

# WalkSense: Classifying Home Occupancy States Using Walkway Sensing

Elahe Soltanaghaei  
University of Virginia  
Charlottesville, USA  
es3ce@virginia.edu

Kamin Whitehouse  
University of Virginia  
Charlottesville, USA  
whitehouse@virginia.edu

## ABSTRACT

Home automation systems can save a huge amount of energy by detecting home occupancy and sleep patterns to automatically control lights, HVAC, and water heating. However, the ability to achieve these benefits is limited by a lack of sensing technology that can reliably detect zone occupancy states. We present a new concept called *Walkway Sensing* based on the premise that motion sensors are more reliable in walkways than occupancy zones, such as hallways, foyers, and doorways, because people are always moving and always visible in walkways. We present a methodology for deploying motion sensors and a completely automated algorithm called *WalkSense* to infer zone occupancy states. WalkSense can operate in both offline (batch) and online (real-time) mode. We evaluate our system on 350 days worth of data from 6 houses. Results indicate that WalkSense achieves 96% and 95% average accuracies in offline and online modes, respectively, which translates to over 47% and 30% of reduced energy wastage, and 71% and 30% of reduced comfort issues per day, in comparison to the conventional offline and online approaches.

## CCS Concepts

•Computer systems organization → Embedded systems; Redundancy; Robotics; •Networks → Network reliability;

## Keywords

occupancy detection, motion sensor, walkway sensing

## 1. INTRODUCTION

Many homes have multiple “zones”, each of which requires a different level of heating and cooling. For example, many homes have a daytime zone (living room and kitchen) where the temperature should be comfortable and a nighttime zone (bedrooms and bathrooms) where the temperature can float. Another example includes two- or three-story homes where heating and cooling should be adjusted based on the floors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*BuildSys '16, November 16-17, 2016, Palo Alto, CA, USA*

© 2016 ACM. ISBN 978-1-4503-4264-3/16/11...\$15.00

DOI: <http://dx.doi.org/10.1145/2993422.2993576>

that are occupied due to temperature stratification. To achieve this, some homes would put a central heating/cooling system into different *modes* based on zone occupancy state, while other homes would have independent thermal conditioning for each zone. Either way, substantial energy can be saved by automatically detecting zone occupancy and adjusting the heating/cooling accordingly. For example, studies have shown that energy usage can be reduced 20-30% by reducing heating and cooling when residents are asleep or away [1]. However, the ability to achieve these benefits is limited by a lack of sensing technology that can reliably detect zone occupancy states.

The most common and intuitive approach to sense zone occupancy states is what we call *activity sensing*: install motion sensors in every zone, and the zones that contain activity are defined to be occupied. This approach is widely used for smart thermostats and home automation systems today [14, 16] even though motion sensors are notoriously unreliable: they can fail to detect a person’s presence if that person is out of view or sitting still (e.g. watching TV or sleep), which creates ambiguity about which zones are occupied. Detecting occupancy in sleeping zones is especially difficult because people remain still for long periods of time while sleeping. In our survey of the occupancy sensing literature, we found no studies that could use motion sensors to reliably detect occupancy in a sleeping zone. Most studies exclude the nighttime hours from their evaluation altogether [3, 13, 20, 22]. Other studies define sleep to be all periods of inactivity between certain hours, such as 10pm-8am [7, 20], even though this approach produces errors every time a person sleeps during the day or goes out at night, both of which are common occurrences for many people, and especially for the nearly 15 million shift workers who work a permanent night shift or regularly rotate in and out of night shifts<sup>1</sup>.

In this paper, we present a new approach to reliably detect occupancy in the zones of a house using motion sensors. For the reasons described above, we do not rely on motion sensors to constantly detect activity in a zone. Instead, we primarily use motion sensors to detect occupancy in the walkways between zones, such as hallways, foyers, or doorways. We hypothesize that motion sensors will work more reliably in walkways than occupancy zones because people are always moving in walkways and do not sit still in them for long periods. In addition, walkways are small enough for the entire area to be within the view of a motion sensor. We therefore propose a new principle for the use of

<sup>1</sup><http://www.cdc.gov/niosh/topics/workschedules/>

motion sensors that we call *Walkway Sensing*, and demonstrate that we can convert walkway sensing into a reliable form of occupancy sensing for zones.

For concreteness, we explain and evaluate walkway sensing in the context of detecting the typical home’s three main occupancy states: (1) *Active*: when at least one occupant presents in the daytime zone (e.g. living room and kitchen), (2) *Away*: when all occupants left the home, and (3) *Sleep*: when all occupants who are at home, are asleep. However, the underlying principles of walkway sensing will generalize to homes with other zone configurations. For example, a home that has a desk in a bedroom could use additional sensors to differentiate between the sleeping zone and the working zone.

To use walkway sensing to detect the active, sleep, and away states, we first *zone* the home into three distinct regions: the outside zone, the sleep zone, and the active zone. Second, we deploy motion sensors in the *walkways* between the three zones, such as the hallway, doorway, or foyer. Third, we deploy a motion sensor covering the main activities in the active zone. Based on this sensor placement, we run the WalkSense occupancy detection algorithm, which comes in two variants. The *offline* variant operates in batch mode on historical data, labeling prior occupancy states with full knowledge of the data produced before and after the state occurred. It identifies sleep (or away) states by looking for inactivity periods that occur during the *interval* between two consecutive sleep (or outside) walkway sensor events. The entire such interval is labeled as a sleep (or away) state. In contrast, the *online* variant operates in real-time, labeling current occupancy states before subsequent data readings are observed. It is executed every time a person is detected in a walkway, which is on every potential transition event into or out of the sleep or away states. It uses a classifier to determine the current occupancy state based on the sensor readings before, after and during the transition event. However, online WalkSense does not require any manual training data; the output of offline WalkSense is used to automatically train online WalkSense.

We evaluate WalkSense on 6 homes resulting in 350 days worth of data. We deployed standard off-the-shelf motion sensors in 5 homes for 3-7 weeks in each home, using daily questionnaires to collect ground truth about the active, sleep, and away states in each home. We also used one public dataset with annotated ground truth that includes 26 weeks of data [8]. Results indicate that offline WalkSense can detect the home occupancy states with 96% accuracy and can reduce occupants comfort loss and energy waste by 71% and 47%, respectively, compared with the conventional activity sensing approaches [14]. The online WalkSense detects occupancy states with 95% average accuracy in real-time, and can reduce comfort issues and heating energy by 30% and 32%, respectively, compared with the conventional online occupancy inference algorithms [16]. In addition, analysis shows that 12% of detected sleeping instances are daytime napping and 11.7% of away periods happened at night, between 9pm to 3am, which highlights the robustness of the proposed method to irregular sleep and away patterns.

