ARTICLE IN PRESS

Applied Energy xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Practical occupancy detection for programmable and smart thermostats

Elahe Soltanaghaei*, Kamin Whitehouse

University of Virginia, Charlottesville, USA

HIGHLIGHTS

- Walkway Sensing as a new principle for using motion sensors to infer occupancy.
- It relies on motion sensors to only detect occupancy in the walkways between zones.
- Walkway sensing is converted into a reliable form of zone occupancy detection.
- The detection model called WalkSense operates in two modes of offline and online.

ARTICLE INFO

Keywords:
HVAC
Energy saving
Occupancy detection
Motion sensor
Walkway sensing

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 implement our system using two types of sensors and evaluate them 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.

1. Introduction

Heating and cooling are the largest sources of residential energy consumption, accounting for an estimated 48% of energy consumption in US homes.[1] However, 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 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 using zone occupancy information to adjust the heating/cooling.

Studies have shown that energy usage can be reduced 20–30% by reducing heating and cooling when residents are asleep or away [1–3]. On a national scale, this would correspond to annual saving of 112 billion kW h in the US and would prevent the emission of approximately 1.2 billion tons of air pollutants [4]. However, studies have shown that occupants do not adjust their thermostats often enough to fully realize these savings [1,5]. To address this problem, the self-programmable [6] and smart thermostats [4] are designed to automatically learn occupants’ schedules and turn the heating and cooling on or off on the user’s behalf. However, the ability to achieve these benefits is limited by lack of sensing technologies that can reliably detect occupancy states [7–9].

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 [10,4] 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

* Corresponding author.
E-mail addresses: es3ce@virginia.edu (E. Soltanaghaei), whitehouse@virginia.edu (K. Whitehouse).

[1] http://www.eia.gov/consumption/residential/.

https://doi.org/10.1016/j.apenergy.2017.11.024
Received 1 August 2017; Received in revised form 17 October 2017; Accepted 2 November 2017
0306-2619/

Please cite this article as: Soltanaghaei, E., Applied Energy (2017), http://dx.doi.org/10.1016/j.apenergy.2017.11.024

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 [11–14]. Other studies define sleep to be all periods of inactivity between certain hours, such as 10 pm–8 am [14,15], 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.[2]

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 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.

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 design an occupancy detection algorithm called *WalkSense*, 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. 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.

The key challenge in the online WalkSense is learning the occupancy pattern changes using the training data. In general, supervised methods use historical data labeled manually for training, which is a time-consuming process. However, we require a dynamic model with continuous learning to identify changes in occupancy patterns of different zones. In this case, using the previous manual methods for annotation requires a constant user involvement. We address this challenge by designing an automatic labeling procedure which is built upon the offline WalkSense. The key insight is that the offline WalkSense uses all data before, after, and during an entire interval and can determine the state of the interval with very high certainty, which is suitable for annotation. Therefore, we iteratively derive the past sensor data and calculate their labels using the offline WalkSense.

We implement WalkSense using two types of sensors. The first implementation uses standard off-the-shelf motion sensors. Time stamp errors due to clock drifts is the main challenge, where we address that by designing a synchronization method. The second implementation called "Back-to-Back (B2B)" is a custom motion sensor package where a pair of PIR sensors are attached in two sides, one covering the walkway zone, and the other one covering the active zone. This design provides additional information about the walking direction into or out of the sleep/away zones. In addition, co-locating the sensors of the two zones provides the opportunity to use a common clock for both sensors through a single micro-controller to solve the issue of time synchronization.

We evaluate offline and online WalkSense with the two implementation designs 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 [16]. 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 [10]. 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 [4]. In addition, analysis shows that 12% of detected sleeping instances are daytime napping and 11.7% of away periods happened at night, between 9 pm and 3 am, which highlights the robustness of the proposed method to irregular sleep and away patterns.

2. Background and related work

2.1. HVAC control systems

The standard goal of an HVAC control system is to keep temperature and air quality in a comfort range while minimizing energy usage. Programmable Thermostats or Rule-Based Control (RBC) systems are one of the most conventional control practice, which schedule different setpoint temperatures at different times throughout the day. However, the expected energy saving is premised on the ability of the occupants in defining the schedules that match the home occupancy patterns. This can be difficult specially for homes with multiple occupants with irregular occupancy patterns. Studies have shown that the risk of comfort loss causes people to reduce their use of setpoint schedules during unoccupied or sleeping periods [17]. Therefore, the setpoint temperature is usually above the safety limit of the house to reduce the risk of comfort loss, thus causing more energy consumption even when the home is vacant. A number of papers proposed the self-programming thermostat to fix this problem by automatically choosing the optimal setback schedules based on occupancy statistics [6,18]. However, this approach still generates static schedules and doesn't react to the dynamic occupancy changes throughout the day.

An alternative approach is to use Reactive Thermostats, which use different sensing modules to turn the HVAC on and off based on occupancy [19,20]. However, they are limited by the lack of a reliable occupancy sensing approach and an accurate occupancy detection algorithm. In addition, HVAC system and thermal properties of the home result in a certain time lag until the temperature setpoint is reached. Therefore, the inability of reactive thermostats to quickly respond to occupancy changes limits their potential energy saving. To address this problem, Smart Thermostats [11,4,21–23] and Model Predictive Control (MPC) systems [24–26] are proposed, which use the current and historical occupancy data to solve an optimal control problem for a finite prediction horizon, and automatically turning off and on the home's HVAC system. Studies have shown that this approach achieves a 28–35% energy saving on average [4]. All of these occupancy-based HVAC control systems require a reliable occupancy sensing system to correctly infer the occupancy data. This paper proposes a new occupancy sensing principle which is compatible with different HVAC control systems including programmable and smart thermostats. In

[2] http://www.cdc.gov/niosh/topics/workschedules/.

addition, we design the detection module in two modes of offline and online so that both programmable and smart thermostats can take benefit of the proposed sensing principle.

2.2. Occupancy detection in residential homes

Several studies have explored new technologies to detect occupancy in residential houses using PIR motion sensor [27,28], electricity usage [15], radio frequency identification (RFID) sensors [11,14,29,30], GPS data [31], WiFi signals [32–34], network usage [35], or light pressure sensors [17]. However, they all ignore sleep periods and nighttime occupancy. For example, PreHeat [11] and ThermoCoach [14] 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 [4] 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 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 [34–39]. 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 [38,41], 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.

