396993.

[67] Yang L, Nagy Z, Goffin P, Schlueter A. Reinforcement learning for optimal control of low exergy buildings. Appl Energy 2015;156:577–86. doi:10.1016/j.apenergy.2015.07.050.

[68] O'Neill D, Levorato M, Goldsmith A, Mitra U. Residential Demand Response Using Reinforcement Learning. 2010 First IEEE Int Conf Smart Grid Commun 2010:409–14. doi:10.1109/SMARTGRID.2010.5622078.

[69] Kara EC, Berges M, Krogh B, Kar S. Using smart devices for system-level management and control in the smart grid: A reinforcement learning framework. 2012 IEEE Third Int Conf Smart Grid Commun 2012:85–90. doi:10.1109/SmartGridComm.2012.6485964.

[70] Koch S, Mathieu JL, Callaway DS. Modeling and Control of Aggregated Heterogeneous Thermostatically Controlled Loads for Ancillary Services. Proc 17th Power Syst Comput Conf 2011. doi:10.1109/DICTA.2007.79.

[71] Morvaj B, Lugaric L, Krajcar S. Demonstrating smart buildings and smart grid features in a smart energy city. Proc 2011 3rd Int Youth Conf Energ 2011:1–8.

[72] Kolokotsa D. The role of smart grids in the building sector. Energy Build 2016;116:703–8. doi:10.1016/j.enbuild.2015.12.033.

[73] Sehar F, Pipattanasomporn M, Rahman S. An energy management model to study energy and peak power savings from PV and storage in demand responsive buildings. Appl Energy 2016;173:406–17. doi:10.1016/j.apenergy.2016.04.039.

[74] Arboleya P, Gonzalez-Moran C, Coto M, Falvo MC, Martirano L, Sbordone D, et al. Efficient energy management in smart micro-grids: ZERO grid impact buildings. IEEE Trans Smart Grid 2015;6:1055–63. doi:10.1109/TSG.2015.2392071.

[75] Xue X, Wang S, Sun Y, Xiao F. An interactive building power demand management strategy for facilitating smart grid optimization. Appl Energy 2014;116:297–310. doi:10.1016/j.apenergy.2013.11.064.

[76] Liang Y, Liu F, Wang C, Mei S. Distributed demand-side energy management scheme in residential smart grids: An ordinal state-based potential game approach. Appl Energy 2017;206:991–1008. doi:10.1016/j.apenergy.2017.08.123.

[77] Hurtado LA, Nguyen PH, Kling WL. Smart grid and smart building inter-operation using agent-based particle swarm optimization. Sustain Energy, Grids Networks 2015;2:32–40. doi:10.1016/j.segan.2015.03.003.

[78] Maasoumy M, Ortiz J, Culler D, Sangiovanni-Vincentelli A. Flexibility of Commercial Building HVAC Fan as Ancillary Service for Smart Grid 2013:1–7.

[79] Hao H, Lin Y, Kowli AS, Barooah P, Meyn S. Ancillary Service to the grid through control of fans in commercial Building HVAC systems. IEEE Trans Smart Grid 2014;5:2066–74. doi:10.1109/TSG.2014.2322604.

[80] Chou JS, Ngo NT. Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns. Appl Energy 2016;177:751–70. doi:10.1016/j.apenergy.2016.05.074.

[81] Raza MQ, Khosravi A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renew Sustain Energy Rev 2015;50:1352–72. doi:10.1016/j.rser.2015.04.065.

[82] Zhao HX, Magoul??s F. A review on the prediction of building energy consumption. Renew Sustain Energy Rev 2012;16:3586–92. doi:10.1016/j.rser.2012.02.049.

[83] Cui C, Wu T, Hu M, Weir JD, Li X. Short-term building energy model recommendation system: A meta-learning approach. Appl Energy 2016;172:251–63. doi:10.1016/j.apenergy.2016.03.112.

[84] Short Term Operating Reserve. Natl Grid UK 2017. http://www2.nationalgrid.com/uk/services/balancing-services/reserve-services/short-term-operating-reserve/ (accessed January 1, 2017).