## 2. BACKGROUND AND RELATED WORK

Several smart thermostat systems, such as the Nest learning thermostat, attempt to control heating and cooling based on occupancy with the goal of both eliminating the need

for manual user adjustment and improving the efficiency of home heating [10, 15, 16, 21–23]. However, they are limited by the lack of a reliable occupancy sensing approach and an accurate occupancy detection algorithm. For example, Pre-Heat [22] and ThermoCoach [20] use RFID tags on each occupant’s house keys to differentiate between active and away states, depending on whether the RFID tags were present in the home, but cannot detect sleep states and ignores possible energy savings during the sleep periods. It should be noted that the sleep periods create more energy saving potential in cold seasons because peak heating load is at night. This highlights the importance of detecting sleep periods in addition to away states. The Smart Thermostat [16] leverages motion sensors to detect the three home occupancy states. However, it defines a fixed temporal boundary between daytime and nighttime activities, which is not effective for houses with irregular or daytime sleep periods. In addition, it requires a full range of coverage to differentiate away periods from low motion activities at home.

Several studies have explored new technologies to detect occupancy in residential houses using electricity usage [7], GPS data [15], or network usage [12]. However, they all ignore sleep periods and nighttime occupancy. For example, Chen et al. [7] proposed a system to estimate occupancy based on electricity usage in the home. However, many activities such as reading or sleeping are not involved with electrical devices. Furthermore, it only detect daytime occupancy due to the lower correlation between occupancy and power consumption at night. Makonin and Popowitch [18] addresses nighttime occupancy by adding ambient light sensors to detect the sleep periods based on illuminance spikes. However, the light level can be greatly affected by the sunlight variations or opening and closing the curtains [17].

Occupancy detection has been also studied for commercial buildings to differentiate occupied and unoccupied states in room zones. Some approaches use camera or Wifi data to detect the presence of occupants [5, 9], but these methods are not practical for residential buildings because of privacy concerns or difficulty in differentiating sleep and away periods. Agarwal et al. [3] uses PIR sensors to monitor the presence of occupants inside the offices and uses magnetic reed switch door sensors to monitor door activities. However, this method assumes that offices are single-person and have a physical door that is closed when nobody is in the room. Walkway Sensing extends this idea to scenarios walkways in general, even if they don’t have a physical door, and to multi-person environments.

Several research papers focus on detecting sleep patterns only. Some studies use load sensors in beds to measure and correlate weight with sleep-related activities such as bed entrances and exits [2, 4]. However, this method does not work properly when two occupants sleep in one bed. Other researches leverage wearable sensors such as acceleration sensors, smart watches, actigraph, and microphones [6], but the usability of wearable sensors cannot be assumed in the context of occupancy sensing for heating and cooling control. In this paper, we propose Walkway Sensing as a novel sensing approach to accurately sense zone occupancy using motion sensors.

## 3. THE WALKWAY SENSING APPROACH

The most obvious solution to occupancy detection is to cover each zone in a home with a motion sensor and corre-

late sensor events to the occupancy of the zones. However, this strategy is only effective if each zone is fully covered by highly sensitive motion sensors to detect presence of occupants even if they are sitting still or have little motions. In addition, it requires that sensing areas of different zones do not overlap with each other to avoid confusions about the location of the occupants. However, these requirements are often impractical and costly with the current motion sensors and the requisite coverage tests. As a result, current approaches [14], only monitor primary home areas such as bedroom, living room, and kitchen to detect specific activities. However, incomplete coverage produces ambiguity about which zone is occupied and is therefore not effective in differentiating sleep, away, and low-motion active periods since they all appear as periods with no sensor event.

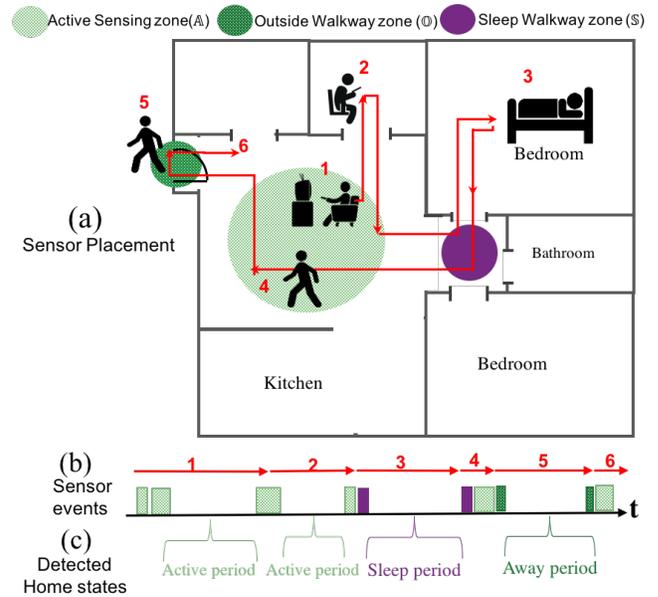
To address these challenges, instead of sensing room zones, WalkSense uses a strategic sensor placement aimed only at sensing the walkways to the bedroom and exit door as shown in Figure 1-(a) to achieve high occupancy accuracy. Sensor events recorded at each walkway will indicate the beginning and the end of sleep or away periods as seen in Figure 1-(b). In addition, to differentiate low motion activities such as watching TV from sleep/away periods, people in the active zone must be detected once during their presence in this zone. Therefore, only the main area of activity must be sensed. Figure 1-(c) shows how the proposed approach classifies the inactive intervals into active, sleep and away states based on the sensor events.

WalkSense can operate in two modes: *Offline* and *Online*. The offline mode determines the occupancy state of the home in the past data and can be used by self-programming thermostats to model sleep and away patterns [11,20]. The offline WalkSense finds the candidate inactive intervals using all the information before, after, and during the intervals, and determines their status based on the walkway events (such as Figure 1-(c)). The online WalkSense determines the current state of the home and performs real-time occupancy detection for smart thermostats to control heating and cooling in real time. It calculates the possibility of transitions in the home status for each walkway event using machine learning models. In the following sections, we first explain the details of sensor placement policies. Then, the offline and online WalkSense algorithms are formally defined.