2.3. Occupancy detection in commercial buildings

Various sensing technologies have been used to detect occupancy in commercial buildings to differentiate between occupied and unoccupied states in room zones. Most approaches combine measurements of motion sensors with CO_2 [42] and humidity [43] sensors to provide occupancy information [44]. However, these data fusion approaches are suitable for small size rooms with a large number of occupants. In addition, they easily miss the presence of the occupants that are sitting still in the room. Cameras and WiFi data are other techniques used to detect the presence of occupants [18,45], but these methods are not practical for residential buildings because of privacy concerns or difficulty in differentiating sleep and away periods. Wang et al. [46] exploits Bluetooth Low Energy (BLE) networks to assess occupant distribution in a large space by leveraging the fingerprints of the received signal strength indicator (RSSI) from BLE beacons for small patches of the physical space. However, this method requires a large database of all occupied scenarios in different locations, which is practically impossible due to random movement behavior of occupants.

2.4. Contributions

In summary, the contributions of Walkway Sensing are:

- To the best of our knowledge, Walkway Sensing is the first practical solution that can detect and differentiate sleep and away inactivity occupancy states.
- Based on the proposed sensing principle, we design an occupancy detection algorithm called WalkSense, which works in two modes of offline and online and provides occupancy data for any types of HVAC control system.
- We propose a novel dynamic annotation model and iterative learning system to identify changes in occupancy patterns of different zones. The proposed method relies on features of offline WalkSense in extracting occupancy information in the historical data.
- We implement WalkSense using two types of sensors: standard off-the-shelf motion sensors, which shows the compatibility of the proposed principle with available sensor modules; and a custom design sensor called "Back-to-Back (B2B)", which provides additional information about the walking directions in the sensing walkway areas, thus improving the inference accuracy.
- The proposed method is evaluated on 350 days worth of data from 6 homes. Results indicate that we achieve over 47% and 30% reduced energy in offline and online modes respectively. In addition, compared to the conventional solution, the comfort loss is reduces to 71% and 30% in average.

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 correlate 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 [10], 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 occupancy zones such as bedroom and exit door as shown in Fig. 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 Fig. 1(b). In addition, to differentiate

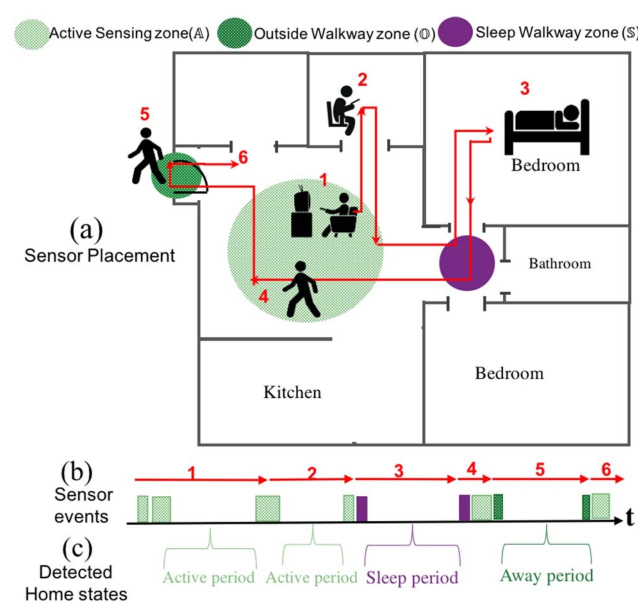


Fig. 1. (a) Sensor Placements in WalkSense. (b) The sensor reading timeline as the occupant performs activities 1–6. (c) Walkway events indicate the start and end of sleep and away periods and differentiate them from low motion active periods.

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. Fig. 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 [6,14]. 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 Fig. 1(c)) in an algorithmic fashion explained in Section 3.2. 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 supervised 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 ($\mathbb{S}$)", "Outside Walkway ($\mathbb{O}$)" and "Active ($\mathbb{A}$)" in each floor plan (as seen in Fig. 1(a)). $\mathbb{S}$ is the transition area between the active zone and the sleep zone, and $\mathbb{O}$ is the transition area through which occupants regularly walk for going out. We define $\mathbb{A}$ 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:

- $\mathbb{S}$ must have a non-empty area not covered by $\mathbb{A}$ or $\mathbb{O}$.
- $\mathbb{O}$ must have a non-empty area not covered by $\mathbb{A}$ or $\mathbb{S}$.
- $\mathbb{A}$ must cover people who are not in the bedroom or outside at least once during their presence in the zone.
- 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 Fig. 1), one sleep walkway zone ($\mathbb{S}$) 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 $\mathscr{K}$ 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 $\mathscr{K}$ 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 $\mathscr{K}$ 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 $\mathscr{K}$ as the minimum duration of a sleep or away state and we expect reasonable values of $\mathscr{K}$ to be between 60 and 150 min [4]. Fig. 2 shows an example of sensor events for a day in three zones. In addition, the candidate sleep and away intervals are represented based on the above definition. If there is a sensor triggering between any of these candidate intervals, that interval will be filtered out. The dotted red lines depict the detected sleep and away intervals during this day.

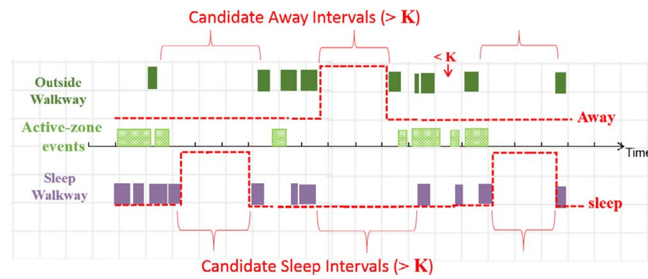


Fig. 2. Sleep and away intervals (shown in dotted lines) are the candidate intervals with no other sensor events in between.

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 $\mathbb{S} = \{s_1,s_2,\ldots,s_{n_1}\}$, where s_i is a motion sensor installed in the sleep walkway. Similarly, the sets of outside walkway and active sensors are defined as $\mathbb{O} = \{o_1,o_2,\ldots,o_{n_2}\}$, $\mathbb{A} = \{a_1,a_2,\ldots,s_{n_3}\}$, where $n_1,n_2,andn_3$ 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,

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

where s_i, s_j, and p are the triggered sensors at time stamps t_{s_i}, t_{s_j}, and t_p, respectively. It should be noted that in a home with multiple occupants, a sleep period is an interval that all occupants are asleep. Therefore, if one of the occupants is asleep while others are active, the above conditions won't be automatically satisfied due to sensor triggering in the active zone and the period will be identified as active. In addition, the sleep period starts from the last triggered walkway sensor to the first one triggered at the end of the sleep period. So, the two walkway events that define a sleep period might not belong to the same walkway zone.

