For office use only
T1 ______________
T2 ______________
T3 ______________
T4 ______________

Team Control Number
9247

Problem Chosen
B

For office use only
F1 ______________
F2 ______________
F3 ______________
F4 ______________

2018
HiMCM
Summary Sheet

Virtually all Americans today use some sort of heating or cooling system in their house, spending a large portion of their utility bill on regulating temperature. Consumers waste unnecessarily large amounts of money and energy when their system regulates temperature for an unoccupied house. Newer smart climate control systems attempt to learn users' schedules; however, they currently require many manual corrections to adjust to irregular schedules, and many people do not possess the time or patience for these adjustments. In this paper, we aim to address this issue and create a smart home model that will learn the user's habits and adjust the temperature appropriately, no matter how irregular their schedule.

The first part of our model deals with learning and predicting the user's schedule. Using GPS technology on residents' phones to track their leave and return as well as built-in features of the smart home system to detect when residents sleep and awaken, we are able to identify regular (consistently recurring) and irregular trips and sleep events. Every time the user leaves the house, the model attempts to match their leave time with known clusters of events with similar leave times and consistent return times. If that fails, it predicts a return time based on the durations of recent non-periodic events.

To test our model, we created a Python program that incorporated these algorithms as well as a set of test data. The test data included a variety of trips a person could take out of the house, including both periodic events (such as work) and spontaneous ones. **For our 10-week data sample, we were highly successful in predicting return times, obtaining an average error of 0.7 hours with over 65% of our predicted return times either early or on time.**

Next, we had to consider the amount of time needed for the system to reach comfort temperatures in anticipation of the user's return. We created an algorithm that calculates the amount of time necessary based on historical rates of change with the same outside temperature. From this we were able to predict when the system should begin changing the temperature based on the predicted return time and temperature outdoors.

Our model also accounts for geographical factors such as pollutants and allergens by increasing air circulation when air quality is low, taking advantage of the built-in filters found in vents throughout the house. The system also takes humidity into account, using the relative humidity to determine the actual temperature needed to make the home feel like the desired temperature. This also includes a system override when the relative humidity reaches dangerous levels, lowering or increasing the temperature accordingly so health is prioritized.

Finally, we expanded our model to account for multiple occupants or heating/cooling zones. Our model tracks the activities of each user separately through their phones' GPS and only changes the temperature if no one is home. The system then uses the earliest predicted return time out of all the occupants to bring temperature back to the desired level. For multiple zones, we include individual room sensors that monitor room occupancy to prioritize which rooms are set to the desired temperature.

Hands-Free Comfort, on Your Schedule

Introducing Enthalpy™, the smart home climate control system of the future

Durham, NC - November 19, 2018 - ComfortLabs Inc., a leading startup in smart home technology, announced today the release of its new Enthalpy™ smart home climate control system.

This next-generation system anticipates when users will arrive, go to bed, or wake, and automatically adjusts temperatures accordingly, even with multiple users and irregular schedules. It transitions seamlessly through a series of carefully designed modes to maximize comfort and savings for a hassle-free home climate control experience.

John Doe, co-founder and CEO of ComfortLabs Inc., stated, "Our Enthalpy™ climate control system is designed with the busy homeowner in mind. No matter how irregular their schedule, Enthalpy™ will recognize regular events and estimate the length of irregular events to flexibly adjust home temperatures and achieve the perfect balance of comfort, ease, and savings. We are proud to be on the forefront of smart home technology and release this new system for our users."

The Enthalpy™ system utilizes multiple built-in modes to regulate a house's internal temperature. It integrates geofencing technology with users' smartphones to accurately measure when all inhabitants have left the house, allowing for energy savings when users are gone, and it anticipates the user's return to bring the house back to a comfortable temperature. The Enthalpy™ system also supports automatic nighttime temperature adjustments, in addition to monitoring humidity, allergens, and air pollution to provide the most comfortable and healthy atmosphere for residents, integrating with existing humidifiers, dehumidifiers, pollutant detectors, or any other available systems within the smart home.

The Enthalpy™ system will be available for purchase in April 2019 on the ComfortLabs website and through select home security providers. The Manufacturer Suggested Retail Price (MSRP) starts at $199.99. More information is available at http://www.ComfortLabs.com/Enthalpy.

1 - Introduction

Air conditioners are used by 90% of Americans and cost homeowners $29 billion a year [1]. Even more Americans use home heating systems, which make up about 42% of the average homeowner's utility bill [2] [3]. Therefore, any means of increasing the efficiency of these devices provides a significant financial benefit for consumers, aside from the environmental boon of their decreased electricity usage. In order to reduce energy use while maintaining user comfort, the thermostat only needs to be set to the consumer's preferred comfortable temperature when there is someone in the house. Smart climate control systems currently in production are able to learn from the user's behaviors and initial manual changes to eventually implement automatic adjustments based on the user's schedule. In this study, we aim to model a smart climate control system that requires no manual changes other than the input of preferred temperatures. This system will allow temperature to vary when the user leaves the house and adjust the temperature appropriately when it expects their return.

Our smart climate system takes physical conditions including humidity and pollutants into account to prevent harmful impacts of extreme conditions on the homeowner as well as the house itself. The system also considers preferred temperatures in the morning and evening, keeping the house cooler when the user goes to bed and warmer when they wake up. Additionally, the system can be applied to homes with multiple inhabitants with different schedules and works regardless of the number of cooling/heating zones

To predict a person's arrival after they have left the house at any regular or irregular time, we created an algorithm that tracks all of their trips over time and optimizes the accuracy of the anticipated arrival time as more data is collected. Furthermore, we tested this model on an example schedule over the course of ten weeks to assess the precision of our algorithm to change the temperature in accordance with the user's preferences and routines.

2 - Restatement/Clarification of the Problem

In order to create the smart home system, we will specify the algorithm it employs to learn and predict users' schedules as well as the means by which these predictions are employed to determine when to begin temperature adjustments. Additionally, we will dictate how it takes into account factors like humidity, outside temperature, and allergens when adjusting temperature or air circulation. Finally, we will address how our model can be extended to encompass dwellings possessing multiple regions whose temperatures can be controlled independently, as well as dwellings with multiple occupants.

Essentially, our model needs to be able to do the following:

1. Anticipate when a person will return after they leave.
2. Adjust temperature while a person is absent, but leave enough time to reach a comfortable temperature before they return.
3. Predict when a person will fall asleep and wake up and adjust temperatures accordingly.
4. Consider and react to factors unrelated to temperature like humidity and air quality.
5. Function with varying amounts of people in the home and number of heating/cooling zones.