### 3.1 Sensor Placement

To detect sleep, away, and active states of a home, we define three zones of “sleep Walkway (S)”, “Outside Walkway (O)” and “Active (A)” in each floor plan (as seen in Figure 1-(a)). S is the transition area between the active zone and the sleep zone, and O is the transition area through which occupants regularly walk for going out. We define A as the areas where occupants generally do their daily activities such as living room where occupants watch TV, read book, or rest. The active sensors must detect people who are not in the bedroom or outside at least once during their presence in the active zone. Therefore, having one key active sensor in the main room is enough for most floor plans. Based on these requirements, the sensor placement policies are defined as follows:

- S must have a non-empty area not covered by A or O.
- O must have a non-empty area not covered by A or S.
- A must cover people who are not in the bedroom or outside at least once during their presence in the zone.



**Figure 1: (a)Sensor Placements in WalkSense. (b)The sensor reading timeline as the occupant performs activities 1 to 6. (c)Walkway events indicate the start and end of sleep and away periods and differentiate them from low motion active periods.**

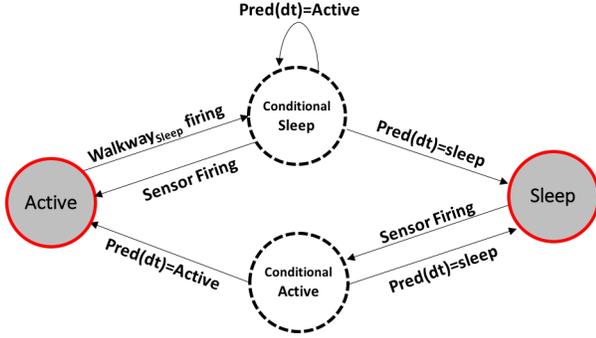
- Sensors can be installed in the doorway if there is no walkway area to the bedroom or outside.

Put simply, these rules can be satisfied for most types of floor plans by installing motion sensors on doorways of bedrooms and exit doors. In addition, motion sensors with short range and limited field of view can be used to avoid overlaps between zones. One of the advantages of this placement is that in floor plans where multiple bedrooms share a common walkway (such as Figure 1), one sleep walkway zone (S) can be defined to minimize number of required sensors. Therefore, for an apartment with one sleep walkway and one exit door, only three sensors are required to cover the defined zones.

### 3.2 Offline WalkSense Method

Offline WalkSense defines a *candidate sleep/away interval* to be the duration between a pair of consecutive sensor events in a given walkway which is longer than  $\mathcal{K}$  minutes. That interval is decided to be in the sleep/away state if no motion is detected in the active zone during this interval. The value of  $\mathcal{K}$  is selected based on the operational delay of the specific HVAC system deployed in each home. From an HVAC perspective, any sleep or away period shorter than  $\mathcal{K}$  minutes can be safely ignored since the HVAC system and thermal properties of the home result in a certain time lag until the temperature setpoint is reached. As a result, the system fails to save energy and may even consume more by reacting to short away or sleep periods. Therefore, we define the parameter  $\mathcal{K}$  as the minimum duration of a sleep or away state and we expect reasonable values of  $\mathcal{K}$  to be between 60-150 minutes [16].

Multiple bedrooms or exit doors define separate walkway zones based on their location in floor plan. Therefore, in general, we define the set of sleep walkway sensors as



**Figure 2:** The online WalkSense follows a state machine to switch between *Active* and *Sleep* (or *Away*) states based on the output of a supervised learning algorithm.

$\mathbb{S} = \{s_1, s_2, \dots, s_{n_1}\}$ , where  $s_i$  is a motion sensor installed in the sleep walkway. Similarly, the sets of outside walkway and active sensors are defined as  $\mathbb{O} = \{o_1, o_2, \dots, o_{n_2}\}$ ,  $\mathbb{A} = \{a_1, a_2, \dots, s_{n_3}\}$ , where  $n_1, n_2$ , and  $n_3$  are the number of sensors in sleep walkway, outside walkway, and active zones, respectively. A candidate sleep interval  $(t_{s_i}, t_{s_j})$ ,  $s_i, s_j \in \mathbb{S}$  is identified as sleep state if there is no active or outside sensor event within that interval. Formally,

$$\nexists t_p | (p \in \mathbb{O} \cup \mathbb{A}) \wedge (t_{s_i} < t_p < t_{s_j}) \quad (1)$$

where  $s_i, s_j$ , and  $p$  are the triggered sensors at time stamps  $t_{s_i}, t_{s_j}$ , and  $t_p$ , respectively. Similarly, a candidate away interval  $(t_{o_i}, t_{o_j})$ ,  $o_i, o_j \in \mathbb{O}$  is identified as an away period if no other sensor triggers during that interval. Formally,

$$\nexists t_p | (p \in \mathbb{S} \cup \mathbb{A}) \wedge (t_{o_i} < t_p < t_{o_j}) \quad (2)$$

where  $o_i, o_j$ , and  $p$  are the triggered sensors at time stamps  $t_{o_i}, t_{o_j}$ , and  $t_p$ , respectively.

### 3.3 Online WalkSense Method

The online WalkSense exploits the concept of walkway sensing to discover the possible transitions to sleep or away zones, which result in state changes. The proposed algorithm tracks sensor firings until a transition to the sleep/away walkway occurs. Then, a supervised learning model runs to calculate the probability of this transition resulting in sleeping or leaving home. The online WalkSense defines two separate *detection modules* in the form of state machines for learning the sleep and away patterns of the occupants. Each module use a classifier to classify the candidate transitions into sleep (or away) and active.

The sleep detection module is explained as a state machine in Figure 2, which has two steady states of *Active*, and *Sleep* and two transient states of *Conditional Sleep*, and *Conditional Active*. The module is in the *Active* state when the home is occupied and at least one resident is active. Then, it switches to the *Conditional Sleep* state if one of the sleep walkway sensors triggers. At this time, the classification model is activated and actively monitors the sensor firings. If the classifier labels the transition event as sleep, the system switches to the *Sleep* state. Otherwise, it stays in the *Conditional Sleep* state and the classifier will be executed every  $\tau$  minutes until one of these conditions happens: (i) a sensor triggers, which indicates that someone is active,

Temporal Features	weekday/weekend
	day of week
	time of day at 2-hour granularity
Transition Features	total number of sensor firings per zone in $dt$
	average gap between sensor firings in $dt$
Mobility Features	total number of sensor firings per zone in time interval $dt_{past}$
	median of inter-arrival times of sensor firings in $dt_{past}$
	time interval between the median (previous feature) and $t_w$

**Table 1:** The online WalkSense uses three groups of features in the sleep/away classifier.

or (ii) the classification model detects the home state as sleep. It should be noted that as long as the system is in the *Conditional Sleep* state, the home state is considered as active.