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,

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

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 Fig. 3, 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

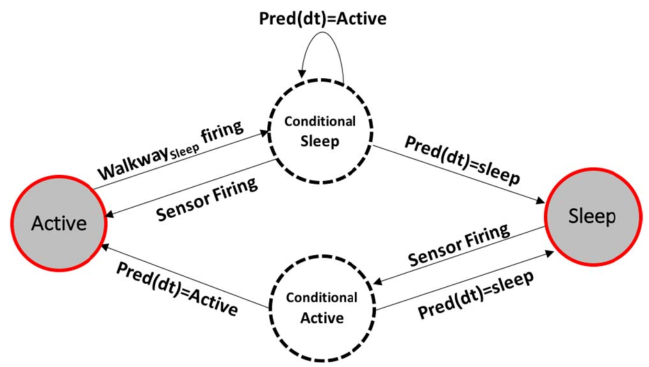


Fig. 3. The online WalkSense follows a state machine to switch between *Active* and *Sleep* (or *Away*) states based on the output of a supervised learning algorithm.

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 τ minutes until one of these conditions happens: (i) a sensor triggers, which indicates that someone is active, 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 Fig. 3, 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 $dt = (t_w, t_w + c\tau), c \geqslant 1$, 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 τ 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.

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

Temporal Features	weekday/weekend day of week time of day at 2-h 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

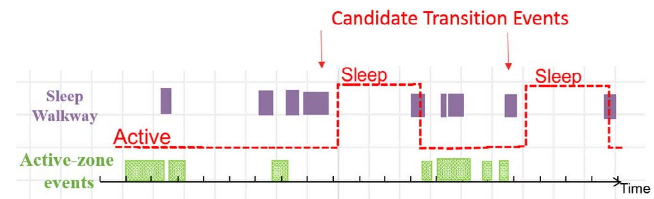


Fig. 4. The online WalkSense tracks sensor events in real-time to find the candidate transition events and it predicts potential state changes using a trained classifier.

Fig. 4 shows an example of the online WalkSense algorithm on sleep detection for a set of sensor events. The candidate transition events are the walkway events with a gap of no sensor event after them (for duration τ). The classifier will be executed for these candidate events which could result in occupancy state change. The red dotted line shows the final occupancy states that are set by the state machine. As can be seen in the figure, there is a short delay for each state change which is due to the conditional states to avoid unnecessary state changes due to false sensor triggering.

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 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 [47] 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. Implementation

WalkSense could be implemented using different types of sensors each with a different trade-off between cost, and performance. In this paper, we explore two possible implementations using motion sensors. The first one uses standard off-the-shelf motion sensors. The main challenge for this implementation is the time-stamp error since the sensors do not have common time stamps and even wireless versions can experience packet loss. To address this challenges we used a sliding method which will be explained in Section 4.1. In the second implementation, we create a custom motion sensor package called "Back-to-Back (B2B)" where a pair of PIR sensor is attached in two sides, one covering the walkway zone, and the other one covering the active zone. This design provides additional information about the walking direction in walkway areas. In addition, the co-location of the sensors in the two zones provides an opportunity to use a common clock for both sensors through a single micro-controller to provide time synchronization and eliminates the issue of packet loss. It also simplifies the deployment process. The details of this implementation is explained in the Section 4.2.

4.1. Standard motion sensors implementation

In this implementation, we use standard motion sensors with local storage to avoid packet loss and transfer delay. However, the sensors deployed in different zones operate independently and their local clocks may not be synchronized with each other due to possible clock skew and drift. The common clock accuracy in off-the-shelf sensors is around ± 20 ppm, which results in 1.7 s of time offset in 24 h. This can cause false detection since we require to integrate the information sensed at different nodes. To address this issue, a clock synchronization mechanism is required to compensate the resulted skews and clock drifts by providing a common notion of time.

Many existing synchronization algorithms use the time information from GPS. However, GPS has a relatively high power demand, which is not available in small cheap motion sensors. Besides hardware-based solutions, the widely-used software-based synchronization method on the Internet is the Network Time Protocol (NTP) [48] due to its scalability, self-configuration, and robustness to failures. However, NTP is not suitable for a wireless sensor network since it could increase the collision of the synchronization packets and the wireless network due to limited bandwidth, which results in higher errors. In sensor networks, there are two types of synchronization algorithms based on the required level of accuracy. The first form of synchronization deals only with ordering of events, while the second one relies on a message exchange to maintain a relative clock by keeping the information about the relative drift and offset of the clocks between the nodes. We build our synchronization algorithm upon these methods by focusing on the special need of our network to disambiguate time-stamps of the active and walkway sensors.

Our disambiguation method relies on the resulted time offset due to clock skew and drifts on time stamps of sequential sensing events. A skew in the clock signal causes a time gap between the expected arrival of the clock and its actual arriving which results in a constant time offset over time. On the other hand, clock drift are caused by clocks counting time at slightly different rates which results in an accumulated time offset over time. However, this rate difference translates into a negligible relative offset between two sequential sensing events. Therefore, we can assume that every two sequential sensor events are shifted in either directions due to the time-stamp offsets, but the time gap between them remains constant. Based on this intuition, we propose a sliding technique for detecting inactive periods in ambiguous scenarios.

In offline WalkSense, the active-zone sensor requires to trigger only once during the presence of the occupants in this zone for an accurate occupancy detection. Therefore, the time-stamp offset between the active and walkway sensors causes ambiguity if it is larger than the minimum time that a person could walk from active zone to any of the walkways. In other words, time synchronization will be necessary if the active and walkway sensors are very close to each other or if their sensing coverage overlaps. The worst case example is shown in Fig. 5, where each sleep/away interval is surrounded by active events. Therefore, any small time offset will cause ambiguity, which results in false detection of actual sleep/away periods as active.

