ThermoCoach: A study of occupancy-based schedule-recommendations on energy costs and user comfort

A Thesis

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements for the Degree

Master of Science (Computer Science)

by

Devika Pisharoty

December 2014

Abstract

The largest portion of a home's energy consumption is attributed to its Heating, Ventilation and Cooling system(HVAC). Since the early 1900s, programmable thermostats have been studied as a potential tool to achieve energy savings in the home. However, studies have shown that conventional programmable thermostats are not used to their full potential due to several factors- difficult to use interfaces, lack of knowledge of working of HVACs and fading user interaction with the thermostats over time. To overcome this, 'Smart' thermostats detect the occupancy trends of a home and auto-generate schedules; thus eliminating the need for users to program their thermostats. Studies indicate that feedback of energy consumption has the potential to keep homeowners engaged with the energy usage in their homes and motivates them to take action to reduce energy consumption. This thesis presents ThermoCoach- An occupancy-based self-programming thermostat with eco-feedback. ThermoCoach uses occupancy sensors to detect occupancy patterns of a home and generates customized recommendations of thermostat schedules for a home. Schedule recommendations are provided to users through an online interface. ThermoCoach is evaluated against conventional programmable thermostats and the Nest Learning thermostat. For this pilot study, sensing systems were installed in thirty nine homes for a period of three months. ThermoCoach schedules reduced energy cost by 5% while Nest schedules increased costs by 7% when compared to programmable thermostats.

Approval Sheet

This thesis is submitted in partial fulfillment of the requirements for the degree of

Master of Science (Computer Science)

Devika Pisharoty

This thesis has been read and approved by the Examining Committee:

Kamin Whitehouse, Adviser

Mary Lou Soffa

Yanjun Qi

Kevin Sullivan

Accepted for the School of Engineering and Applied Science:

James H. Aylor, Dean, School of Engineering and Applied Science

December 2014

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Importance of Energy Use

Energy, in its various forms, plays an important role in our lives today. Energy is used to heat/cool our homes and offices, power our devices and machines and fuel our cars. Energy is also consumed majorly by manufacturing and other industries and also for transportation of goods. In our day to day lives, we take the availability of energy for granted and we are not always conscious of the impacts of wasteful energy use. In 2011, in the United States, total energy use per person (or per capita consumption) was 312 million British thermal units (Btu) [1]. For bituminous coal, this translates to approximately between 24,000-50,000 lbs of coal per person(or per capita consumption).

Over the years, energy consumption has increased faster than energy production. Main sources of energy include coal, natural gas and petroleum(Oil). Increasing demand on these resources is increasing the pressure to produce energy from these depleting resources, to meet tomorrow's needs. Significant amount of energy is wasted annually, costing homeowners and businesses financially. Hence energy conservation is vital.

Excessive burning of fossil fuels has lead to carbon pollution. Large concentrations of greenhouse gases have caused global warming. Global warming is causing glaciers to melt and sea levels to rise. Weather patterns have been altered and climate change is also having an effect on wildlife. Climate change is affecting agriculture and other industries. It has impacts on human health and more countries are at risk of water shortages as temperatures rise. The primary sources of greenhouse gases is the burning of fossil fuels for energy (3/4th's of total emissions) [2]. 32% of greenhouse emissions are from electricity production. In 2013, 39% of electricity was generated from Coal and 27% from Natural gas in the US [1]. Aggressive extraction of natural resources is causing deforestation which increases carbon dioxide percentages. As the planet is getting warmer, temperatures are on the rise throughout the globe [3]. Erratic weather, hurricanes, droughts are side-effects of climate change and increasing consumption of energy are putting a strain on energy resources around the world.

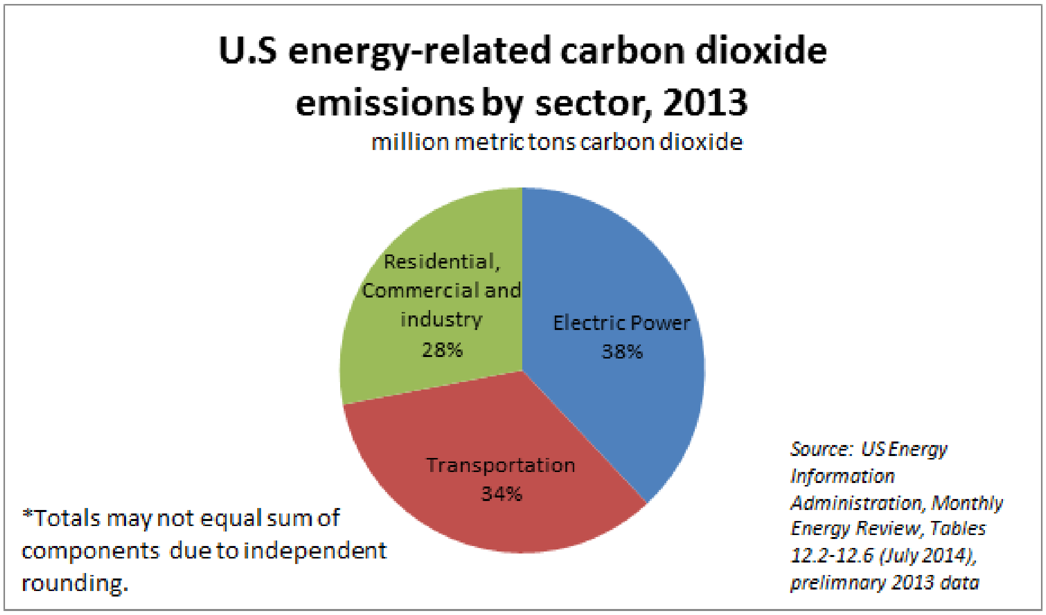

Figure 1.1: Carbon Dioxide Emissions by Sector

The Stern Review [4] predicts the global gross domestic product (GDP) to fall several percent due to climate change. Increasing energy demands have caused rising conflicts of resources. If domestic production is insufficient to satisfy energy needs of a country, it depends on foreign reserves. Out of the total energy needs, 84% of the US needs were satisfied by domestic sources [1](2013).

In order to address these issues, efforts are being made to address climate change and energy consumption. Renewable resources are being explored as alternative sources of energy. Technology is being improved to make appliances, infrastructure and vehicles more energy efficient than before. Several policies are in effect to curb greenhouse gas emissions. People are being made aware of the problems associated with excessive energy consumption. How energy is consumed impacts the environment and efforts are being made to reduce energy consumption.

1.2 Heating, Ventilation and Cooling(HVAC) Energy Usage

In 2010, the United States total energy consumption was about 19% of world total primary energy consumption. In 2011, 18% of the total energy was consumed in the world by the residential buildings while 12% of the total energy was consumed by commercial buildings, as shown in Figure 1.2. In the US, 40% of total energy consumption was consumed in residential and commercial buildings(2013) [5]. Homes are major source of energy usage. 49% of homes use natural gas and 15-20% use coal [6]. Out of the total global greenhouse gas emissions, 8% are from residential and commercial buildings. Homes and offices are good candidates for employing energy saving habits and mechanisms, at an individual level.

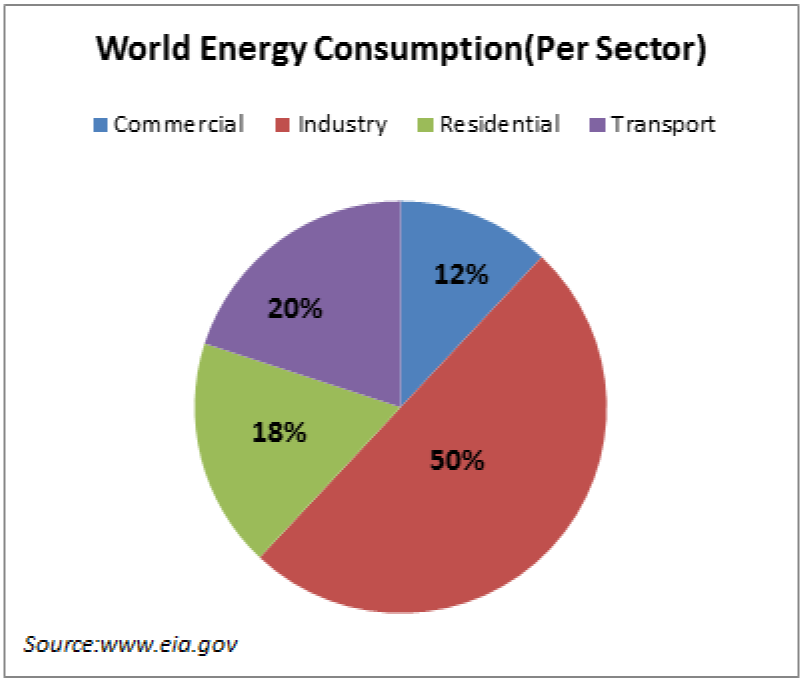

Figure 1.2: World Energy Consumption by Sector

In 2013, 11% of the US's energy consumption was from cooling residential and commercial buildings. Heating and Cooling homes consumes a significant amount of energy. 45% of a home's energy usage is attributed to Heating and Cooling and 6% to cooling alone [7]. Currently 2/3rd of the homes in the US have air conditioning which translates to about 85 million homes. Households have spent more than $11 billion annually on powering their Heating,Ventilation and Cooling (HVAC) systems. Homeowners pay anywhere between $700 - $2500 on their electricity bill [6] and average electrical prices in the US are on the rise, increasing 3.2% between 2013 and 2014 [7].

Significant amounts of energy can be conserved by reducing the amount of energy spent on heating and cooling spaces. Billions of dollars can be saved. Energy can be conserved in buildings and homes in a number of ways. Good design, better insulation and energy efficient equipment will reduce a home or building's energy needs. One of the cost effective ways to reduce the impact of Heating, Ventilation and Cooling (HVAC) systems is through the use of Programmable thermostats. Programmable thermostats have been around since the 1900s. The Residential Customer Characteristics Survey 2009 reported that programmable thermostats were installed in approximately 51% of households [8]. Programmable thermostats allow users to set temperature schedules to heat or cool their homes at different temperatures through the day.Programmable thermostats have been long thought to be a major source of energy savings in homes and studies have shown that annual energy usage in a home can be reduced by 10% to 30% [7].

Programmable thermostats however require users to program them manually. Studies have shown that occupants often do not have a good understanding of their daily patterns [9] and are thus unable to program their thermostats. Most people cannot remember the exact times that they typically wake up or leave the house and identifying occupancy patterns for multi-person homes is more difficult. Over the years, several features have been added to programmable thermostats. However, the complex user interfaces have made the thermostats difficult to use and often the programming feature is not used [10]. Often people lack motivation to adjust settings on their thermostats and keep track of energy usage. This is especially the case if the thermostat is installed in a room in the home that is not used. Most occupants do not bother changing the thermostat settings when they change their schedules, either temporarily or permanently. This leads to discomfort and users generally turn off the programming feature and begin to operate it manually. In addition to design issues, there are additional issues hindering potential energy savings, including lack of understanding of how HVAC systems work [10].

To overcome some of the challenges in the adoption and continual use of programmable thermostats, several studies have made a few common recommendations. The recommendations include better user interfaces, tutorials and guidelines on usage, enhanced user support, feedback on energy consumption and *smart* thermostats that can program themselves [10] [11]. *Smart* thermostats aim to gather information about a home's daily lifestyle and auto generate setpoint schedules. Ideally, these thermostats are designed to be able to generate tailored setpoint schedules for a home. Thus these system are designed to change the temperature when the home is unoccupied, without an input from homeowners. However, any defects in such systems would cause occupants to be uncomfortable and frustrated with the system and ultimately may lead to discontinuation of use.

1.3 Overview of Proposed Approach

This thesis presents ThermoCoach- a pilot study on the effects of occupancy-based thermostat schedule recommendations on energy cost and user comfort levels, in homes. ThermoCoach uses occupancy sensors such as motion sensors to detect and infer occupancy patterns of a home. The learned trends are then used to generate schedules that are presented to users through an online interface. Three schedules were presented to users with varying energy costs and comfort levels, in addition to feedback on energy consumption. The claim is that eco-feedback in the form occupancy-based schedule recommendations is more effective in keeping users involved in their home's energy consumption, leading to more energy savings as compared to no feedback or only energy-based eco-feedback. 39 volunteer homes were recruited for the study and custom sensors were installed in each of them. Data was collected for about twelve weeks. In this thesis, ThermoCoach is evaluated against a conventional programmable thermostat and the Nest Learning Thermostat- a state of the art, *learning* thermostat.

Chapter 2

Background & Related Work

Several programmable thermostats have been in the market since the early 1900s. The basic approach is to program a comfortable temperature when people are home and a more energy efficient temperature when they are away or asleep. Studies have shown that more than 50% of energy consumption in a home is by the HVAC, and programmable thermostats have the ability to reduce usage and costs by 20-30% [7]. In 2009 more than 33 million of U.S. households, had a programmable thermostat. Survey results conducted by the Department of Energy suggest that 14.5 million of these households do not currently use their thermostat for daytime setbacks and 11.6 million do not use nighttime setbacks [1] [12].

Programmable thermostats require users to set various parameters. Users are burdened with having to set their thermostats based on their schedules. Most people cannot remember the exact times that they wake up or leave the house over time and identifying occupancy patterns for multi-person homes is more difficult. A study by Krumm and Brush [9] show that people are not good at predicting their daily patterns. Participants were asked to carry a GPS device with them. Any GPS point within 100 meters of a participant's home was considered to be part of the participant's home. Participants of the study were asked to fill out a schedule of when they thought they were at home, away from home or sleeping. GPS data was used as ground truth. The authors found that their participants predicted that they would be home when they were actually away about 68% of the times. They concluded that most participants were good at predicting their bedtime but were poor at predicting home/away patterns.

Some households do not bother changing the thermostat settings when they change their schedules, either temporarily or permanently. This leads to discomfort and the users generally turn off the preset schedules and begin to operate it manually. Studies show that 20-30% of households that have programmable thermostats do not use setback temperatures when sleeping or away from home [13] [10]. Over time, people stop interacting with their thermostat. They are less interested in updating the schedules. User expectations and understanding of these devices do not line up with actual functioning of them, leaving users frustrated and ultimately leads to discontinuation of use of these devices. Studies show that a large number of people often do not know how to program their thermostats. [14] [13]

In the past few years, WiFi connected thermostats have come into the market. These allow users to remotely program and control their thermostat. This feature has been found to be useful to users but with time, homeowners lose interest. [11] Previous research has shown that very few people program their thermostat and the most common reason is that they find it difficult to do so because of poorly designed user interfaces. Only 50% of programmable thermostats are actually programmed to adjust temperatures at night or unoccupied times during the day, and thus they do not save much energy [10] [12]. Thus, programmable thermostats are not being used to their full potential.

This chapter introduces some terms associated with programmable thermostats and their schedules and the current state of programmable thermostats.

2.1 Background

Programmable thermostats allow users to set temperature schedules. In addition to setting the temperature, programmable thermostats today have complex user interfaces that provide a number of features, for example, In addition to being able to control their HVAC on some thermostats, homeowners can also view their past energy usage, view their home's past temperature settings, humidity levels and so on.

2.1.1 Terminology

A *setpoint* is a target temperature value an HVAC system should reach. With a programmable thermostat, a setpoint schedule can be set. Setpoints can be set at different times of the day to achieve different temperatures during the day. Thus a *thermostat schedule* consists of one or more setpoints scheduled to take effect at specific points.

Programmable thermostats can be used to set temperature schedules. A different schedule can be set for every day of the week. Alternatively, different schedules can be set for weekdays and weekends.

When a cooling system reaches a setpoint value, it cycles off. Temperature *setbacks* help save cost by reducing how often the heating/cooling system runs. A setback allows the home to slowly drift into a higher(during warm seasons) temperature or lower(during cold seasons) temperature. *Setbacks* can be set when the occupants of a home are away or when they are asleep. When occupants are away or asleep, a more energy efficient setpoint can be used. According to a study by the US Department of Energy, it is estimated that 1% energy savings can be achieved for every one degree Fahrenheit setback for an eight hour period. Thus a 5 degree setback at night and when the home is unoccupied, has the potential of 10% savings of utility bills. [7]

2.1.2 Current User Interfaces

Early thermostats had a simple dial with two needles to indicate current and target temperatures. User interfaces for programmable thermostats have evolved over time and today they often have electronic and a large amount of information is provided- the current and target temperatures, outside temperature, humidity levels, past energy usage, interface to set schedules, system status information and so on. Various new features are constantly being added by manufacturers.

As thermostats became more sophisticated, however the user interface also became complex and users often did not know how to program the thermostat and use its advanced features. Studies show that most participants find thermostat controls very difficult to use [12] [15] [13]. Users end up using it in HOLD mode (as a manual thermostat) and true potential savings are not achieved. Studies have highlighted some of the issues of complicated interfaces, including terms and abbreviations that homeowners are not familiar with, poorly layed out and hard-to-navigate interfaces. A review by the US Environmental Protection Agency(EPA), along with other studies have indicated that people find programmable thermostats difficult to program. [10]. Users lack the motivation to understand how to use their programmable thermostat. [16] Peffer et al. show that homeowners sometimes did not know how to override their settings when their schedule changed [10]. They ended up using the thermostat as a manual thermostat, relying on themselves to setback the temperature when they left home. Studies show that homeowners want devices with advanced features but an easy-to-use interface that does not need a lengthy user manual. Lack of understanding of setpoints and complex user interfaces are causes for programmable thermostats to not be used effectively.

S. Karjalainen [17] found that good thermostat controls are essential for user comfort in addition to energy savings. Research into better interface design has been conducted [13] [18]. Some of the recommendations include: grouping settings into basic and advanced features, providing energy usage feedback, simplistic controls, identifiable and enhanced symbols that are intuitively understandable.

2.1.3 Energy Feedback

Household energy consumption is *invisible* to consumers and is a major cause of waste. Occupants typically have only a vague idea of their energy fingerprint which makes energy management difficult; Most homeowners are ignorant of their energy use, making it harder for them to adopt energy conserving measures. Research has been conducted into energy-feedback or eco-feedback [19] [20] [10] as a source of providing users with their energy usage data to motivate them to improve their energy usage. Knowledge of energy use will allow homeowners to make changes to their daily behaviors. Without any feedback on energy usage, most homeowners are 'blind' to their energy consumption. [21] The only feedback generally is in the form of their monthly utility bill showing the cost and total energy used in terms of number of Kilowatt-hours(kWh) Energy feedback gives households the flexibility on how energy savings can be achieved. Energy feedback was evaluated as early as in 1979 by McClelland & Cook [22]. Early research focused on the type of feedback that is useful in motivating remedial action. Later on the type of delivery method for the information was studied. [20]

Mcclelland et al. [22] found that homes with energy feedback saved on average 12% more than homes without any feedback. More recent studies have shown that feedback has the potential for 4-14% energy savings depending on the technology used. [20] Several studies [19] have indicated that homes with energy feedback tend to discuss their energy use and change their usage patterns. A survey conducted by Wood et al. [23] showed that 80-93% of their participants changed their behavior pattern by reducing the use of the air conditioner or turned down the temperature of their heaters. Another study by Lutzenhiser et al. showed that approximately 48% of the households in their survey modified their heating/cooling behavior. [24]

Behavioral research [25] [26] has been conducted to study the relationship between energy consumption and behavior. One analysis [25] suggests that feedback has three stages. A learning stage during which the occupants become aware of their energy consumption and they make a number of minor changes. Minor changes lead to habits (the second stage) and finally 'internalization of behavior' when people behave in an energy conserving way without being actively conscious of it. Feedback can be direct or indirect feedback. Direct feedback provides real time input on how changes to settings will affect energy cost. Sources of indirect feedback include utility bills, energy audits and contain average or summarized usage value. Historical feedback of previous energy usage has been found to be useful. [27] Studies have shown that feedback in the form of utility bills, energy audits or weekly feedback can lead to consumption savings of up to 10% [27]. Neenan B et al. [25] show that on thermostats, feedback on how changes to the temperature affect energy usage and energy bills is useful to homeowners when they make decisions on their thermostat use. Thus continual direct and indirect energy feedback has energy saving potential.

2.1.4 Commercial Buildings

HVAC systems in commercial buildings consume a lot of energy. Commercial spaces are significantly different from homes. Occupancy patterns of homes and commercial buildings is very different.Most commercial spaces are occupied during fixed working hours and are unoccupied most of the other times. Most commercial buildings have centralized heating and cooling. In some cases, room level control is provided. Large buildings have zoning systems allowing different zones to be heated/cooled independently. The majority of the commercial buildings have motion sensors and other sensing systems to identify when a building or specific room is occupied or unoccupied. These sensors could be coupled with zoning systems to only condition occupied spaces. Unoccupied rooms can be set to a setback temperature. Occupancy patterns in commercial buildings do not change often and generally an operator is in charge of controlling the settings and modifying schedules on the thermostat(s). Irrespective of whether occupancy data is used, there is always an operator that has knowledge of the building usage or has access to occupancy data. This operator can then make informed decisions on the thermostat schedules. On the other hand, most homes do not have zoning systems and occupancy behavior in homes changes often. Changes in occupancy behavior need to be reflected in thermostat schedules and this needs to be done by one of the home's occupants. Many people do not update their thermostat schedules, ultimately leading to discomfort and/or energy wastage.

2.2 Related Work

To overcome the challenges in the use of programmable thermostats, several researchers developed and presented *Smart* thermostats that were able to detect and learn occupancy patterns of a home and eventually be able to predict occupancy, which was then used to program thermostats. Each of these approaches used different sensing systems and machine learning algorithms to learn and predict occupancy.

2.2.1 Research Prototypes

Mozer et. al present NeuroThermostat [28] an adaptive control algorithm that uses Neural Nets to predict occupancy. The tradeoff between energy savings and user comfort is combined into a single metric which is optimized with their model. The system was evaluated on simulated data. Real occupancy was collected from their NeuralHouse, a smarthome instrumented with 75+ sensors, results of which were not presented. In the evaluation presented, NeuroThermostat required 5+ months of training data. NeuroThermostat performs well under the assumption that occupants have regular, weekly occupancy patterns. Thus it does not respond quickly to changes in schedules.

Gupta et al. used live data from mobile phones or in-vehicle GPS devices to control home heating and cooling [29]. Their system heats the home during time it takes a person to travel home from the current location. However this method requires the users to always carry their smartphones or GPS device with them. While GPS today has improved, GPS enabled devices still drains power and require users to carry the device or a GPS enabled smartphone with them. This system is not sensitive to different states of occupancy within the home. Activity levels of the home are not available and its not possible to identify times when the occupants of the home may be asleep. Knowledge of sleep times can be used to program setbacks during sleep time.

J. Scott et al. [30] present a system called PreHeat which uses occupancy prediction to create efficient setpoint schedules. PreHeat used motion sensors and RFID tags to detect occupancy. Participants of the study were made to carry around RFID tags and these tags were attached to the keys for the home. Guests and visitors were also given tags. Participants were requested to take their keys with them whenever they left their home. Thermostats in their test homes were replaced by custom hardware. All the units (sensors and custom hardware) had ZigBee radio modules. When a space is not occupied, the system tries to predict when it will be occupied next using collected historical data. K-nn was used to classify a new instance. K was set to three. Hamming distance was used as the evaluation metric. The mean prediction accuracy of the system for the whole house lies between 80-85%.

Lu et al. present SmartThermostat, an HVAC control algorithm that uses Hidden Markov models [31]. Occupancy is predicted from occupancy data collected from 8 homes using X10 motion and door sensors. The approach was evaluated using simulation. The approach works for two-stage HVAC systems with two stage setbacks- shallow and deep.

Gao et al. present the Self-Programming Thermostat that automatically generates schedules based on a home's occupancy patterns. x10 motion sensors were used to detect occupancy. The generated schedules with varying energy use and comfort levels were presented to users and users could then accept one of the schedules. Users could were given feedback on energy use and comfort levels of the schedules as users made modifications to a schedule, in real-time [32].

2.2.2 WiFi Thermostats

Recently several thermostats connect to WiFi are available commercially. The first WiFi thermostat was released in the market in the early 2000s. These thermostats connect to WiFi and thermostat usage data is collected. Web and mobile interfaces are provided to users, thus allowing them to control their thermostats remotely- change the temperature, turn their AC on or off, set schedules and even begin heating or cooling before they reach home. In addition to remote availability, some thermostats also provided periodic reports of energy usage. Knowledge of how many hours their system was running, a history of energy usage is useful in helping occupants make informed decisions to reduce their energy usage. These thermostats however require users to manually program schedules and are not capable of auto generating schedules tailored to a home. Ecobee, Honeywell, Bayweb along with other manufacturers WiFi enabled versions of their thermostats.

2.2.3 Learning Thermostats

The Nest Learning Thermostat, is one of the few learning thermostats available in the market today [33]. In addition to providing users with remote control of their thermostat, the Nest thermostat automatically generates setpoint schedules for the home. It learns from temperature changes made by users in an initial learning phase that lasts about fourteen days. It uses this information to automatically generate schedules for a home. The thermostat has an inbuilt motion sensor which is used to detect when occupants typically leave home and automatically sets back the temperature to an `away` value. This feature is called as Auto-Away. The details of the learning algorithms are not available.

Nest has several additional features such as monthly energy reports, eco feedback on the thermostat, making users more conscious of their interaction with the Nest [33].

Some of these features are listed below along with a brief description [33]:

1. Auto Away: Nest senses that occupants may have left the house and will adjust temperature to avoid conditioning an empty home. It sets a *setback temperature* when it thinks the home is unoccupied. For Auto-Away to work,the Nest is required to be installed in a room where it can detect activity whenever occupants are home [33].

2. Auto Schedule: Nest generates setpoint schedules automatically by learning from temperature changes made by users in the past. The thermostat is then programmed to follow the setpoint schedule generated.

3. Time-to-Temp: Nest estimates how long it takes to heat or cool the home and it shows how long it will take to reach the target temperature. For users this may reduce the temptation to set a new target temperature if the previous target temperature has not been reached but users know how long it would take to reach it.

4. Early On: Based on the Time-to-Temp and the weather, Nest begins pre-heating or per-cooling in order to achieve the target temperature at the right time. This allows the home to conditioned before the specified time and ensures that home is comfortable, reducing the number of manual temperature changes made by occupants.

5. CooltoDry: If this feature is on, Nest will turn on air conditioning to decrease humidity if it senses high humidity in the home.

6. SunBlock: This feature is set only if direct sunlight falls on the thermostat . If the thermostat is in direct sunlight, Nest adjusts the indoor temperature it reads.

7. Airwave: If this feature is set, Nest turns the compressor off a few minutes before the target temperature is attained, using only the fan to cool.

8. Lock: Users can set a specified temperature range in which the target temperature should be. This will ensure that the thermostat is not set to a temperature not in that range accidently, unless a 4-digit pin is entered.