If the state machine switches to the *Sleep* state in Figure 2, the classification model stops running. The home remains in this state until a sensor triggers. However, this sensor firing may happen because of sensor failures, or short interrupts caused at midnight for drinking water or going to the bathroom. Therefore, the home status should not be changed aggressively and the system should not react to every short-term sensor triggering. To avoid these fluctuations, if a sensor triggers in *Sleep* state, the state machine switches to the *Conditional Active*. Then, the classifier runs again to classify this event. If the classifier output is active, the state machine switches to the *Active* state. Otherwise, it turns back to the *Sleep* state. Although this mechanism causes a short delay in switching back to the *Active* state, thereby resulting in a small occupant comfort loss, it tries to make a balance between the energy saving and occupants comfort. We investigate this trade-off in the evaluations. It should be noted that the away detection module works similarly by tracking occupants walking to the away walkways and classifying these transitions to *Active*, and *Away* states.

The classification models include a group of observed features that are calculated for the transition interval  $dt = (t_w, t_w + c\tau)$ ,  $c \geq 1$ , where  $t_w$  is the timestamp of detecting a transition to sleep/away walkway,  $c$  is the number of loops in the *Conditional Sleep/Away* state, and  $\tau$  is the wait time between loops in the *Conditional Sleep/Away* state. The features can be categorized into three groups as shown in Table 1: (i) temporal features: to discover the correlation of occupants activities with time (ii) transition features: to indicate the activity patterns during the transition time interval  $dt$ , (iii) mobility features: to measure how active the occupants are before entering the walkway zones. The mobility features are calculated for the time interval  $dt_{past} = (t_w - 60', t_w)$ . It should be noted that features (ii) and (iii) are calculated per zone to show the sensor firing patterns in sleep walkway, outside walkway, and active zones.

One of the challenges of using supervised models is the required training set. The previous approaches use manual labeling by asking occupants to report their sleep and away times for few days, however, this is prohibitively a time-consuming process and causes user frustration. In addition, to learn changes in occupancy patterns and improve the occupancy models over time, they require a constant

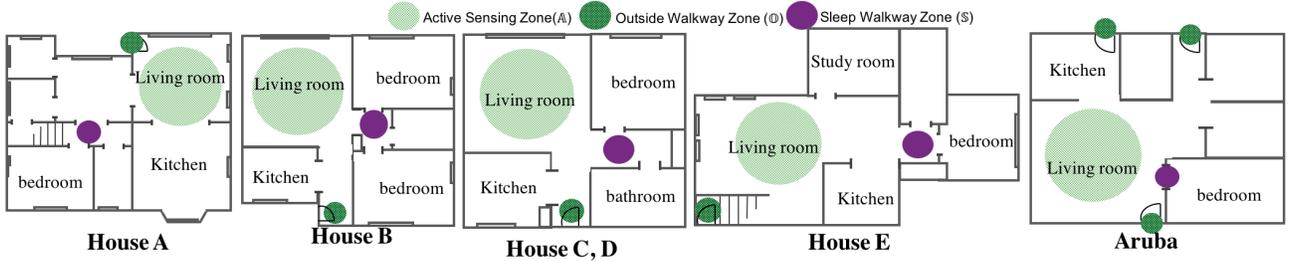


Figure 3: Floor plan of the six homes in the study with the placement of deployed sensors in each home

user involvement. We address this challenge by exploiting the offline WalkSense as an automatic labeling technique and derive the training set from the past sensor data and calculate labels by using the offline WalkSense algorithm. The insight behind this idea is that the offline WalkSense can consider all data during the entire interval and provides the state of the interval with very high certainty, thus suitable for creating the training set. The other advantages of this automatic labeling technique is that it provides incremental training [24] over time to use all the past sensor data and improve the occupancy models, as opposed to the manual methods such as the Nest thermostats which can only use the user inputs for learning new patterns.

#### 4. EXPERIMENTAL SETUP

To investigate the impact of occupancy patterns and different floor plans on the performance of WalkSense, we use the sensor reading collected through two means: (i) the in-situ data traces from IRB-approved studies in 5 instrumented homes, and (ii) one public dataset [8] called Aruba with sleeping and away annotated activities. All homes are instrumented with standard off-the-shelf motion sensors and the sensors are placed as shown in Figure 3 for each floor plan. In general, we deploy one motion sensor in the hallway toward the bedroom, one on entryway to the home, and one sensor in the living room. Two of the homes have the identical floor plans with one and two-person occupancy. The Aruba public dataset includes 30 sensors in the entire house, but we only uses five sensors to define  $\mathbb{S}$ ,  $\mathbb{O}$ , and  $\mathbb{A}$  zones. The Aruba floor plan has three exit doors and one sleeping bedroom resulting in 5 required motion sensors. We use all 30 sensors for analyzing the effect of number of sensors and sensor locations in Section 6.1.

Participants were given no special instructions and followed their routine activities and home occupancy functions. They also had guests or multi-day trips during the

Home	number of people	number of rooms	number of sensors	number of weeks
A	2	8	3	3
B	2	6	3	7
C	2	5	3	6
D	1	5	3	6
E	1	6	3	2
Aruba	1	8	5	26

Table 2: Details of the 6 data traces used in the evaluation

study. This can help investigate the effect of irregular sleep and away patterns on accuracy of the proposed approach. The homes include both single-person and multi-person occupants and the people living in the home include students, professionals, and homemakers. For example, one home includes two students with similar sleep and away patterns (home A), while the other one includes a professional couple with different work hours, but they sleep in the same room (home C). In addition, one of the single-person homes includes a young professional with late night shifts (home E). Table 2 summarizes the information about the homes.

The duration of sensor deployments varied from three to six weeks with the total number of 170 days, and the Aruba dataset includes 26 weeks of sensor data, resulting in 350 days worth of data from 6 houses. We collected ground truth for the five instrumented homes using manual daily reports of the residents. We created an online form on which residents would enter their sleep and away times every night. These self-reported times are not expected to be perfectly accurate, but the errors are estimated less than 30 minutes. The ambiguous and questionable data was clarified by interviews every week. The public Aruba dataset includes annotated sleeping and leaving events which are used as ground truth.

The offline WalkSense has the input parameter  $\mathcal{K}$ , which shows the minimum length of sleep or away intervals. We select the value of 2 hours for the main results and investigate the impact of  $\mathcal{K}$  on the WalkSense performance in Section 6.3. By exploratory analysis of the collected data, we found that a sleep period may be interrupted by occupants awaking at midnight to go to the bathroom. However, the HVAC system should not react to such short-term state changes. Therefore, we merged the sequential detected sleep (or away) periods that have a time gap shorter than  $\mathcal{K}/2$  in between to avoid unnecessary state fluctuations. In the online WalkSense, we use decision tree classifier as the classification model because of its inherent capability of handling combinations of mix type data (categorical and numerical) and its capability of finding the best split points of numerical features for multi-class problems [19]. To evaluate the models, we also use leave-one-out cross validation.