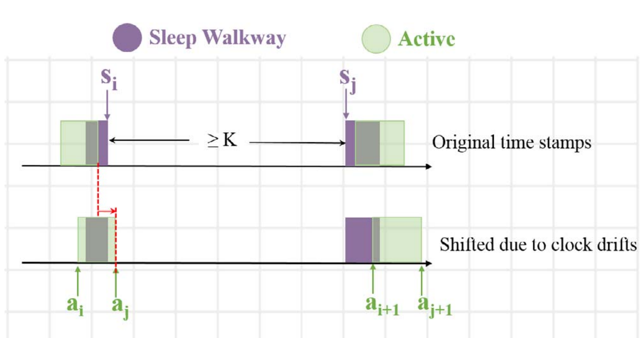


Fig. 5. The time-stamp offsets due to the clock skew and drift of the sensors shifts the sensing events and could cause ambiguity in detecting sleep/away intervals.

To address this issue, we first require to identify these ambiguous scenarios. Let's consider a candidate sleep interval as (t_{s_i},t_{s_j}), where t_{s_i} and t_{s_i} denote the beginning and end of the corresponding interval. The two active events around this interval are represented as (t_{a_i},t_{a_j}) and $(t_{a_{i+1}},t_{a_{j+1}})$, where the first tuple shows the beginning of sensor triggering and the second one shows the end of triggers. It should be noted that off-the-shelf motion sensors merge sequential triggers with a minimum gap to reduce the number of required communications and storage. A candidate sleep/away interval (t_{s_i},t_{s_j}) is defined as ambiguous if it is surrounded by two active events and has time overlap with them. Formally,

$$\exists\ (t_{a_i},t_{a_j}) \wedge (t_{a_{i+1}},t_{a_{j+1}}) | (t_{a_i} < t_{s_i} < t_{a_j} \wedge t_{s_j} < t_{a_{i+1}}) \vee (t_{a_{i+1}} < t_{s_j} < t_{a_{j+1}} \wedge t_{s_i} > t_{a_i}) \tag{3}$$

To disambiguate these cases, we rely on the constant gap between two sequential events in the existence of clock drifts. Therefore, if an ambiguous interval is sleep/away, it should be longer than the interval between the two sequential active events surrounding it since there shouldn't be any active sensor event within the sleep/away interval. In other words, the ambiguous sleep interval (t_{s_i},t_{s_j}) will be defined as sleep if

$$|t_{s_j} - t_{s_i}| < |t_{a_{i+1}} - t_{a_j}| \tag{4}$$

This is equivalent to sliding the ambiguous sleep interval in either directions in the boundary of the active interval surrounding the sleep event. If at least one position is found where the active logs are placed outside of the sleep interval, the ambiguous interval will be labeled as sleep. It should be noted that the same conditions will be applied to candidate away interval to find the ambiguities and make the decision based on the relative time-stamp values.

Addressing time-stamp offsets in online WalkSense is much easier, since the decision is made for every time window τ instead of the whole interval. Therefore, the time stamp offset only delays the transition to sleep/away state. To compensate this delay, we consider a safe boundary around the transition interval dt (defined in Section 3.3) to make sure that the potential time-stamp offsets won't filter out sleep intervals. Instead, the state machine remains in the *Conditional Sleep/Away* state for a longer duration until gaining more information from the next time windows.

4.2. Back-to-Back Implementation

In this implementation, we design a custom motion sensor package called "Back-to-Back (B2B)" where a pair of PIR sensors are attached back to back, one covering the walkway zone, and the other one covering the active zone. This design provides information about the walking direction into or out of the sleep/away zone, which results in reduced sensing demand in active zone and higher accuracy of the online detection module. To determine the walking direction, the B2B sensors will be installed in the sleep/away walkways such that one PIR sensor observes the active zone and the other one observes the sleep/away walkway zone as shown in Fig. 6. Occupants passing through the walkway would enter into both sensors' fields of view in sequence

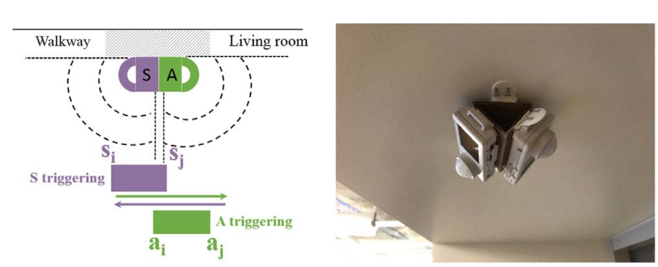


Fig. 6. (Left) The direction of occupants passing from the walkway can be detected by B2B sensing modules based on the order of logs generated in the sensors, (Right) the B2B sensor installed on a bedroom doorway.

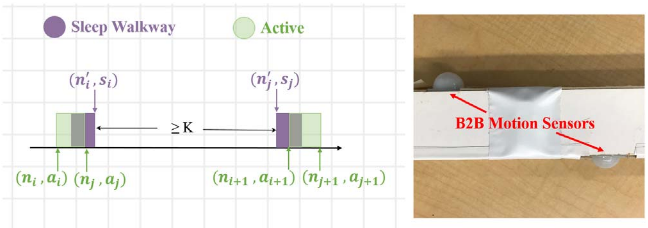


Fig. 7. (Left) The B2B sensing module uses both time-stamps and counter numbers to detect sleep/away intervals in the presence of loss. (Right) Using a single micro-controller connected to both sensors eliminate time synchronization issues.

which allows us to detect the movement direction based on the order of occurrence of sensor events. Let's consider (t_{a_i}, t_{a_j}) and (t_{s_i}, t_{s_j}) as the active and sleep walkway sensing events, respectively, for a crossing sample. The crossing will be a potential sleep transition if

$$t_{a_i} < t_{s_i} \wedge t_{a_j} < t_{s_j} \tag{5}$$

However, this design is more sensitive to packet loss and time-stamp offsets due to the small time gap between sensing events of the two zones. Therefore, the previous solution for time synchronization cannot disambiguate all of the sleep/away intervals. To deal with this challenge, we design a new custom package in which the two B2B motion sensors are connected to one micro-controller, thus using the same clock (shown in Fig. 7). Therefore, the time-stamps of the two sensors in a walkway are synced with each other. In addition, we define a software-base counter which assigns a sequence number to each sensor event to deal with the packet loss. Therefore, we can firstly recognize the occurrence of a packet loss and then interpolate the time stamp of a lost packet based on the previous and next packet in the sequence. In summary, a candidate sleep interval (t_{s_i}, t_{s_j}) is identified as sleep state if the following condition is satisfied:

$$(t_{s_i} < t_{a_j} \wedge n_i' < n_j) \vee (t_{s_j} > t_{a_{i+1}} \wedge n_j' > n_{i+1}) \tag{6}$$

where (t_{a_i}, t_{a_j}) shows the active-side sensor event and n is the assigned sequence numbers for the start and end of the active-side triggers. n' shows the corresponding sequence numbers assigned to the sleep-side sensor events. The away intervals will be identified similarly.