3 - Assumptions with Rationale/Justification

Assumption 1: *Our smart home system has access to the lights, appliances, and measurement devices (such as thermometers and hygrometers) in the home as well as a thermometer outside.*

We know the house must have access to thermometers and hygrometers in order to adjust the temperature and humidity, but we must assume that it also has access to the lights and appliances as the glossary definition of a smart home suggests. Our system utilizes the use of lights and appliances as indication that users are awake in the morning and asleep at night.

Assumption 2: *All residents of the home possess smartphones and consistently take them along when they leave the home*.

Without making this assumption it is hard to determine if the house is occupied. We do not rely on other components such as lights and motion sensors because someone could be at home without moving enough to trigger the motion sensors or turn on any lights, or there could be pets moving around in the house. While this assumption will introduce error into the model, these ideas should hold true for most households. On average, children now get their first smartphone at the age of 10.3 [4] and children under that age typically are not left home alone, so even younger children should be covered under the model.

Assumption 3: *Residents of the home are asleep when all lights and appliances are off, and no motion has been sensed for at least 10 minutes.*

In order to track sleep events, our model needs to determine when users have gone to bed. While this assumption fails to consider people sleeping with small lights or appliances such asf the TV on, it removes the need for people to input their sleep schedule or indicate to the system they have gone to bed.

Assumption 4: *If a resident consistently wakes up within the same interval on particular days, they have an alarm set and need to wake up at the same time on those days*

Since people like sleeping at cooler temperatures [5], our system needs to adjust the temperature for when they are waking up. To predict when users are waking up, the system will determine whether an alarm is set or not. If an alarm is set, which we ascertain based on wake-up times consistently within a five minute interval, then the user will always wake up around the same time, and our system can adjust the temperature for that specific time.

Assumption 5: *If a resident does not consistently wake up within the same interval on particular days, they do not have an alarm set and do not need to wake up at a specific time on those days*

If the user does not wake up at a consistent time, then we assume the amount of time they sleep is more important than the time they wake up. Instead of adjusting the temperature for a specific time that morning, such as days where they have an alarm, the system finds the average amount of time they sleep on those days. Then, the system will adjust the temperature based on the time they went to bed and the amount of time they will be sleeping.

4 - Model Design and Justification

4.1 - Anticipating Return

The first part of our model is focused on predicting when the user will return given any departure time. Our smart climate system tracks the user with their smartphone using geofencing to determine when the user leaves the house. The model is based on a system of stored events. The departure time and arrival time of each “leave occurrence” are stored in the system.

In order to predict return times, our system stores the data on previous outings; specifically, for each outing, we record its starting time, ending time, and day of the week. These data for each outing are stored in “events”, representing collections of occurrences which are hypothesized to be instances of some recurring event; we differentiate somewhat arbitrarily between events containing at least three occurrences and those with less than three; the former are referred to as “regular events”, because the evidence is fairly compelling that they actually represent a recurring event, whereas the latter are “irregular events”, as there is insufficient data to determine whether the event us truly repeating.

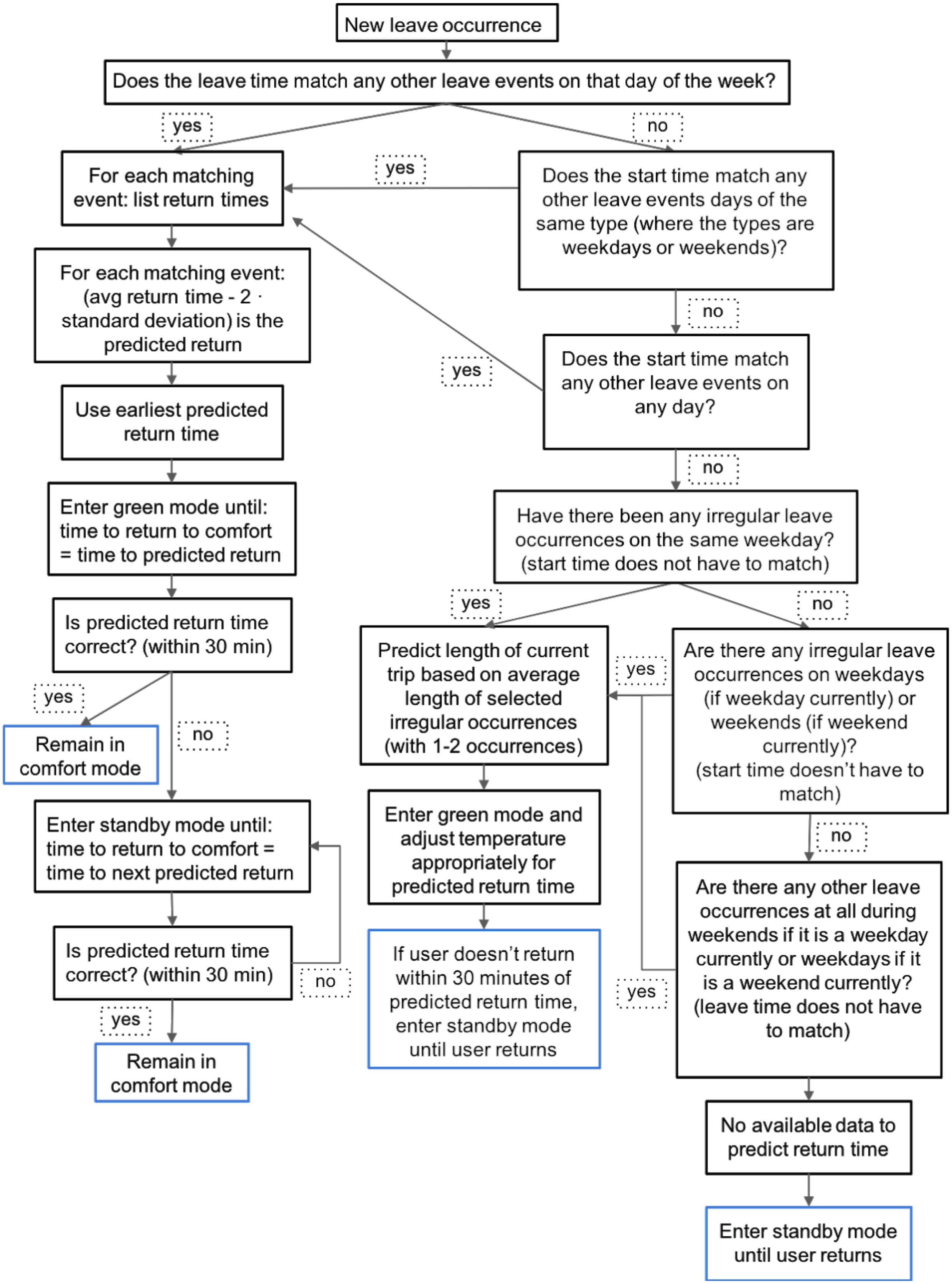
When a new leave time is recorded, it is compared to leave times of existing events in an increasingly broad search; first it is compared with events containing at least one occurence on the same day of the week as the leave time; then those containing at least one occurrence on a day of the same type, either weekends or weekdays as is appropriate, and finally with all recorded events, irrespective of day. In each case, for each event meeting the criteria, it is determined whether the new leave time falls within two standard deviations of the average of the leave times of the occurences making up that event (ignoring the standard deviation and substituting 0.25 hours, if the event contains less than three occurrences). During each of these three stages if we find any events which fulfill this, we take those events as the ones we will use to generate return times do not move on to any of the broader-search stages that would otherwise succeed it.