#### 4.1 Baseline

Both the offline and online WalkSense approaches are compared to existing methods that aim to fulfill their respective goals [14, 16]. As the accuracy of the existing methods highly depends on the number of sensors and their placements, to provide a fair comparison, we consider the same sensors for these baselines and WalkSense by including the

walkway sensors in the baselines. We discuss the impact of sensor placements in Section 6.1. In addition, to avoid false activity detection in the bedroom for the baselines, motion sensors are not installed in the bedrooms and the sleep walkway sensors are used as a notion of bedroom activity.

The offline baseline (called *Offline-Activity*) is based on the activity sensing approaches [14], which detects the activities of leaving home and sleeping by finding periods of inactivity. The extracted inactive periods are classified to sleep or away based on a fixed interval (e.g. 10 PM-8 AM), which indicates the sleep hours. The best sleep interval is selected based on the training data for each house. Although the Offline-Activity baseline leverages the benefit of sensors placed in walkways, it does not have any knowledge about walkway sensing and considers every detected inactive interval either as sleep or away.

The online baseline (called *Online-HMM*) is an occupancy inference algorithm based on Hidden Markov Model (HMM) [16], which estimates the probability of the home being in one of the away, active, or sleep status. The Online-HMM transitions to a new state every ten minutes based on the observed variables defined in [16]. To consider a similar comfort/energy sacrifice with the Online-HMM baseline, we select ten-minute intervals for the value of  $\tau$  (the wait time between *Conditional Sleep/Away* loops) in the online WalkSense.

It should be mentioned that the PreHeat approach [22] classifies home states to occupied/unoccupied states by using RFID tags. Therefore, it doesn't have any mechanism to differentiate between sleep and active periods. On the other hand, the Nest learning thermostats only use one motion sensor to adjust the temperature after detecting a period of inactivity. It cannot differentiate between sitting still, sleeping and being away with one sensor since all appear as periods of inactivity. Therefore, it has to rely on the user inputs to learn the occupancy patterns. For this reason, we do not include PreHeat and Nest in the comparison, as detecting sleep periods and differentiating them from away periods is one of the main goals of this paper.

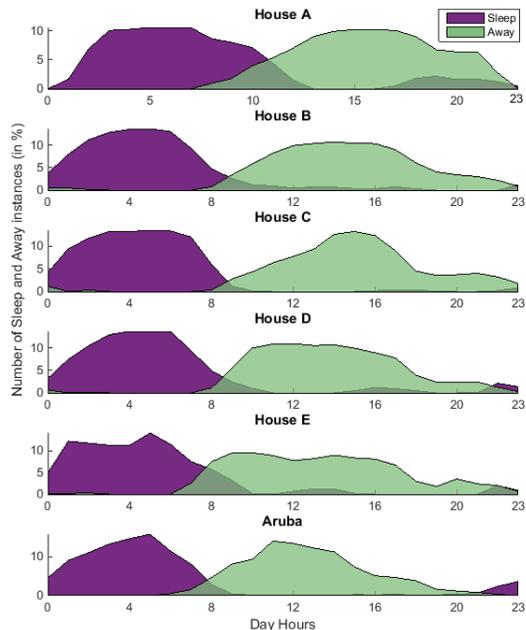
## 4.2 Evaluation Metrics

We evaluate the offline and online WalkSense based on two quantitative metrics of Energy Penalty and Comfort Penalty. *Energy penalty* is defined as the average time when the occupants are away or sleep, but the system wrongly detects it as active (equation 3). *Comfort penalty* is defined as the average time when the home is in the active state, but the system wrongly detects it as sleep or away (equation 4). In addition, the HVAC system can be scheduled to a lower temperature when people are away, compared to the periods when people are asleep. Therefore, if the away periods are wrongly detected as sleep, it causes energy load, while if the sleep periods are misclassified as away, it results in comfort issues. Therefore, the Energy and Comfort penalties are defined as follows:

$$EnergyPenalty = \frac{1}{d} \sum_{i=1}^d (t_i^{OasA} + t_i^{SasA} + t_i^{OasS}) \quad (3)$$

$$ComfortPenalty = \frac{1}{d} \sum_{i=1}^d (t_i^{AasO} + t_i^{AasS} + t_i^{SasO}) \quad (4)$$

where  $d$  is the total number of experiment days for each



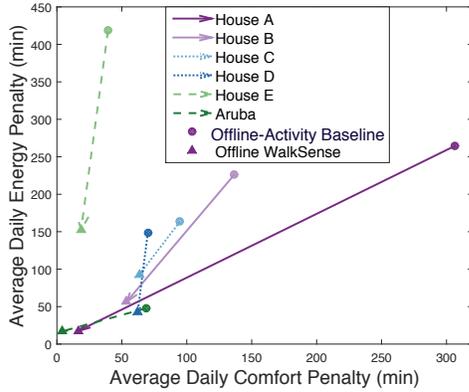
**Figure 4: The sleep and away instances overlap in different times due to irregular occupancy patterns. Therefore, defining a sleep interval for differentiating sleep and away periods is expected to fail.**

house,  $t^{OasA}$  is the fraction of time that Outside ( $O$ ) instances are wrongly detected as Active ( $A$ ),  $t^{SasA}$  indicates the total time period that Sleep ( $S$ ) is detected as Active ( $A$ ), and similarly other parameters are defined. In addition to these two metrics, the confusion matrices of offline and online WalkSense approaches are compared with the baselines in Section 5.3. The values in the confusion matrices are calculated in time and converted to percentage to provide easy comparison between the offline and online approaches as well as WalkSense and the baselines.

## 5. EVALUATION

The collected data traces in the 6 homes show that people involved in the experiments regularly spend 24% of the day outside home, and 37% in asleep. Thus, the energy can be saved by reducing the temperature setpoints in these intervals. In addition, 12% of sleeping instances are daytime napping with average duration of 2 hours and 11.7% of away instances are between 9PM to 3AM with average duration of 3.17 hours. Furthermore, in 44% of nighttime sleep instances, residents woke up at midnight, which caused fragmented sleep intervals. From HVAC perspective, the sequential fragmented sleep intervals should be considered as one sleep instance and HVAC should not react to these short state changes. We address this problem by merging fragmented sleep periods (Section 4).