The Nest learns changes to settings made by users and does not take into account the occupancy of the home. Thus it does not learn occupancy patterns of the home and is sensitive to the changes made by the users. Thus even a temporary change may be picked up by the learning mechanism and reflected in the schedule. The learning algorithm seems to rely on users to update settings. Overtime however the novelty of the device wears off and occupants stop interacting with thermostat. A single motion sensor is not sufficient to detect when a home is typically unoccupied.This method assumes that people pass by the thermostat before exiting the home, which may not always work. One of the major problems in the use of the Nest is the misunderstandings people have with its working. Yang et al. [11] present findings from a year-long study in which participants with Nest thermostats in their homes, were interviewed periodically about their experience with it. The study shows that after some time, the novelty of the product wears out and occupants interaction with their thermostat reduces. The motivation to make changes to the schedule faded and users ended up using the thermostat as a manual thermostat. Unless the system made a significant change, people began to ignore the device, expecting it to work on its own. Users over-relied on the thermostat to identify changes in their schedule. They expected it identify when they went on vacation and turn their HVAC off. Hence some of those homes were conditioned even when occupants were away on vacation since the occupants did not turn off the system before they left home. This lead to increase in energy usage in those homes over time.

Chapter 3

ThermoCoach

ThermoCoach is a *self-programming* thermostat which creates thermostat schedules based on occupancy patterns of a home, while providing eco-feedback to users. In addition to giving users remote access to their thermostats, ThermoCoach provides setpoint schedule recommendations to users periodically, thus keeping them engaged in their energy usage. The claim of this work is that periodic eco-feedback in the form of schedule recommendations, along with the features available in most WiFi thermostats today, will help in maintaining energy savings using programmable thermostats.

ThermoCoach relies on occupancy sensors that help the system infer when a home is typically occupied, unoccupied, when most occupants are asleep and when they typically wake up. Occupancy data collected over time are used to detect activity patterns which are then used by the system to make schedule recommendations through an online interface.

3.1 Occupancy State Detection

To generate customized schedules for homes, it is essential to identify states of the home through the day. Once historical data of a home's state is gathered, schedules can be generated. ThermoCoach defines the home to be in one of three states: `Away`, `Asleep` or `Active`. A home is in `Away` state when all the occupants of the home are not at home and the home is unoccupied. `Asleep` is a state that represents when people are asleep at night time. When a home is neither in `Away` state nor `Asleep` state, it is in `Active` state. Thermocoach detects and infers the state of the home at 15 minute intervals. During a twenty four hour period, there are 96, 15-minute long, intervals. For each home, an `Occupancy` vector is created. The vector is a two dimensional array. Each row is indexed by the date the data corresponds to. Each row of the two dimensional vector is a vector itself, with 96 elements, each representing 15-minute intervals. Each entry in the Occupancy vector can take one of the three values: `Asleep` , `Away` and `Active`.

3.1.1 Away State

Occupancy sensors are used to detect if a home is occupied or not. From the data of occupancy sensors the location of a person with respect to the sensor can be inferred. These sensors can also been used to infer periods of time when there's no activity in the home. There are several types of occupancy sensors that could be used- Motion sensors that detect movement in their field of view, GPS devices can be used to know how far occupants are from their homes or RFID tags can be used to detect when an occupant wearing the tag is within range of a receiver installed in the home.

A home is identified as occupied or not based on occupancy sensor data and from that, an `Away` event is defined as a 15 minute interval such that the interval before and after it are unoccupied. If an interval is not labelled as `Away`, it is labelled as `Active`.

Once `Occupancy` vector for a home is processed, an `Away` percentage vector is computed. It defines the percentage of times the home was unoccupied at the corresponding 15-minute interval.

$$Away_i = \frac{\sum_{day\ \epsilon awayDays}[\forall \text{Occupancy[day][i]} = \texttt{Away}\ ]}{\text{numAway}} * 100 \tag{3.1}$$

where numAway is the number of days `Away` events were detected for the home and awayDays is the set of days when both `Away` and `Active` events were detected

During the study, home occupants went on vacations. Days when they left for vacation and days when they came back from vacation do not represent their typical patterns. Data loss also prevented detection of `Away` states on certain days. In order to exclude these days in further analysis, any days on which the home was not occupied for atleast 10 hours were removed.

3.1.2 Sleep Detection

Sleep detection is defined as inferring periods in a home when occupants are asleep. This information can be used to detect changes in lifestyle and can also is useful for smart thermostats that can set back to a temperature suitable when the home's occupants are asleep. ThermoCoach uses this information to generate thermostat schedules in which the temperature is set back to a `sleep` temperature-defined by the participant-when the home is in sleep state.

If a home is occupied at an interval i, it is labelled as `Asleep` if it is night time and there is no activity in the home. A `Sleep event` is defined as the last interval before 3AM that the home was active. A `Wake` event is defined as the first interval after 4AM that shows activity. The period between sleep and wake is when the home is `Asleep`. Detection of any of the above events occurs only if the occupants are home at the corresponding points in time. If the home is unoccupied, no sleep period is detected. `Asleep` is defined as the period between `Sleep` and `Wake`. Occupancy vector is populated with these states. Due to data loss, on some days, it is possible to detect only one of `Sleep` or `Wake`.

`Asleep` Percentage vector is computed as the percentage of times the occupants of the home are asleep at interval i

$$Asleep_i = \sum_{day\ \epsilon\ sleepDays} \frac{[\forall \text{Occupancy[day][i]} = \texttt{Sleep}]}{numSleep} - \sum_{day\ \epsilon wakeDays} \frac{[\forall \text{Occupancy[day][i]} = \text{'Wake'}]}{numWake} \tag{3.2}$$

where numSleep is the number of days `sleep` *events were detected for the home and* `sleepDays` *is the set of days when* `sleep` *events were detected , numWake is the number of days* `wake` *events were detected for the home and* `wakeDays` *is the set of days when* `wake` *events were detected*

Due to data loss from sensor failures, on some days only one the two events- *sleep* or *wake*, were detected. The data pre-processing step is also responsible for discarding certain events that are not typical for a home, for example on days when occupant's are returning from vacation. Thus *numSleep* and *numWake* may not have the same value.

In addition to the above vectors, two more are computed-

1. Asleep': Defines the percentage of times occupants were home and alseep at interval i

$$Asleep'_i = (100 - Away_i) * Alseep_i \tag{3.3}$$

2. Active Percentage: Defines the percentage of times the home is occupied and active at interval i

$$Active_i = 100 - Away_i - SleepPrime_i \tag{3.4}$$

3.2 Schedule Generation

Recommended schedules are generated using an implementation of the algorithm in the Self-Programming Thermostat [31]. The algorithm outputs schedules with four setpoints. The algorithm generates all possible schedules with four setpoints. Four setpoints are used, corresponding to the four states a home can be in- `Asleep` setpoint, `Wake` setpoint, `Leave` and `Arrive` setpoints. The times these setpoints are set at are denoted as tSleep, tWake, tLeave and tArrive. The algorithm generates schedules with varying tWake, tLeave, tArrive and tLeave setpoint times. This order of the four setpoints is maintained in all of the schedules generated.

For each time interval, the algorithm calculates the percentage of times the home was in a particular state at that interval. The interval was chosen to be 15 minutes long, since Nest allows scheduling at the same granularity. The algorithm uses this data while trying to optimize two functions: *Energy Cost* and *Miss Time*(defined in the section below) and generates three schedules with varying Energy Cost and Miss Time values. The three schedules have decreasing energy cost and increasing miss time. This gives users the choice of selecting an energy efficient schedule with high Miss Time, a energy costly schedule with lower Miss Time or something in between.

To generate the schedules for each home, ThermoCoach determines the percentage of times a home was `Away`, `Asleep` and `Active` at different intervals throughout the day as described above.

3.2.1 Miss Time Function

Miss Time is used to indicate the number of minutes during the day that occupants may be uncomfortable. Alternatively, it is number of minutes the temperature is not what it should be for the state of the home in consideration. ThermoCoach uses two setbacks- a four degree setback for `Asleep` states and an eight degree setback when the home is unoccupied. These setbacks are relative to the temperature the home's occupant set in the `Active` period. Thus this value was obtained from the home's manually set temperature setting.

Miss Time is used to calculate how many times the schedule expects the home to be unoccupied(or asleep) but the home is actually occupied at that time.

Away_MissTime is defined as the number of minutes the home is considered unoccupied by the schedule, even though it is not.

$$Away\_MissTime = \sum{}_{\mathrm{i=(tLeave,tArrive)}} [\frac{(Active_i * 15) + (SleepPrime_i * 7.5)}{100}] \tag{3.5}$$

Asleep_MissTime is defined as the number of minutes the home is considered to be 'asleep' by the schedule, even though it is active.

$$Asleep\_MissTime = \sum{}_{\mathrm{i=(tSleep,tWake)}} [\frac{(Active_i * 7.5)}{100}] \tag{3.6}$$

The sleep time setpoint(tSleep) can be either higher than or lower than the setpoint occupants use when they are active.(twake) When the sleep time temperature can be the same temperature or a temperature higher than tWake, Asleep Miss Time can be calculated using Equation 3.6. In some cases, homes may prefer their sleep temperature to be lower. Thus occupants may be uncomfortable if tWake is before the actual time they wake up. Similarly, if tSleep is much later than a home's bedtime, some amount of discomfort will occur. For such cases, Miss Time for sleep is calculated using the equation below.

$$Asleep\_MissTime' = (\sum\nolimits_{\text{i=(tWake,tLeave)}}[\frac{(SleepPrime_i * 7.5)}{100}]) + (\sum\nolimits_{\text{i=tArrive to tSleep}}[\frac{(SleepPrime_i * 7.5)}{100}]) \tag{3.7}$$

Finally, Miss Time is the sum of Away and Asleep Miss Times, calculated as-

$$MissTime = \text{Away\_MissTime} + \text{Asleep\_MissTime} \tag{3.8}$$

3.2.2 Energy Cost Function

This cost function scores the energy use of a schedule. Cooling a home at lower temperatures and for longer periods is expensive. The function estimates a cost of a schedule by adding up the costs of cooling a home at the setpoints for the corresponding durations.

The lowest temperature in a schedule is calculated as-

$$lowestTemp = min\{\text{all temperature setpoints in a schedule}\} \tag{3.9}$$

The cost function for a schedule is defined as-

$$cost = \sum[(1 - (0.06 * (setpointTemp_i - lowestTemp))) * (setpointTime_{i+1} - setpointTimei)] \tag{3.10}$$

where i = 1 to n , where n is the number of setpoints in a schedule, setpointTemp is the temperature value of the setpoint and, setpointTime is the time the setpoint is set at.

A schedule that wastes energy the most, by keeping a constant temperature throughout the day, has an energy cost of 1. As the efficiency of schedules increases, the cost value decreases. As energy cost decreases, Miss Time increases.

3.3 Schedule Recommendations

For each generated schedule its *Miss Time* and *Energy Cost* is computed. Once the set of schedules are generated, three schedules are selected to be presented to the user. These schedules have minimum *Energy*

Cost and their *Miss Time* is less than a defined upper bound. The upper bound is defined as a certain percentage of the typical, daily total `Active` time for a home. Three schedules called `High Comfort`, `Energy Saver`, `Super Energy Saver` are generated with the minimum costs and miss times within the bounds below.

$$totalActiveTime = \sum_{i=(0,95)} [Active_i * 15] \tag{3.11}$$

1. Upper bound of acceptable MissTime for High Comfort

$$0.10 * totalActiveTime \tag{3.12}$$

2. Upper bound of acceptable MissTime for Energy Saver

$$0.20 * totalActiveTime \tag{3.13}$$

3. Upper bound of acceptable MissTime for Super Energy Saver

$$0.30 * totalActiveTime \tag{3.14}$$

Figure 3.1 shows a sample historical plot. The dark regions depict percentage of times the home is asleep at each 15 minute interval. `Away` events are depicted by the light curve and the region below it. The schedules generated vary in beginning and ends of `Sleep` events and `Away` events. The different times are selected along the curve.

In Figure 3.1, the home typically is unoccupied from 10AM(interval number 40 in the figure) until 4:30PM(interval = 66),about 70% of the time. During 12:30AM(interval = 2) to 5AM (interval = 20) the home is typically is asleep.

Schedule recommendations are emailed to homeowners. The email contains snapshots of four categories of schedules. One is the home's current schedule and the remaining are recommended schedules. The recommendations vary in Energy Cost and Miss Time. Each category of recommendations has two schedules- for the week and one for weekends. Each recommendation is annotated with possible energy saving percentages and the impact on comfort level. Figure 3.2 shows a sample email generated. Users can accept a recommendation as is or edit a recommendation. If they choose to Edit a recommendation, they are redirected to the ThermoCoach webpage. Figure 3.3 shows a webpage from one of the recommendations for a home. In addition to the schedules, estimated energy savings over keeping the same temperature, are displayed. An

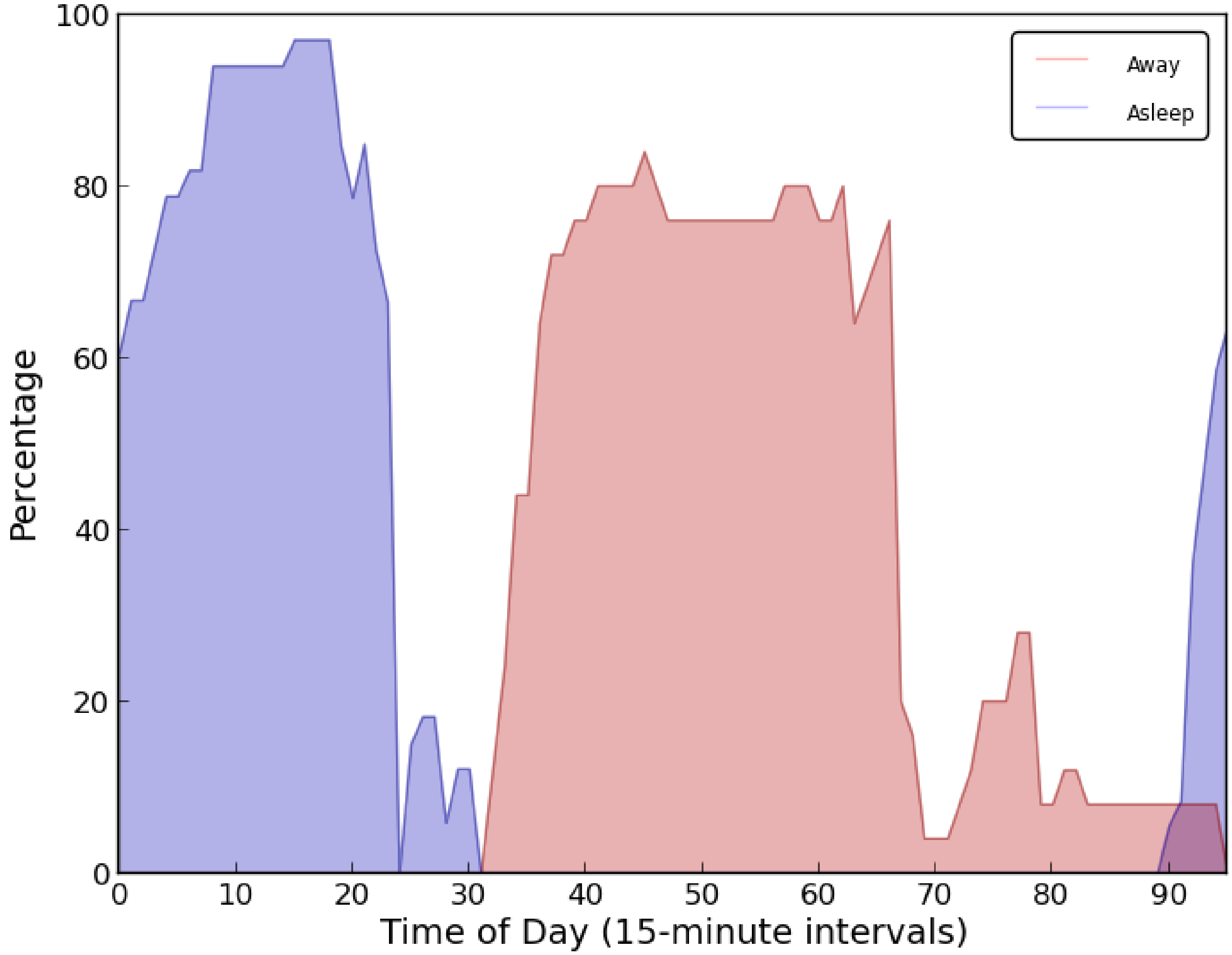

Figure 3.1: Historical State Information

occupancy graph is also displayed. The graph depicts the occupancy trends of the home. It displays how often a home was active at different times through the day calculated from Equation 3.4.

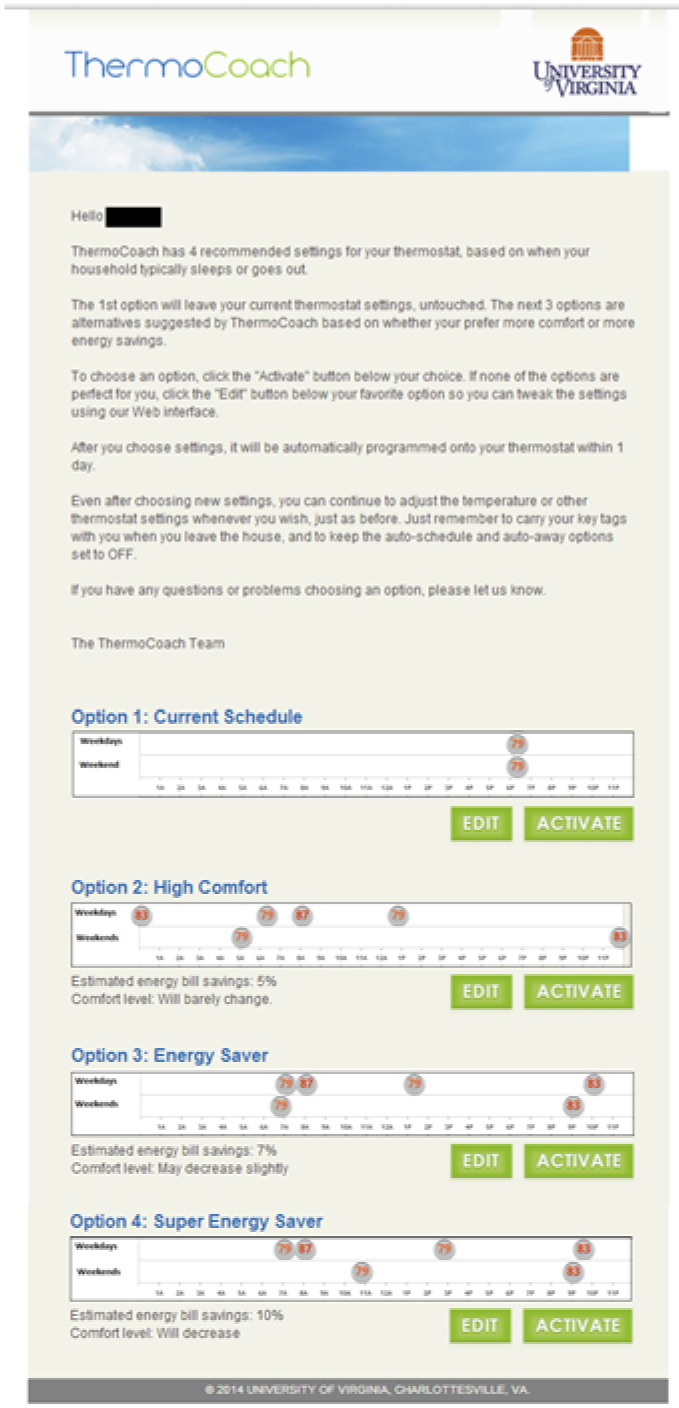

Figure 3.2: ThermoCoach Email Recommendations

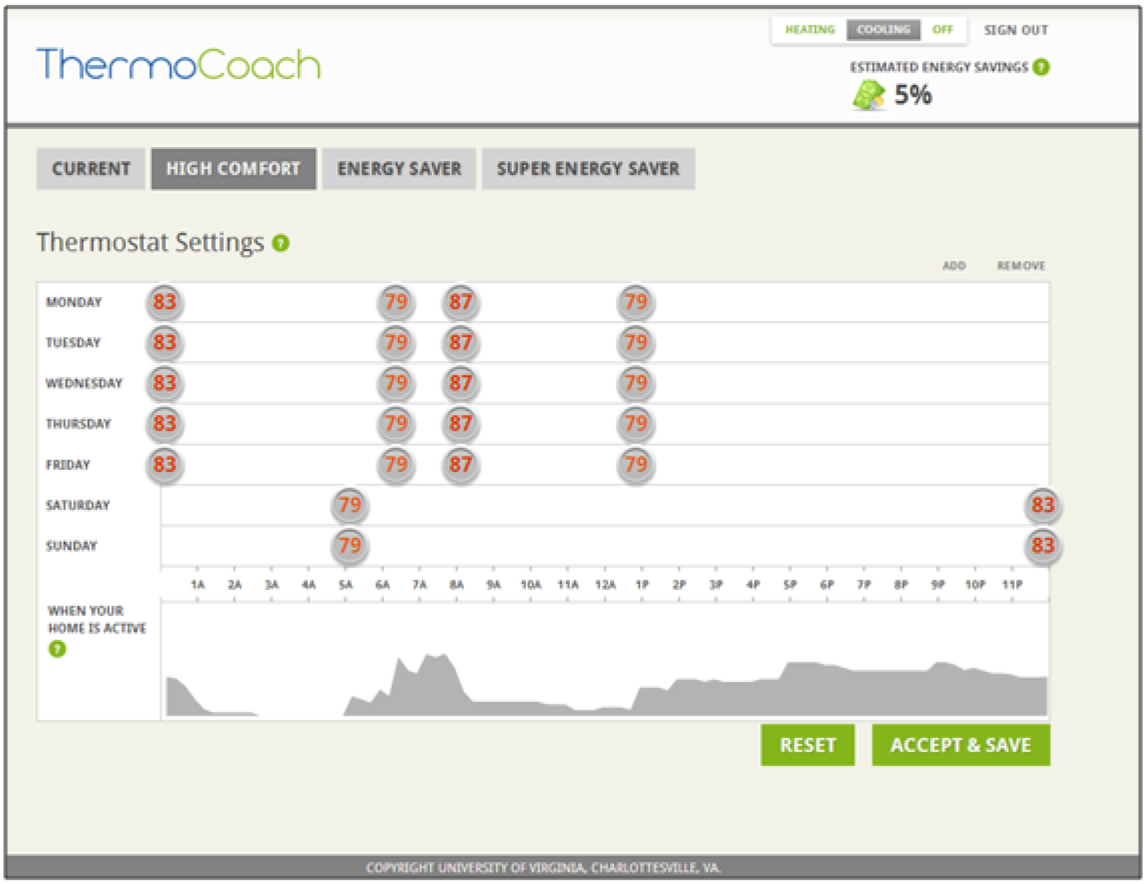

Figure 3.3: ThermoCoach Webpage

Chapter 4

Implementation

While occupancy data can be obtained from a number of different sensors, for this study, ZWave Motion Sensors and Bluetooth 4.0 key fob sensors were used. Instead of using expensive devices and data collection platforms such as HomeOS [34], a new platform Piloteur was designed and implemented as a part of this study [35]. Data was collected using Raspberry Pi's running Piloteur. A Nest thermostat was installed in each home and usage data was collected. ThermoCoach currently does not include design or implementation of the hardware of the thermostat or the control algorithm. The commercially available Nest Learning Thermostat was used. The system relies on Nest's Time-to-Temp feature to precondition a home such that it reaches the target temperature gradually just as the time of the setpoint is reached. The ability to turn on or off the various features of the Nest thermostat allowed for a comparison between ThermoCoach and the Nest thermostat.

4.1 Occupancy State Detection

4.1.1 Motion sensors

To detect active periods in a home, commercial off-the-shelf, ZWave PIR motion sensors were used. Passive Infrared (PIR) based motion sensors sense movement within their field of view. A differential in the received infrared radiation indicates a change of state. PIR sensors have found popular use in home automation systems and security systems. PIR sensors generally have a field of view between 110° to even 360°. They are small, inexpensive and low power devices.

ZWave is a propriety wireless protocol used for home control and monitoring. ZWave also allows applications to be built in such a way that ZWave devices talk to each other, thus enriching the home environment in many ways. ZWave requires a gateway or controller to be installed in a home which communicates with the ZWave products in the home. Applications can be written that interface with the gateway, making it possible for homeowners to control the appliances in their home from PCs, tablets, smartphones, etc. Besides its use for home automation, ZWave devices have been used in applications to conserve energy; for example, various ZWave thermostats can be controlled via the Web and in home security systems.

This study used Schlage S200HC V N N SL motion sensors. These are battery operated PIR sensors that communicate with a ZWave controller. They have a detection area is approximately 9 x 12 meters, with about 120° wide angle detection pattern. They can see up to 100 feet (30.5 meters) line-of-sight. The sensors are event driven and *sleep* to conserve battery. They wake up periodically, when polled by the ZWave controller, or when their state has changed. [36]

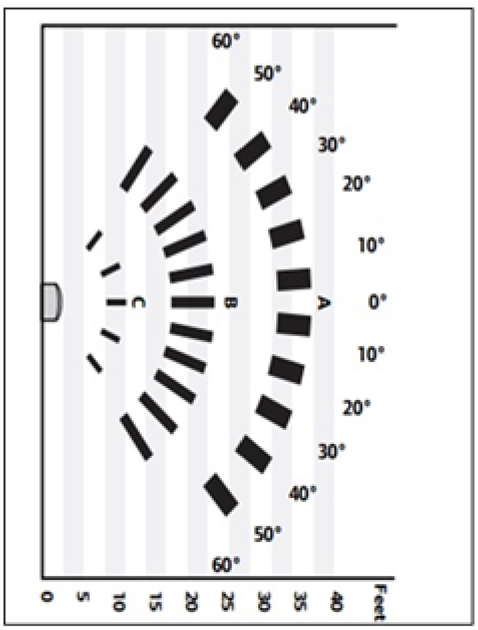

Figure 4.1: Top View of Motion Sensor Range

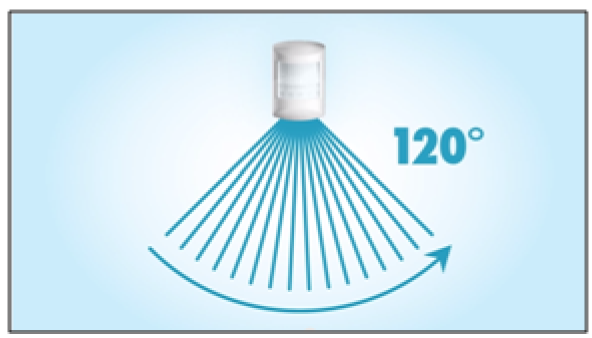

Figure 4.2: Front View of Motion Sensor Range

4.1.2 Bluetooth Low Energy Sensors

Bluetooth Low Energy(BLE) wireless technology consumes only a small percentage of the power of classic Bluetooth radios. [37] These small, low power and low cost, coin-cell battery operated sensors were created with the vision to be used in wearable devices, human interface devices such as keyboards and other *smart* devices. The battery is designed to last about a year without recharging. ThermoCoach uses these Bluetooth 4.0 sensors to detect occupancy. Participants in the study were asked to carry these sensors with them whenever they left home. StickNFind Bluetooth 4.0 tags [38]- small, quarter-sized, battery operated BLE sensors, were used in the study. They have a range of approximately fifty meters, line of sight. However, the range depends on the range of the Bluetooth Adapter used and most of the commonly available Bluetooth 4.0 USB adapters have a range of seven to eight meters. Bluetooth Low Energy technology has a *advertising* functionality that makes it possible for a slave devices(sensors) to announce that it has something to transmit to other devices that are *scanning*. *Advertising* messages can also include an event or a measurement value, Media Access Control (MAC) addresses, a device name, etc. [37] The StickNFind tags were programmed to advertise their MAC addresses.