If no matching event is found in any of these stages, we move to a still less stringent selection process. Since no events have leave times acceptably close to the new one, we predict the return time based on the average duration of recent irregular events. Again, we proceed in three stages, first looking for events containing occurrences on the same day, followed by those containing occurrences on the same day-type, finally extending our net to all events. The major difference is that we are now considering only irregular events, that is, events with one or two occurences. We disregard regular events because we consider it unlikely to be the case that this new leave time represents an instance of an existing regular event, but at a completely different time than is usual for that event. If any irregular events are found at any stage of this process, we do not continue on to the additional stages, but advance to predicting the return time based on the durations of these past events.

For each event we ended up with in the previous step, we predict a return time independently. If we are predicting based on return times, we do as follows: if the event has only one data point, we simply predict that time; otherwise, we take the average of the existing data minus two standard deviations, so as to err on the side of returning to comfort mode unnecessarily early rather than unpleasantly late. If the new leave time matches multiple events, a return time will be predicted for each event. This collection of times is then input to an algorithm described elsewhere for determining which modes should be employed at different times.

In the other case, in which no past events with sufficiently close leave times were available, we simply take the average of the durations of the irregular events we gathered in the last step. We predict the return time as the average duration plus the observed leave time.

Figure 1. Diagram depicting algorithm used for predicting return time of a new leave occurrence



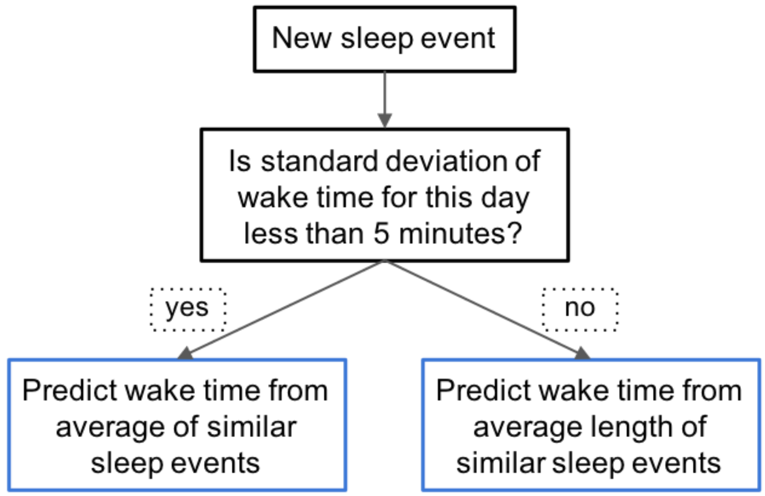


Figure 2. Algorithm used to determine wake time for a new sleep event.

We categorize sleep events into two types based on whether we think the user usually sets an alarm for the next day or not. Without looking at alarm data from the user's phone, to which we would likely not have access, we determine if an alarm is set based on the standard deviation of the times that they normally wake up the next day. We set the boundary at 5 minutes, so if the standard deviation of the times that they wake up is less than 5 minutes, then we assume they have set an alarm and will be getting up at that time the next day.

The motivation for choosing five minutes comes from probability considerations. Since we cannot assume that the distribution of times is normal, we use Chebyshev's theorem, which is true for any probability distribution we might feasibly encounter, and states that at least $1 - \frac{1}{K^2}$ of the data must be within K standard deviations of the mean. Using $K = 3$, we find that at least 89% of the data must be within 3 standard deviations from the mean. This led to our choosing 5 minutes because it creates a 30 minute range ($3 \cdot 5 = 15$ +/- the mean) in which at least 89% of the wake up times must occur. If someone usually indicates that they have woken up (via turning on lights or triggering motion sensors) within the same 30 minute time period, then they likely have a consistent alarm set for that day of the week. However, we use one standard deviation in determining whether an alarm has been set to increase the precision of our predicted wake time.

If the user usually has an alarm set on that day, then we assume they will wake up at that time, regardless of when they go to sleep. However, on days where the user normally doesn't have an alarm set, then it is hard to predict exactly what time they will wake up. Therefore, a better measure for those days is the average length of sleep. If they do not set an alarm then they will sleep as long as their body needs or until they are disturbed. Thus, on days where our model does not think that they have an alarm set, we average the length of time they normally sleep on those days and add it to the time they went to sleep in order to predict when they wake up.