4.3. Intermittent sleep/away intervals

Based on our observations with extensive in-situ experiments, a sleep or away period may be interrupted by short presence of the occupants in the active zone for example awaking at midnight to drink water, or to go to the bathroom, or getting back home to pick an item left behind. This could cause unnecessary reactions of the heating and cooling system if the time lag of the HVAC system is more than the duration of the active interval. In addition, the fragmented parts of a large sleep/away period may not satisfy the threshold value $\mathscr{K}$ as the minimum duration defined in offline WalkSense, thereby getting filtered out. To address this challenge, we consider a merging mechanism which looks for sequential sleep/away periods that are shorter than $\mathscr{K}$ individually, but satisfy the minimum duration if combined. In addition, we define a threshold $\mathscr{T}$ for the time gap between two fragmented intervals to avoid comfort issues. Therefore, the two candidate fragmented intervals will be merged with each other if the active interval between them is shorter than $\mathscr{T}$. Based on our exploratory analysis, we consider $\mathscr{T} = \frac{\mathscr{K}}{2}$. It should be noted that the fragmented sleep/away intervals do not cause any problem in online WalkSense due to defined conditional sleep/away states, which are originally designed to deal with aggressive HVAC reactions to short transitions.

5. 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 [16] 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 Fig. 8 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. In addition, Home A has both the standard and B2B implementations at the same time for comparison. 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 7.1.

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 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 min. 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 $\mathscr{K}$, which shows the

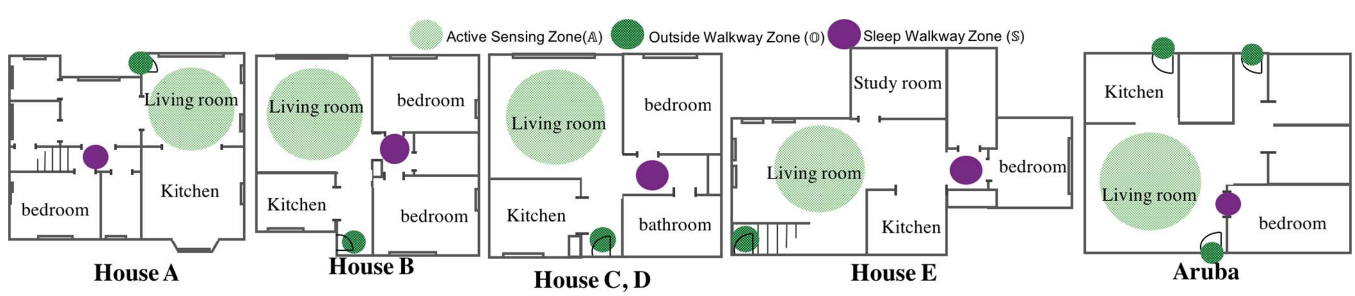


Fig. 8. Floor plan of the six homes in the study with the placement of deployed sensors in each home.

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

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

minimum length of sleep or away intervals. We select the value of 2 h for the main results and investigate the impact of $\mathscr{K}$ on the WalkSense performance in Section 7.5. 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 $\mathscr{K}/2$ 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) [49,50] and its capability of finding the best split points of numerical features for multi-class problems [51,52]. To evaluate the models, we also use leave-one-out cross validation.

5.1. Baseline

Both the offline and online WalkSense approaches are compared to existing methods that aim to fulfill their respective goals [10,4]. 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 7.1. 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 [10], 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 schedules (e.g. 10 PM–8 AM) as in programmable thermostats. For each house, different schedules are tested on the historical data and the optimal schedule is selected to differentiate sleep and away intervals. 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) [4], 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 [4]. To consider a similar comfort/energy sacrifice with the Online-HMM baseline, we select ten-minute intervals for the value of τ (the wait time between *Conditional Sleep/Away* loops) in the online WalkSense.

It should be mentioned that the PreHeat approach [11] 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.

5.2. 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 (Eq. (7)). *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 (Eq. (8)). 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:

$$EnergyPenalty = \frac{1}{d}\sum_{i=1}^{d}(t_i^{OasA} + t_i^{SasA} + t_i^{OasS}) \tag{7}$$

$$ComfortPenalty = \frac{1}{d}\sum_{i=1}^{d}(t_i^{AasO} + t_i^{AasS} + t_i^{SasO}) \tag{8}$$

where d is the total number of experiment days for each 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. It should be noted that the energy and comfort penalty metrics are defined in a general format, so the results could be applied to any types of HVAC control system. The actual energy saving is a function of the setback temperature, the properties of the home (e.g. size of the home) and HVAC system (e.g. the thermal lag). Therefore, we keep the energy and comfort metrics as a percentage number for the generality purposes.

We also compare the performance of the inference algorithm with the baselines in Section 6.3 using confusion matrices. 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. In addition, we define *detection rate* as a metric to show the overall performance of the proposed methods in correctly detecting occupancy states of sleep, active, and away. Therefore, detection rate is the fraction of time that the three occupancy states are correctly detected among the total duration of experiments averaged between houses:

$$DetectionRate = \sum_{i=1}^{N}\frac{t_i^{SasS} + t_i^{OasO} + t_i^{AasA}}{t_i^{total}} \tag{9}$$

where t_i^{SasS},t_i^{OasO},and t^{AasA} are the fractions of time that Sleep, Outside, and Active instances of house i are correctly detected, respectively. t_i^{total} shows the total duration of the experiment in house i.