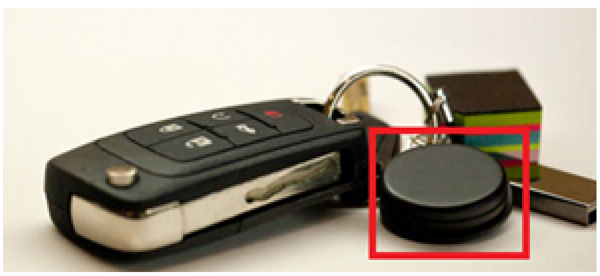

Figure 4.3: Bluetooth 4.0 Key fob sensor

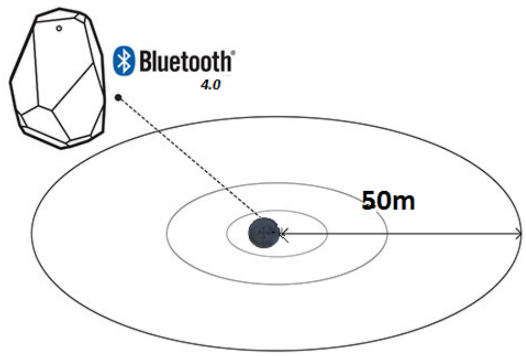

Figure 4.4: Bluetooth 4.0 sensor Range

4.1.3 Data Pre-Processing

During preliminary analysis of Bluetooth data, it became apparent that in a few homes, the script logging Bluetooth data was logging a number of different MAC addresses. Each home had about 3-4 BLE tags. In addition to the tags, other devices such as smartphones, tablets etc. may have their Bluetooth turned on. Some of the MAC addresses of electronic devices of home occupants may have been logged as well. To filter these out, stray MAC addresses that occurred only a few times (order of tens and hundreds), were removed. Since BLE data collection script was always running, typically on an 8 hour workday, any given tag was logged 1000+ times in data files. Tags that were seen only on one day and never seen again in consequent weeks, were filtered out too. Participants were requested to carry their tags with them. Even then, often an extra tag was left behind, especially in homes that had attached them to their car keys and had multiple cars. Tags that were always at home were filtered out and feedback from participants was used to verify that they did indeed leave a key tag behind. Participants were also periodically reminded via email to carry their tags with them.

4.1.4 Removal of Partial Data

Throughout the study, some homes had intermittent data. Days that had *partial data* were removed from future analysis as they are not representative of a home's daily patterns. A day is considered to have *partial data* if only `Leave` or `Arrive` events were detected but not both. A day is also considered to have *partial data* if no `Asleep` state was detected on the day. Days on which occupants were on vacation or returning from vacation were also removed. Days on which the home was unoccupied for less than 10 hours were removed from the data for that home. The assumption is that most homes in the study were unoccupied for at most 8 hours a day. This resulted in removal of days when participants were on vacation and also days where a sensor failed. In some cases, occupants left their keys behind when they left home. This violates the assumption that participant's carry their keys with them when they leave home. For a home, MAC addresses that were logged by the Bluetooth 4.0 driver on less than 50% of the study duration were filtered out as noise and were not considered in any analysis.

ZWave motion sensors have their own set of false positive and false negative rates. ThermoCoach assumes that in each home, there was at least a three hour period when all occupants of the home were asleep, between 9pm at 1pm. Homes with pets were identified before the study began. Any movements at night time which caused a sleep period to be less than three hours was attributed to movement by a pet and were ignored. Days with sleep periods less than three hours were considered days with *partial data*. A filter was created to parse occupancy data and remove days with *partial data* for the homes.

Figure 4.5 shows the Bluetooth data from different homes in the study. Each sub-figure represents data for one day. (A) shows a case where multiple Bluetooth addresses were logged. The home had only two tags but the graph shows a total of five Bluetooth Low Energy devices being detected. (B) shows a typical day where tags were taken by participants when they left home. (C) is an instance when a tag was left at home all day. In the absense of any additional information, it impossible to say if a participant left the tag behind or was genuinely home on a day. For this reason, tags are filtered over multiple days.

4.1.5 State Detection

Bluetooth Low Energy (BLE) tags were used to determine if a home is occupied or not. When tags were present, the system assumes that the home was occupied and assumes that it was unoccupied when no tags were present. Each participant was given a BLE tag to place on their keys or the one item they always carry with them when they leave their home. Participants were asked to always keep their keys in the same place when home. A Bluetooth receiver was placed at that location to detect the tags. Each home had three receivers that collected BLE data, to mitigate data loss due to Bluetooth range limitations and sensor failures.

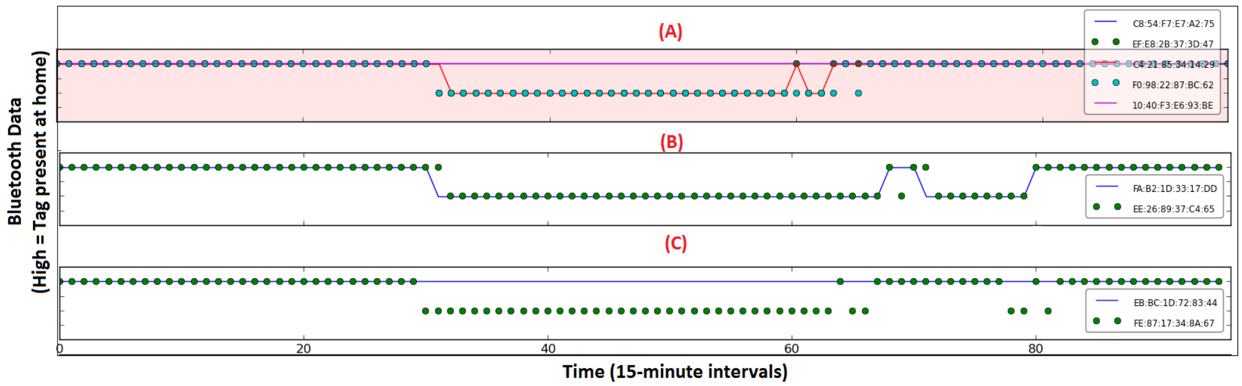

Figure 4.5: Raw Bluetooth Data

Each BLE data entry in the Bluetooth data collected contained the timestamp, MAC address of the tag and signal strength. Each entry was passed through a filter which determined whether to keep or discard the entry. If the entry was not discarded by the filter described in the previous section, the timestamp wass converted to the corresponding fifteen minute interval number (each interval is 15 minute long, there are 96 such intervals in 24 hours). An interval was labeled as '1' for occupied and '0 for unoccupied. If a Bluetooth tag was present in a home at an interval , then that home was labelled as occupied at that interval. This was then used to detect `Away` events as described in Chapter 3. An interval is labelled as `Away` if the home is unoccupied in the intervals before and after it.

4.1.6 Sleep Detection

Sleep detection is defined as inferring periods in a home when occupants are asleep. This information can be used to detect changes in lifestyle and can also be useful for self-programming thermostats that can set a setback when the homes occupants are asleep. ThermoCoach uses this information to generate setpoint schedules in which the temperature is set back to a *sleep* temperature.

The first step to detect sleep time is to determine if the occupants of the home are asleep and inactive or if the home is unoccupied. In the second step, the system proceeds to detect an `asleep` state for a night only if the home is occupied during that night.

Similar to Bluetooth data, ZWave data is processed to populate a 96-element array representing if the home is active at a particular fifteen-minute interval or not. The interval corresponding to the timestamp of a motion sensor firing is set to '1'. All others are '0' by default, indicating no activity. ThermoCoach assumes that all sleep events occur between 9PM to 1PM. While this assumption has limitations, use of better sensors could provide better data and will only make the results better. Thus these results for sleep detection could be considered as a lower bound for sleep detection accuracy.

Due to significant false negative detection rates of motion sensors, on some days, only one of sleep events and wake events were detected. Thus the total number of sleep events detected may not be the same as the as the number of wake events detected.

Detecting the time when people go to bed is challenging, especially in multi-occupant homes. Presence of pets further complicates the problem. To avoid false postives, motion sensors were not installed inside bedrooms where they can see a large part of the inside of the bedroom. This however means that sleep time is denoted as the time when people enter their bedrooms for the night and not actual sleep time. Thus, in cases where people have a study table or office inside their bedroom or if they watch television at night before bed, the beginning of that period is labeled as asleep. With PIR motion sensors alone, it is not possible to distinguish between sensor events triggered by humans versus pets.

Figure 4.6 show's Bluetooth data, ZWave data and `Asleep` and `Active` states. The top two lines represent ZWave data and the other's represent BLE data. The pink shaded regions indicates when homes are occupied and the purple shaded regions indicate `Asleep` states of a home.

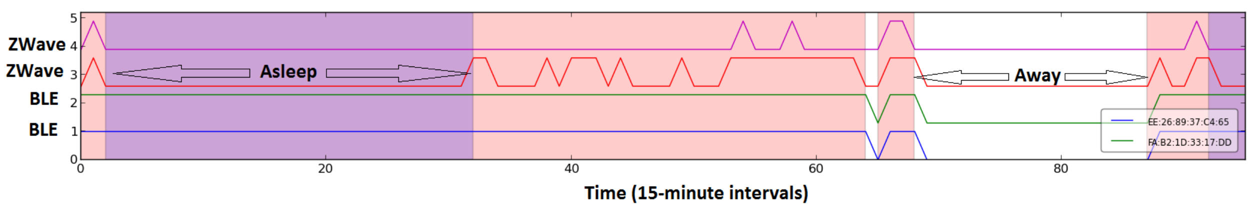

Figure 4.6: Raw Data and State Inference

4.2 Hardware Platform

Occupancy sensors described in section 4.1 were used to obtain occupancy data. To collect data from these sensors in homes, a Raspberry PI running a smarthome platform Piloteur [35]was used.

4.2.1 ZWave Data Collection

To collect ZWave data from Motion sensors, a ZWave controller is needed to communicate with the sensors. Aeon Lab S-2 Z-Stick connected to a Raspberry PI running a custom Open-ZWave [39] *driver* was used to communicate with the sensors. The *driver* ran continuously waiting for messages from the motion sensors. When a motion sensor detects motion, an *On* state is logged along with system's timestamp.

RF signal strength decreases as it passes through dense objects such as ceramic tiles, concrete, granite or other hard stone and large metal objects. Too many wireless devices may saturate the environment and it's advised to place these sensors at least 5 feet away from devices with radios such as wireless controls and sensors, security systems, cameras, cell phones, stereo receivers, TV's, baby monitors, cable boxes, game systems, microwave ovens, etc. Thus location of the sensors played an important role in reliable data collection.

4.2.2 Bluetooth 4.0 Data Collection

IOGear Bluetooth 4.0 USB adapters connected to Raspberry PI's were used to detect BLE tags. The driver/interface used Linux's *hcitool*'s *lescan* feature to scan for Bluetooth Low Energy devices and 'hcidump' was used to get Received Signal Strength Indication(RSSI) values, for tags within the range of the adapter. Timestamp, MAC address and RSSI signal values were recorded every second. Thus when an occupant entered their home with tags attached to their key chains, information about the tag was logged. The study assumed that occupants carried their tags with them whenever they left their home. Participants were reminded to carry their keys with them.

4.2.3 Raspberry PI

To collect data from the sensors in a home, Raspberry PI model B boards were used. Each PI ran Debian optimized for Raspberry PI. Additionally, Piloteur was installed on each PI. A PI running Piloteur is called a *endpoint* [35]. The platform is responsible for running and monitoring scripts that collect data from sensors. It also has various monitoring and fault analysis scripts. These endpoints connect to WiFi and sync data to a backend server. Raspberry PI boards are small and powerful and are noiseless, making them convenient for deploying in homes for data collection. The model B uses between 700-1000mA depending on what peripherals are connected to the PI and the maximum power the Raspberry PI can use is 1 Amp. A micro USB cable connected to a 5V USB power adapter from Enercell was used to power the Pi's. endpoints were enclosed in plastic project boxes. The boxes had vents to prevent the PI's from overheating. All USB attachments were secured with tape to prevent them from being accidently pulled out.

4.2.4 The Nest Thermostat

Each home was installed with a Nest 2nd Generation Learning Thermostat. The Nest thermostat is a state-of-the-art *Learning* thermostat that learns from changes to temperature settings made by the user. It then generates schedules for the home. It connects over WiFi and allows users to control their thermostat remotely through the web or their mobile app. It tries to learn the time taken by the HVAC system to heat/cool.

Figure 4.7: An endpoint

Each Nest used in the study was set up with a custom email account that researchers had access to. Each Nest had a unique home number as an identifier and a unique password. A script was written to log thermostat data through an un-official REST API since Nests official API had not been released at the time of hardware design. The current setpoint schedule, target temperature values, current conditions of the HVAC and states of the advanced features of the Nest were logged along with timestamps. The script continouously logged Nest data. Ten accounts were processed at a time to avoid being rate limited by Nest's servers.

The Nest Learning thermostat was installled in homes by the installer, with ease. In some homes, the thermostat had trouble maintaining WiFi connectivity. The main reason for this is was that the thermostat was located further away from the router in those homes. The Nest thermostat currently does not work well with routers set up as access points, hence use of access points is not recommended. Two of the homes in the study found ice in their air filters a few weeks into the study causing the homes to not cool. It was not clear if the Nest thermostats were responsible but nonetheless one particpant dropped out of the study due to this.

4.3 Software Platform: Piloteur

Homes are hazardous environments to sensors. There are several causes of hardware failure: poor wireless connectivity, power outages and crashed software. Homes are remote environments and researches do not have regular access to them. Visiting a participant's home entails scheduling and planning and results in several days and often even weeks of data loss. Thus, a reliable, easy-to-use platform for collecting sensor data is needed. Piloteur takes into consideration the guidelines presented in The Hitchhiker's guide to successful residential sensor deployments [40] and includes additional features that address issues that were brought to light by this study. The Hitchhiker's guide [40] presented several issues typically faced during a typical large scale deployment in homes and guidelines to overcome those issues along with general advice on the architectural design of sensing systems.

Plitoeur [35] is an open-source platform that was designed and implemented to satisfy the requirements of this study and is also available to other researchers for modification and use. The platform was responsible for collecting and managing sensor data. Piloteur is a user-space software that is installed on top of a machine's base operating system. Piloteur currently supports two platforms: Amazon Web Services (AWS) EC2 instances and Raspberry PI B embedded computers. It must be installed on an endpoint machine, that will be installed in a home, to physically interface with the sensors and controllers and a server machine that will backup, manage, and monitor the endpoints.

Piloteur was installed on all the PI's deployed in the homes. The platform's monitoring service monitored the installed sensors and alerts were generated when sensors failed. The sections below list the important issues to be considered in deployments and the features of Piloteur that help mitigate, if not eliminate, a number of these issues.

4.3.1 A Simple Architecture for large scale and long term deployments

Large scale pilots may involve dozens or hundreds of houses and devices. Management of these systems becomes difficult. Management is needed to ensure that all sensors are operating as expected. During a long term study, software may require updates, licenses may need to be updated or sometimes a major change in a software component is needed. Sensing systems require continuous maintenance and a platform should be designed to deal with these issues. Using different platforms in a single deployment increases the complexity of the system and adds to possible failure points. Piloteur currently runs on Linux and can be used to interface with a number of different sensors, depending on the hardware on which it runs. The need for different software platforms is eliminated since Piloteur can run on any hardware running a Linux distribution, from Laptops and Tablets to Raspberry PI's.

4.3.2 Simplified Setup

Before performing a deployment, the developer needs to create two directories that are accessible online: one to hold software *driver*s used during the deployment, and another to hold configuration files for all *endpoints*. In this study, repositories on GitHub were used. Piloteur provides a simple interface on the endpoint in the form of three "magic directories": a software directory, a data directory, and a logs directory. Any executable that is put into the software directory is guaranteed to be executed upon startup, monitored, and restarted if it crashes.

Piloteur's configuration service is used to set up and maintain the Piloteur environment on an endpoint. It ensures that all software dependencies are installed and that data and code is downloaded and installed. Before starting a new deployment, the user needs to create two directories that can be accessed via the Internet: one to store the scripts for hardware *drivers* used during the deployment and another to store configuration files for all the endpoints in the deployment. Configuration parameters are specified as key-value pairs in JSON format. Top level directory configuration applies to all endpoints in the deployment. New *classes* of endpoints are created with subdirectories with the class name. Subdirectories can have directories within them, running a number of levels deep, defining a hierarchy of *endpoint* classes. Configuration for a specific *endpoint* can be defined by creating additional sub-directories with the node's unique ID and in the case of conflict of parameter names, values from subclasses override values from super-classes. These hierarchical *endpoint* classes allow the use of a single configuration file for parameters that are shared among an entire class of endpoints. In this study, all of the endpoints had a top level configuration file, specifying details about the backup server; all the ZWave endpoints were grouped under a ZWave class and Bluetooth endpoints under a BLE class. Each *endpoint* had its own configuration specifying details such as home identifier, WiFi credentials and so on.

To create a new endpoint, Piloteur's configuration needs to be executed from a host machine such as a laptop. The user points it towards the new *endpoint* machine, and either gives a unique identifier(ID) for the new *endpoint* or a unique ID is generated automatically by a tool called *uuidgen* which creates a random id. An Ubuntu 12.4 virtual machine was used as the host machine in this study. Each *endpoint* was given a unique id based on the home number, *endpoint* number and class of endpoint(ZWave,BLE,etc). The configuration service downloads the files associated with that ID and configures the new Piloteur *endpoint* accordingly, installing Piloteur services, software dependencies and all hardware *drivers* in a *drivers* directory (in this case the Bluetooth *driver*, ZWave *driver* and Nest *driver*) on the *endpoint* automatically. The configuration service itself is installed.

The configuration service runs itself periodically on an endpoint, to keep the system up-to-date. It periodically updates the hardware *drivers*, configuration files, and system files. If one or more *drivers* have been updated, those *drivers* are gracefully terminated and then restarted. If the change is to any other service, all active *drivers* are gracefully terminated. The configuration service rolls the system back if it thinks there's a problem with the new configuration, but it can detect only a limited set of problems. Authentication information and data location is specified through configuration files. Piloteur allows researchers to add their own scripts to monitor *drivers*. The tests are executed by a monitoring server for each *endpoint* that has the corresponding *driver* enabled according to its configuration.

Thus Piloteur's configuration design makes it easier to set up large scale deployments and manage them by allowing a hierarchy of configuration file classes.

4.3.3 System Monitoring

A watchdog service runs entirely on the *endpoint* to ensure that the hardware *drivers*, *endpoint* hardware and Piloteur services are always running. For each *driver* loaded on an endpoint, the watchdog validates that the *driver* is located in the software directory and ensures that the *driver* is running and starts it if it isn't. If the *driver* process crashes or fails , the watchdog will restart it. It checks to see if all the *drivers* that need to be running on a *endpoint* are operational, every minute. The watchdog also periodically checks the network connection and tries to resolve the issue. The watchdog checks the network connection by opening a socket to the server. In the case of interface failure, the watchdog resets the network interfaces. Additionally, the watchdog periodically generates operational logs of the *endpoint* platform, including CPU utilization, RAM utilization, running processes, status of hardware peripherals, core temperature, and kernel failure or hardware reboots, cpu utilization, disk usage, RAM usage, etc.

4.3.4 Remote Access

One of the major issues in deployments is that homes are remote environments and researches do not have regular access to them. While certain failures such as a broken sensor or battery failure require human intervention, in many cases bugs could be fixed if researchers have access to the remote endpoints.

Reverse Tunnel

Each *endpoint* is connected to WiFi in order to sync data to backend servers. Additionally, for each *endpoint* a reverse ssh tunnel was set up, allowing researchers to log in to the *endpoint* remotely even though it sat behind the home's firewalls. This feature allows researchers to debug the actual *endpoint* without having to retrieve the *endpoint* from the home. It significantly reduces an endpoint's down time.

Automatic code updates and continuous deployments

For long term and large deployments, the ability to update code remotely is a very useful feature. Bugs can be fixed and new features can be added without having to stop data collection during an ongoing deployment. The platform uses Ansible, an automation tool, to download the newest code pushed to our online repository. The endpoints can also be rolled back to previous code. *Endpoints* check for new updates periodically.

This feature was found to be useful during this study, especially when new code had some bugs and was then rolled back to a previous stable version in order to analyze the cause of the bug.

Ease of interfacing new sensors

The platform allows researchers to interface new *drivers* easily. A *driver* is a script that interfaces with sensors and their adapters(wherever applicable). The platform redirects output of *stdout* to log files and manages the rotation and syncing of the files making it possible to add new *drivers* in short period of time. While the new sensors may have to be remotely deployed, new *drivers* can be added to the *endpoint* remotely through the automatic update feature.

Time Syncing

For time series data, it is essential that the timestamps on all the sensors and data receivers are synced. In this study, Raspberry PI's running Piloteur were used and the PI's do not have an internal clock and use Network Time Protocol (NTP) to set the time. Instead of using the Net work Time Protocol (NTP) which creates unrecorded shifts in system time, the watchdog periodically logs the local system clock simultaneously with the remote server's system clock for post-facto time synchronization to be performed.

Monitoring and Altering

The watchdog service attempts to diagnose and recover from problems that require autonomous corrective actions on an endpoint. Piloteur's monitoring service attempts to recognize failures that may require manual intervention and, when detected, it emails an alert message to the operator. The monitoring service infers *endpoint* state from the data and log files using a predefined ruleset. For example, if no logs are synced for a day, it can be assumed that the *endpoint* is down. If all endpoints in a home are down, network issues could be assumed. The monitoring service runs on a separate monitoring server and processes the most recent log files and returns the endpoint's status. In addition to the monitoring service, Piloteur provides a RESTful Web interface that details the software versions and operating status of the endpoints. This was useful during deployments to assess success/failure of installs. The overall status of a *endpoint* is reported as a color. The following states were reported:

- Red: needs manual repair; no network connection.
- Yellow: needs remote repair; network connection is up and all hardware peripherals are working, but there is a software error.
- Green: does not need repair; end-to-end data delivery is operational.

When the *endpoint* is being deployed by a non-expert, the color status can help quickly decide whether an *endpoint* should be deployed or returned to the lab.

4.4 Deployment

190+ endpoints, 40 Nest thermostats, over 250 motion sensors and 135+ Bluetooth low energy(BLE) tags were deployed across thirty nine homes. A third party installer was hired to do the installation in participant's homes. A commercial online tool called ScheduleOnce was used to schedule installations. Participants were asked to choose suitable times and the installer would coordinate with them. For each home a kit was created containing necessary equipment needed for deploying the sensors in homes. Each kit consisted of-

1. 4 endpoints (Rapberry PI Boards running the Smarthome platform, Piloteur)
2. 6 ZWave motion sensors
3. A minimum of 2 BLE tags and key fobs
4. Router
5. Nest Thermostat
6. Wire clips, 3M Command strips, power extension cords and expansion plugs
7. Deployment Information Sheet
8. Smartphone

4.4.1 Deployment of endpoints

The ZWave *endpoint* was installed in the heart of the home to ensure that it was within range to most of the motion sensors. One BLE *endpoint* was installed near each external door, to detect occupants when they entered their homes and one *endpoint* was placed in the room where occupants typically kept their keys. The location of this *endpoint* varied from home to home.

4.4.2 Motion Sensors Deployment

To keep cost low and also to keep the ZWave network at a manageable size, the aim was to use as few sensors as possible to determine the state of the home effectively. In majority of the homes no motion sensors were installed in bedrooms. If sensors were needed to be installed in bedrooms they were installed in a way that they looked at the door entrance, away from the interior of the room. A home can be thought to have two zones- an Active Zone and a Transition Zone. The Transition Zone may be Hallway leading to bedroom, a common wall outside all bedroom doors, or stairway leading to upstairs bedrooms. ThermoCoach assumes that whenever a person enters the transition zone in order to go to the bedroom(s), a sensor in the transition zone can detect them. The Active Zone is commonly used areas when participants are not sleeping, e.g. living room, kitchen, dining room, study etc. Sensors were placed on a wall so that it is facing the most active portion of the room. (Eg. Wall opposite couch, opposite the range or sink in kitchen, etc) Sensors were installed in the *Active* zone and one sensor in the *Transition* zone of the home. The sensor in the *Transition* zone captured people walking in and out of bedrooms, defined as the *Sleep* zone. The ideal location of a sensor in a room is such that it can always capture activity in the room within an interval of 2 hours or less. Details on where these sensors were placed and the reasoning for doing so are in the section on sensor deployment in a home(on the next page).

The sensors were placed at least five feet above the floor, but not closer than two feet from the ceiling, for maximum coverage. Placing a sensor close to the ground or ceiling increases the likelihood of interference from furniture, support structures and other objects. To avoid wireless competition, the installer was instructed to place the sensor about 5ft feet from electronic devices with RF radios such as microwave ovens, televisions, etc, whenever possible to do so. Sample floor plans with sensors mapped on them, were provided to the installer to explain expected sensor locations.

Sensor Deployment Guidelines

Motion sensors should be installed to cover the Active and Transition Zones

1. *Transition* Zone Sensors (One or two sensors)

2. *Active* Zone Sensors (4 in total)

The *Active* zone sensor should be placed on the wall approximately one foot higher than the tallest occupant (About five-six feet on average but not closer than two feet from the ceiling.) Face the sensors away from transition and sleep zones if possible.

Transition Zone Sensor/Hallway Sensor should be placed on a wall facing the bedroom and facing away from the Active Zone if possible. Sensor should face bedroom doors if possible or close to doors. If there is no wall opposite the bedroom door, the *Transition* Zone sensor can also be placed inside the bedroom facing the doorway, but facing away from the bed. The sensor can also be placed on the wall above the door frame, in line with the door.