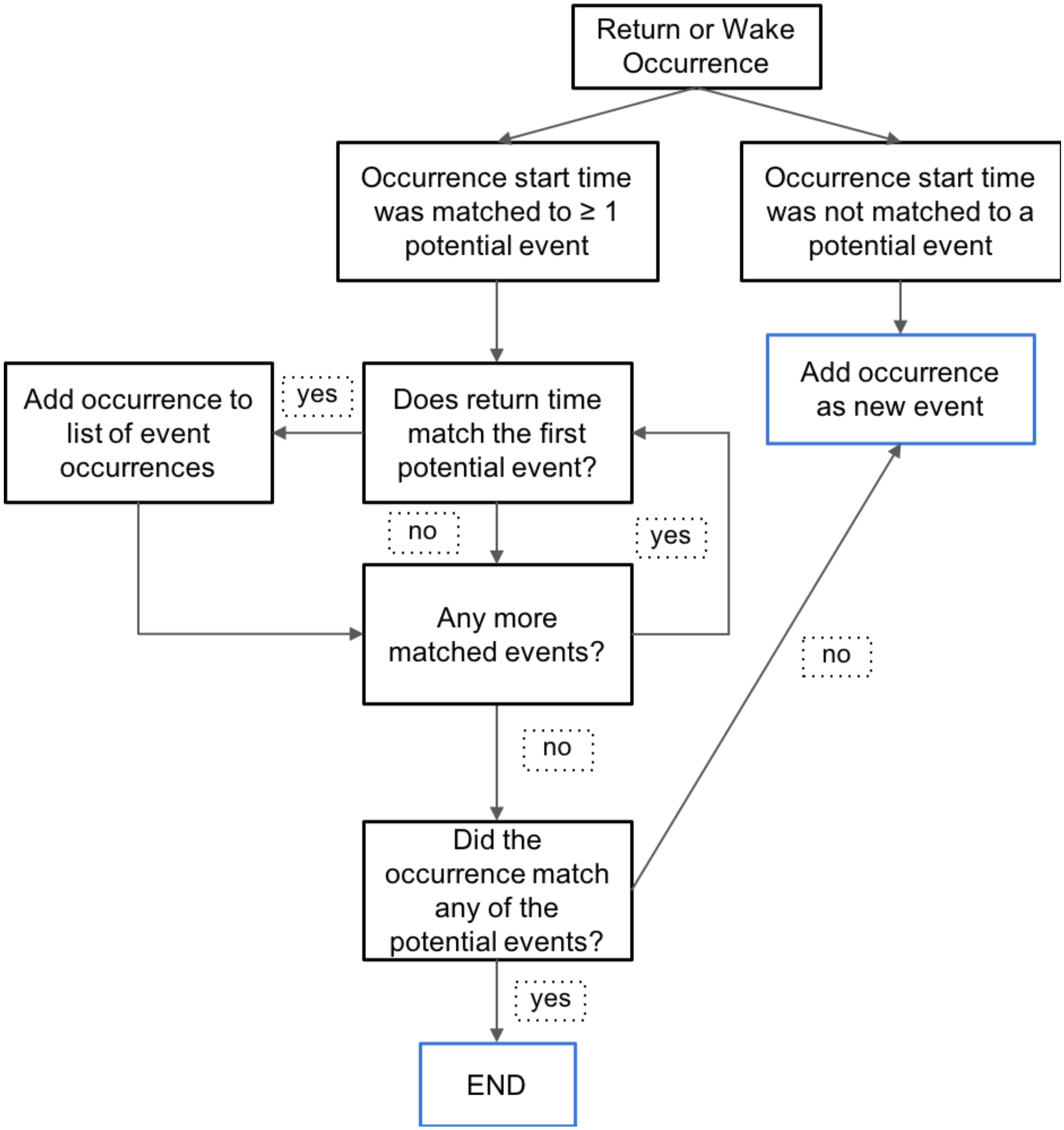


Figure 3. Diagram of the big-picture algorithm used to classify an occurrence once the user has returned.

4.2 - Temperature/Time Determinations

In order for the system to determine how long it will take to attain the comfort temperature given the current internal temperature, the system stores in its memory recent pairs of times and temperature changes and uses these to approximate the time required. Temperature transition data is stored according to the outside temperature at the beginning of the transition. For each interval of 2°F, we store the five most recent heatings and five most recent coolings that had an initial outside temperature in that range. Whenever a new heating or cooling is completed, if it occurs entirely within one of the intervals, the oldest heating or cooling respectively for that period is replaced by the duration and change in temperature of the new event.

To predict the time necessary to effect a temperature change under the same outside temperature as some previous temperature transition with a known temperature difference and duration, we assume that, with heating and cooling events considered separately, the time our system takes to carry out a temperature change is proportional to the magnitude of the temperature change (assuming a constant outside temperature). Support for this assumption is provided in **4.3**. Under this assumption, considering two temperature transitions with identical outside temperatures, the first having temperature difference ΔT_1 and taking time t_1, while the second has temperature difference ΔT_2 and takes time t_2, we then have

$$t_2 = \frac{\Delta T_2}{\Delta T_1} \cdot t_1$$

and so t_2 would be the duration predicted for a temperature change ΔT to occur based on a previous transition in which a temperature change of ΔT_1 required the duration t_1.

However, because the outside temperature does vary, we do not compute the expected time of the current temperature change from only one prior temperature transition. Rather, we use the average of the predicted times given all the heat or cooling events, as is appropriate, which fall within the same outside temperature interval as the new event. If there is no past data at all for any similar outside temperatures, a default rate of change of 1°F/18 minutes, based on [6], is employed to estimate the duration needed to return to comfort mode.

4.3 - Justification for Linear Heating and Cooling

To predict the time necessary to attain comfort temperature from the current internal temperature, we based our computations proportionally on an average of recent temperature transitions. With these proportional calculations, we assume that (under a constant external temperature) the relationship between temperature change ΔT and duration t is linear: regardless of the initial or final temperature, the same change in temperature will always take the same amount of time, and a smaller or larger change will scale proportionally.

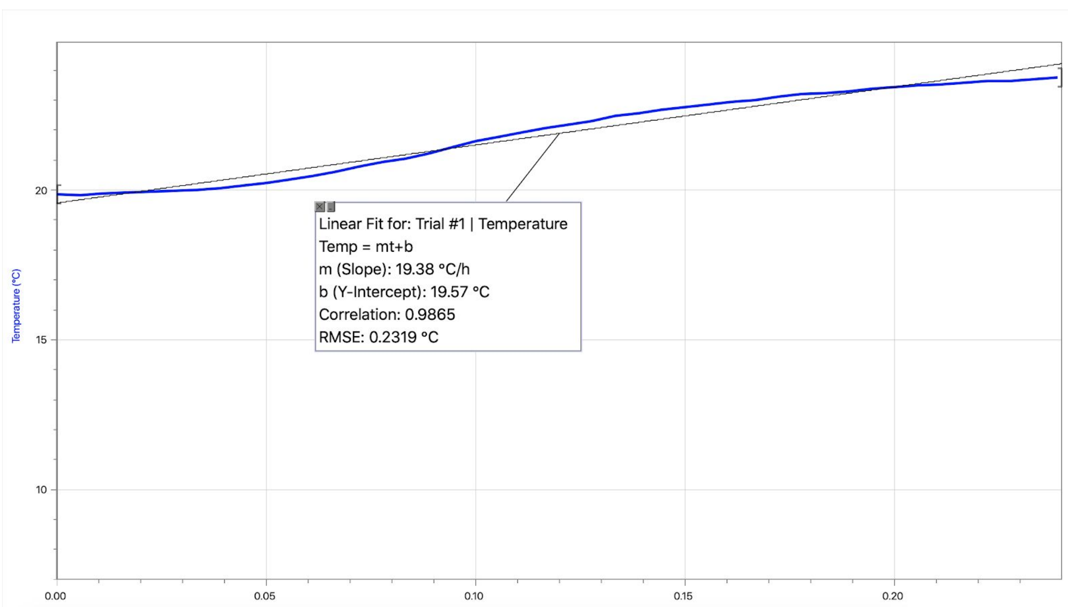
We began with solely considering the temperature change affected by the air conditioner or heater. Based upon the thermochemical equation $q = mC\Delta T$, we know that for the constant mass of air (m) and specific heat of air (C) within a house, the change in temperature (ΔT) is proportional to the amount of heat required (q). Since an air conditioner or heater will be manufactured for a specified constant heating or cooling output per second, given in Watts (W), dividing the amount of heat required (q) by the heat produced per second (W) gives the total time to effect a given change in temperature (ΔT), as follows:

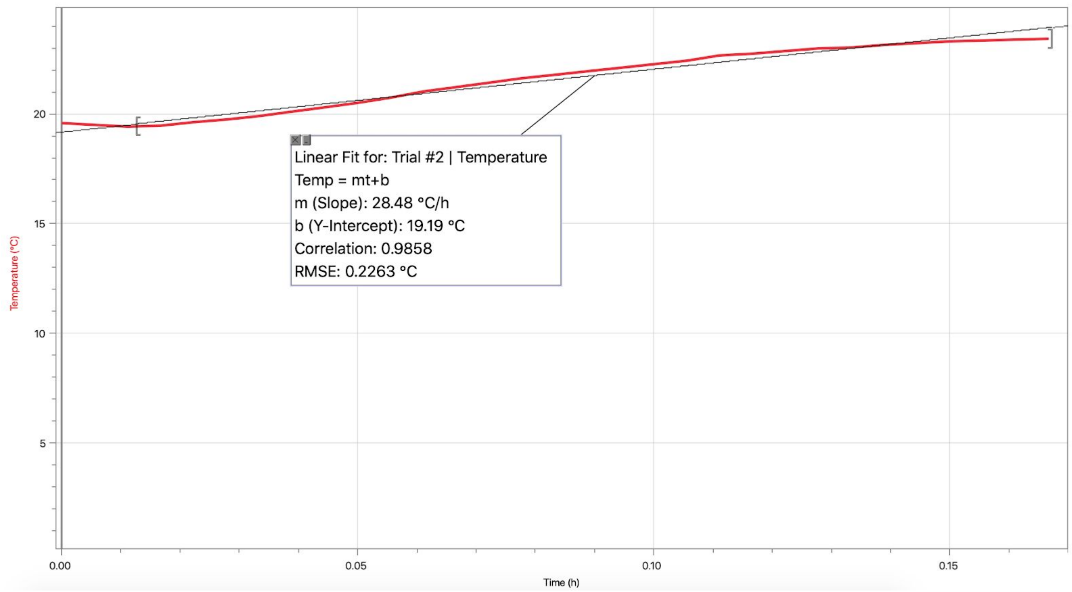
$$t = \frac{mC\Delta T}{W}$$

Thus, the time necessary to effect a given temperature change forms a linear relationship, where time (t) is the product of change in temperature (ΔT) and a set of constants associated with the amount of air in the room and the power of the heater or air conditioner.

However, this calculation assumes that no heat is lost or gained from any external source. In a home, some heat will be lost or gained to the outdoors regardless of how well the house is insulated, and heat may also be gained from sources such as fireplaces, space heaters, and kitchens. Since fireplaces, space heaters, and kitchens remain unused while a home is empty, our main additional consideration was heat lost or gained to the outdoors.

In order to test whether the heating or cooling of a room remains linear while accounting for heat lost or gained to the outdoors, we conducted two experimental trials. We used Vernier temperature probes to record data in LoggerPro as we turned a heater on its “HIGH” setting and allowed it to heat a room. The following graphs of data from our trials each demonstrate positive linear correlations of 98.6%, supporting the claim that the relationship between temperature and time will remain very nearly linear linear even with heat lost or gained to the outdoors.





Figures 4 and 5: Heat gain over time from an HVAC system in an experimental setting. The approximately linear data supports our assumption that heating or cooling time can be calculated in proportion with previous heatings or coolings.

4.4 - System Modes

Our model uses a variety of modes to accommodate the needs of the user in various circumstances.

Comfort mode - The function for this mode is simple: to keep the house at the normal desired temperature set by the user. While the system is in this mode it functions like a typical thermostat. The system enters this mode when anticipating the user's return and remains in this mode while people are home.

Green mode - This mode was created in order to save the user energy and money by maintaining minimal function of the system when it is not needed. By tracking the location of the user's phone, the system determines when the user leaves the house, prompting it to enter green mode. In green mode, temperature boundaries are set at 10 degrees Fahrenheit above and below the desired comfort mode temperature (these boundaries can be edited based on preferences). Then, the thermostat turns off until the temperature reaches one of these boundaries. Once the temperature has reached one of the boundaries, the thermostat will turn on to keep the temperature from changing further. It then remains in this state, continually calculating the amount of time it would take to return to the comfort mode temperature through the method described above in **4.2** and comparing that to the expected amount of time until any occupant of the house returns home. If the former is greater than or equal to the latter, the system begins returning to comfort temperature.

Standby mode - Standby mode was created as a compromise between savings and comfort. This mode functions in the same way as green mode, but the boundaries are set only 5 degrees Fahrenheit above and below the desired temperature. This mode is used when the system encounters an irregular event and the user does not return within 30 minutes of the predicted time. Since the system is already in comfort mode and given that the user could be back at any time, from minutes to hours later, there are three options.

1. Return to green mode until the user's return, which saves money but would make the temperature uncomfortable when they return.
2. Remain in comfort mode, which wastes energy and money but ensures that they will be comfortable when they get back.
3. Enter standby mode where the system has a chance to save energy by not working at full capacity, but the boundaries are closer to the desired temperature so that it won't take as long to get back to comfort mode when the user does eventually return.