[85] Ma O, Alkadi N, Cappers P, Denholm P, Dudley J, Goli S, et al. Demand Response for Ancillary Services. IEEE Trans Smart Grid 2013;4:1988–95. doi:10.1109/TSG.2013.2258049.

[86] Xue X, Wang S, Yan C, Cui B. A fast chiller power demand response control strategy for buildings connected to smart grid. Appl Energy 2015;137:77–87. doi:10.1016/j.apenergy.2014.09.084.

[87] Jung D, Krishna VB, Temple WG, Yau DKY. Data-driven evaluation of building demand response capacity. 2014 IEEE Int Conf Smart Grid Commun SmartGridComm 2014 2015:541–7. doi:10.1109/SmartGridComm.2014.7007703.

[88] Mathieu JL, Price PN, Kiliccote S, Piette MA. Quantifying changes in building electricity use, with application to demand response. IEEE Trans Smart Grid 2011;2:507–18. doi:10.1109/TSG.2011.2145010.

[89] Illuminance Based Lighting Control from BEMOSS 2017. https://github.com/bemoss/BEMOSS3.5/blob/master/Applications/code/IlluminanceBasedLightingControl/illuminancebasedlightingcontrol/agent.py (accessed January 1, 2018).

[90] Galasiu AD, Veitch JA. Occupant preferences and satisfaction with the luminous environment and control systems in daylit offices: a literature review. Energy Build 2006;38:728–42. doi:10.1016/j.enbuild.2006.03.001.

[91] Shen E, Hu J, Patel M. Energy and visual comfort analysis of lighting and daylight control strategies. Build Environ 2014;78:155–70. doi:10.1016/j.buildenv.2014.04.028.

[92] Dubois MC, Blomsterberg Å. Energy saving potential and strategies for electric lighting in future north european, low energy office buildings: A literature review. Energy Build 2011;43:2572–82. doi:10.1016/j.enbuild.2011.07.001.

[93] Motegi N, Piette MA, Watson DS, Kiliccote S, Xu P. Introduction to Commercial Building Control Strategies and Techniques for Demand Response 2007:1–107. doi:10.2172/1004169.

[94] Weng T, Balaji B, Dutta S, Gupta R, Agarwal Y. Managing plug-loads for demand response within buildings. Proc Third ACM Work Embed Sens Syst Energy-Efficiency Build - BuildSys '11 2011:13. doi:10.1145/2434020.2434024.

[95] Arnold D, Sankur M, Auslander DM. An architecture for enabling distributed plug load control for commercial building demand response. 2013 IEEE PES Innov Smart Grid Technol Conf ISGT 2013 2013. doi:10.1109/ISGT.2013.6497850.

[96] Hong SH, Yu M, Huang X. A real-time demand response algorithm for heterogeneous devices in buildings and homes. Energy 2015;80:123–32. doi:10.1016/j.energy.2014.11.053.

[97] Zhuang P, Liang H. Energy Storage Management in Smart Homes Based on Resident Activity of Daily Life Recognition. 2015 IEEE Int Conf Smart Grid Commun Archit Control Oper Smart Grids Microgrids 2015:641–6. doi:10.1109/SmartGridComm.2015.7436373.

[98] Commercial Building Energy Consumption Survey 2012, Table E3. Electricity consumption (Btu) by end use. US Energy Inf Adm 2016. https://www.eia.gov/consumption/commercial/data/2012/c&e/cfm/e3.php.

[99] Xu P, Haves P, Piette MA. Peak demand reduction from pre-cooling with zone temperature reset in an office building 2004.

[100] Katipamula S, Lu N. Evaluation of Residential HVAC Control Strategies for Demand Response Programs. ASHRAE Trans 2006;112.

[101] Lee K, Braun JE. Development and Application of an Inverse Building Model for Demand Response in Small Commercial Buildings. SimBuild 2004 310AD:1–12. doi:10.1.1.163.9316.

[102] Yin R, Xu P, Piette MA, Kiliccote S. Study on Auto-DR and pre-cooling of commercial buildings with thermal mass in California. Energy Build 2010;42:967–75. doi:10.1016/j.enbuild.2010.01.008.