The variations of sleep and occupancy patterns of the homes can be seen in Figure 4, which indicates the percentage of sleep and away instances (with minimum duration of 2 hours) occurred in each hour of the day. Some participants sleep early at night (such as home A) and some others wake up late in the morning (such as home B). Furthermore, the sleep and away distributions overlap in some hours and there is no clear boundary between them to define a fixed



**Figure 5: The offline WalkSense outperforms the Offline-Activity baseline by 47% lower energy load and 71% reduced comfort penalty.**

sleep interval for differentiating between sleep and away periods. On the other hand, the distributions of sleep hours have longer tail in some homes, which can be due to irregular sleep patterns, or specific plan changes based on the day of the week (e.g. weekend versus weekday). These variations in different homes necessitate an occupancy model which can learn the changes of sleep/away pattern over time.

### 5.1 Offline WalkSense Performance

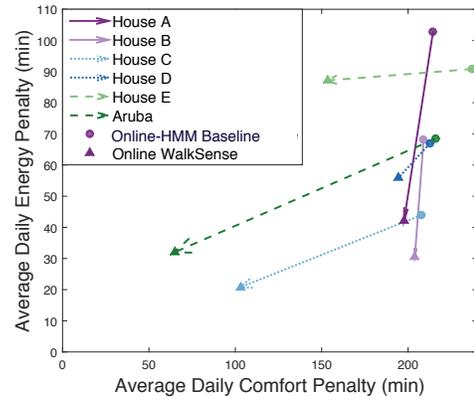
Figure 5 summarizes how the measured energy and comfort penalty metrics differ between the Offline-Activity baseline and WalkSense in 6 homes. The start point of each arrow shows the penalty values of the baseline and the end points are the corresponding values for the offline WalkSense. In all homes, WalkSense outperforms the baseline on both metrics. WalkSense saves energy by 47% with correctly differentiating sleep and away periods from active intervals. The comfort penalty also reduces 71% on average by correctly detecting low motion activities in the active zones from sleep and away intervals.

Among 6 homes, comfort penalty reduces with a small factor in home D because the occupant has a study desk in the bedroom and spent some active periods in the bedroom. Therefore, WalkSense detects these periods as sleep resulting in a small comfort issue. However, in other homes, sleeping is the only long activity that occurs in the bedroom. On the other hand, in home E, in spite of having a large improvement in saving energy, WalkSense still have higher energy penalty than in other homes. The occupant of home E woke up several times at midnight. So, a couple of sleep instances were fragmented into short periods ( $< \mathcal{K}$ ) and did not satisfy the conditions of the candidate sleep intervals.

It should be noted that the performance of the Offline-Activity baseline is highly dependent on the defined sleep hours, which makes it less applicable for homes with irregular occupancy patterns, or even for one home over time. To provide a fair comparison, we consider the best sleep interval for each house based on the historical data and the best results are illustrated in Figure 5.

### 5.2 Online WalkSense Performance

We evaluate the performance of the online WalkSense and compare it with the Online-HMM baseline in terms of en-



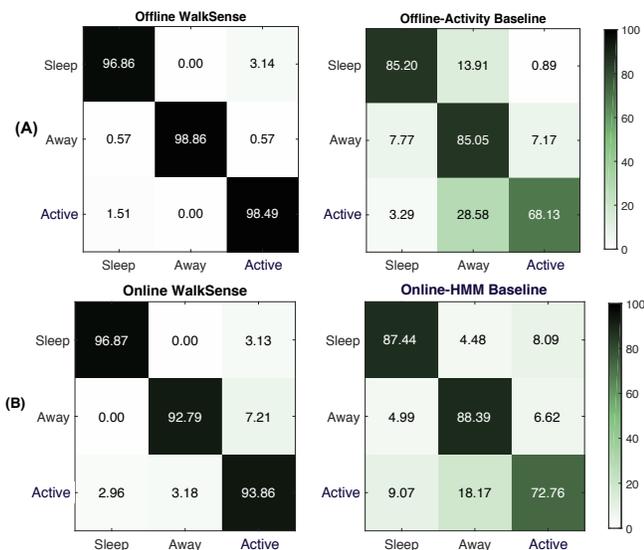
**Figure 6: The online WalkSense has 30% lower energy penalty and 32% lower comfort penalty compared with the Online-HMM baseline.**

ergy and comfort penalty metrics in Figure 6. In order to maintain the validation with the limited data, we perform leave-one-out cross validation over the number of days of the deployment in each home. The online WalkSense succeeds in decreasing comfort penalty by 32% and energy penalty by 30% on average. Compared with the offline WalkSense, the comfort penalty decreases by a small factor in online WalkSense because the proposed algorithm has a short delay in switching to the active state to avoid state fluctuations. Nevertheless, the online WalkSense in overall decreases the comfort penalty compared to the Online-HMM baseline by using rich semantic information from walkways.

### 5.3 Inference Accuracy

Figure 7 shows the performance of the sleep and away detection modules in the offline and online WalkSense and compares them with the baselines. We expect the offline WalkSense to outperform the Offline-Activity baseline, since WalkSense incorporates rich information by sensing the walkways through the sleep and away zones, in contrast to the baseline that only uses number of sensor firings in different zones. The results in Figure 7-A show that the offline WalkSense outperforms the baseline by 96.1% correctly detecting sleep, away, and active periods, compared with 79.5% in the baseline. We see that offline WalkSense misclassifies fewer sleep and away instances as active, leading to lower energy wastage. In addition, by sensing the walkways, the offline WalkSense can differentiate little motion activity periods from sleep/away intervals, resulting in lower comfort penalty.

The confusion matrices in Figure 7-B compare the online WalkSense with the Online-HMM baseline. We see that the online WalkSense can completely differentiate between sleep and away instances because it has disjoint walkway sensors with independent detection modules for sleep and away. In contrast, the Online-HMM approach misclassifies 10% of sleep and away events in place of each other. In addition, the HMM-base baseline misclassifies more active instances as sleep and away because number of sensor firings is the main used information. The other important feature of the online WalkSense is the incremental training by using offline WalkSense algorithm for automatic labeling of the training set. Analysis shows that the accuracy of the online WalkSense can improve from 94.5% to 96% by



**Figure 7: (A) The offline WalkSense correctly detects home states by 96.1% compared with the activity sensing baseline with 79.5% accuracy. (B) The online WalkSense also outperforms the smart thermostat with 94.5% accuracy compared with 82.9%.**

using the ground truth instead of offline WalkSense for labeling training set, which is almost negligible compared to the provided convenience.