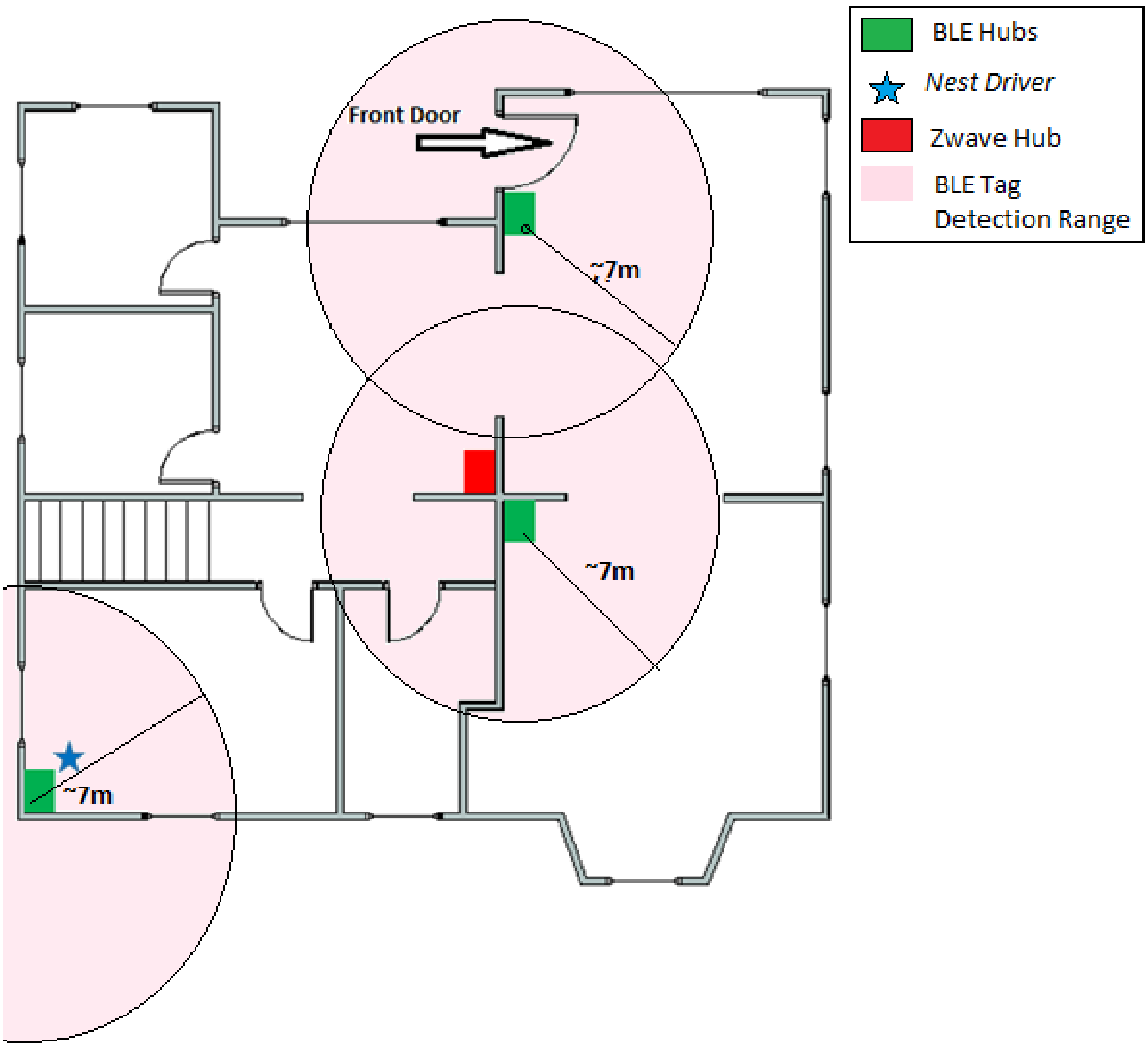

Figure 4.8: Example floor plan and sample *endpoint* placement. The * indicates a BLE *endpoint* that also collected Nest thermostat data for the home.

The Transition Zone sensor should not cover any of the *Sleep* Zone(Bedrooms). The area of the *Active* Zone covered by the transition sensors should be minimal; no overlap is ideal. To ensure that this sensor does not see any of the *Active* Zones of the house, a blinder could be added to the sensor to restrict its vision to the left or the right. A blinder may be needed only in some homes and will depend on the home's layout. In some homes, it may be necessary to use more than one *Transition* sensor to cover sleep areas. *Active* Zone sensors must not cover any of the Bedrooms and may cover the *Transition* Zone if necessary.

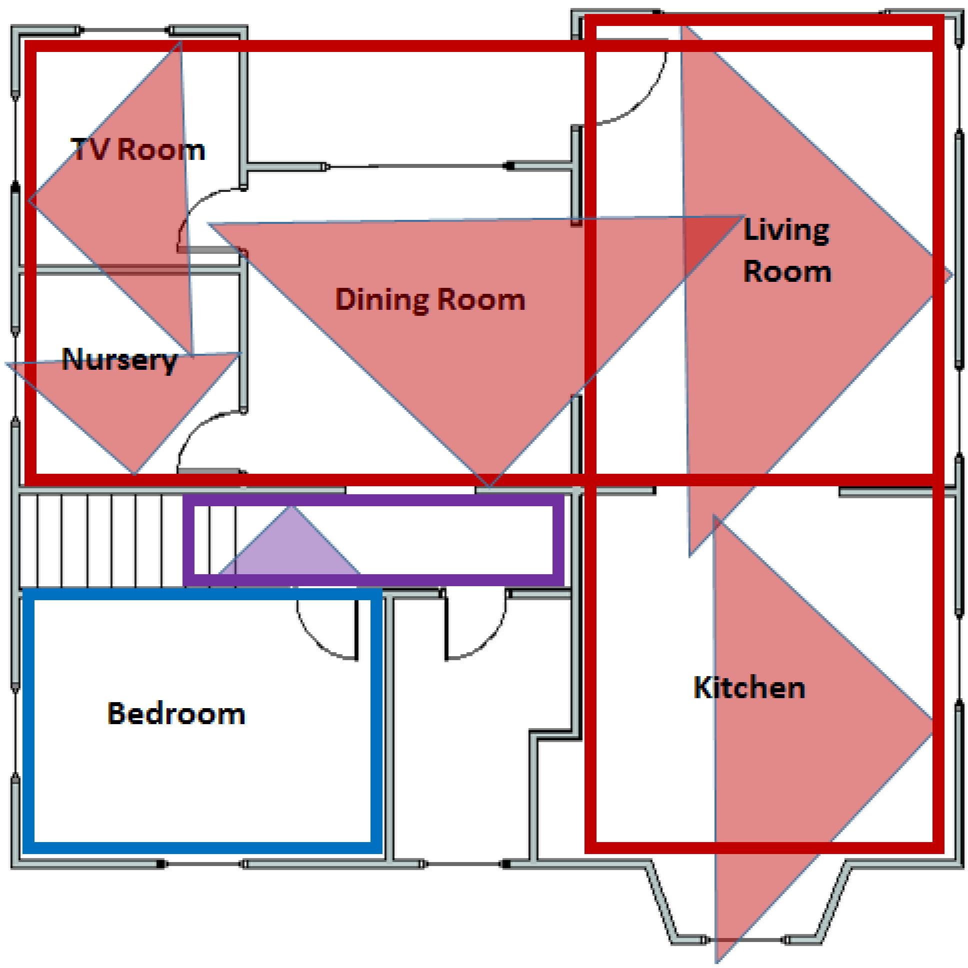

Figure 4.9: A sample floor plan indicating Motion Sensor locations and ranges. Red regions are considered a homes *Active* region. As shown, most active regions are covered by at least 1 motion sensor. The purple region shows a Transition region. A transition region is typically a hallway, separating the living space from the bedrooms.

4.4.3 BLE Tags

BLE tags in advertising mode were attached to key Fobs and given to participants to be attached to their key chains. Participants were requested to carry it with them whenever they left home.

4.4.4 Nest Thermostat

Each kit also contained a 2nd Generation Nest Learning Thermostat. Each Nest was configured in the lab before deployment to connect to the router provided as part of the study. Nests data was retrieved and logged and the account information was also provided to the participant. A central *endpoint* running Piloteur collected data from the thermostats. Each participant was given access to the Nest thermostat's online interface and mobile app.

Figure 4.10: A sample *endpoint* install

4.4.5 Deployment Tools

Each kit contained a router that was configured to work with a home's existing router. This allowed for endpoints to be configured to connect to WiFi before the actual deployment and eliminated the need to obtain the participant's WiFi network credentials.

For aesthetic purposes and to prevent the power adapters of the endpoints being unplugged, wire clips, extension cords and expansion plugs were used to attach the endpoints to walls. Based on an outlet's usage in a home, furniture around the outlet and location of the outlet, the installer used a combination of solutions provided to mitigate the chances of the adapter being disconnected. To avoid leaving marks on the walls, 3M Command strips were used to install all the sensors on the walls of a home.

A *deployment sheet* was used as a checklist by the installer and to note down any issues faced during deployment. The installer was instructed to note down the home's floor plan and location of installed sensors for researchers to verify that sensors were installed in desired locations. An example can be found in the Appendix.

The installer was also given a smartphone that was used to connect to Piloteur's online tool that provides the status of an endpoint, given an endpoint's identifier or a similar search string. This allowed the installer to verify quickly if data collection had started and was working, before he left the home. The installer was also instructed to note down model numbers of the home's HVAC system and its power usage in various modes.

4.5 Maintenance

Continuous maintenance of sensors was performed through the study. Right after deployment, endpoints had a 17% failure rate and the rate of failure remained anywhere between 7-17% through the three months of the study. The cause of failure is uncertain and could be due to poor WiFi connectivity, disconnection of a *endpoint* from power or a software or hardware failure. Some endpoints were offline because they were located in parts of the home with weak WiFi connectivity especially since other endpoints running the same software stack in the home were working as expected. The locations of endpoints in a home were constrained by the requirements of the study. As a result, some endpoints were offline quite often and synced data intermittently whenever they came back online. This issue was detected at the time of initial deployment and to address this issue the deployment tool was modified to display the WiFi signal strength at the location of deployed endpoint. This helped the installer make better decisions on location of installation of endpoints in later deployments and also helped determine cause of *endpoint* failure.

For endpoints that were offline, visits were scheduled to switch out the failing endpoints. WiFi signal strength was a consideration when re-deploying replacement endpoints. 95% of the endpoints retrieved worked on reboot. This indicated that the PIs needed to be rebooted periodically. In about two or three cases one or more endpoints were unplugged in a home. Participants informed the installer of this and the endpoints were connected back to power. This often happened in homes with pets.

ZWave data was not consistent in a number of homes in the study. On many days, the controller and sensors failed to log any motion. There could be several possible reasons- A malfunctioning ZWave adapter, faulty sensors, or even the type of home(ZWave signal attenuates through concrete, etc.) and location of the sensors from the ZWave adapter. Wireless signals are attenuated by concrete, cement, thick walls and suffer from interference from large electronic devices such as televisions, microwaves and so on. In most cases, the ZWave network repaired itself in a couple of days. In nine homes, ZWave sensors and the Aeon Lab ZStick were replaced. In these cases, the endpoints were online, but the ZWave network failed to log activity detected by the motion sensors. When restarting the *driver* failed, the motion sensors and ZWave *endpoint* were replaced since they were hard to debug remotely. Throughout the study period only 1/3rd of the homes had consistent ZWave data.

In total, about twenty six visits were made to fifteen different homes for switching out motion sensors and endpoints. 1/3rd of the endpoints were online again once they were rebooted. In at least three visits, some endpoints were disconnected from power when the installer visited the home for a repair. In five endpoints, sections of the SD card were found to be corrupted and had to be replaced.

Seven participants lost their key fob sensors. After one month, participants reported that their key fobs had fallen off. New key fob sensors were mailed to participants. In one case, the BLE tag began pinging loudly, probably due to a dying battery. The participant was informed to discard the sensor and replacements were mailed to them.

4.6 Hardware Removal

At the end of the study, participants were asked if they would like to keep the sensors and the Nest thermostat. If they wished to return the hardware, they were given the option of either setting up an appointment for the installer to visit their home and retrieve the sensors or alternatively, they were mailed pre-paid shipping materials and were asked to take down the sensors themselves.

4.7 Analysis of Piloteur

Right after deployment endpoints had 17% failure rate, i.e. 83% of the endpoints collected data on the server. As mentioned earlier, dailure rate remained between 7% to 17% per week. Analysis of the operation logs indicate that, all of the 156 endpoints deployed experienced potentially fatal failures. All thirty eight Z-Wave endpoints had at least one hardware *driver* failure, if not more than one. In total, the Z-Wave *driver* was restarted 15,206 times across thirty eight end-points. About twenty six endpoints experienced problems with the Ansible system that is used to auto-update the endpoint.

30 endpoints had configuration errors. Thirteen endpoints had power failures. Twenty four endpoints had hardware failures. On thirty eight endpoints the hardware *drivers* were fixed by remotely updating the code, where the Open ZWave code was updated to another version. In thirty cases, a node was repaired or revived by remotely updating the configuration files which contained account information for Nest *drivers* that was no longer valid. Forty two endpoints were manually revived through the reverse SSH tunnel. Fifty two endpoints with hardware or critical software failures were fixed with a total of twenty six maintenance visits across 15 homes.

Maintenance visits were tedious to schedule and were usually scheduled several days in advance by a combination of email and phone calls with the study participants. Physical maintenance visits to all failed endpoints would not have been feasible.

All 156 endpoints had network failures at some point and the sync service actually failed to connect to the server over 21,000 times across all endpoints. Most endpoints simply had network drops due to intermittent network connectivity and regained the connected shortly after. Throughout the study, endpoints lost connection with the server when the house's network connection was dropped, but regained connectivity.

Forty endpoints had poor WiFi signal strength at the time of deployment and eventually lost the network connection permanently. Ten of them were moved to locations with better signal strength. Buffering data locally on the *endpoint* was extremely useful since network failures were common and in some cases rsync failed to sync data to the server. Even when a manual repair of the *endpoint* was required, data was synced manually through the SSH tunnel before any repairs were attempted, to minimize the loss of data.

All endpoints running the BLE *drivers* also had hardware *driver* failures and the BLE *driver* was restarted 33,268 times over all endpoints. However, this failure rate is in part due to the nature of the *driver* that required the hardware to be periodically reset. It was assumped that Piloteur's watchdog would always restart the *driver*.

Due to sensor failures, *endpoint* failures, network failures etc., days of training data were removed by the data pre-processing filter. Days with partial data, days with detection of too few events, days when occupants were away on vacation, were removed. Figure 4.11 shows the number of actual training days used to generate schedule recommendations for homes.

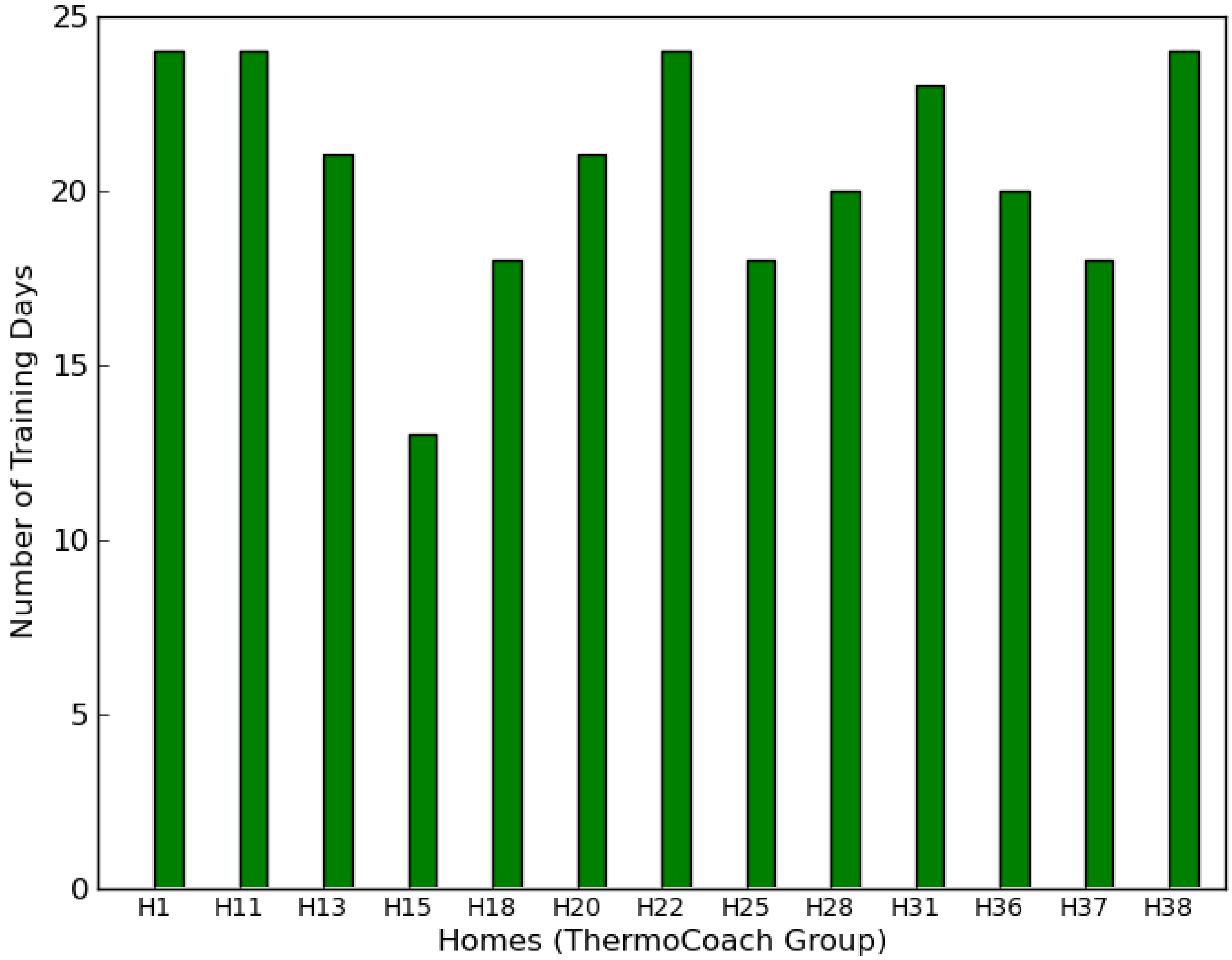

Figure 4.11: Number of training days used for homes in ThermoCoach group are shown here. These days represent days with ‘good’data as described in section on data pre-processing

Chapter 5

Evaluation

To evaluate ThermoCoach, a three month long study was conducted and for the evaluation, it is essential to compare ThermoCoach's performance with the current state-of-the-art thermostats. ThermoCoach is compared against manually programmable thermostats with energy feedback and the Nest Learning Thermostat. Thirty nine volunteer homes were recruited. Participants that were recruited had homes in neighboring areas with similar weather conditions. All participating homes were single family homes and most participants worked outside the home for at least 8 hours during the day. Participants were randomly divided into three groups. Group1 represented users of manually programmable thermostats, Group 2 represented Nest thermostat users and Group 3 represented users of ThermoCoach. After 6 weeks, an *intervention* was performed. At the time of *intervention* homes in Group 3 were given ThermoCoach recommendations. In addition, all homes in the study were given energy feedback in the form of Nest's monthly energy report. Each home was emailed with the previous months monthly report. The report contained information on the total number of hours the air conditioning was on in a home during the previous month, general energy savings tips and promotional material from Nest. The design of the study, response to recommendations and their effectiveness are described in this chapter.

5.1 Study Design Overview

A Nest thermostat was installed in each home and its settings were modified based on the group the homes belonged to. Thus the same hardware and HVAC control algorithms were used in all the homes. Features of the Nest were enabled or disabled to emulate the type of thermostat represented by the group. Auto-Schedule(schedule learning feature) and Auto-Away(feature that learns when the home is typically unoccupied) were turned off for Groups 1 and 3. All other features were left on. Participants were provided with information about Nest's features and a list of features they could use during the study. During the study, any overrides made to settings were logged. Occupancy sensors were installed in homes by the installer and key fob sensors were given to participants. On the day of installation, participants were provided account credentials for their thermostat and instructions on how to use the thermostat. When the system was initially installed, the thermostat was not programmed with a setpoint schedule.

Feature	**Programmable**	**Nest Learning Thermostat**	**ThermoCoach**
Nest's Auto Schedule (Schedule Learning)	Off	On	Off*
Early on (Pre-heat feature)	On	On	On
Auto Away	Off	On	Off
Flexible Scheduling	On	On	On
Usage History	On	On	On
TimetoTemp	On	On	On
Mobile + web interface	available	available	available
Eco-feedback	available	available	available
Schedule Recommendations	un-available	available	available

Table 5.1: Nest Settings Across Groups. *ThermoCoach recommendations provided. Explanation of features can be found in section 4.2.4

Table 5.1 describes the Nest thermostat settings for each group. The thermostats were programmed with these settings during time of install. Participants were also emailed instructions containing a summary of features they could and couldn't use. Group 1 and 3 were asked to use the Nest as a manual programmable thermostat. No suggestions were made to participants regarding scheduling. They were asked to keep Nest's Auto-Away(feature that learns when the home is typically unoccupied) and Auto Schedule(schedule learning feature) features turned off. They were given access to Nest's web and mobile interface, allowing them to access their thermostat remotely. On similar lines, Group 2 homes were sent instructions and told to use all the features of the Nest. 'Auto Schedule' was turned on for Group 2 homes during installation of the Nest thermostat.

At the time of recruiting all participants were informed that they may receive recommendations on energy savings but were not informed that the recommendations could be setpoint schedules. Occupancy data was collected from all the homes even though recommended schedules were sent only to participants in Group 3. All homes had the same hardware installed and were asked to use their Nest as a manual programmable thermostat. The exception to this is Group 2 homes in which Nest automatically set schedules for the thermostats. Hence in the training period, homes in Group 1 and Group 3 behaved similarly.

Most homes were had the common features described in Table 5.2. Two weeks after all homes were instrumented, entry interviews were conducted to understand people's interaction with their thermostat, energy needs, weekly schedules, etc. Participants were reminded via email to carry their key fob sensors with them whenever they left home. Besides this, no other feedback was provided and participants were told to interact naturally with their thermostat.

Groups 1 and 2 were the control groups. Group 3 formed the treatment group and received three schedule recommendations: an energy efficient one, a comfort based one and a balanced one.

After a six week long training period, the *intervention* was performed and all three groups were emailed their Nest energy reports that outline the number of hours the home was cooled/heated. It also provides a "Leaf" rating. A home earns a "Leaf" if energy usage is not excessive. Occasionally, a tip for the home's schedule is given along with promotional information. This kind of energy feedback has been found useful in making the participants aware of their energy usage. Participants in Group 3 were additionally emailed three schedule recommendations in addition to the Nest energy report. Participants were asked to make a selection within forty eight hours of receiving the email. All participants received the email on the same day. Participants were allowed to change the thermostat or schedule at any point during the study. Once a selection was made by the participant, their thermostats were programmed manually by the researchers based on their selections within twenty four hours. At the time of writing, an automatic mechanism for programming the Nest thermostat was not available.

Features	**Type**
Type of home	Independently owned
Type of household	Single family
HVAC equipment	Single stage heat pump for heating and cooling
Pets	Yes/No
Children(Below 4 years)	Yes/No
Number of hours a home is typically unoccupied	8-10 hours per day
Number of occupants per home	2-6

Table 5.2: Common features across homes

5.2 Intervention

Out of the thirteen homes in Group 3, twelve participants had responded to the email within forty-eight hours. Eight homes out of the twelve selected schedules other than their current schedule. Out of those eight, four selected ThermoCoach recommendations as is, while the rest made modifications to the recommended schedules. One of the four homes that modified a schedule changed the setback temperature from 87 to 81 degrees. Two homes made minor changes to the times of the setpoint and the third changed both the times and temperatures to a highly comfort-oriented schedule with an `Away` setback that differed only by two degrees from the temperature set when the home is occupied. Out of the homes that kept their current schedule, only one home did not have a schedule before *intervention*. At time of *intervention*, three homes changed the time of the setpoints in the chosen recommended schedule. One home moved the end of the `Asleep` period by one hour for weekdays. One home moved the end of the `Asleep` period by one hour for weekends. One home changed the period when the home was unoccupied by a total of one hour. Since there is no ground-truth for occupancy data, the accuracy of state detection and the schedules can be estimated as the number of changes(overrides) made to the schedule.

One participant had seen the email but not made a selection. One participant did not respond. The response to the emails was high. Eco-feedback seems promising though the response rate may be biased since the participants were conscious of the fact that they were participating in a study.

The temperatures for the different states of the home were set based on participant's preferred temperature settings that they made before the *intervention*. Setbacks were determined based on these values. At time of *intervention*, one participant reduced the setback temperature when the home was unoccupied. All others made no changes to the temperature values in the schedule. Four participants modified the time of a setpoint in their respective schedules. Two participating homes kept their current schedules. One of the homes had a good setpoint schedule before the *intervention* period. Three homes choose HighComfort(or some variation of it); the schedule that maximized comfort and had least energy savings. Energy Saver was chosen by three homes, with slightly more Miss Time and higher energy savings. Two homes chose Super Energy Saver.

Figure 5.1 shows the schedules for Group 3 (ThermoCoach Group). Three recommended schedules along with pre- and post-*intervention* schedules of homes, are shown. The schedules are plotted against energy cost on the x-axis and Miss Time on the y-axis. Schedules with energy cost as "1" are energy expensive schedules. As the Miss Time increases, comfort decreases. The energy cost of a home's pre-*intervention* schedules are plotted. For home's that did not have a schedule, miss time is considered zero. The assumption is since only one setpoint is used, the occupants are always comfortable. Miss Time is considered as in indication of comfort; as Miss Time increases , comfort levels decrease.

The graph shows that Super Energy Saver schedules are the most energy efficient of the three recommendations and energy efficiency decreases as Miss Time decreases. In some of the homes with a schedule before *intervention*, there is significant Miss Time. This indicates that the schedules were not accurately programmed and a reason for this could be that occupants may not have known how to accurately do so. Even though about seven of homes either kept their own schedules or accepted comfort oriented recommendations, majority of them had a 4 to 8 degree setback when the home was unoccupied.