[103] Lin Y, Barooah P, Meyn S, Middelkoop T. Experimental Evaluation of Frequency Regulation From Commercial Building HVAC Systems 2015;6:776–83.

[104] Hao H, Middelkoop T, Barooah P, Meyn S. How demand response from commercial buildings will provide the regulation needs of the grid 2012:1908–13.

[105] Rotger-griful S, Jacobsen RH, Nguyen D, Sørensen G. Demand response potential of ventilation systems in residential buildings. Energy Build 2016;121:1–10. doi:10.1016/j.enbuild.2016.03.061.

[106] Derrible S, Reeder M. The cost of over-cooling commercial buildings in the United States. Energy Build 2015;108:304–6. doi:10.1016/j.enbuild.2015.09.022.

[107] Sehar F, Pipattanasomporn M, Rahman S. A peak-load reduction computing tool sensitive to commercial building environmental preferences. Appl Energy 2016;161:279–89. doi:10.1016/j.apenergy.2015.10.009.

[108] Piette MA, Watson D, Motegi N, Kiliccote S, Xu P, Watson D. Automated Critical Peak Pricing Field Tests : Program Description and Results - Appendices Automated Critical Peak Pricing Field Tests : Program Description and Results Appendices. 2006.

[109] Piette MA, Sezgen O, Watson DS. Development and evaluation of fully automated demand response in large facilities. 2004.

[110] Piette MA, Watson DS, Motegi N, Bourassa N. Findings from the 2004 fully automated demand response tests in large facilities. 2005.

[111] Yang L, Yan H, Lam JC. Thermal comfort and building energy consumption implications - A review. Appl Energy 2014;115:164–73. doi:10.1016/j.apenergy.2013.10.062.

[112] Ghahramani A, Zhang K, Dutta K, Yang Z, Becerik-Gerber B. Energy savings from temperature setpoints and deadband: Quantifying the influence of building and system properties on savings. Appl Energy 2016;165:930–42. doi:10.1016/j.apenergy.2015.12.115.

[113] Afram A, Janabi-Sharifi F. Theory and applications of HVAC control systems - A review of model predictive control (MPC). Build Environ 2014;72:343–55. doi:10.1016/j.buildenv.2013.11.016.

[114] Bianchini G, Casini M, Vicino A, Zarrilli D. Demand-response in building heating systems: A

Model Predictive Control approach. Appl Energy 2016;168:159–70. doi:10.1016/j.apenergy.2016.01.088.

[115] Kwak Y, Huh J-H, Jang C. Development of a model predictive control framework through real-time building energy management system data. Appl Energy 2015;155:1–13. doi:10.1016/j.apenergy.2015.05.096.

[116] Oldewurtel F, Ulbig a., Parisio a., Andersson G, Morari M. Reducing peak electricity demand in building climate control using real-time pricing and model predictive control. Decis Control (CDC), 2010 49th IEEE Conf 2010:1927–32. doi:10.1109/CDC.2010.5717458.

[117] Sanyal J, Nutaro JJ, Fugate D, Kuruganti T, Olama M. SUPERVISORY CONTROL FOR PEAK REDUCTION IN COMMERCIAL BUILDINGS WHILE MAINTAINING COMFORT. ASHRAE IBPSA-USA SimBuild 2016 Build. Perform. Model. Conf., Salt Lake City, UT: 2016.

[118] Nutaro J, Fugate D, Kuruganti T, Sanyal J, Starke M. An Inexpensive Retrofit Technology for Reducing Peak Power Demand in Small and Medium Commercial Buildings. Int. High Perform. Build. Conf., 2014. doi:10.1080/23744731.2015.1047719.

[119] Spencer B, Alfandi O. Forecasting Internal Temperature in a Home with a Sensor Network. Procedia Comput Sci 2016;83:1244–9. doi:10.1016/j.procs.2016.04.259.