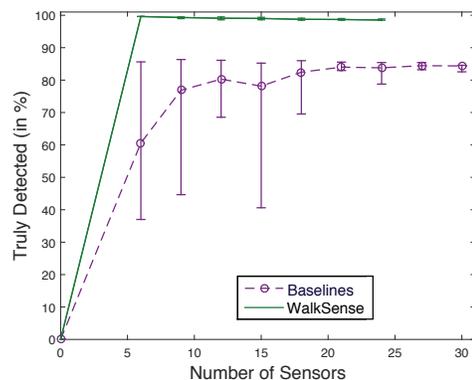
## 6. ANALYSIS

In this section, we analyze how much each component of WalkSense contributes to its accuracy and discuss the impact of sensor placements, training size and input parameters on the performance of WalkSense.

### 6.1 Sensitivity to Number of Sensors

To have a fair comparison, we used the same deployed sensors in the baseline and WalkSense in previous results. In this section, we perform a simple analysis to study the impact of the number of sensors and their locations on the performance of the baselines and WalkSense. The Aruba dataset includes 30 motion sensors located above the key items in the home such as a chair, bed, toilet, and stove, which are deployed for activity recognition purposes. We consider different combination of random sensors starting from 5 sensors to 30 sensors in total including the bedroom sensors and calculate the accuracy of the baseline and WalkSense. However, WalkSense requires 5 specific sensors shown in Figure 3 and no sensor should be placed in the bedroom. So, in WalkSense, we selected the extra random sensors from a subset of sensors (the non-bedroom sensors) and count them as active sensors.

Figure 8 shows the average accuracy of WalkSense compared with the baseline. The error bars indicate the minimum and maximum of calculated accuracies for different sensor placements. We see that the accuracy of the baseline increases by using more sensors and the increase of sensor coverage. However, it still has lower accuracy than WalkSense because it cannot differentiate the away periods from low motion activities since they both appear as periods of inactivity in the active zone. On the other hand, using more



**Figure 8: The WalkSense detection accuracy is robust even with only a small number of sensors.**

active sensors does not impact the performance of WalkSense considerably because the active sensors should only detect people in the active zone once during their presence in the active zone. Then, one active sensor is sufficient for the majority of floor plans.

### 6.2 Sensitivity to the Training Size

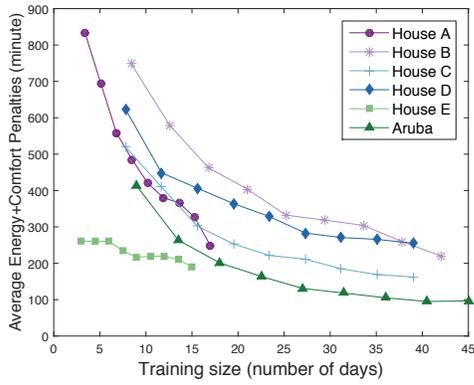
The accuracy of the online WalkSense algorithm varies with the size of the training set. The more data we have in the training set, the higher accuracy we can obtain in the online detection. We investigate the effect of training size by drawing the learning curves of 6 homes in Figure 9. The x-axis shows the number of days in the training dataset and y-axis is the sum of energy and comfort penalties. Note that homes have varying lengths of data. Based on this figure, the online WalkSense achieves the best results with around ten days to one month of training data on average. Considering that the offline WalkSense can automatically produce the training set, the incremental training can be applied to the classification models over time without any user involvement. In addition, in most of the homes, the overall penalty continues to decrease beyond one month, indicating that WalkSense can achieve even higher accuracy in the long run.

### 6.3 Sensitivity to the value of $\mathcal{K}$

Figure 10 illustrates the impact of  $\mathcal{K}$  on the energy and comfort penalties. Generally, smaller values of  $\mathcal{K}$  cause more errors and higher comfort/energy penalty because there can be more number of non-sleeping bedroom activities with the same duration. In addition, higher sensing accuracy is required in the active zones. Therefore, as  $\mathcal{K}$  increases, lower comfort penalty is resulted. However, the impact of  $\mathcal{K}$  on energy penalty is not linear. The reason is that 44% of sleep instances are fragmented into shorter periods when residents wake up at midnight. Therefore, the resulted intermittent sleep periods are filtered out in the candidate interval selection for larger values of  $\mathcal{K}$ , which increases the number of undetected sleep periods and energy wastage.

### 6.4 Sensitivity to Sleep Patterns

Homes C and D have the same floor plan, but the number of occupants and their sleep patterns are different. Therefore, by comparing these two houses, we can isolate the floor plan effect to investigate the impact of sleep patterns more



**Figure 9: Larger training set provides higher accuracy and the automatic labeling feature of the WalkSense provides incremental learning over time.**

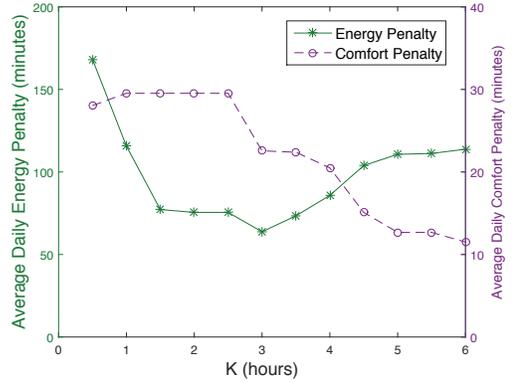
precisely. The people in home C are a professional couple with irregular sleep and away patterns, while the occupant in home D is a graduate student with regular patterns. The average accuracy of WalkSense for home C is 95% and for home D is 93.33%, which indicates that WalkSense can work properly regardless of the occupancy and sleep patterns.

In addition, in home B and E, sleep and away distributions highly overlap with each other and have larger variances (as seen in Figure 4). So, they can be considered as case studies with more irregular occupancy patterns over time. WalkSense detects sleep and away periods of these two homes with 98% accuracy, which is comparable with other case studies.

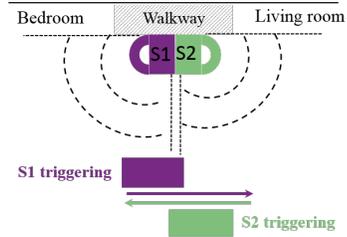
## 7. DISCUSSION

The analysis in this paper assumes that every zone always has the same heating and cooling needs, but this is not always the case. For example, some people may watch TV or use a computer for long time periods in their bed, and these activities cannot be separated from sleeping through zoning. Thus, a single zone has different heating/cooling needs at different times, depending on the activity. A similar situation may be encountered in studio apartments, where all activities occur in the same room. In such situations, extra sensors are required to differentiate these states. For simplicity, the analysis in this paper also assumes that a hallway can be instrumented as a walkway to the sleep zone, even if it is connected to a bathroom because the bedroom is the only room where the occupants stay longer than  $\mathcal{K}$  minutes. However, this assumption may not work for smaller  $\mathcal{K}$  values or for all floor plans. In these situations, the walkway sensors should be installed directly on the doorways of bedrooms. Otherwise, the sensor lenses could be occluded to limit their field of view, as needed.