We chose for our system to use this third option, a compromise between the other two.

Sleep mode - Most people like sleeping at a lower temperature than the temperature during the day. To account for this preference, our model includes a sleep mode to adjust the temperature while residents are asleep. The system tracks what time the user goes to bed and wakes up in the morning based on a number of factors. The system determines the user is asleep when all lights and appliances are off, and there has been no motion for at least 10 minutes. Then, the system determines that a person has woken up when a light is turned on or an appliance is used in the morning. By tracking when the user goes to bed and wakes up, it can determine when to enter sleep mode and when to return to comfort mode.

Vacation - This is not a mode, but rather a consideration for when the user goes on vacation. Since a majority of vacations are almost completely unpredictable in timing and length, it would be extremely difficult for our model to foresee a vacation event. Therefore, we include an option for the user to input when they are going on vacation and for how long, so our system can enter green mode for that period of time.

4.5 - Humidity Control

Humidity is a very important condition to maintain, because it endangers the user's health if it reaches extreme levels. When humidity is too low, it can make skin dry and itchy, damage wood furniture and floors, and the threat of bacteria, viruses and infections increases. On the other hand, if humidity is too high, it can feel uncomfortable, promote mold growth, and also increase the threat of bacteria, viruses, infections, mites and fungi [7].

The following chart shows the suggested relative humidity based on outside temperature [8].

Outdoor Temperature (T)	Suggested Relative Humidity (H)
T > 50 °F	H < 50%
50 °F > T > 20 °F	H < 40%
20 °F > T > 10 °F	H < 35%
10 °F > T > 0 °F	H < 30%
0 °F > T > -10 °F	H < 25%
-10 °F > T > -20 °F	H < 20%
T < -20 °F	H < 15%

Table 1. Ideal relative humidities based on outdoor temperature.

Humidity is also an important condition to consider when varying temperature, because it affects what the temperature actually feels like. There are various products that can control humidity, from whole-house humidifiers and dehumidifiers to small or single room products. Our system adjusts to humidity in different ways depending on what it has access to:

Humidifier and Dehumidifier (Whole-house)
With access to a whole-house humidifier and dehumidifier, the system has complete control over humidity. The house tracks the temperature outside, then sets the humidity inside accordingly. Once it has humidity held constant by using the humidifier/dehumidifier, the system changes the temperature inside so that the heat index (formula defined in the following section) will be the desired temperature.

Dehumidifier (Whole-house)
Without a humidifier, the system cannot raise the humidity if it falls to dangerous levels. However, we implement a system override when the humidity is too low. In the event that the humidity falls into dangerous levels (10% below ideal relative humidity), the system shuts off the A/C, regardless of temperature, in attempt to prevent the humidity from decreasing any further.

In addition, the system uses the dehumidifier to prevent humidity from rising above a certain value. Therefore, if humidity is higher than desired, the system can hold the humidity constant and then use the heat index formula to determine what temperature to set the thermostat to.

If the humidity isn't too high, but also not constant, then the system just needs to keep recalculating the temperature it needs to set the home to, and since humidity doesn't change rapidly, it only needs to recalculate this value every 30 minutes.

Humidifier (Whole-house)
Without a dehumidifier, the system can't lower the humidity if it rises to dangerous levels. However, the use of A/C naturally draws liquid out of the air, lowering the humidity. Therefore, we can implement a system override, where if the humidity rises beyond the ideal values outlined above, the system will turn on the A/C until the humidity falls to the desired level, even if it causes the temperature to be lower than desired. This is because while temperature is for comfort, high levels of humidity can actually be dangerous.

With a humidifier, the system can prevent humidity from falling below a certain level, so if humidity is lower than desired, the system can bring the humidity up and hold it constant, and then use the heat index formula to determine what temperature to set the thermostat to.

If the humidity isn't too low, but also not constant, then the system just needs to keep recalculating the temperature it needs to set the home to, and since humidity doesn't change rapidly, it only needs to recalculate this value every 30 minutes.

No Humidifier or Dehumidifier
If the house doesn't have either a humidifier or dehumidifier, then the system can't do much. The only thing within the system's control is turning on or off the A/C if humidity is too low or high. Therefore, if the humidity rises beyond the values outlined above, the system can turn on the A/C disregarding temperature to prioritize humidity. Similarly, if the humidity drops to a dangerous level (10% below ideal levels), the system can disregard temperature and shut the A/C off to try and stop it from going any lower.

Single-room Humidifier or Dehumidifier
Some people utilize small or single use humidifiers and dehumidifiers to control humidity in their homes. Our system likely wouldn't have access to these products, so it would have to react to the humidity the user sets. In this case the system would measure humidity every 30 minutes in order to recalculate the desired temperature.

4.6 - Heat Index

As mentioned previously, relative humidity can affect what the temperature feels like. In our model, users can input their preferred temperatures, but the system won't just set that as the temperature. By monitoring the humidity, the system can input the relative humidity and desired "feel-like" temperature in order to derive an actual temperature to set in the house from the following equation in which T = temperature in degrees Fahrenheit and rh = relative humidity:

$$\text{Heat index (feel-like temperature)} = -42.379 + (2.04901523 \times T) + (10.14333127 \times rh) - (0.22475541 \times T \times rh) - (6.83783\times10^{-3}\times T^2) - (5.481717 \times 10^{-2} \times rh^2) + (1.22874 \times 10^{-3} \times T^2 \times rh) + (8.5282\times10^{-4} \times T \times rh^2) - (1.99\times10^{-6} \times T^2 \times rh^2)$$ [9]

4.7 - Air Quality Control

One condition that our system has to consider is air quality, including the presence of allergens such as dust, pollen and mold, and air pollution. Since our model already includes sensors in each room for motion and temperature, we will also build pollutant and allergen sensors into each one. Once a sensor measures higher amounts of air pollution or allergens, the system can activate the fan in the AC unit to circulate air throughout the home. As the air circulates, any pollutants and allergens will be caught in air filters [10].

One problem with this method is that it can distribute warm or cool air through the home. If it is warm outside, then air sitting in air conditioning ducts will heat up. Likewise, cold ducts will cool the air when it is cold outside. If the system is then turned on and this air is circulated without being cooled by the system, then this will affect the temperature in the house [11]. To combat this issue, the fan will only be turned on for short periods at a time. If these short bursts (approximately 15 minutes at a time) are not enough to remove the air pollution, then the system will run the fan for as long as needed. Extended fan use should not cause major issues because the thermostat will detect the changes in temperature and adjust accordingly. If the house heats or cools too much due to air circulation, the heating or air conditioning system will turn on to keep temperature the household temperature at the desired level, which runs the fan and circulates the air automatically.