[120] Spencer B, Al-Obeidat F. Temperature Forecasts with Stable Accuracy in a Smart Home. Procedia Comput Sci 2016;83:726–33. doi:10.1016/j.procs.2016.04.160.

[121] Marvuglia A, Messineo A, Nicolosi G. Coupling a neural network temperature predictor and a fuzzy logic controller to perform thermal comfort regulation in an office building. Build Environ 2014;72:287–99. doi:10.1016/j.buildenv.2013.10.020.

[122] Huang H, Chen L, Mohammadzaheri M, Hu E. A new zone temperature predictive modeling for energy saving in buildings. Procedia Eng 2012;49:142–51. doi:10.1016/j.proeng.2012.10.122.

[123] Zamora-Martínez F, Romeu P, Botella-Rocamora P, Pardo J. On-line learning of indoor temperature forecasting models towards energy efficiency. Energy Build 2014;83:162–72. doi:10.1016/j.enbuild.2014.04.034.

[124] Ruano AE, Crispim EM, Conceição EZE, Lúcio MMJR. Prediction of building's temperature using neural networks models. Energy Build 2006;38:682–94. doi:10.1016/j.enbuild.2005.09.007.

[125] Mustafaraj G, Lowry G, Chen J. Prediction of room temperature and relative humidity by autoregressive linear and nonlinear neural network models for an open office. Energy Build 2011;43:1452–60. doi:10.1016/j.enbuild.2011.02.007.

[126] Harb H, Boyanov N, Hernandez L, Streblow R, M??ller D. Development and validation of grey-box models for forecasting the thermal response of occupied buildings. Energy Build 2016;117:199–207. doi:10.1016/j.enbuild.2016.02.021.

[127] Mustafaraj G, Chen J, Lowry G. Development of room temperature and relative humidity linear parametric models for an open office using BMS data. Energy Build 2010;42:348–56. doi:10.1016/j.enbuild.2009.10.001.

[128] Kim D, Braun JE, Cai J, Hu J. Development of a Plug-and-Play Multiple RTU Coordination Control Algorithm for Small / Medium Commercial Buildings. Acc 2015 2015:1659–64.

[129] Kim D, Braun JE, Cai J, Hu J. Development and experimental demonstration of a plug-and-play multiple RTU coordination control algorithm for small/medium commercial buildings. Energy

Build 2015;2015–July:1659–64. doi:10.1109/ACC.2015.7170971.

[130] Yang R, Wang L. Development of multi-agent system for building energy and comfort management based on occupant behaviors. Energy Build 2013;56:1–7. doi:10.1016/j.enbuild.2012.10.025.

[131] Freire RZ, Oliveira GHC, Mendes N. Predictive controllers for thermal comfort optimization and energy savings. Energy Build 2008;40:1353–65. doi:10.1016/j.enbuild.2007.12.007.

[132] Yang R, Wang L. Multi-zone building energy management using intelligent control and optimization. Sustain Cities Soc 2013;6:16–21. doi:10.1016/j.scs.2012.07.001.

[133] Ferreira PM, Ruano AE, Silva S, Conceição EZE. Neural networks based predictive control for thermal comfort and energy savings in public buildings. Energy Build 2012;55:238–51. doi:10.1016/j.enbuild.2012.08.002.

[134] Garnier A, Eynard J, Caussanel M, Grieu S. Low computational cost technique for predictive management of thermal comfort in non-residential buildings. J Process Control 2014;24:750–62. doi:10.1016/j.jprocont.2013.10.005.

[135] Zhao Q, Zhao Y, Wang F, Wang J, Jiang Y, Zhang F. A data-driven method to describe the personalized dynamic thermal comfort in ordinary office environment: From model to application. Build Environ 2014;72:309–18. doi:10.1016/j.buildenv.2013.11.008.

[136] Tanabe SI, Kobayashi K, Kiyota O, Nishihara N, Haneda M. The effect of indoor thermal environment on productivity by a year-long survey of a call centre. Intell Build Int 2009;1:184–94. doi:10.3763/inbi.2009.0005.