One key limitation of WalkSense is what we call *multi-person ambiguity*: two or more people in different zones causing the detection of an erroneous occupancy state. For example, in a multi-person home, if one occupant is sleeping and a second occupant transitions from active to away, the household should transition from active to sleep but no sleep transition would be detected. Our results indicate that, although such ambiguities are possible, the timing requirements make them exceedingly rare. That being said, however, more experiments are required to fully understand the



**Figure 10: Smaller values of  $\mathcal{K}$  causes more energy wastage and comfort issues because of higher sensing sensitivity in the active zone.**



**Figure 11: The B2B sensor package can detect the direction of occupants traveling through walkways.**

robustness of WalkSense over longer time periods and in a more diverse set of multi-person homes.

## 8. FUTURE WORK

The concept of walkway sensing can be implemented by using different types of sensors. In this paper, WalkSense uses standard off-the-shelf motion sensors, but we also design a custom motion sensor package called “Back-to-Back (B2B)” where a single micro-controller connects to a pair of PIR sensors such as Figure 11. The B2B design provides information about the walking direction into or out of the sleep/away zone, which results in lower required sensing accuracy in active zone and higher accuracy of the online detection module. To determine the walking direction, the B2B sensor will be installed in the sleep/away walkway such that one PIR observes the active zone and the other observes the sleep/away zone as shown in Figure 11. Hence, occupants passing through the walkway would enter into both sensors’ fields of view in sequence. We created a prototype of B2B package and installed in home A. The preliminary results show that B2B design outperforms the commercial PIR sensors. In addition, with B2B sensors, we can apply walkway sensing in commercial buildings to identify room occupancy. In our future work, we plan to design a low power B2B sensor network and test it in residential and commercial buildings.

## 9. CONCLUSIONS

In this paper, we present the concept of walkway sensing to accurately detect and differentiate home occupancy states as sleep, away, and active. We also propose Walk-

Sense as an implementation of the walkway sensing, which is cost-effective by requiring small number of sensors and non-intrusive by not requiring user involvement. The WalkSense algorithm operates in two modes of offline and online, which provides the inference occupancy information for both types of self-programming and smart thermostats. In addition, the offline WalkSense can be efficiently used for automatic labeling of the training set required in the online WalkSense. We evaluate the proposed system by analyzing the data traces of 6 homes. The results show that the proposed system outperforms the existing solutions by providing the average accuracy of 96% in offline mode and 94% in online mode. We believe that the proposed system will show more significant improvements if it applies for bigger floor plans with irregular occupancy patterns. Currently, we are working on a new design of the walkway sensing approach, which provides the walking direction information and can be used in commercial buildings.

## 10. REFERENCES

- [1] Environmental protection agency. summary of research findings from the programmable thermostat market. washington, dc: Office of headquarters. 2004.
- [2] A. M. Adami, A. G. Adami, G. Schwarz, Z. T. Beattie, and T. L. Hayes. A subject state detection approach to determine rest-activity patterns using load cells. In *Engineering in Medicine and Biology Society (EMBC)*, pages 204–207. IEEE, 2010.
- [3] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng. Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 1–6. ACM, 2010.
- [4] D. Austin, Z. T. Beattie, T. Riley, A. M. Adami, C. C. Hagen, and T. L. Hayes. Unobtrusive classification of sleep and wakefulness using load cells under the bed. In *Engineering in Medicine and Biology Society (EMBC)*, pages 5254–5257. IEEE, 2012.
- [5] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal. Sentinel: occupancy based hvac actuation using existing wifi infrastructure within commercial buildings. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 17. ACM, 2013.
- [6] M. Borazio and K. Van Laerhoven. Combining wearable and environmental sensing into an unobtrusive tool for long-term sleep studies. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 71–80, 2012.
- [7] D. Chen, S. Barker, A. Subbaswamy, D. Irwin, and P. Shenoy. Non-intrusive occupancy monitoring using smart meters. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8. ACM, 2013.
- [8] D. J. Cook. Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems*, 2010(99):1, 2010.
- [9] V. L. Erickson, M. Á. Carreira-Perpiñán, and A. E. Cerpa. Observe: Occupancy-based system for efficient reduction of hvac energy. In *10th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 258–269. IEEE, 2011.
- [10] K. Farrahi and D. Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):3, 2011.
- [11] G. Gao and K. 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*, pages 67–72. ACM, 2009.
- [12] M. Gupta, S. S. Intille, and K. Larson. Adding gps-control to traditional thermostats: An exploration of potential energy savings and design challenges. In *Pervasive Computing*, pages 95–114. Springer, 2009.
- [13] W. Kleiminger, C. Beckel, T. Staake, and S. Santini. Occupancy detection from electricity consumption data. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8. ACM, 2013.
- [14] N. C. Krishnan and D. J. Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.
- [15] J. Krumm and A. B. Brush. Learning time-based presence probabilities. In *Pervasive Computing*, pages 79–96. Springer, 2011.
- [16] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, and K. Whitehouse. The smart thermostat: using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 211–224, 2010.
- [17] S. Makonin, L. Bartram, and F. Popowich. A smarter smart home: Case studies of ambient intelligence. *IEEE Pervasive Computing*, (1):58–66, 2013.
- [18] S. Makonin and F. Popowich. Home occupancy agent: Occupancy and sleep detection. *Journal on Computing (JoC)*, 2(1), 2014.
- [19] M. Pal and P. M. Mather. An assessment of the effectiveness of decision tree methods for land cover classification. *Remote sensing of environment*, 86(4):554–565, 2003.
- [20] D. Pisharoty, R. Yang, M. Newman, and K. Whitehouse. Thermocoach: Reducing home energy consumption with personalized thermostat recommendations. In *Proceedings of the 2th ACM International Conference on Embedded Systems For Energy-Efficient Buildings*. ACM, 2015.
- [21] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Pervasive Computing*, pages 152–169. Springer, 2011.
- [22] J. Scott, A. Bernheim Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, and N. Villar. Preheat: controlling home heating using occupancy prediction. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 281–290. ACM, 2011.
- [23] S. Tominaga, M. Shimosaka, R. Fukui, and T. Sato. A unified framework for modeling and predicting going-out behavior. In *Pervasive Computing*, pages 73–90. Springer, 2012.
- [24] P. E. Utgoff. Incremental induction of decision trees. *Machine learning*, 4(2):161–186, 1989.