5 - Model Testing and Sensitivity Analysis

To test our model, we implemented the portion of our model which predicts return and wake-up times in a Python program. We continued to create a sample 10-week schedule for a person who has a job with fairly standard hours on the weekdays. We added random trips on some days of the week and especially on the weekend, as many people have errands to run after work and different things to do every weekend. The schedule of one of the weeks is shown below:

	Occurrence	Leave/Sleep Time	Arrival/Wake Time	Duration
Monday	work	9:06 AM	5:12 PM	8.1 hrs
	go out	6:00 PM	9:06 PM	3.1 hrs
	sleeping	10:30 PM	7:00 AM	8.5 hrs
Tuesday	work	8:48 AM	5:06 PM	8.3 hrs
	go out	7:30 PM	10:00 PM	2.5 hrs
	sleeping	10:42 PM	7:06 AM	8.4 hrs
Wednesday	work	9:18 AM	7:06 PM	9.8 hrs
	sleeping (overslept)	10:12 PM	8:30 AM	10.3 hrs
Thursday	work (got up late)	10:06 AM	5:00 PM	6.9 hrs
	sleeping	10:36 PM	7:00 AM	8.4 hrs
Friday	work	9:12 AM	12:30 PM	3.3 hrs
	go out	3:36 PM	6:12 PM	2.6 hrs
	sleeping	12:30 AM	10:30 AM	10 hrs
Saturday	go out	11:06 AM	2:30 PM	3.4 hrs
	go out	4:06 PM	8:30 PM	4.4 hrs
	sleeping	12:24 AM	10:18 AM	9.9 hrs
Sunday	church	10:00 AM	2:06 PM	4.1 hrs
	sleeping	10:30 PM	7:06 AM	8.6 hrs

***Note:** *trips in red are irregular occurrences*

Table 2. Sample of simulated data set used to test model.

We entered this data into our Python program and were able to track how the program learned the person's tendencies over time. For each sleep or leaving event, the program predicted a return or waking time using our model's algorithm, which we compared with "true" time by taking the absolute value of their difference. We then averaged together all of these errors, resulting in an average error of 0.7 hours. The model predicted a return or wake time which was early or on time for over 65% of our test events. The algorithm was designed to often predict early arrival times, to ensure that the user would often return to a comfortable temperature, so a percentage this high is reasonable.

6 - Strengths and Weaknesses

6.1 - Strengths

Once the user has left the house at least once, our system is then able to predict return times for even the most irregular schedule. When predicting return times, our model has the capacity to recognize that an entirely non-periodic set of outings are part of the same event, as long as their start times are sufficiently consistent. Furthermore, our model can recognize instances of the same event on multiple weekdays if their start times are sufficiently similar. This allows it to more quickly refine its predictions, as it has a greater abundance of data for each event. More than just predicting the user's schedule, our model is able to adjust to the desired temperature using the least amount of energy possible by predicting how much time it will take to change the temperature back to comfort mode from the monitored outside temperature. Our model also takes into account the homeowner's safety by considering humidity and modifying its settings as necessary in order to keep the humidity of the house from reaching levels dangerous to the user's health. Similarly, the system also tracks air quality factors including allergens and pollen and adjusts air circulation accordingly to keep the house comfortable.

6.2 - Weaknesses

If our model is unable to predict a return time based on the new outing's leave time, it rather crudely combines all of the durations of recent irregular events into a single predicted duration, where more sophisticated techniques might allow us to distill recent irregular events' durations into several clusters, in turn providing us with several likely return times that represent a more integrated reflection of recent trends. Our model also assumes that the temperature will change in a linear fashion, and while this estimate may be adequate for more mild climates, it could be unsuitable for extreme conditions. The general error that comes with a system that makes predictions like ours results in energy waste that would not occur as much if the user inputted their schedule manually. In addition, our model looks at the the turning off of lights and appliances to determine when the user goes to sleep, which could introduce error as some people fall asleep with the light or other appliances on. Another weakness of our model is that we were unable to test it on real data and our tests were based on simulated data, which could be somewhat unrepresentative of an actual person's life.

7- Comparison with Similar Products

There are lots of similar products currently on the market, including some that employ similar techniques to our own model and others that use different methods.

7.1 - Ecobee4 Smart Thermostat

One smart home thermostat system that is very popular is the Ecobee4 Smart Thermostat. The Ecobee4 is an example of the older generation of thermostats that require the user to input their schedule in order to know when to save energy [12]. One benefit to this method is that it is very precise. You know your schedule better than any model could predict, so the Ecobee4 should never be off or on when it isn't supposed to be. However, one downside is the amount of manual input that is required from people with irregular schedules. This becomes especially complex when multiple people with different schedules are coming and going from the house, making it difficult to determine when the thermostat should be shut off.

7.2 - Nest Learning Thermostat

Another smart home thermostat that shares similarities with our model is the Nest Learning Thermostat. Nest is an example of the newer generation of smart home thermostats that learn from the user's behaviors and manual inputs during an initial "training period". After this brief period, the thermostat begins to automatically adjust the temperature in response to the user's preferences and schedule. This product is different than Ecobee4 because it doesn't require as many manual changes. However, since many people have irregular schedules, the Nest system introduces more error into its predicted schedule when it tries to learn when to adjust the temperature. The only way to overcome this error is to make manual changes to the schedule, making the user experience of the Nest ultimately similar to the Ecobee4. The difference between each method results in a clear disparity in energy savings. Ecobee claims their product saves an average of 23% on heating/cooling costs, while Nest claims to save between 10-12% on heating costs, and 15% on cooling [13].

7.3 - Comparing all three

There are some features that Nest, Ecobee4 and Enthalpy all have in common, including geofencing, room sensors, and saving of energy. Geofencing is the method used by all of the products to determine when a person has left their home. The idea is to create a virtual zone around the user's house and track their phone's location to see if they are inside the zone. The zone is fairly small, so if a phone is in the zone, the systems consider that person to be at home and will switch modes appropriately.

Another feature that all these systems share is room sensors, but some of them work differently. The Nest thermostat uses the sensors to keep track of each room's temperature, which it in turn uses to adjust temperature as needed. Also, with the use of these sensors, Nest allows different rooms to have different desired temperatures [14]. Our model and Ecobee4 also use similar sensors to determine the temperature of each room, but they are also used to see which rooms are occupied via built-in motion sensors. This allows both our model and Ecobee4 to prioritize the temperature of some rooms and save energy in rooms that aren't being used.