Finally, we define a merging mechanism in the implementation section to combine the fragmented sleep or away intervals due to short presence of people in the active zone. This will avoid unnecessary reactions of HVAC to short state transitions. To study the effectiveness of this mechanism, we define a new metric called *Miss-Interval Rate (MIR)*, which calculates the percentage of the sleep/away intervals that are not detected due to intermittent intervals. In other words, the average of undetected sleep/away intervals that are shorter than $\mathscr{K}$,

$$MIR = \frac{count(I_f^{SasA}) + count(I_f^{OasA})}{N_f + M_f} \tag{10}$$

, where $ISasA_f$ is the number of fragmented sleep intervals ($\frac{\mathscr{K}}{2} < I_{SasA_f} < \mathscr{K}$) that are wrongly detected as active, and $IOasA_f$ is the number of fragmented outside instances ($\frac{\mathscr{K}}{2} < I_{OasA_f} < \mathscr{K}$) that are wrongly detected as active. N_f and M_f show the total number of

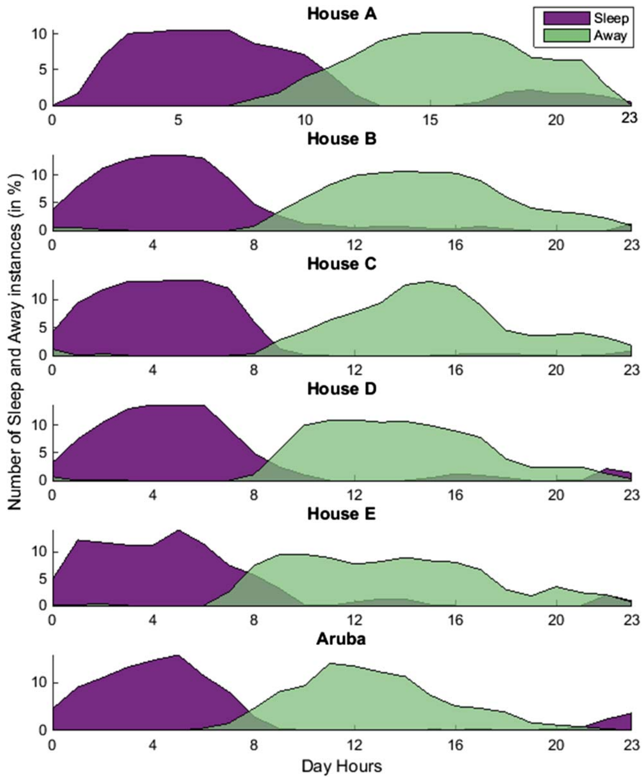


Fig. 9. The sleep and away instances overlap in different times due to irregular occupancy patterns. Therefore, defining a sleep interval for differentiating sleep and away periods is expected to fail.

fragmented sleep and outside instances, respectively.

6. 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 h and 11.7% of away instances are between 9 PM to 3 AM with average duration of 3.17 h. 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 5).

The variations of sleep and occupancy patterns of the homes can be seen in Fig. 9, which indicates the percentage of sleep and away instances (with minimum duration of 2 h) 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 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.

6.1. Energy vs. comfort penalty: offline WalkSense

Fig. 10 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. On the other hand, the results indicate that in average WalkSense misses only 63 min of potential energy saving conditions per day. This reflects into 4% of total time in a day and implies that WalkSense achieves near to optimal energy saving.

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 ($<\mathscr{K}$) 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 Fig. 10.

6.2. Energy vs. comfort penalty: online WalkSense

We evaluate the performance of the online WalkSense and compare it with the Online-HMM baseline in terms of energy and comfort penalty metrics in Fig. 11. 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. Online WalkSense further reduces the daily missing energy saving situations to 44 min in average, which corresponds to 3% of daily time. 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.

6.3. Inference accuracy

Fig. 12 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 Fig. 12-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

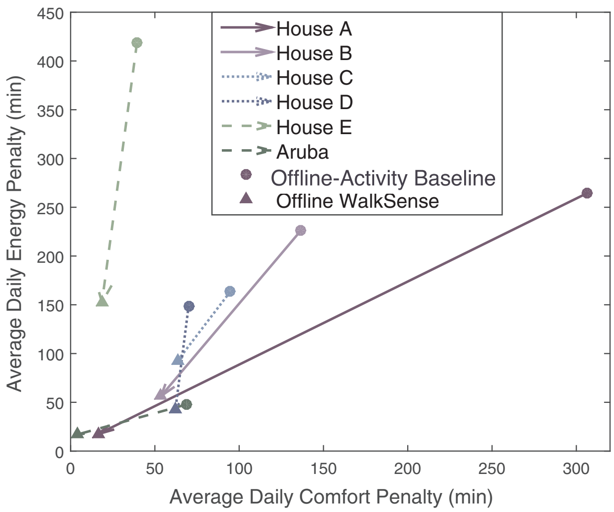


Fig. 10. The offline WalkSense outperforms the Offline-Activity baseline by 47% lower energy load and 71% reduced comfort penalty.

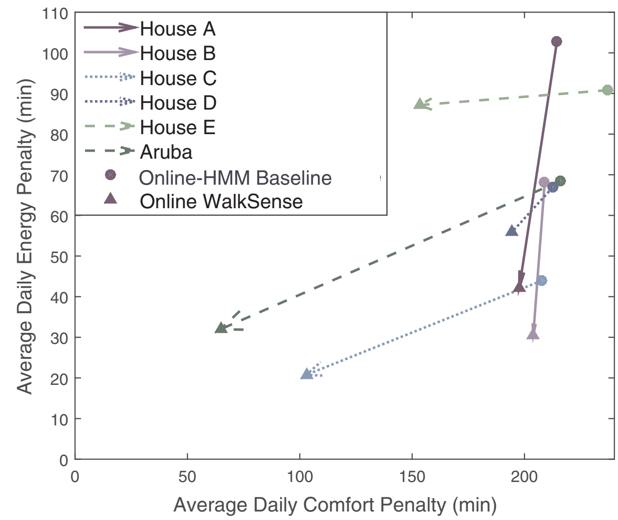


Fig. 11. The online WalkSense has 30% lower energy penalty and 32% lower comfort penalty compared with the Online-HMM baseline.

from sleep/away intervals, resulting in lower comfort penalty.

The confusion matrices in Fig. 12-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 using the ground truth instead of offline WalkSense for labeling training set, which is almost negligible compared to the provided convenience.

6.4. Implementation accuracy

In this section, we compare the two implementations of WalkSense in house A: the standard off-the-shelf motion sensors (COTS), and Back-to-Back (B2B) design. The B2B sensors can precisely detect the walking direction in the walkway areas, which can disambiguate the walking events toward the sleep/outside area from those toward the active zone, thus reducing the noise in the training data set. As shown in Table 3, the two implementations have comparable performances in the offline method since the sensor events before, after and during a candidate sleep/away interval can work efficiently for disambiguation of the walking directions even in COTS implementation. However, the B2B sensors work more accurate than COTS in the online mode since the direction information helps to filter out the ambiguous candidate walking events, thus reducing noise in the training set. This results in a better trained model which translates into 40% lower comfort penalties by using the B2B sensors instead of COTS, and 7% lower energy penalty.

7. 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.