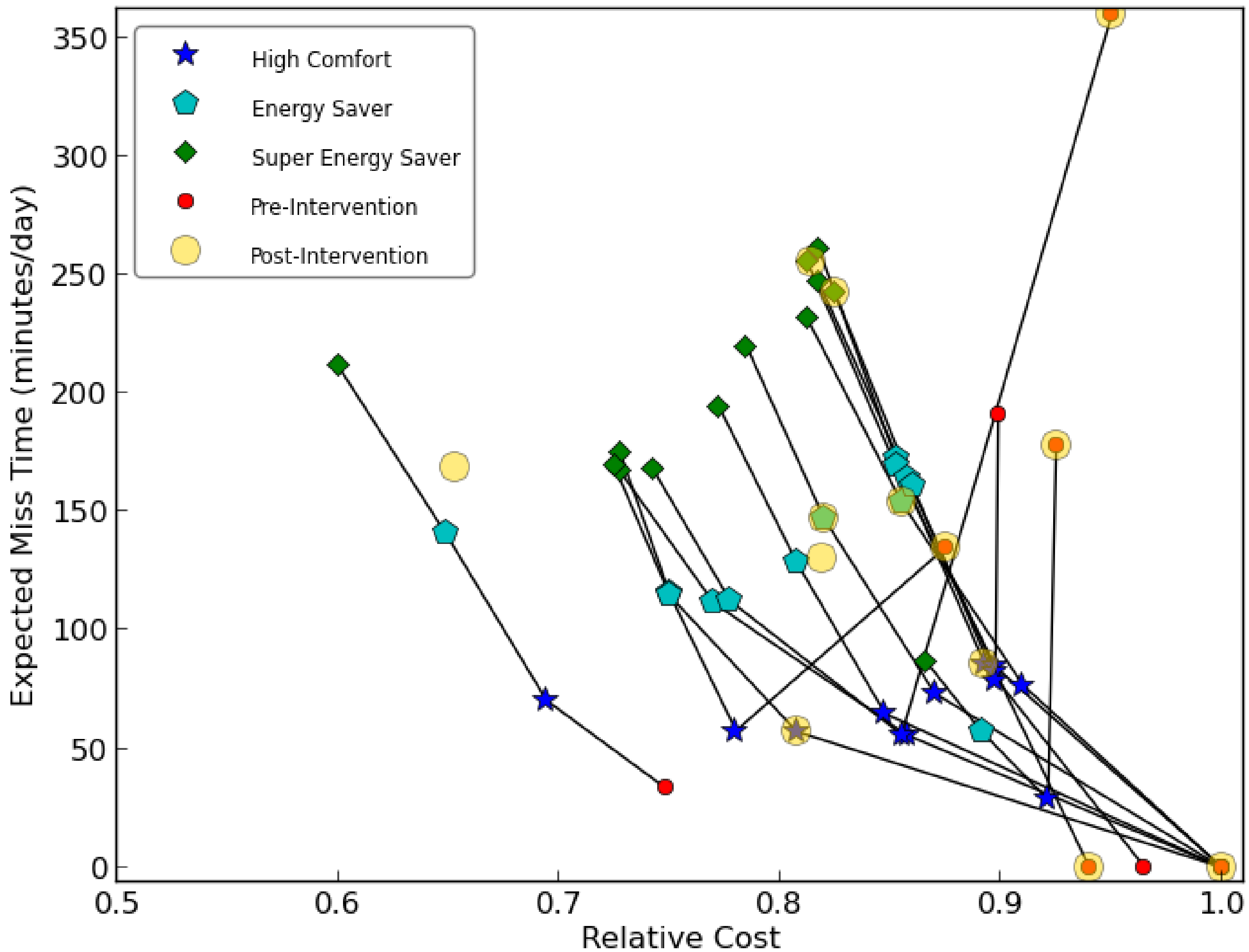

Figure 5.1: Schedule Recommendations for Group3. *Relative Cost* values are estimated from Equation 3.10

5.3 Panel Data Regression Analysis

Impact analysis was performed to analyze the effectiveness of the treatment applied in the study. A randomly controlled trial approach was used to randomly allocate participants to treatment and control groups. With panel studies one can model for changes in weather and time-invariant unobservable factors that may impact energy usage and costs. A panel dataset measures changes in one or more characteristics in the same set of individual samples over time. Weather effects and other fixed effects can be modeled using regression analysis of the panel data.

5.3.1 Regression Model

The model used to estimate the impact is taken from the guidelines on Measurement and Verification of behavior-based energy-efficiency programs. [41]

$$
\begin{aligned}
Ln\_cost_{\mathrm{it}} = \alpha_0 + \alpha_1 * Treatment_i + \alpha_2 * Post_t + \alpha_3 * TreatmentxPost_{\mathrm{it}} + \beta_1 * CDD_t + \beta_2 * CDDxTreatment_{\mathrm{it}} \\
+ \beta_3 * CDDxPost_t + \beta_4 * CDDxTreatmentxPost_{\mathrm{it}} + v_i + u_i
\end{aligned}
\tag{5.1}
$$

where :

$\mathrm{Ln\_cost_{it}}$: Natural log of the cost of conditioning home i on day textitt

$\mathrm{Treatment_i}$: Dummy variable that takes the value of 1 if home i is a treatment home

$\mathrm{Post_t}$: Dummy variable that takes the value of 1 if t is in the treatment period

$\mathrm{TreatmentxPost_{it}}$ Dummy variable that takes the value of 1 if home i is measured in the treatment period

$\mathrm{CDD_t}$: Cooling degree days for day t

$\mathrm{CDDxTreatment_{it}}$: Interaction of $\mathrm{CDD_t}$ with $\mathrm{Treatment_i}$

$\mathrm{CDDxPost_t}$: Interaction of $\mathrm{CDD_t}$ with $\mathrm{Post_t}$

$\mathrm{CDDxTreatmentxPost_{it}}$: Interaction of $\mathrm{CDD_t}$ with $\mathrm{TreatmentxPost_{it}}$

5.3.2 Schedule Cost Analysis

Regression Analysis was performed using the model above on data that represented the daily cost of conditioning the home. All the temperature setpoints used on a particular day for a home were used to calculate the cost of conditioning the home using Equation 3.10. Data was obtained from Nest logs for the homes. The average change in the cost of conditioning a home in the pre-*intervention* and post-*intervention* period is compared across Groups 1,2 and 3. The beginning of the Post-*intervention* period was the day by which most thermostats in Group 3 were programmed with selected schedule recommendations, as opposed to the day the recommended schedules were emailed to participants of Group 3.

Panel Study Between Group 1 and 3

Equation 5.1 is estimated using data from both treatment and control groups, during the training and treatment periods. Homes from Group 1 formed the control group and homes from Group 3 formed the treatment group. Data from the entire study period was used.

Data was converted to the format of variables in Equation 5.1. Cooling Degree Days is defined as the amount of energy required to cool a space relative to a baseline temperature, derived with respect to the outside temperature, on a given day. The base temperature is a typically an indoor temperature suitable for human comfort. The Cooling Degree Days (CDD) values for the period of the study are available from a number of online resources. CDD values used in the estimation here were obtained from [42]. The natural log of the daily cost was used as the dependent variable.

Parameter	**Coefficient Estimate**	**Standard Error of Coefficient**
Constant	-0.4912	0.0249
$\mathrm{Treatment_i}$	-0.0211	0.0354
$\mathrm{Post_t}$	-0.0325	0.0534
$\mathrm{TreatmentxPost_{it}}$	-0.0829	0.0755
$\mathrm{CDD_t}$	0.00334	0.00367
$\mathrm{CDDxTreatment_{it}}$	-0.00890	0.00525
$\mathrm{CDDxPost_{it}}$	-0.0028	0.00760
$\mathrm{CDDxTreatmentxPost_{it}}$	0.0063	0.0108

Table 5.3: Estimated Coefficients

Regression Analysis was performed using MiniTab. The estimated coefficients of the model are presented in Table 5.3 The baseline temperature for CDD estimation used in the model was 70 degrees Fahrenheit.

The model in Equation 5.1 measures the cost of conditioning a home, given a set of independent variables. To measure the impact of recommendations, Average Treatment Impact is used. Treatment Impact is the measure of the causal effect of a treatment on the outcome variable, in this case, energy cost. The treatment indicator is generally a binary variable indicating whether treatment was performed on the sample or not. For an individual home, the treatment effect is given by costi1 - costi0

The Average Treatment Impact across all individual samples(ie. homes) is the measure of the difference in the expected values of cost(the outcome variable) before and during treatment(post-*intervention* period). The Average Treatment Effect/Impact is denoted as E[$\mathrm{cost_{i1}}$ - $\mathrm{cost_{i0}}$] where $\mathrm{cost_{i1}}$ is the cost in the post-*intervention* period and $\mathrm{cost_{i0}}$ is the cost in the pre- *intervention* period(without the treatment being evaluated). The Average Treatment Impact is the expected effect of the treatment for a randomly drawn individual sample(in this case a home) from the population.

The resultant estimated parameters of the terms involving the treatment variable in the regression model, indicate the estimated average change in the dependent variable due to the treatment.

Average Treatment Impact(ATC) is estimated using the equation:

$$\widehat{ATC} = \hat{\alpha_2} + \hat{\beta_3} * CDD_t \tag{5.2}$$

$\mathrm{CCD_t}$ is the average Cooling Degree Days in the treatment period.(post-*intervention* period) $\mathrm{CDD_t}$ was 5.72 for the treatment period. Substituting the values from Table 5.3 and converting to the impact into percentages(since change in natural log is an approximation of percentage change), the Average Treatment Impact is -4.86%. A negative percentage indicates decrease in the value of the dependent variable. Thus, schedule costs decreased by 4.68% by ThermoCoach when compared to programmable thermostats.

Confidence intervals of this point estimate can be calculated as-

$$\widehat{ATC} = \widehat{ATC} \pm c * se(\widehat{ATC}) \tag{5.3}$$

For a 95% confidence interval c is the 97.5th percentile in a $\mathrm{t_{df}}$ distribution, Degrees of Freedom(df) is calculated as

$$df = n - k - 1 \tag{5.4}$$

where n is the number of samples and k is the number of independent variables in the model

For the model, k = 7 and n = 1678. On substituting the values, c = 1.96. The standard error obtained from the model is 0.319.

Thus for the specified model, the confidence interval for ATC is

Lower Bound = 3.9 (in percentages)

Upper Bound = 5.45 (in percentages)

In addition to the panel study above, two more studies were performed with Groups 1 and 2 forming the control groups and Group 3 forming the treatment Group. The summarized results of the study are shown in the table Table 5.4

Treatment Group	**Control Group**	**ATC**** [1]
Group1	Group 2	+7.79%
Group2	Group 3	-12.39%
Group1	Group 3	-4.686%

Table 5.4: Average Treatment Impact on Schedule Cost [All Days]

Row 1 of Table 5.4 is the result of a panel study of the energy usage of Group 2 is compared to that of Group 1. Group 1 formed the control group and Group 2 formed the treatment group. Group 1 represents homes with programmable thermostats and Group 2 represents homes with Nest thermostats.

Table 5.4 contains results from models that used data from the entire study duration. The Nest thermostat's Auto Schedule feature(scheduling learning) takes about two weeks to learn a home's schedule. Thus the panel studies above were repeated with data from the first two weeks and last two weeks of the study. The first

[1] **alpha=0.01

two weeks formed the training period where all three groups behaved like users of manually programmable thermostats and the last two weeks of the study formed the treatment period. This allows for a comparison between Nest's Auto schedule feature, manually programmable thermostats and ThermoCoach. The results in Table 5.5 follow similar trends of Table 5.4.

Treatment Group	**Control Group**	**ATC**** [2]
Group1	Group 2	+9.96%
Group2	Group 3	-10.37%
Group1	Group 3	-1.069%

Table 5.5: Average Treatment Impact for Group-wise Panel Study on Schedule Cost [4 weeks]

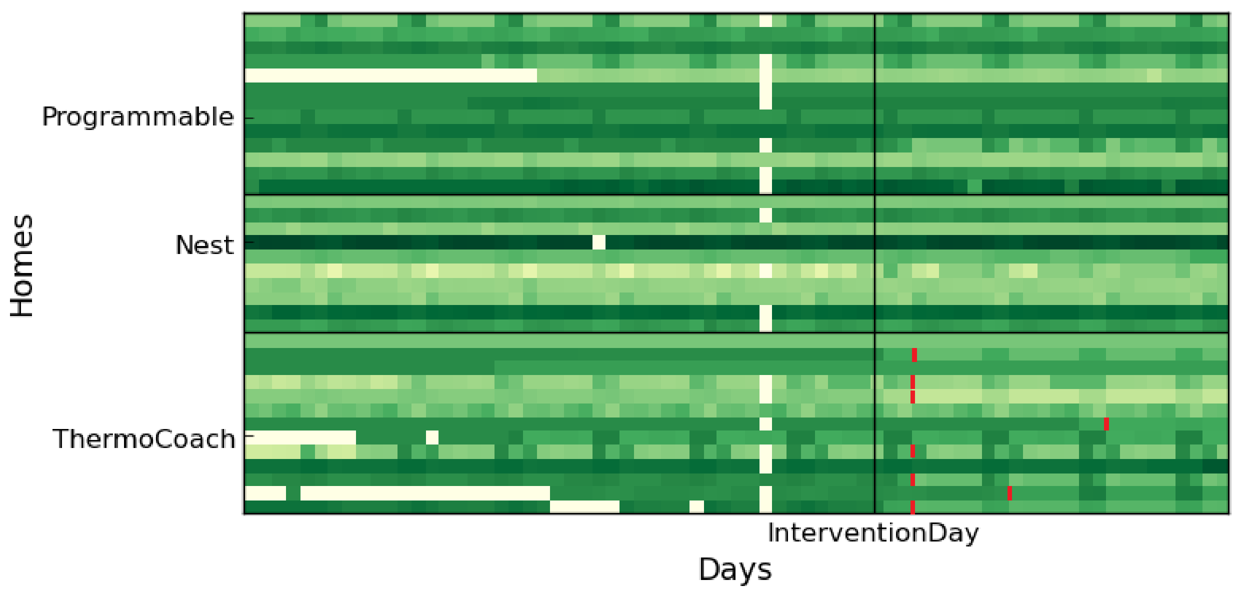

Figure 5.2: Energy Usage ColorMap across all homes

Figure 5.2 represents the daily energy cost estimated in a similar manner to equation 3.10. Each cell in Figure 5.2 represents the energy cost of the thermostat schedule used on a particular day. Each day's thermostat schedule was parsed from Nest data being logged for each home's thermostat. Each horizontal line represents data for a home. The homes are arranged based on the group they belong to. Energy costs over time for all homes in the study are estimated. Darker shades indicate higher energy cost and lower shades are ideally desired. The graph also indicates the *intervention* day. All days after the vertical line at *intervention* day includes data from only the post-*intervention* period. White cells indicate days of when data loss was significant.

[2] **alpha=0.01

It is clear from the graph that thermostat recommendations reduced energy costs significantly for seven out of eight homes that chose recommended schedules at time of *intervention* . One home choose a schedule that looked very similar to their pre-*intervention* schedule. Thus significant change in energy usage is not seen for that home. Homes that did not change their schedules continued to have similar energy usage as compared to their pre-*intervention* usage. For the homes that selected recommended schedules, change in energy consumption is visible soon after the *intervention* date. For two homes however, the effect of the schedule is seen a few days later. The Nest thermostats in these homes were having WiFi connectivity issues and were offline at time of *intervention* . This prevented the schedules from being programmed remotely. Once this issue was fixed, the effect of the schedules is seen. Red bars in Figure 5.2 indicate when significant reduction in energy usage was first seen for a home.

After *intervention* , energy usage of only one home changed in groups 1 and 2 and Group 1 seems to have consumed more energy than Group 2 and 3. The one home that decreased energy use post-*intervention* , did not have a schedule before *intervention* and a schedule with a *Away* setback was set soon after *intervention* . Overall, it seems that energy based eco-feedback alone may not be sufficient to keep users engaged in their energy utilization. Participants may have looked at their energy usage but failed to do anything about it in the absence of any actionable feedback. This is however impossible to ascertain due to several independent factors and variability of energy usage across homes. From looking at Group 2, it is also apparent from the figure that Nest's schedules have not improved energy usage over time.

It is interesting to note that about 50% of the homes across all groups had some schedules even though some may have used more energy than others. A weekday-weekend pattern and increase in energy usage on weekends is quite clear.

5.3.3 OnTime Analysis

Similar to the analysis described in the previous section, panel studies were performed on data that represented the fraction of the day the air-conditioning was on and actively cooling the home. The same model as in Equation 5.1 was used. The dependent variable in this case was the log values of the fraction of the day the air conditioning was on. Table 5.9 contains the resultant treatment effect values. All days of the study were used in the model.

Treatment Group	**Control Group**	**ATC*** [3]
Group1	Group 2	-1.5%
Group2	Group 3	-4.63%
Group1	Group 3	-6.15%

Table 5.6: Average Treatment Impact on OnTime [All days]

OnTime analysis was also performed with data from the last two and first two week of the study. These results give greater confidence for the impact involving Group 2, ie. the Nest group.

Treatment Group	**Control Group**	**ATC*** [4]
Group1	Group 2	1%
Group2	Group 3	-5%
Group1	Group 3	-3.89%

Table 5.7: Average Treatment Impact on OnTime [4 weeks]

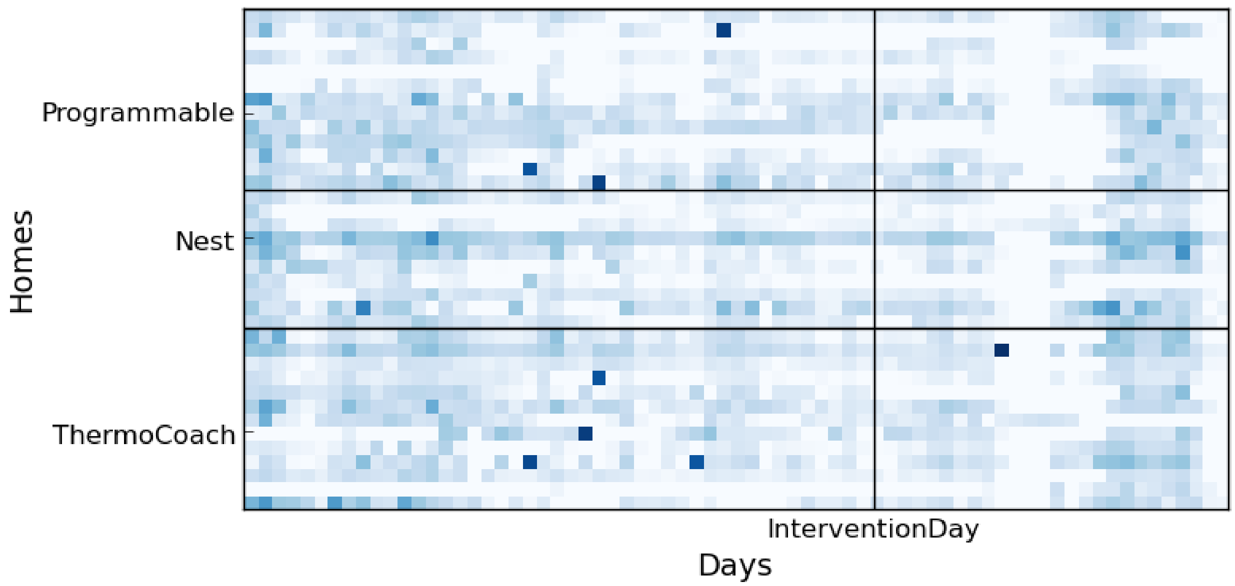

Figure 5.3: OnTime ColorMap accross all homes

Figure 5.3 shows the fraction of the day the air conditioning was on. Days are plotted on the x-axis and homes are plotted on the y-axis. Each row represents data from one home. Each cell represents the fraction of the day the air conditioning was on that day. Cells that are completely white indicate days when data was lost hardware issues.

5.3.4 Overrides Analysis

Throughout the study, participants made manual changes to the target temperature in their homes. These changes were anywhere between 1-8 degrees Fahrenheit changes. Instead of the number of temperature changes made, the average degrees changed daily can be considered as an indication of a user's comfort level for the thermostat schedule. This measure does not penalize homes for making a one degree change to the

[3] *alpha=0.05
[4] *alpha=0.05

temperature which is not really indicative of user comfort. A change in target temperature that is not part of the thermostat's schedule is considered an override. The panel study data indicated the average change in degrees made on a day. Group 1, 2 and 3 were evaluated and the results are presented in Table 5.8. The models used data from the entire study period.

Treatment Group	**Control Group**	**ATC*** [5]
Group1	Group 2	-3.29%
Group2	Group 3	-45.67%
Group1	Group 3	-47.43%

Table 5.8: Average Treatment Impact on Average Degrees Changed [All days]

Similar to the analysis in previous sections, panel studies represented in Table 5.9 were performed with data from the first and last two weeks of the study. The average daily degrees changed by ThermoCoach users is compared to that of Nest and manually programmable thermostat users.

Treatment Group	**Control Group**	**ATC*** [6]
Group1	Group 2	-3.13%
Group2	Group 3	-68.34%
Group1	Group 3	-61.8%

Table 5.9: Average Treatment Impact on Average Degrees Changed [4 weeks]

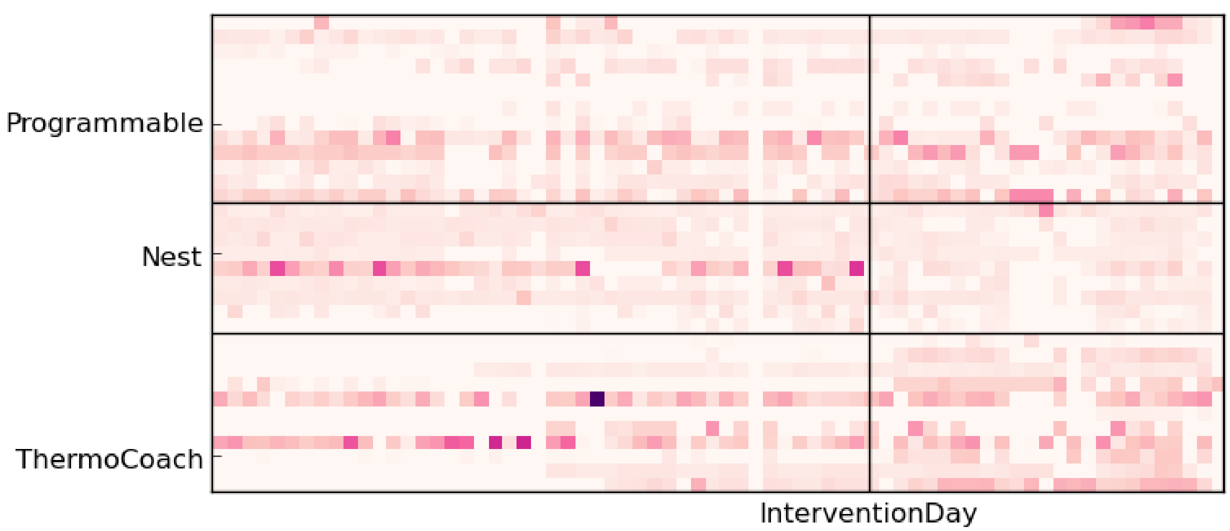

Figure 5.4: Colormap of daily Average Degrees Changed across all homes

Figure 5.4 shows the average degree change per day when compared to the day's setpoint schedule. In some homes in the ThermoCoach group, manual overrides were made post-*intervention*. These changes were

[5] *alpha=0.05
[6] *alpha=0.05

changes made to the target temperature at a particular time in the day and not changes made to the schedule. Each row in the figure represents data from a home. Days are plotted on the x-axis and and homes on the y-axis. Each cell represents the number of manual changes made on a particular day. These changes could have been made on the actual thermostat or through its web or mobile interface. In many homes changes made were reverted back to reflect the setpoint schedule. In most cases, the temperature was changed by one or two degrees Fahrenheit, while in some cases the temperature was changed by four to six degrees Fahrenheit. Changes in the treatment period can be considered an indication of comfort levels. It can also be attributed to warmer weather in the post-treatment period.

Chapter 6

Discussion

The results indicate that ThermoCoach's schedules are more energy efficient compared to Nest's schedules and those set on programmable thermostats. In addition, ThermoCoach reduced the OnTime by 6%. The number of overrides decreased post-*intervention* indicating that residents were comfortable with their selected schedule recommendation.

6.1 Cost of Conditioning a home

Post-*intervention*, the energy cost to condition a home(derived from setpoint values) decreased for Groups 1(programmable thermostats) and 3(ThermoCoach) and energy usage did not change for Group 2(Nest users). ThermoCoach reduced the amount of energy needed to condition a home by about 5% when compared to programmable thermostats. In the pre-*intervention* period, T-tests between energy cost of Group 1 and 3 indicate that Group 1 homes used more energy than Group 3 homes. This difference increased in the post *intervention* period by 65% in the post *intervention* period. Thus even though Group 1 homes did use more energy than Group 3 homes in the pre-*intervention* period, schedule recommendation significantly reduced the energy usage of Group 3 homes. Two homes in group 1 had significant reduction in their energy usage post-*intervention*. One home modified their existing schedule to make it more energy efficient while the other home did not have a schedule pre-*intervention* and set a schedule after receiving energy feedback. Thus energy feedback did motivate two homes to reduce their energy usage. In Group 3, eight homes choose a schedule recommendation or some variation of it. Even though two of these homes had schedules pre-*intervention*, they choose recommendations that were more energy efficient. The other homes that accepted schedules did not have any schedule prior to *intervention*. Thus significant reduction in energy usage is seen for these homes. Energy usage of did not vary in homes that did not change their schedule post *intervention*.

Cost of conditioning the home with ThermoCoach decreased by 10% when compared to the Nest thermostat. Pre-*intervention*, there was no statistically significant difference between energy use between homes in Group 2 (with Nest thermostats)and homes in ThermoCoach group (Group 3). However, in the post *intervention* period, T-tests confirm that energy usage in Group 3 homes was significantly less. Thus schedule recommendations did have an impact on energy savings and did better than Nest's Auto Schedule feature.

When compared to Group 1, Nest schedules increased the cost of conditioning a home by 9.96% During the pre-*intervention* period, Group 1 homes used 13.6% more energy to condition their homes(schedule cost) when compared to Group 2 homes. In the treatment period, energy cost of homes in Group 1 decreased significantly. Thus difference in the mean energy costs between homes in Group 1 and 2, decreased in the treatment. Group 1 used 6% more energy than Group 2. This caused the results to indicate that the Nest group did worse than Group 1(conventional programmable thermostats). Hence the Nest(Group 2) did not use more energy, only the impact of the treatment(Nest's Auto Schedule) decreased in the treatment period when compared to conventional thermostats. A paired T-Test for samples from Group 2 before and after *intervention* indicate that there was no significant change in energy usage for the Nest group.

6.2 OnTime Analysis

ThermoCoach decreased the average time for which the air conditioning was *on*, on a given day(onTime) by 6% when compared to conventional programmable thermostats. The setbacks in ThermoCoach's recommended schedules ensured that the air conditioning was not *on* when occupants were `Away` and the nighttime setback in some homes also reduced energy use. These results indicate that ThermoCoach did indeed reduce the actual energy use of homes.

Nest reduces energy use by about 1-2% when compared to programmable thermostats. This result is applicable to the model that used data from the entire study duration. Nest's Auto-Schedule(learning) feature becomes available about fourteen days after the thermostat has been installed. The limitation of this model is that it does not really compare Auto-Schedule and programmable thermostats, instead it compares Auto-Schedule before and after *intervention*. Because of this, analysis was performed using the first two weeks of the study as the training period and the last two weeks as the treatment period. This analysis indicates that Nest increases onTime by 1%. However, varying lifestyles and other factors may have biased the result. The first week of the training period included a long weekend and there may have been an unequal distribution of homes between Group 1 and 2 that went on vacation in the training period(and hence used less energy) may have biased this result.