Additionally, in order to save energy all three systems use the concept of turning off the thermostat until the temperature reaches a high or low boundary, where the system then turns on to prevent further change in temperature past the boundary.

Finally, the systems share many smaller features including different modes for when people are home, away, sleeping, etc. and the ability to make manual adjustments to the schedule/temperature through a smartphone app.

While these three systems share much in common, there are also some key differences. The largest difference between the models is the level of input required for the system to adjust temperature. Ecobee4 requires the user to input their schedule completely manually while Nest attempts to learn the user's schedule by reacting to consistent manual corrections [15]. On the other hand, our system learns their preferences and then adapts to a changing schedule without requiring any manual changes at all.

8 - Multiple Occupants

While our model is effective in the simple scenario of a single occupant in a small home, we also have to consider larger homes with multiple inhabitants.

To account for multiple people in the house, we don't need to adjust the way our model tracks leaving and returning times, but rather how it uses them. In order to determine when the user has left the house, the system uses the location of the user's phone. Once the phone has left a small region surrounding the house, the system records the time and treats it as the start of an event. This is useful for expanding our model to include multiple people, because the system can track each phone as its own user, allowing it to track each person's departure and arrival times separately. The system will only enter green mode after all phones are out of the zone, meaning no one is home.

Once all users are out of the house, our system enters green mode and determines when to transition back to comfort mode by preparing for the earliest predicted return time out of all the schedules. The way our system finds predicted return times is outlined in detail above. Then, similarly to how it functions for just one person, if the first person doesn't return at the predicted time, the system will prepare for the next earliest predicted return time out of all the schedules.

9 - Multiple Heating/Cooling Zones

In a house with multiple heating/cooling zones, our model requires only slight adjustments. We introduce a new mode, called vacant mode, in order to save more money and energy. Using motion sensors in each zone, the system can monitor which zones are in use, and therefore which zones don't need to be heated or cooled. Any zones where people have not been present for more than 10 minutes will enter this mode, which follows the same logic as both our green and standby modes. In this mode, the thermostat would let the temperature change naturally until it reaches a high or low boundary, which in this mode would be set at 2°F above and below the desired temperature.

The idea behind this mode is that there are periods where the thermostat is not in use, and therefore not using energy, but it doesn't let the temperature change by so much that it would take a long time to return to a comfortable temperature if they re-enter the zone. In addition, since people can move from zone to zone very quickly, and it would be pointless to attempt to predict the movement of inhabitants through the zones, we want to make sure that if they re-enter a zone at any point, in the period it takes to return to the desired temperature, it isn't uncomfortably warm or cold. A difference in 2°F is noticeable, which is why the system returns to the comfortable temperature when someone enters the zone, but it isn't so different that it would be unpleasant to stay in the zone until it returns to comfort mode.

One thing that is important to note is that the other modes will override these motion sensors. For example, if you have a pet running around the house while you are gone, the motion sensors won't think that someone is home and therefore move into comfort mode. The transition into green mode is solely based off of phone location. Only once someone is home and the house is in comfort mode will individual rooms start to enter this new mode as people move between the zones.

10 - References

[1] "Air Conditioning." *Department of Energy*, www.energy.gov/energysaver/home-cooling-systems/air-conditioning.
[2] "Home Heating Systems." *Department of Energy*, www.energy.gov/energysaver/heat-and-cool/home-heating-systems.

[3] Lawrence, Chrishelle. "U.S. Households' Heating Equipment Choices Are Diverse and Vary by Climate Region." *U.S. Energy Information Administration* , 6 Apr. 2017, www.eia.gov/todayinenergy/detail.php?id=30672.
[4] Donovan, Jay. "The Average Age for a Child Getting Their First Smartphone Is Now 10.3 Years." *TechCrunch*, TechCrunch, 19 May 2016, techcrunch.com/2016/05/19/the-average-age-for-a-child-getting-their-first-smartphone-is-now-10-3-years/.
[5] Ware, Arista. "Best Temperature for Sleep." *Sleep.Org*, Sleep.Org, 10 Nov. 2014, www.sleep.org/articles/temperature-for-sleep/.
[6] "How Long Should It Take My AC to Cool Down My Home?" *Sansone AC*, Sansone Air Conditioning, 20 Aug. 2018, sansone-ac.com/how-long-should-it-take-ac-to-cool-down-home/?fbclid=IwAR1rk_3OBp_R59y2mApX6g2TMjXtljXp02zd6fIlCwz6kThM_czrPmcIwKM.
[7] Unsdorfer, Stewart. "Managing Ideal Indoor Humidity for Maximum Comfort." *Central Heating and Air Conditioning*, Central Heating & Air Conditioning, 29 Mar. 2018, www.centralhtg.com/blog/managing-home-humidity-for-maximum-comfort.
[8] "What Is the Recommended Humidity Level for My Home?" *HVAC.com*, HVAC.com, www.hvac.com/faq/recommended-humidity-level-home/.
[9] "Heat Index." National Weather Service. https://www.weather.gov/media/epz/wxcalc/heatIndex.pdf
[10] Rodgerson, Corey. "Allergies and Your Air Conditioner - Purge the Pollen." *Climate Heating & Cooling*, 26 Feb. 2018, www.climateinc.com/allergies-air-conditioning/.
[11] "Should You Run Your Blower Fan without the Air Conditioner? – AND Services." *AND Services*, 27 Mar. 2018, www.andservices.com/blog/run-blower-fan-without-air-conditioner/.
[12] Dirks, Brent. "How to Set Up and Use the Ecobee4 Smart Thermostat." *MakeUseOf*, 15 Dec. 2017, www.makeuseof.com/tag/setup-ecobee4-smart-thermostat/.
[13] Sage, Simon. "How Much Money Can You Save with a WiFi Thermostat?" *IMore*, IMore, 11 June 2015, www.imore.com/wifi-thermostat-save-you-money.
[14] "Nest Temperature Sensor | The Right Temperature, Right Where You Want." *Nest*, Nest Labs, Inc., nest.com/thermostats/nest-temperature-sensor/overview/.
[15] "How Nest Thermostats Learn." *Nest*, Nest Labs, Inc., nest.com/support/article/An-introduction-to-learning#nest-thermostat-learns-a-week.