7.1. Sensitivity to number of sensors

To have a fair comparison, we used the same deployed sensors in the baseline and WalkSense in the 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

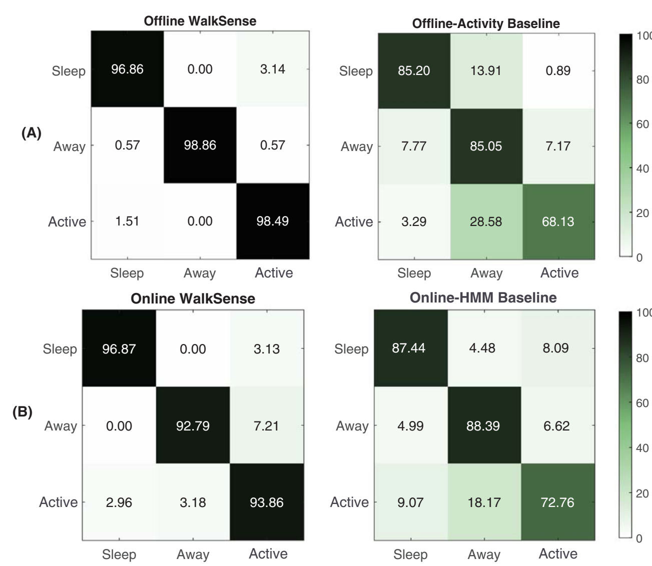


Fig. 12. (A) The offline WalkSense correctly detects home states by 96.1% compared with the activity sensing baseline with 79.5% accuracy. (B) The online WalkSense also outperforms the smart thermostat with 94.5% accuracy compared with 82.9%.

Table 3
The B2B design achieves higher detection rates since it can detect the crossing direction in walkways.

Detection	Sleep		Away		Active	
Rate	COT	B2B	COT	B2B	COT	B2B
Offline	99	100	98.6	98	98.9	100
Online	97.2	99	98	99.3	93.1	100

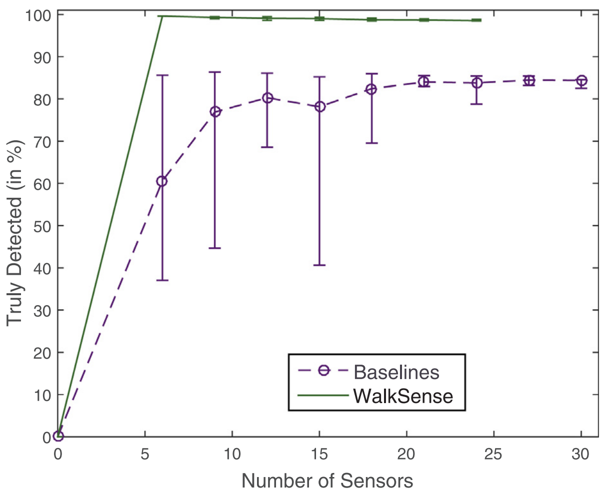


Fig. 13. The WalkSense detection accuracy is robust even with only a small number of sensors.

sensors shown in Fig. 8 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 senors) and count them as active sensors.

Fig. 13 shows the average detection rate of WalkSense compared with the baseline. The error bars indicate the minimum and maximum of calculated accuracy 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 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.

7.2. 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 Fig. 14. 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

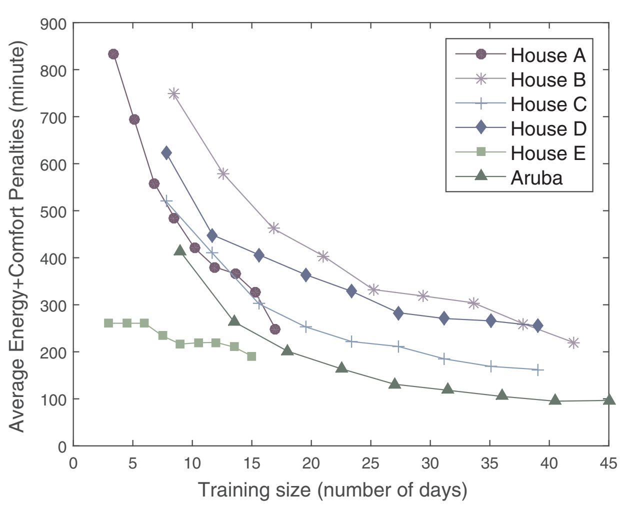
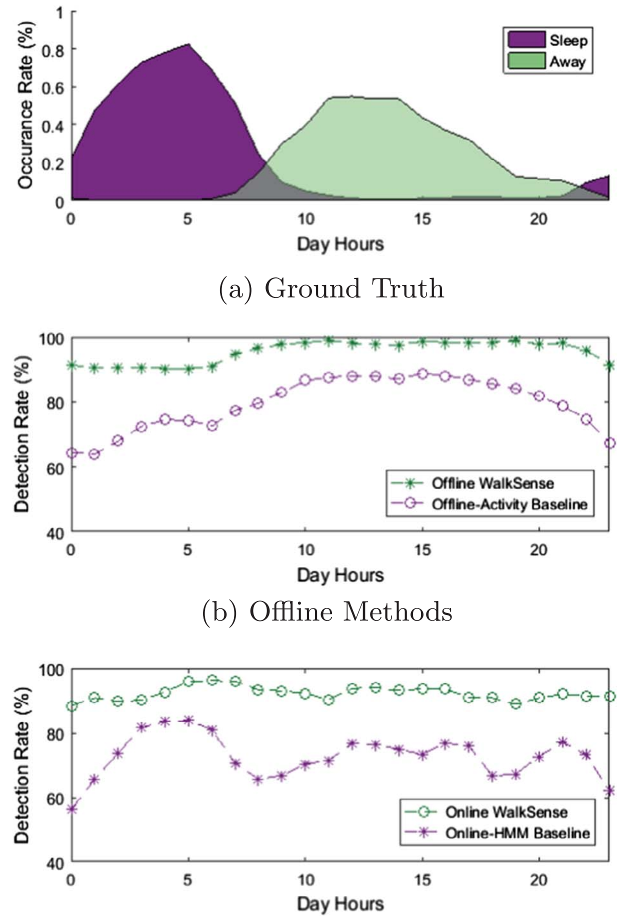


Fig. 14. The automatic labeling of the WalkSense provides incremental learning over time, thus increasing the accuracy.



(a) Ground Truth

(b) Offline Methods

(c) Online Methods

Fig. 15. WalkSense is robust to irregular behavior patterns, while the baselines have aggressive reaction to any changes in the pattern.