Auto-Away may have also affected energy use in Group 2 homes and may have played an important factor in the analysis. For Auto-Away to work effectively, one or both of the following are essential [33]:

- The Nest thermostat is installed such that it can sense activity when occupants are home. It then tries to infer when a home is typically unoccupied during the day.
- The time occupants leave their home in the mornings is predictable.

This understanding of Auto-Away's functioning implies that Auto-Away may not have been effective in some homes in which the thermostat was installed in a less frequently used part of the home. In such homes the Nest may have failed to detect activity whenever the home was occupied. Auto-Schedule learns from changes made to settings by users. If users did not set a *setback* when they left home during the learning period, it is possible that the automatically generated schedules may not have had a *setback* for times when the home was unoccupied. Because of all these factors it cannot be said with certainty that Nest increased energy usage in homes when compared to programmable thermostats. T-tests indicated that Nest used less energy than the programmable thermostats before and after *intervention*.

6.3 Number of Overrides

The number of overrides made on a thermostat can be an indication of the extent to which residents are comfortable with their thermostat schedule. An override is any change made to the temperature that is not part of the thermostat's setpoint schedule. Instead of the number of overrides, the average change in degrees made on a day is computed and evaluated. This eliminates cases where the temperature was changed by just 1 degree Fahrenheit on a given day. Average change in degrees made is referred to as an *override* in the following discussion. The average number of overrides increased for all the three groups after *intervention*.

Five homes in Group 3 always made significant changes to their thermostat throughout the study period. For these homes it is unclear if changes were made because occupants were uncomfortable or if it was just due to their habit of interacting with the device. The number of overrides decreased for Group 2 over time. A reason for this could be that the Nest was always keeping the homes at a cooler temperature, compared to ThermoCoach's schedules. The Nest schedules rarely had a setback of more than 3-4 degrees Fahrenheit. If the homes were always conditioned, it seems clear why the number of overrides decreased for the Nest group over time. Another reason could be that since the Nest learns from changes made by the user, overtime, it learned not to keep the temperature lower.

The daily average temperature change due to overrides reduced for Group 3 by 47% post-*intervention* when compared to conventional programmable thermostats. There was no statistically significant difference in the number of overrides between Group 1 and Group 3 in the pre-*intervention* period. These results indicate that most home in Group 3 were comfortable with the recommended schedules they chose.

The average degrees overridden by Nest group was 3.13% lower compared to programmable thermostat users(Group 1). In the pre-*intervention* period, the T-tests indicate that the number of overrides in Group 2 were higher than the number of overrides made by Group 1 homes. However, after *intervention*, the number of overrides increased for Group 1 when compared to Group 2. The reason for this could be that energy feedback made homeowners in Group 1 more conscious of their energy usage and instead of modifying or setting a schedule, they changed the target temperature on their thermostat. Paired T-tests for Group 2 before and after *intervention* indicate that there was no statistically significant difference in the number of overrides for Group 2. Thus though the impact analysis indicate that Nest group had lower number of overrides, the number of overrides increased for Group 1 post-*intervention*, which resulted in 3.13% decrease in the number of overrides by the Nest group.

Chapter 7

Limitations and Future Work

7.1 Regression Analysis

The regression analysis performed in the previous chapter has certain limitations. One of the most critical issues with using regression analysis is the assumption that specified model is correct. If variables are omitted from a model and those missing variables are correlated with one or more independent variables, the estimates could be biased. Any measurement error will lead to biased and inconsistent estimates of parameters. If the error terms are auto-correlated(errors in the treatment and pre-treatment are correlated), the standard error estimate may be biased [43]. The P-values of many coefficients in the models used for the different datasets in the analysis in Chapter 6, indicate that those coefficients were insignificant. In most cases, only the variables indicating treatment, the CDD values and their interactions were influential on the value of the dependent variable.

The models in the previous chapters had high residuals and this may have caused the estimates to be biased. R^2 ranged between 5-30% for the different models. T-Tests were performed on the three metrics for all three groups. The tests validate the general trends seen in the impact analysis. The T-tests compared the mean values for groups before and after *intervention* for all combinations of the three groups in the study.

7.2 Other Limitations

During the training period, there was some confusion about what features of the Nest could be used amongst a few homes in Group 2. Some participants were under the assumption that they could not use Nest's 'Auto-Away' feature. This feature estimates when a home is typically unoccupied and sets the temperature to a *setback* value. A few participants did not change the temperature as they thought that the settings were provided by researchers as part of the study, which was not the case. Because of this confusion about Nest's settings, the training period was extended from four weeks to six weeks. All participants were sent an email reminding them that they could change the temperature and schedule at any point during the study.

At time of *intervention*, the system that processed the selections made by participants was not functioning. Selections were manually verified from participants of Group 3, over the phone. Each participant was asked if they had viewed the email and made a selection. Then they were asked to repeat the selection they made. Eleven out of thirteen participants had a made a selection before the phone conversation, thus it is likely that the response to eco feedback was not biased by the phone call. However, participants were more likely to view the email since they were a part of the study. Thus the response rate to eco-feedback in the form of recommendations may vary in longer use cases, but the results are promising and worthy of further investigation.

The recommended schedules had only four setpoints. While this made the recommendations easier to understand, it is limited and some homes may require flexible scheduling options. The Nest thermostat also has an advantage over ThermoCoach since it allows for flexible scheduling which has been found to be useful to users [11]. The current implementation of the scheduling algorithm creates two schedules: one for the week and one for weekends. ThermoCoach also does not generate separate schedules for each day of the week. Flexible scheduling is extremely useful to have since some homes may have a three day schedule during the week and may need a different schedule based on the day of the week. The ThermoCoach UI allows users to modify and set different daily schedules. Thus even though the system did not generate different schedules based on the day of the week, homes could have flexible schedules if the users manually made changes to the ThermoCoach's suggestions. Hence, the absence of this feature does not affect the energy saving potential of ThermoCoach. The only concern would be user comfort. Users are given a chance to modify the schedules before they are programmed on the thermostat. Additionally, users can always override schedule setpoints. The current algorithm can be easily expanded to add this feature. In this pilot study however, the data was too noisy and insufficient for the system to generate any meaningful daily schedules.

ThermoCoach currently is not capable of programming the Nest with new schedules automatically and requires human intervention. At *intervention*, schedules for homes in Group 3 were manually programmed. Future work includes use of thermostat that can be better interfaced with.

Occupancy data can be improved with better sensors. Many participants occasionally forgot their key tags at home. This was particularly prevalent in homes with multiple vehicles. ZWave motion sensors too have their drawbacks. Since these ZWave battery operated sensors sleep, they often miss events or fail to report an event to the controller. Also as the number of nodes increases, the performance of the ZWave network decreases. Motion sensors and ZWave controller had to be replaced in about nine homes. In others, the network would repair itself in a couple of days. Only 1/3rd of the homes had good consistent ZWave data. The Hubs often required to be manually rebooted by disconnecting and reconnecting them to power. Since the system stored data on *tmpfs* partition, data was lost on reboot in cases where it was not successfully synced before the *endpoint* went down. The platform could be modified in the future to copy and restore data from tmpfs partition at every reboot. Failures of Hubs or weak WiFi lead to loss of data, including energy data from Nest, allowing for only limited energy usage analysis before and after *intervention*.

This pilot study lasted for only one season over a period of three months. The summer season during which the study conducted was a relatively mild summer for the region. This may have caused people to use their air-conditioning less often. A study over multiple seasons would help to reduce the effect of any other external factors that may have affected energy usage behavior. A study of longer duration would give useful insights into the performance of the system. It would be useful to analyze the minimum number of training days needed to detect changes in occupancy patterns with significant confidence. Future work would include a study that lasts twelve to eighteen months.

ThermoCoach did not present to users a history of their setpoint schedules used in the past, due to the short duration of this study. However, the current thermostat schedule was presented to users at the time of *intervention*. Additionally, the Nest web interface allows users to browse their energy usage history where they can view how many hours the system was cooling over the last one month and day by day usage of the past week. Studies have shown that users find it useful to have the ability to view past schedules when they make decisions about their current settings [11]. At time of writing, the web interface did not give a dynamic estimation of schedule cost while users modified a recommended schedule. Future work includes real-time feedback with estimates of change in cost and comfort as users increase or decrease temperatures and change setpoints.

Current evaluation of ThermoCoach uses an estimation of energy cost and use. Actual energy used was not measured. A more realistic cost evaluation could be done by using models that include information about the type of equipment in a home and other factors that may affect energy use.

ThermoCoach currently does not generate recommendations daily and it does not immediately respond to changes in a home's occupancy patterns. Future work will include analysis of the minimum number of training days needed to make a credible detection of change in patterns. ThermoCoach also does not perform room-level occupancy prediction. The number of sensors needed would increase with the number of rooms. This would be worth exploring once room-level control of HVAC becomes prevalent in homes. ThermoCoach is currently designed to work in homes. For ThermoCoach to be useful in offices and commercial buildings, significant changes to the sensing systems, control algorithms and schedule generation algorithms will be needed.

Chapter 8

Conclusion

Self-programming thermostats have the potential to reduce HVAC costs. Self-programming thermostats detect a home's occupancy patterns and generate tailored setpoint schedules. They reduce the amount of human effort needed to program or update a thermostat schedule. ThermoCoach uses low cost, off-the-shelf motion sensors and Bluetooth 4.0 sensors to detect occupancy in a home. Setpoint schedules based on learned occupancy patterns are generated and presented to users through a web interface. The schedule recommendations vary in user comfort levels and energy usage. This form of eco-feedback makes homeowners aware of their energy consumption and motivates them to modify their behavior to conserve energy.

ThermoCoach was evaluated against a programmable thermostat and the Nest Learning Thermostat. 39 participant homes were recruited for the study. The system was trained for six weeks and data collection lasted three months. Energy savings in homes with ThermoCoach were found to be statistically significant. ThermoCoach schedules reduced cost by 5% and reduced energy use(onTime) by 6% when compared to programmable thermostats.

Actionable eco-feedback that puts users in control, in addition to energy usage data, has the potential to save energy in homes by allowing homeowners to make conscious decisions about their energy consumption. In this study too, eco-feedback in the form of schedule recommendations was found to be useful in achieving lower energy costs. Homes and HVAC systems in particular, are major consumers of energy. With a system like ThermoCoach, it will become easier to achieve the true energy saving potential of programmable thermostats, in the long term. Annually, such systems can reduce energy cost by 10-20% and billions of dollars can be saved.

Acknowledgement

I offer my gratitude to my advisor, Dr. Kamin Whitehouse, for his support and guidance through the Masters program. His advice, patience and encouragement helped me overcome the challenges faced in this project. In addition, I would also like to thank my committee members Dr. Mary Lou Soffa, Dr. Yanjun Qi and Dr. Kevin Sullivan for their suggestions and guidance on my thesis.

Dr. Mark Newman and Rayoung Yang helped with the study design and their valuable insights were extremely useful while conducting the study. They also designed and conducted user interviews to understand participant's response to the system.

Special thanks to John Feminella who built Piloteur and without whom it would have been extremely difficult to implement the project within the short time frame. I would like to thank Eva Potter and her team who helped design and implement the front-end of ThermoCoach's website. I would also like to acknowledge Duct Rite Mechanical who installed the sensors in homes. I am grateful to my lab-mates for their suggestions and guidance.

Finally, I would like to thank my family and friends for their constant love and support.

Bibliography

[1] Frequently asked questions. Energy Information Administration. http://www.eia.gov/tools/faqs/index.cfm#generalenergy.

[2] Energy explained. Energy Information Administration. http://www.eia.gov/energyexplained/index.cfm?page=environment_home.

[3] Climate change. U.S. Department of Energy. http://www.energy.gov/science-innovation/climate-change.

[4] The stern review on the economic effects of climate change. *Population and Development Review*, 32(4):793–798, 2006.

[5] Residential energy end-use splits by fuel type (quadrillion btu). table 2.1.6. U.S. Department of Energy, 2014. http://buildingsdatabook.eren.doe.gov/TableView.aspx?table=2.1.6.

[6] Household energy consumption and expenditures, consumption by end use, 2009. annual energy review. Energy Information Administration, 2009. http://www.eia.gov/totalenergy/data/annual/index.cfm.

[7] Energy efficiency. U.S. Department of Energy. http://www.energy.gov/science-innovation/energy-efficiency.

[8] Residential energy consumption survey: preliminary housing characteristics tables. Energy Information Administration, 2009. http://www.eia.doe.gov/emeu/recs/recs2009/hc2009_tables/detailed_tables2009.html.

[9] Krumm J. and Brush. Learning time-based presence probabilities. *Proceedings of Pervasive*, 2011.

[10] T. Peffer, M. Pritoni, A. K. Meier, C. Aragon, and D. Perry. How people use thermostats: A review. *Building and Environment*, 46(12):2529–2541, 2011.

[11] Rayoung Yang, Mark W. Newman, and Jodi Forlizzi. Making sustainability sustainable: Challenges in the design of eco-interaction technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 823–832, New York, NY, USA, 2014. ACM.

[12] Olga Sachs, CSE Fraunhofer, and Building America. *Field Evaluation of Programmable Thermostats*. US Department of Energy, Energy Efficiency & Renewable Energy, Building Technologies Program, 2012.

[13] Alan Meier. How people actually use thermostats. *ACEEE Summer Study on Energy Efficiency in Buildings. Pacific Grove, Calif.: American Council for an Energy Efficient Economy.*, 2012.

[14] Nevius M and Pigg S. Programmable thermostats that go berserk: taking a social perspective on space heating in Wisconsin. In *Proceedings of the 2000 ACEEE Summer Study on Energy Efficiency in Buildings*, page 8, 2000.

[15] Lisa Gobio-Lamin. Usability of thermostat controls an example of the ucl energy institute. In *People and Buildings*, 2011.

[16] K. Rathouse and B. Young. *Market Transformation Programme - Domestic Heating: Use of Controls.* Department of Environment, Food and Rural Affairs, UK, 2004.

[17] S. Karjalainen. *Why It Is Difficult to Use a Simple Device: An Analysis of a Room Thermostat.* VTT, 2007.

[18] S. Rubens and J. Knowles. *What people want from their heating controls: a qualitative study.* A report to the Department for Energy and Climate Change, 2013.

[19] Karen Ehrhardt-Martinez. Changing habits, lifestyles, and choices: The behaviours that drive feedback-induced energy savings. *Proceedings of the 2011 ECEEE Summer Study on Energy Efficiency in Buildings, Toulon, France*, pages 6–11, 2011.

[20] Allen D and Janda K. The effects of household characteristics and energy use consciousness on the effectiveness of real-time energy use feedback: a pilot study. In *Proceedings of the 2006 ACEEE Summer Study on Energy Efficiency in Buildings*, 2006.

[21] Corinna Fischer. Feedback on household electricity consumption: a tool for saving energy? *Energy Efficiency*, 1(1):79–104, 2008.

[22] L. McClelland and S. W. Cook. Energy conservation effects of continuous in-home feedback in all-electric homes. *Journal of Environmental Systems*, 1980.

[23] Wood G and Newborough M. Energy-use information transfer for intelligent homes: enabling energy conservation with central and local displays. *Energy and Buildings*, 2007.

[24] Lutzenhiser L. *Social and behavioral aspects of energy use.* Annual Review of Energy and Environment, 1993.

[25] Neenan B, Robinson J, and Boisvert RN. Residential electricity use feedback: a research synthesis and economic framework. *Electric Power Research Institute (EPRI)*, 2009.

[26] Van Raaij, W. F., and T. M. M. Verhallen. A behavioral model of residential energy use. *Journal of Economic Psychology*, 3:39–63, 1983.

[27] Darby S. The effectiveness of feedback on energy consumption. *Oxford: Environmental Change Institute*, 2006.

[28] M. C. Mozer, L. Vidmar, and R. M. Dodier. *The Neurothermostat: PreHeat Optimal Control of Residential Heating Systems.* Advances in Neural Information Processing Systems 9, 1997.

[29] Manu Gupta, StephenS. Intille, and Kent Larson. Adding gps-control to traditional thermostats: An exploration of potential energy savings and design challenges. In Hideyuki Tokuda, Michael Beigl, Adrian Friday, A.J.Bernheim Brush, and Yoshito Tobe, editors, *Pervasive Computing*, volume 5538 of *Lecture Notes in Computer Science*, pages 95–114. Springer Berlin Heidelberg, 2009.

[30] James Scott, A.J. Bernheim Brush, John Krumm, Brian Meyers, Michael Hazas, Stephen Hodges, and Nicolas Villar. Preheat: Controlling home heating using occupancy prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, pages 281–290, New York, NY, USA, 2011. ACM.

[31] Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 211–224, New York, NY, USA, 2010. ACM.

[32] Ge Gao and Kamin Whitehouse. The self-programming thermostat: Optimizing setback schedules based on home occupancy patterns. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 67–72, New York, NY, USA, 2009. ACM.

[33] Nest learning thermostat. Nest Labs. https://nest.com/.

[34] Neal Rosen, Rizwan Sattar, Robert W Lindeman, Rahul Simha, and Bhagirath Narahari. Homeos: Context-aware home connectivity. In *International Conference on Wireless Networks*, pages 739–744, 2004.

[35] John Feminella, Devika Pisharoty, and Kamin Whitehouse. Piloteur: A lightweight platform for pilot studies of smart homes. In *Proceedings of the 1st ACM International Conference on Embedded Systems For Energy-Efficient Buildings*. ACM, 2014.

[36] C. ; Paradells Gomez and J. Wireless home automation networks: A survey of architectures and technologies. *Communications Magazine, IEEE*, 48(6):92, June 2010.

[37] Bluetooth 4.0. http://www.bluetooth.com/Pages/Basics.aspx.

[38] Sticknfind. https://www.sticknfind.com/.

[39] Open-zwave. Google Code. https://code.google.com/p/open-zwave/.

[40] Timothy W Hnat, Vijay Srinivasan, Jiakang Lu, Tamim I Sookoor, Raymond Dawson, John Stankovic, and Kamin Whitehouse. The hitchhiker's guide to successful residential sensing deployments. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 232–245. ACM, 2011.

[41] Sanem Sergici and Ahmad Faruqui. Measurement and verification principles for behavior-based efficiency programs. 2011.

[42] Degree days. BizEE Software. http://www.degreedays.net/.

[43] RLW Analytics. The california evaluation framework. 2004.

[44] Sami Karjalainen. Why it is difficult to use a simple device: An analysis of a room thermostat. In JulieA. Jacko, editor, *Human-Computer Interaction. Interaction Design and Usability*, volume 4550 of *Lecture Notes in Computer Science*, pages 544–548. Springer Berlin Heidelberg, 2007.

[45] Load impact estimation for demand response: Protocols and regulatory guidance (2008). California Public Utilities Commission, Energy Division, 2008. http://www.calmac.org/events/finaldecision_attachementa.pdf.

[46] Chris Elbers and Jan Willem Gunning. Evaluation of development programs: Using regressions to assess the impact of complex interventions. Technical report, Tinbergen Institute Discussion Paper, 2012.

Appendix

A sample *Deployment Sheet* is included. It was used by the installer to note down details during deployment. Email recommendations to homes in Group 3 are also presented here. Each image contains four schedules. The `Current` schedule is the schedule that was set by participants before intervention. The three schedules- `High Comfort` , `Energy Saver` and `Super Energy Saver` are ThermoCoach's recommendations for a home.

Thermocoach: Deployment Information Sheet

ThermoCoach

Participant Name: ABC UVA

Phone Number: 434-xxx-xxx

Address: xxx,yy,zz

Hub Ids: thermocoach-hx-x-ble , thermocoach-hx-y-ble, , thermocoach-hx-x-nble, thermocoach-hx-x-zwave

Returned Thermostat to Participant: ☐ Yes ☐ No

Nest Install Successful: ☐ Yes ☐ No

HVAC Power Reading (State: Off): ____________________

HVAC Power Reading (State: On): ____________________

IDs of Hubs Not Working: ____________________

Number of Tags Given: ____________________

Notes:

1

Figure 1: Deployment Sheet

Thermocoach: Deployment Information Sheet

Floor Plan

2

Figure 2: Deployment Sheet

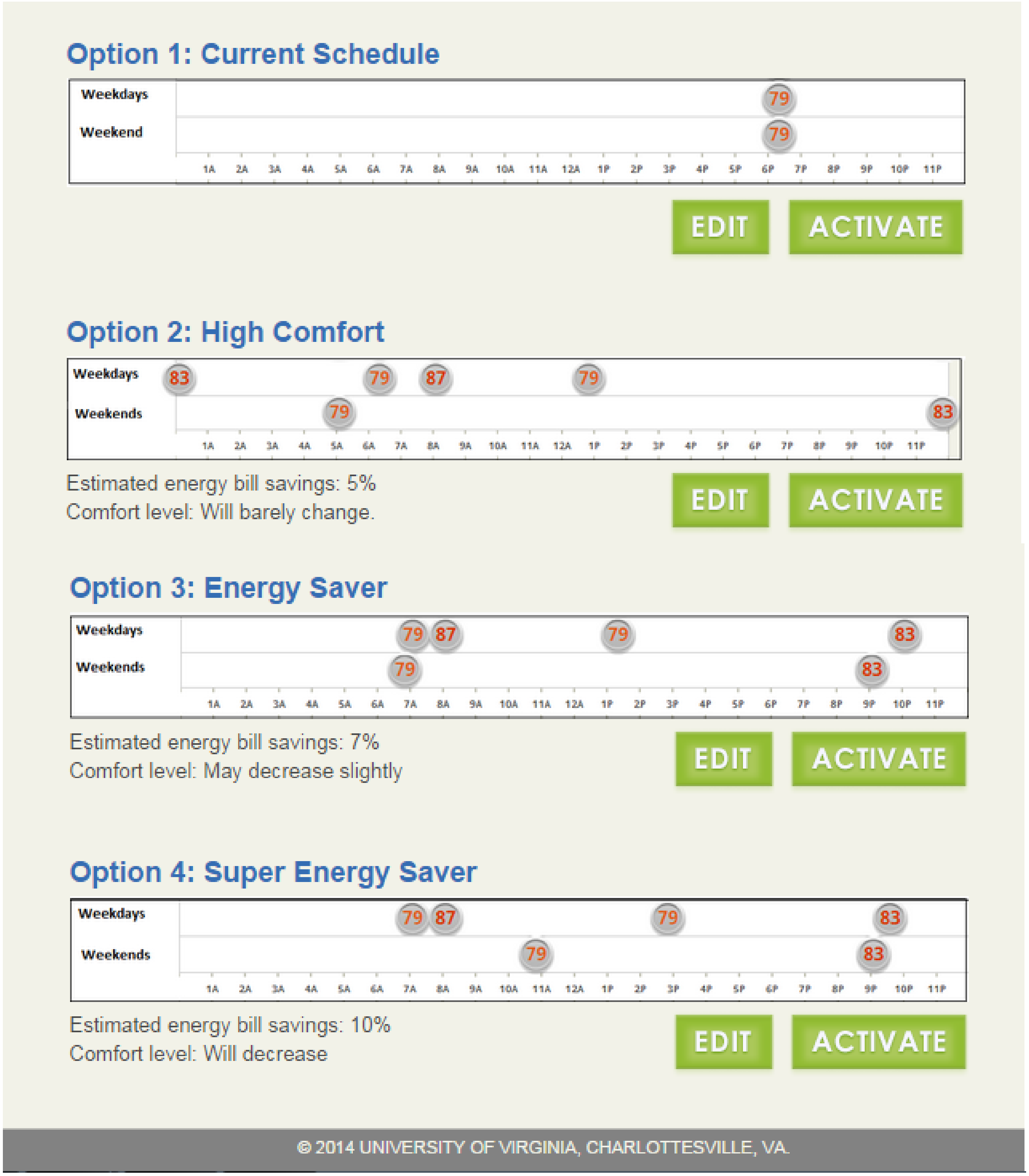

Figure 3: Recommendations For Home 1

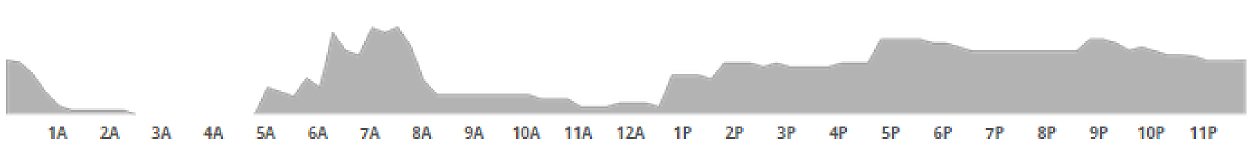

Figure 4: Occupancy Graph for Home 1

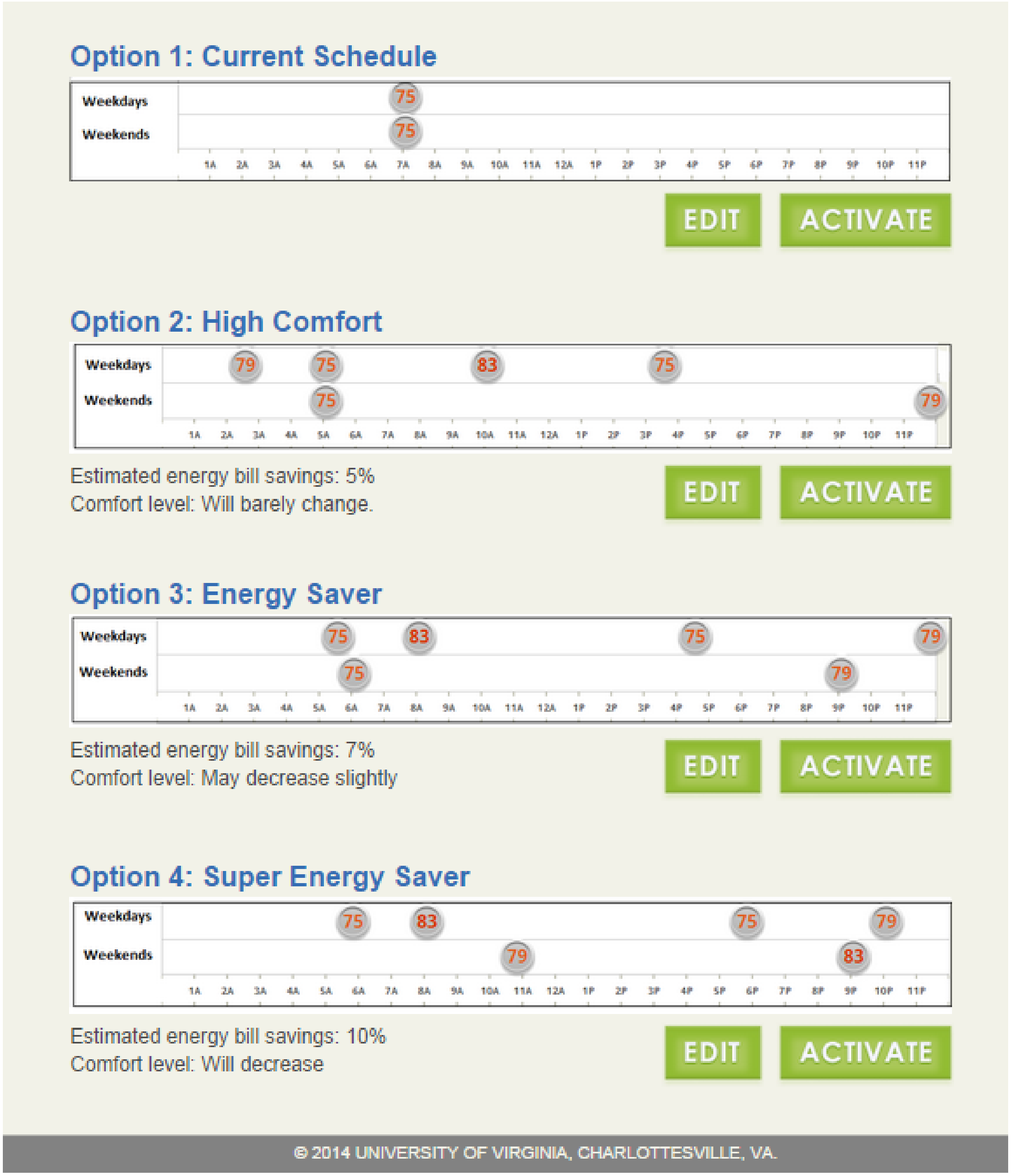

Figure 5: Recommendations For Home 11

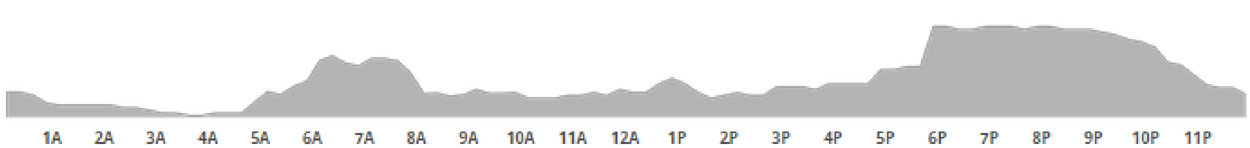

Figure 6: Occupancy Graph for Home 11

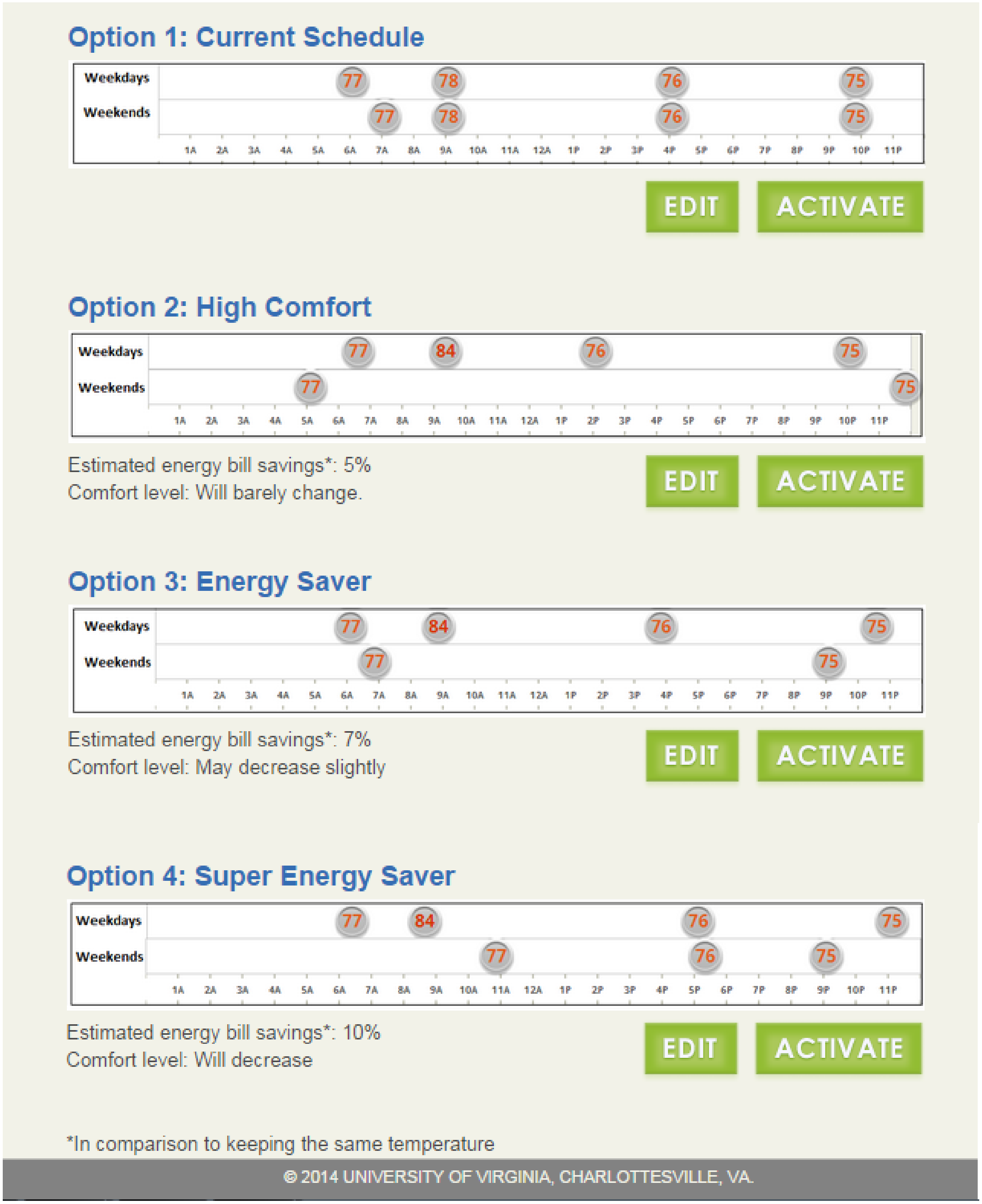

Figure 7: Recommendations For Home 13

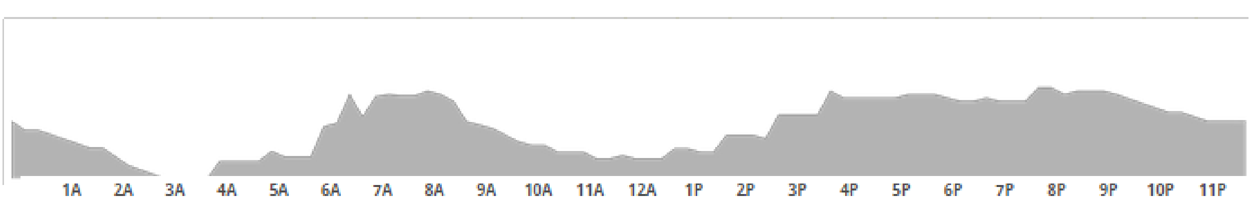

Figure 8: Occupancy Graph for Home 13

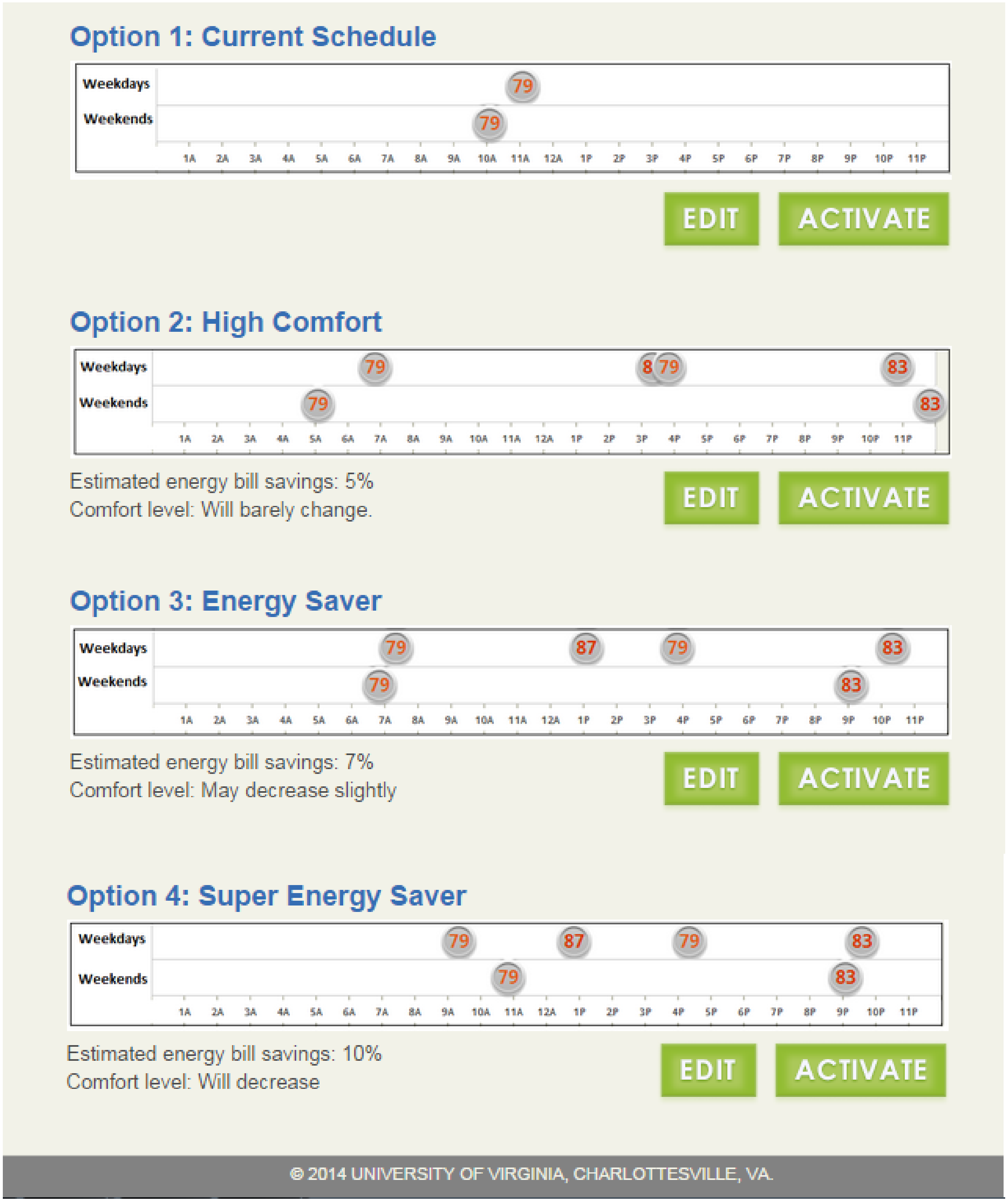

Figure 9: Recommendations For Home 15

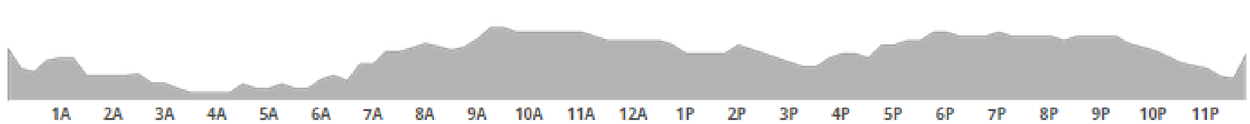

Figure 10: Occupancy Graph for Home 15

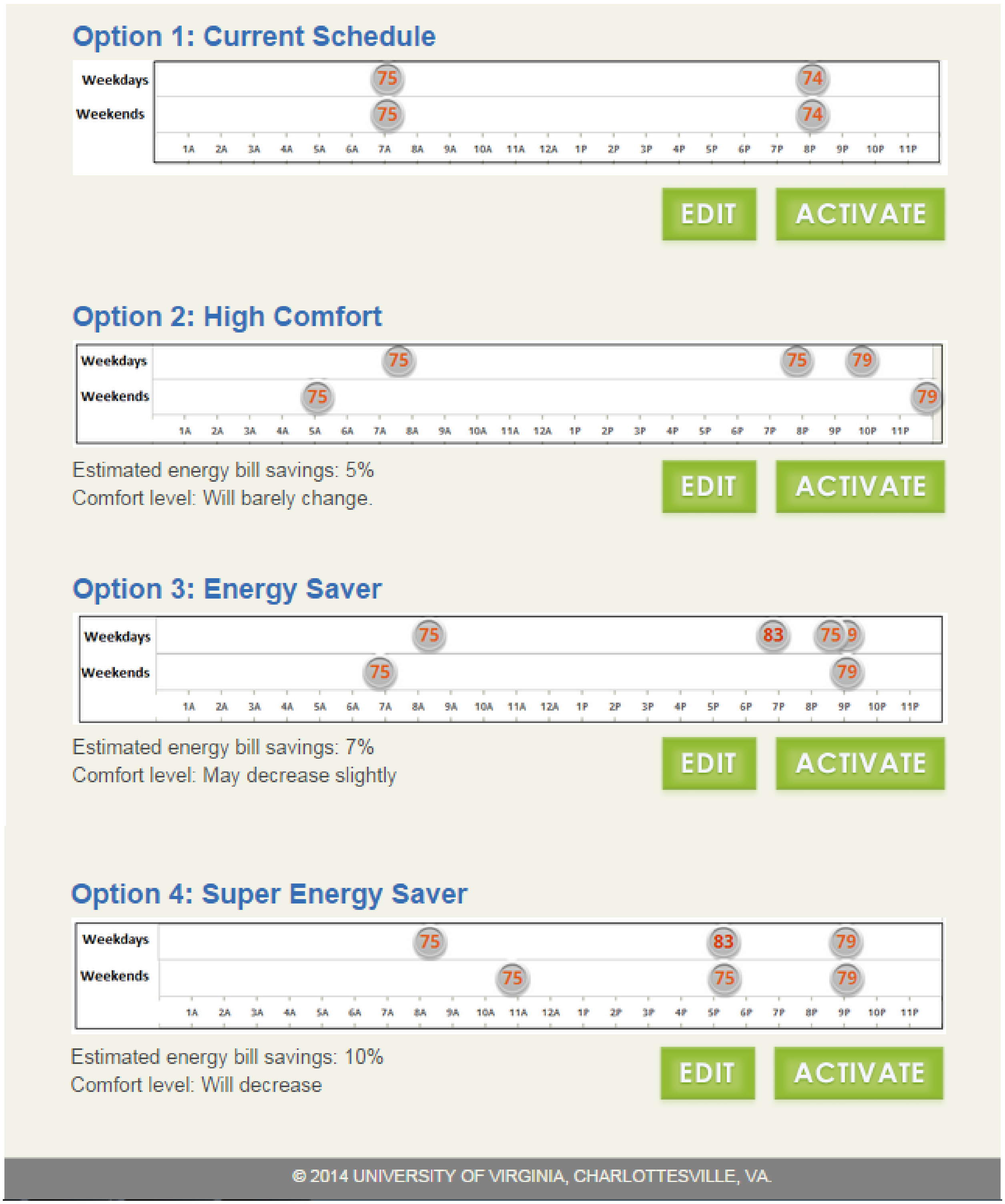

Figure 11: Recommendations For Home 18

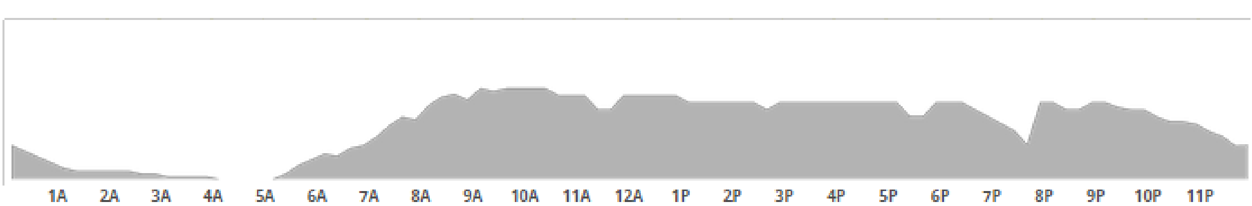

Figure 12: Occupancy Graph for Home 18

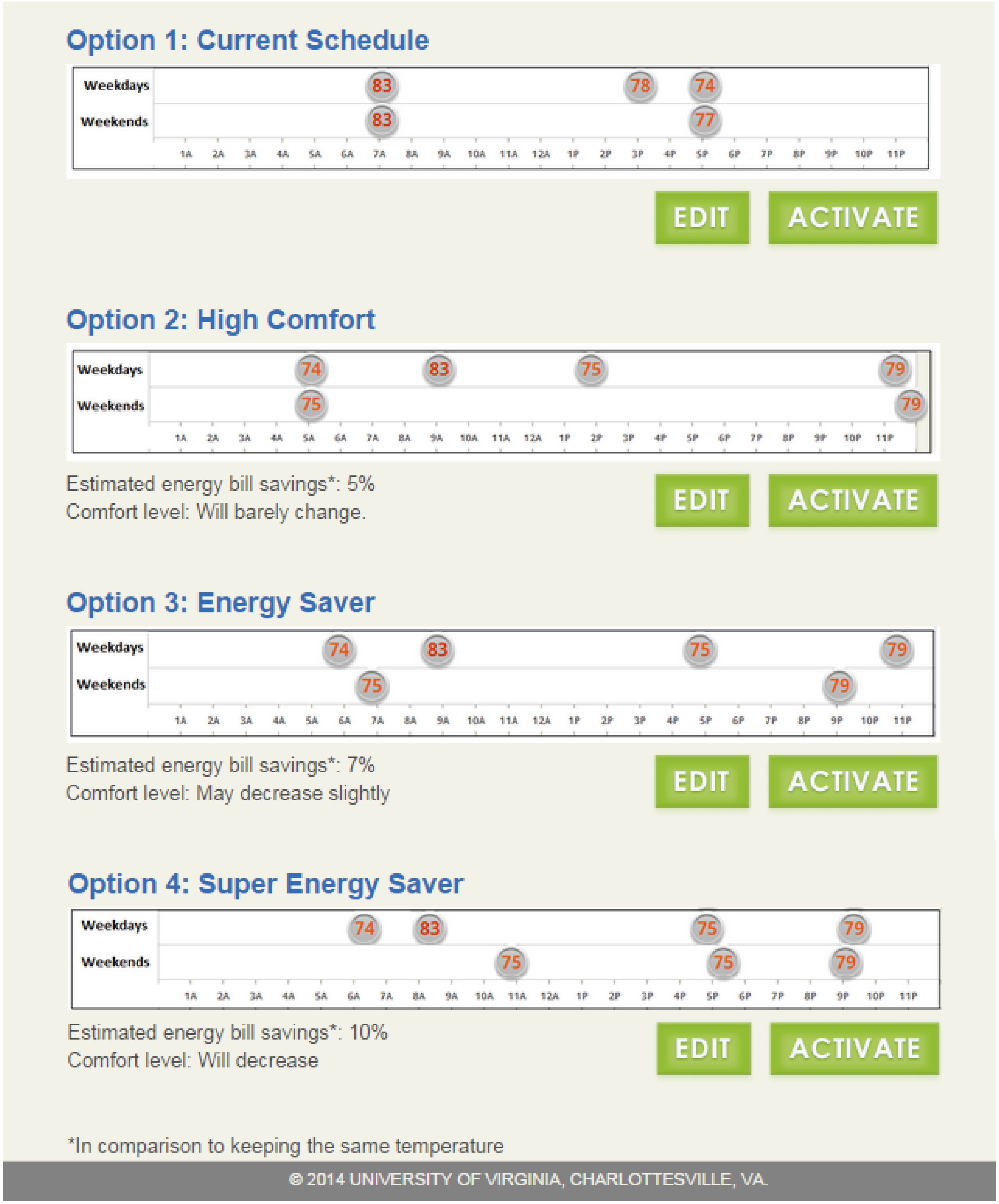

Figure 13: Recommendations For Home 20

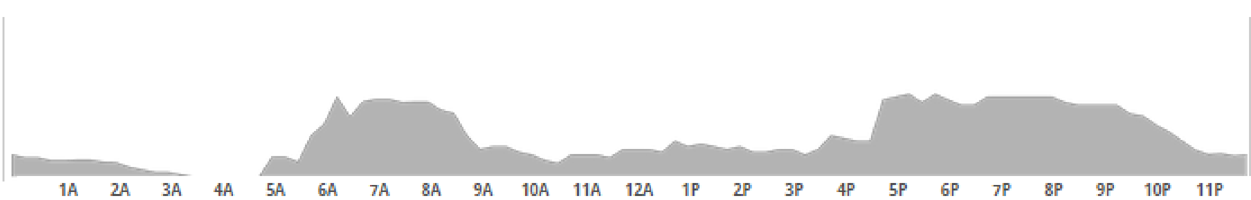

Figure 14: Occupancy Graph for Home 20

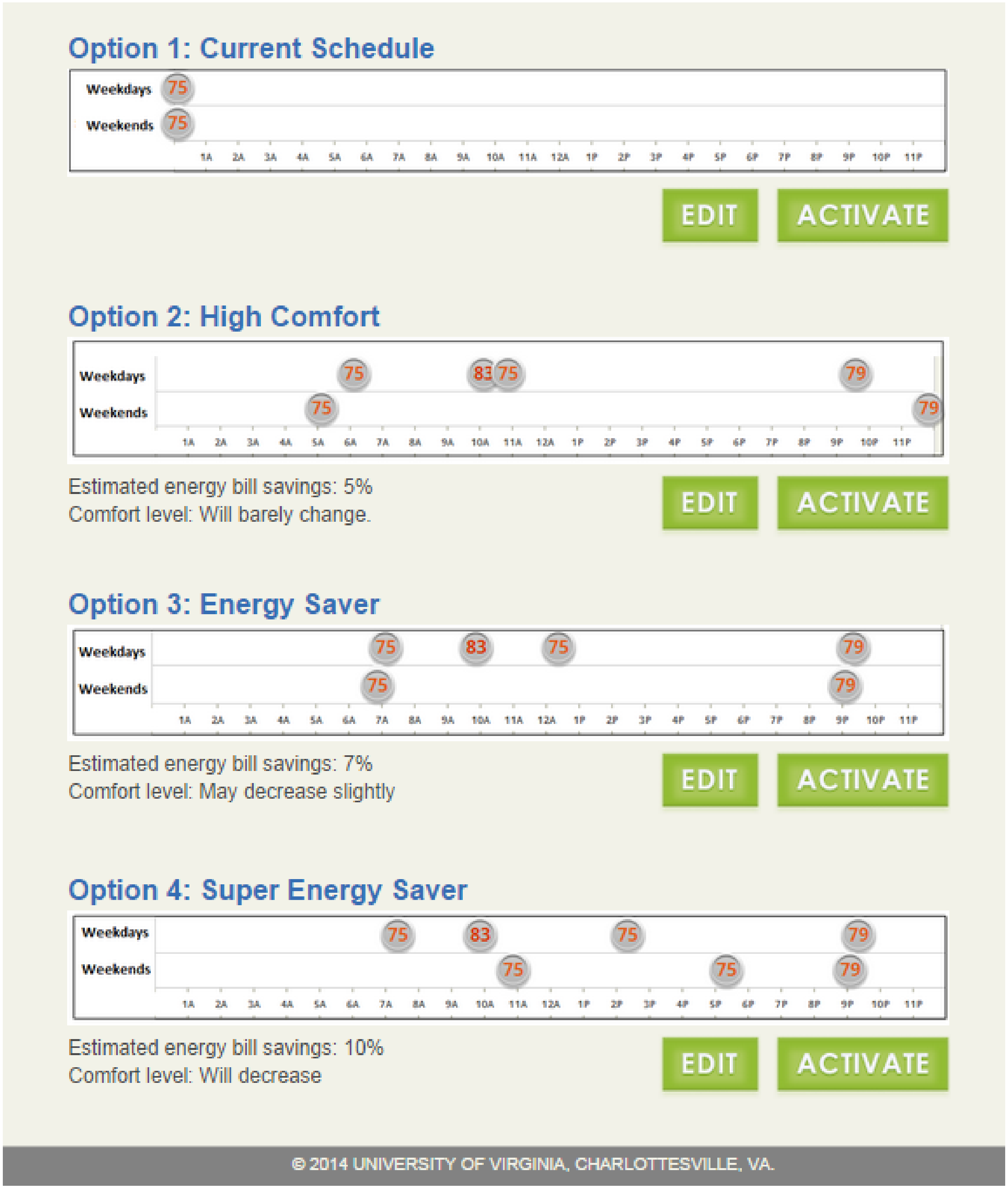