[137] Cui W, Cao G, Park JH, Ouyang Q, Zhu Y. Influence of indoor air temperature on human thermal comfort, motivation and performance. Build Environ 2013;68:114–22. doi:10.1016/j.buildenv.2013.06.012.

[138] Rupp RF, Vásquez NG, Lamberts R. A review of human thermal comfort in the built environment. Energy Build 2015;105:178–205. doi:10.1016/j.enbuild.2015.07.047.

[139] Doty S, Turner W. Energy Management Handbook, Seventh Edition. 7th ed. The Fairmont Press, Inc., 2004; 2004.

[140] Zhang X, Pipattanasomporn M, Rahman S. A self-learning algorithm for coordinated control of rooftop units in small- and medium-sized commercial buildings. Appl Energy 2017;205:1034–49. doi:10.1016/j.apenergy.2017.08.093.

[141] BEMOSS Version 3.5 GitHub Repository n.d. https://github.com/bemoss/BEMOSS3.5 (accessed January 1, 2018).

[142] HardKernel Odroid XU4 Product Description n.d. http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825 (accessed January 1, 2018).

[143] Cubieboard4 (CC-A80) Product Description n.d. http://cubieboard.org/model/cb4/ (accessed January 1, 2018).

[144] Wandboard Quad Product Description n.d. https://www.wandboard.org/products/wandboard/WB-IMX6Q-BW/ (accessed January 1, 2018).

[145] Shao S. An Approach to Demand Response for Alleviating Power System Stress Conditions due to Electric Vehicle Penetration. Virginia Polytechnic Institute and State University, 2011.

[146] Shao S, Member S, Pipattanasomporn M. Development of Physical-Based Demand Response-Enabled Residential Load Models 2013;28:607–14.

[147] Radio Thermostat CT50 7-Day Programmable Thermostat (WiFi Enabled) n.d. https://www.amazon.com/Radio-Thermostat-Programmable-Enabled-Controls/dp/B00KQS35XA/ref=sr_1_1?ie=UTF8&qid=1514866193 (accessed January 19, 2018).

[148] Weather Underground Forecast Data. Weather Undergr 2016. https://www.wunderground.com/about/data.

[149] Juban R, Ohlsson H, Maasoumy M, Poirier L, Kolter JZ. A multiple quantile regression approach to the wind, solar, and price tracks of GEFCom2014. Int J Forecast 2016;32:1094–102. doi:10.1016/j.ijforecast.2015.12.002.

[150] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning (Second Edition). Springer; 2013. doi:10.1007/b94608.

[151] Joachims T. Learning To Classify Text Using Support Vector Machines. 2001.

[152] Kohonen T. An introduction to neural computing. Neural Networks 1988;1:3–16. doi:10.1016/0893-6080(88)90020-2.

[153] Mcculloch WS, Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bull Mothemnticnl Biol 1943;5:115–33. doi:10.1007/BF02478259.

[154] Zhang X, Cai M, Pipattanasomporn M, Rahman S. A Power Disaggregation Approach to Identify Power-Temperature Models of HVAC Units. IEEE Int. Smart Cities Conf., Kansas City: 2018.

[155] Lan L, Wargocki P, Lian Z. Quantitative measurement of productivity loss due to thermal discomfort. Energy Build 2011;43:1057–62. doi:10.1016/j.enbuild.2010.09.001.

[156] Lakshman A, Malik P. Cassandra-A Decentralized Structured Storage System. ACM SIGOPS Oper Syst Rev 2010;44:35. doi:10.1145/1773912.1773922.

[157] AWS. AWS Lambda Pricing n.d. https://aws.amazon.com/lambda/pricing/ (accessed November 11, 2018).

[158] Sutton RS, Barto AG. Reinforcement Learning : An Introduction. MIT Press; 2018.

[159] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing Atari with Deep Reinforcement Learning. arXiv 2016;2016–Janua:2315–21. doi:10.1038/nature14236.

[160] Nair A, Srinivasan P, Blackwell S, Alcicek C, Fearon R, De Maria A, et al. Massively Parallel Methods for Deep Reinforcement Learning 2015. doi:10.1109/IJCNN.2010.5596468.