Figure 15: Recommendations For Home 22

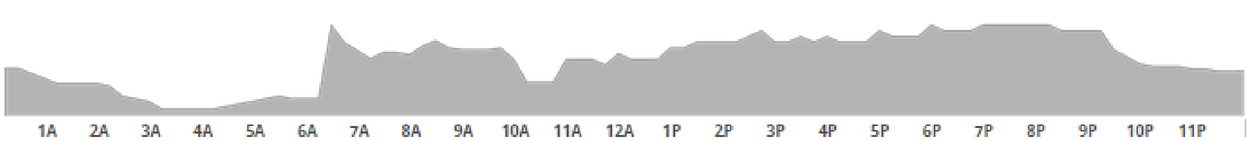

Figure 16: Occupancy Graph for Home 22

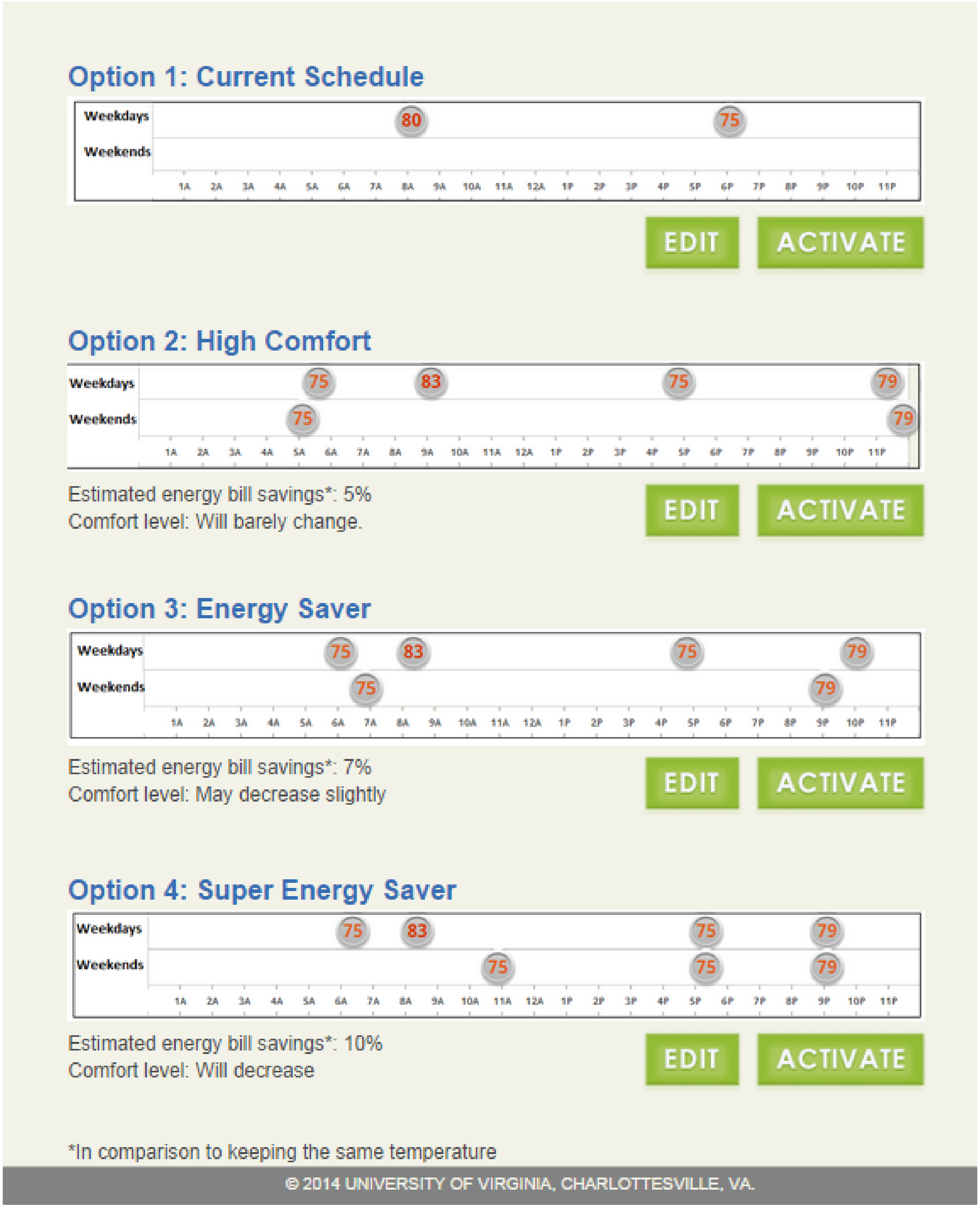

Figure 17: Recommendations For Home 25

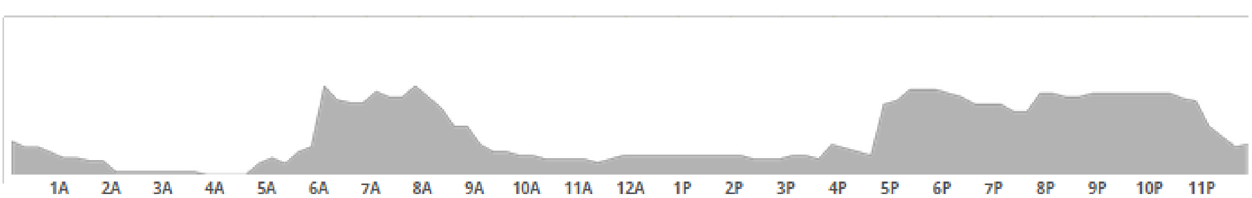

Figure 18: Occupancy Graph for Home 25

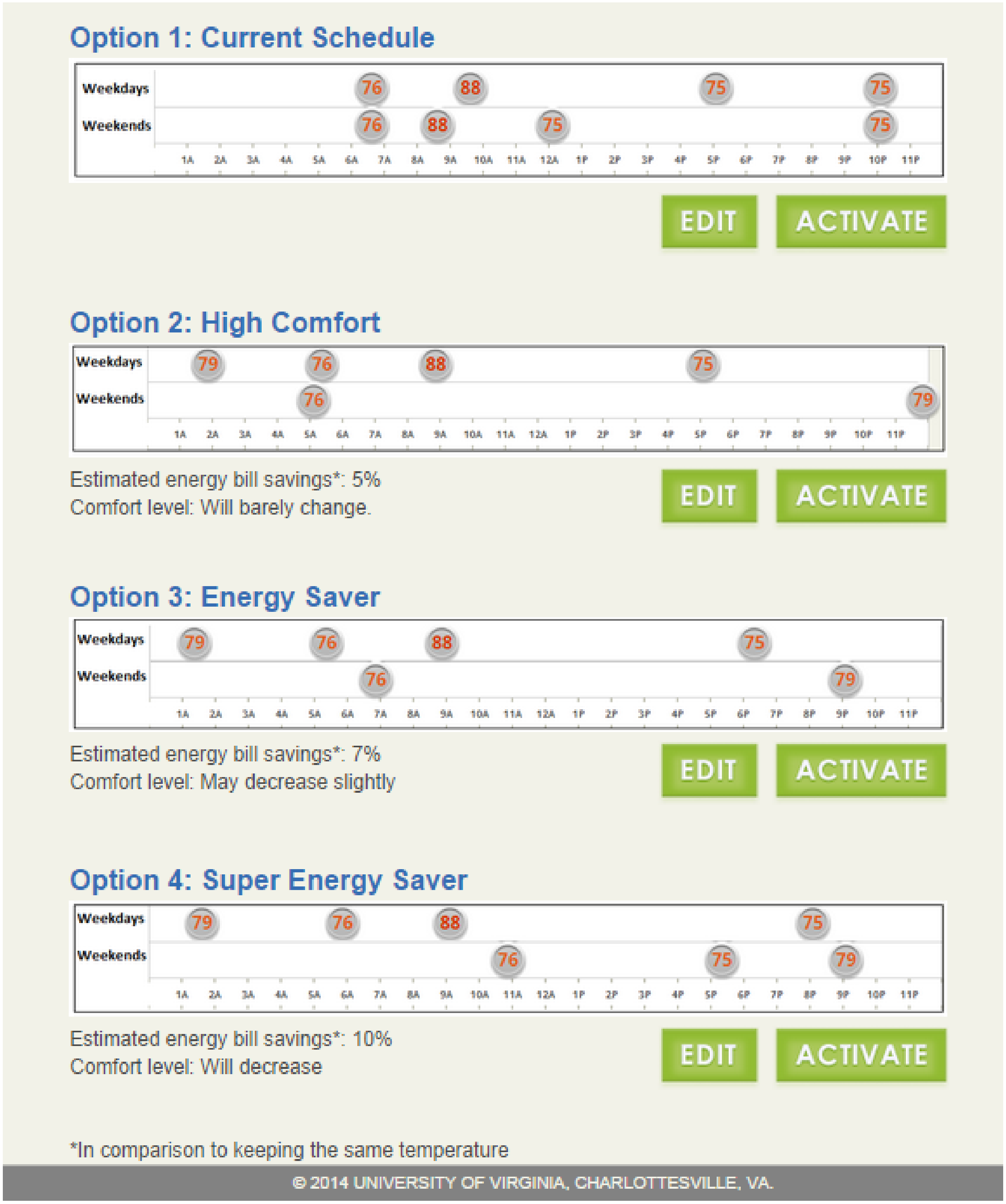

Figure 19: Recommendations For Home 28

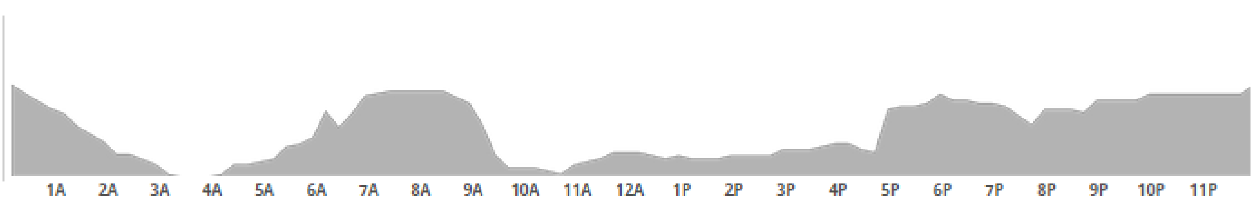

Figure 20: Occupancy Graph for Home 28

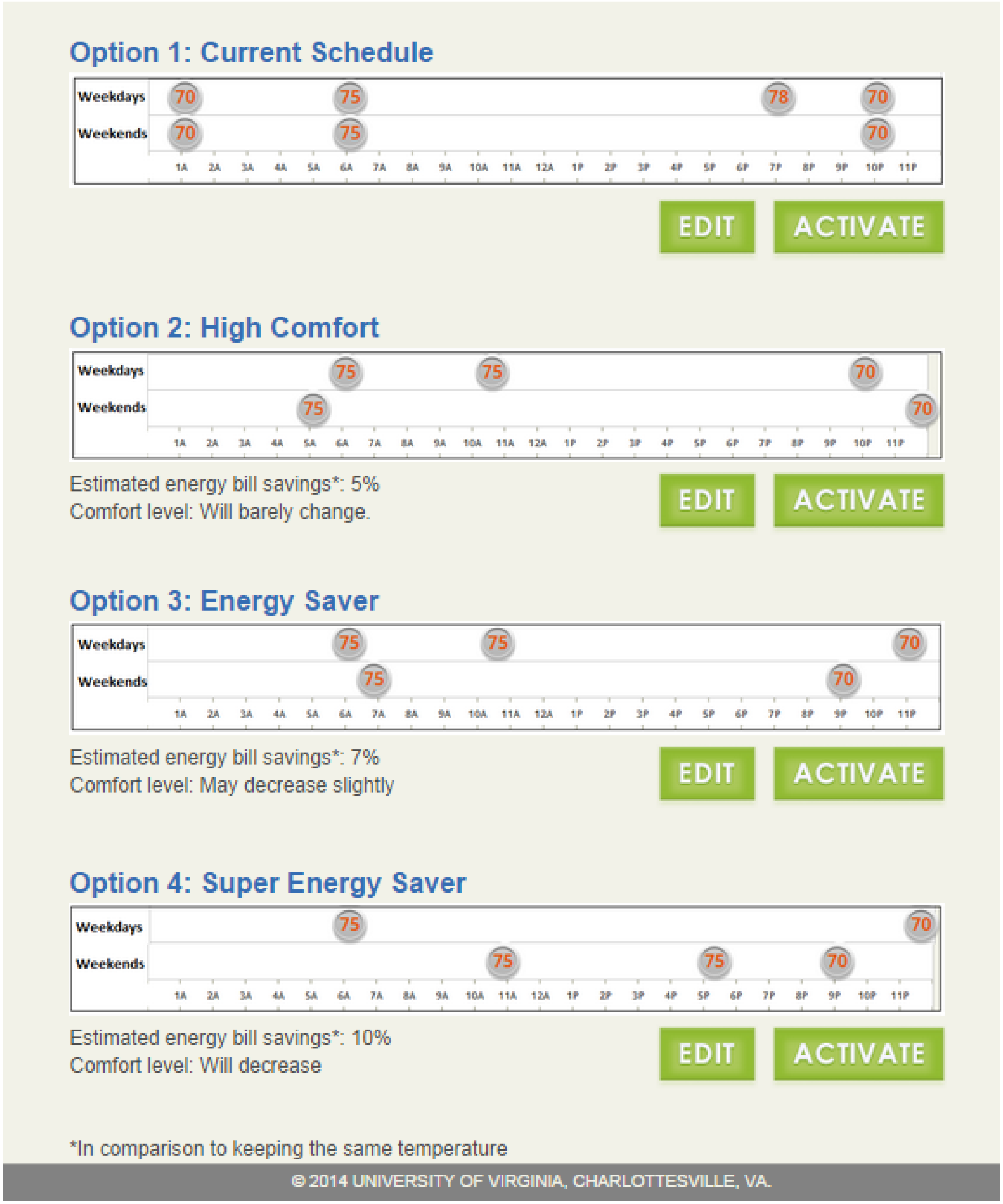

Figure 21: Recommendations For Home 31

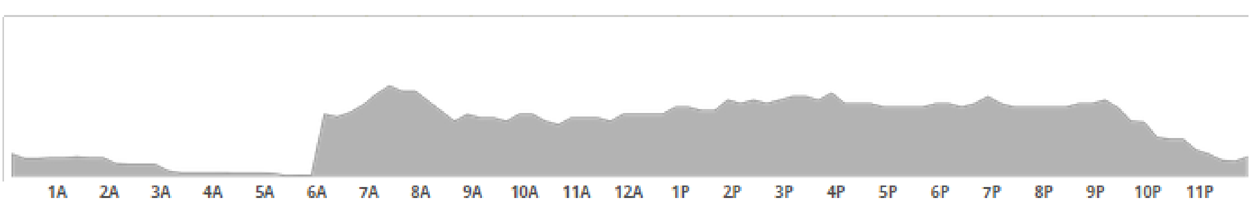

Figure 22: Occupancy Graph for Home 31

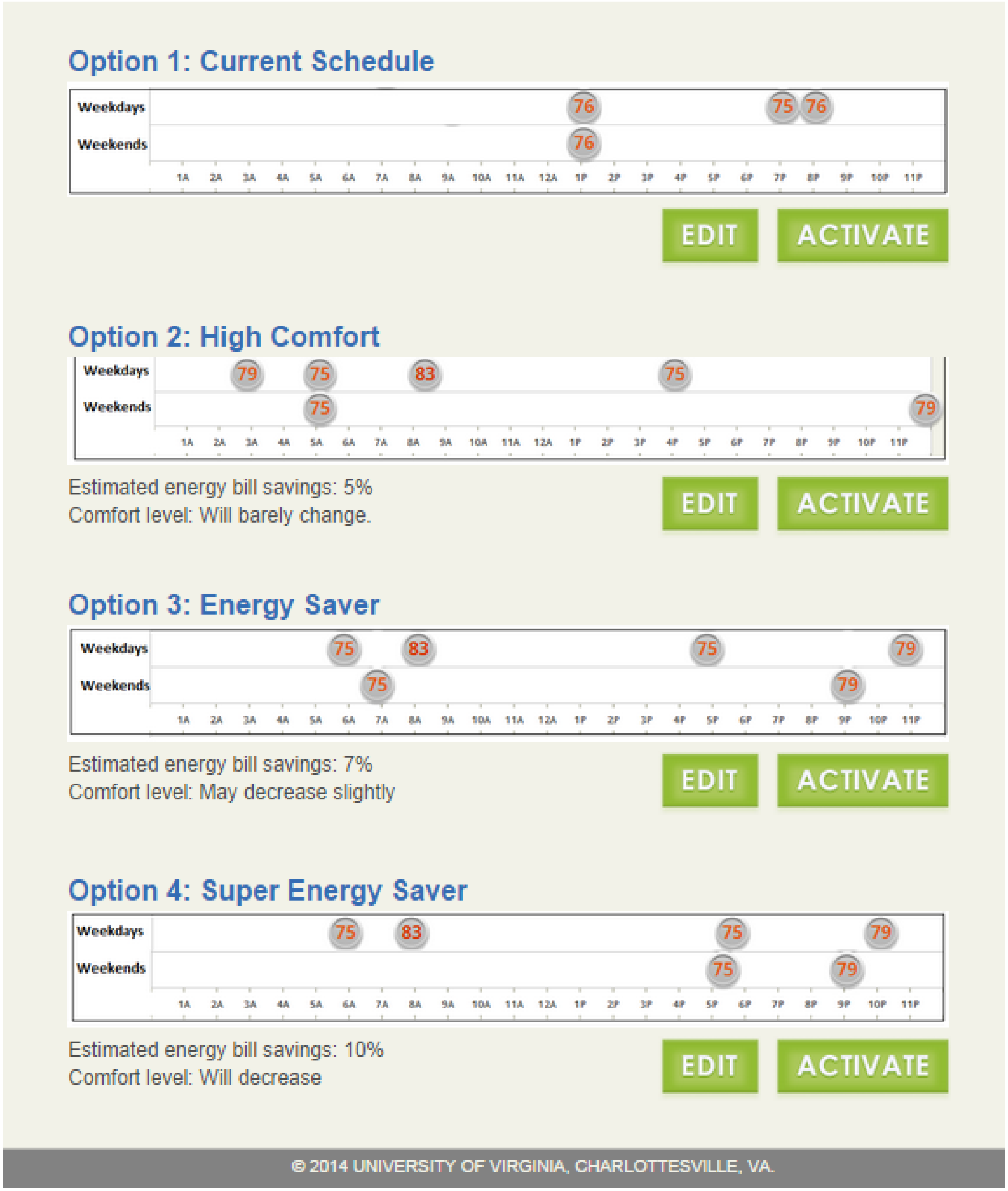

Figure 23: Recommendations For Home 36

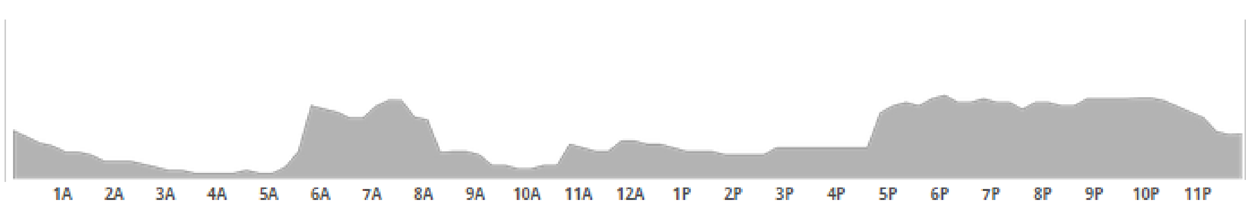

Figure 24: Occupancy Graph for Home 36

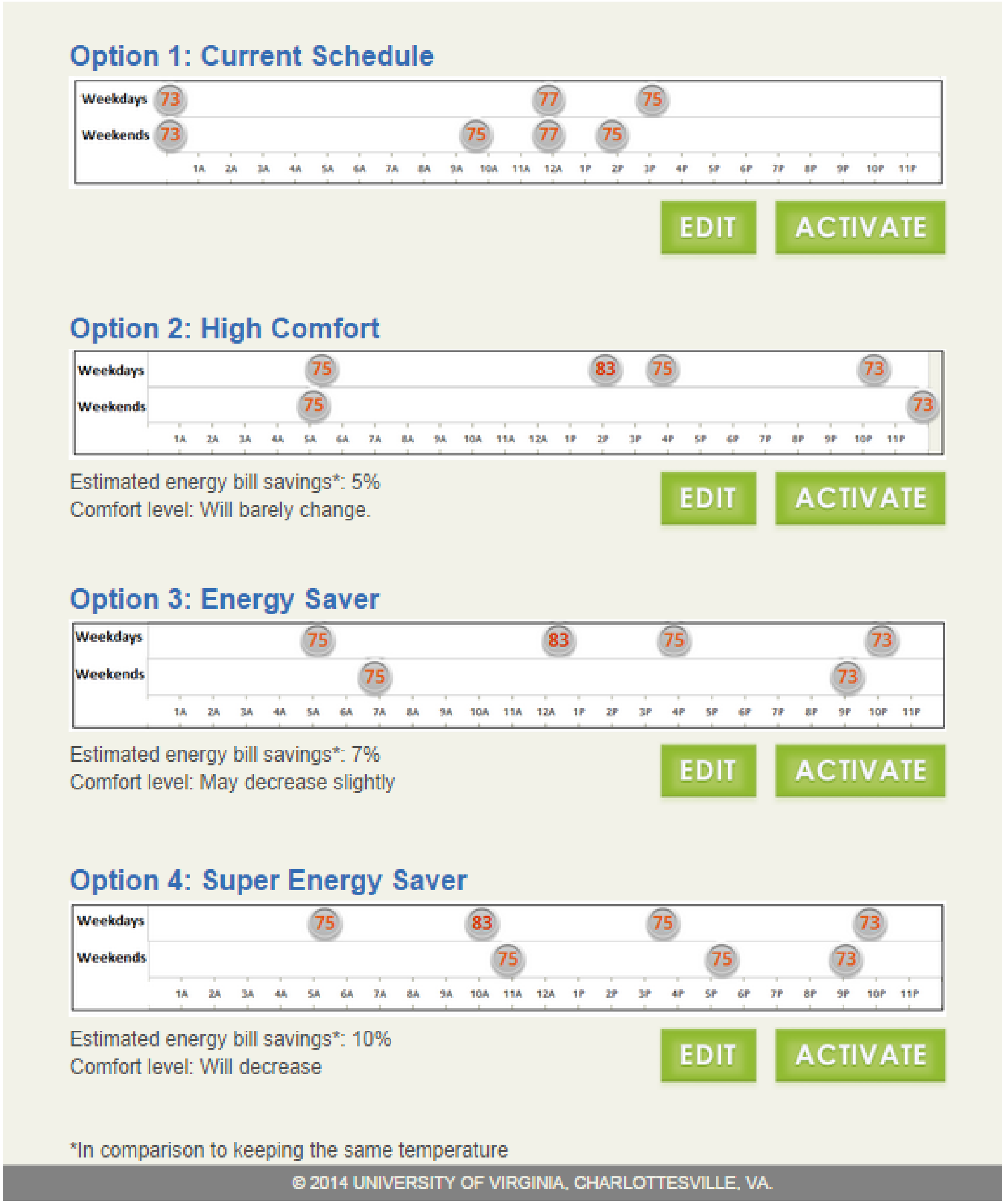

Figure 25: Recommendations For Home 37

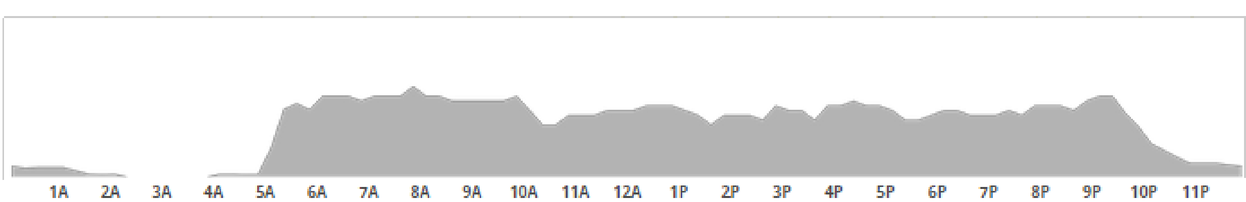

Figure 26: Occupancy Graph for Home 37

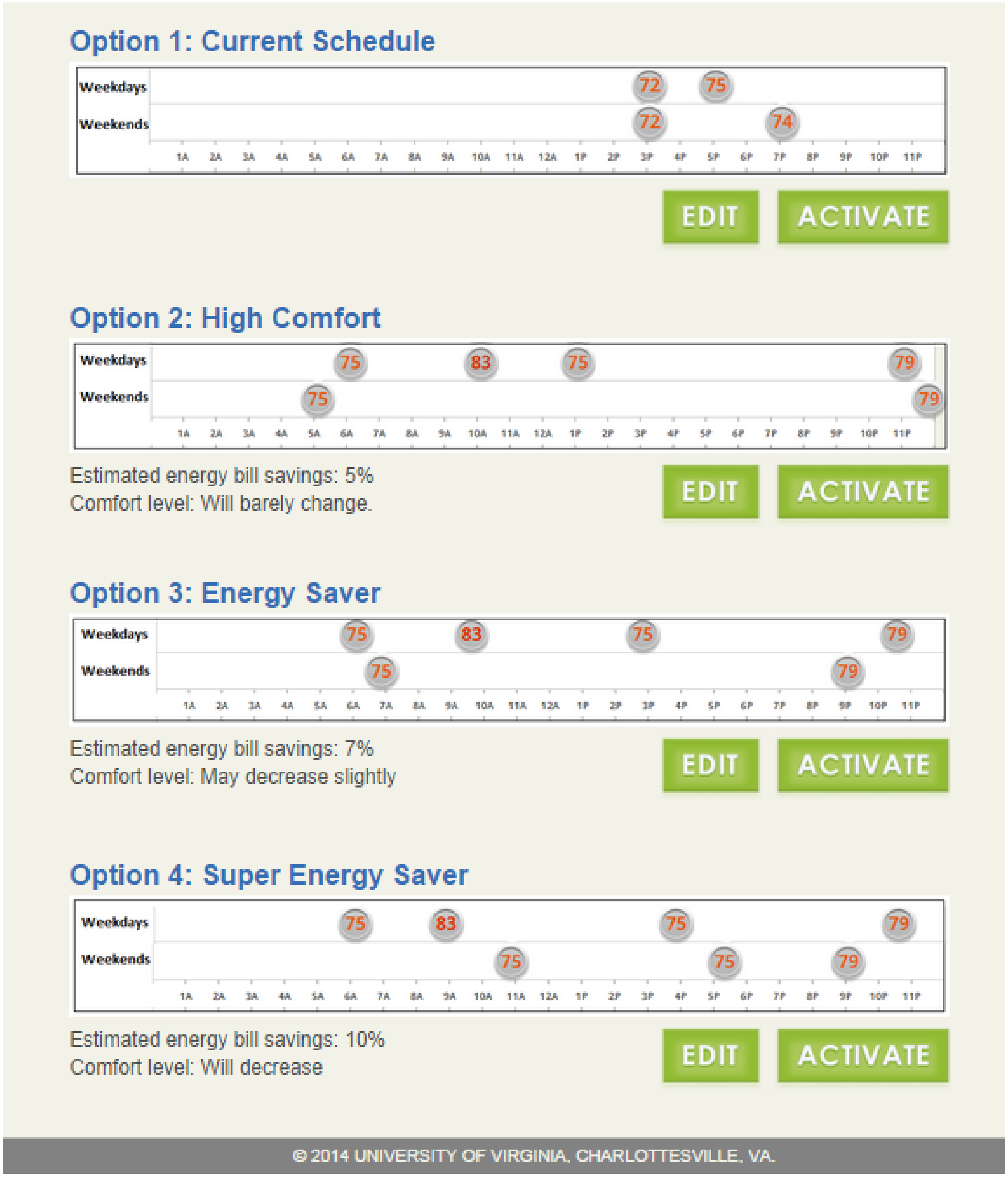

Figure 27: Recommendations For Home 38

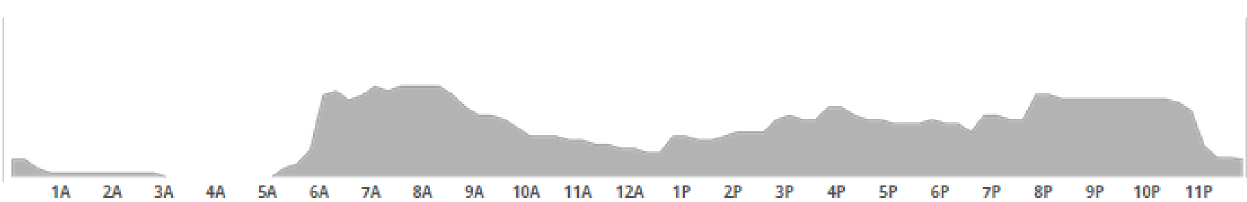

Figure 28: Occupancy Graph for Home 38