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. Recall that the duration of studies varies in different homes. For example, the experiments lasted 43 days in home B, but 15 days in home E. Therefore, some curves do not have any values for larger training sizes. In addition, the Aruba dataset includes 219 days of data, but the first 50 days are shown in this graph for brevity.

7.3. Sensitivity to behavior patterns

To study the robustness of WalkSense against different behavior patterns, we compare the averaged detection rate of WalkSense with the baselines for different hours of the day. Fig. 15a shows the averaged ground truth occurrence rate of sleep and away instances in different hours of the day. It shows a consistent behavior between 3 and 6 am, when 75% of the time people were asleep and between 12 and 15 pm, when people were outside of home in 60% of cases. However, sleep, away, or active states all happen between 7–11 am and 7–11 pm and it varies based on the occupants' schedules. As Fig. 15b and c represent, the WalkSense method is robust in different schedules and hours of the day with negligible changes over days. However, the baseline, specially Online-HMM, is not capable of adjusting its learning based on these

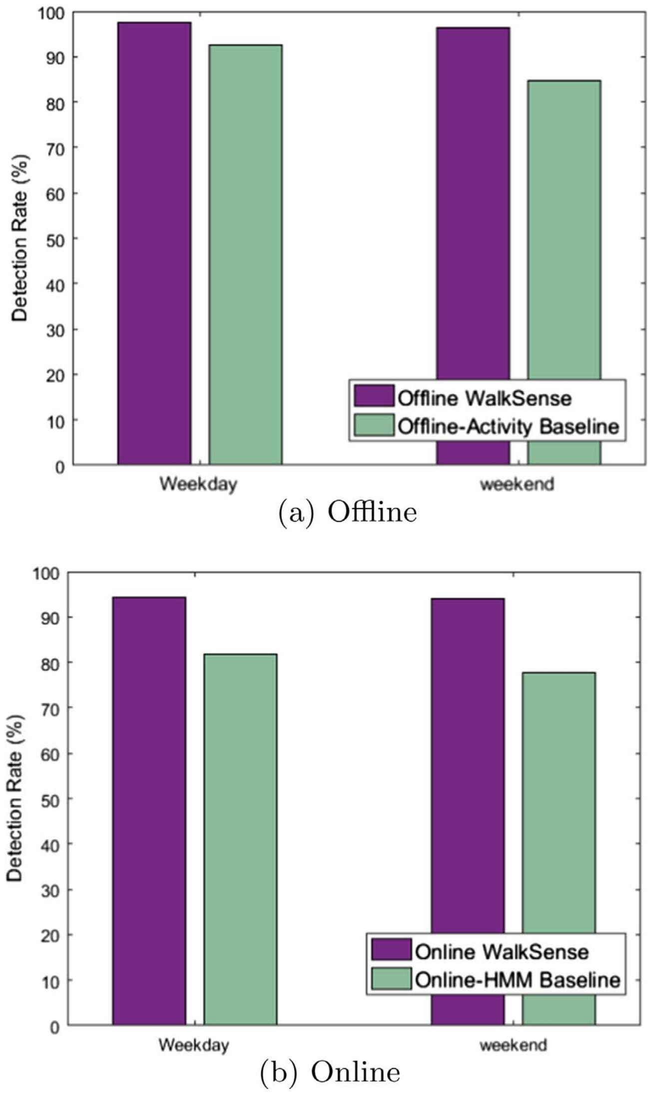


Fig. 16. Sensitivity to occupancy patterns over weekday and weekends.

changes in the occupants schedule, which results in lower detection rate during hours with overlapped sleep and away instances.

Another metric to measure behavior patterns is the day of the week. Occupants usually have a regular schedule in weekdays specially those who work outside. However, people might spend most of their time at home during the weekend or have occasional plans in different weekends. Based on this, we categorize the averaged detection rates based on the day of occurrence to weekday and weekend to better study the effect of these changes. Fig. 16 illustrates how the Online and Offline WalkSenses perform for different types of days in a week. The results indicate that, for both weekdays and weekends, the WalkSense provides higher detection rate than the baselines and it is robust to irregular occupancy patterns. For weekends, the WalkSenses achieve 97.2% averaged detection rate, while the baselines provide 79.1% accuracy. We observe that the WalkSense approach achieves very similar accuracies in weekdays and weekends, while the baselines have 10% lower accuracy during the weekends. This is because the baseline cannot learn the changes in the behavior patterns as fast as the WalkSense.

To better study the effect of sleep patterns, we compare homes C and D which have the same floor plans, but different numbers of occupants and sleep patterns. Therefore, we can isolate the effect of the floor plan to investigate the impact of sleep patterns more 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 homes B and E, sleep and away distributions highly overlap and have larger variances (as seen in Fig. 9). Therefore, 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.

7.4. Sensitivity to number of occupants

We compare the performance of WalkSense on the single-person and multi-person homes. In our experiments, homes D, E, and Aruba are single-person and homes A, B, and C have two occupants. The averaged accuracy of the two-person experiments is 98.1% with 3.7% false negative, compared to 99.3% and 1.11% for single-person case studies. We observe 1.2% lower accuracy in the multi-person homes compared with single-person homes. However, our extensive analyses reveal that this reduction is not due to number of occupants, but the sensor failures or intermittent sleep periods caused by occupants waking up at midnight.

Theoretically, there are very specific situations that multiple people may cause ambiguities for inferring home occupancy. For example, if the entrance of a person to the sleep zone coincides with the exit of another occupant from home, the home status would be inferred incorrectly. However, these situations are very rare and do not have a significant impact on the WalkSense performance. In spite of the rare potential multi-person ambiguities, the presence of multiple occupants at home increases the activities and the physical dynamics, which results in more sensor firing and richer information. For example, even if one of the occupants is sitting still (potential false positive), the movements of other occupants can trigger sensors and filter out this potential misdetection. On the other hand, the feature sets defined in the online WalkSense are based on the type of walkways (sleep/away), not per walkway. Therefore, the larger number of occupants increases the training size while the feature set remains constant. This results in higher accuracy with no effect on the complexity of the model. However, we were not able to evaluate WalkSense on homes with more than 2 occupants due to difficulties in groundtruth collection, which is explained in Section 8.

7.5. Sensitivity to the value of $\mathscr{K}$

Fig. 17 illustrates the impact of $\mathscr{K}$ on the energy and comfort

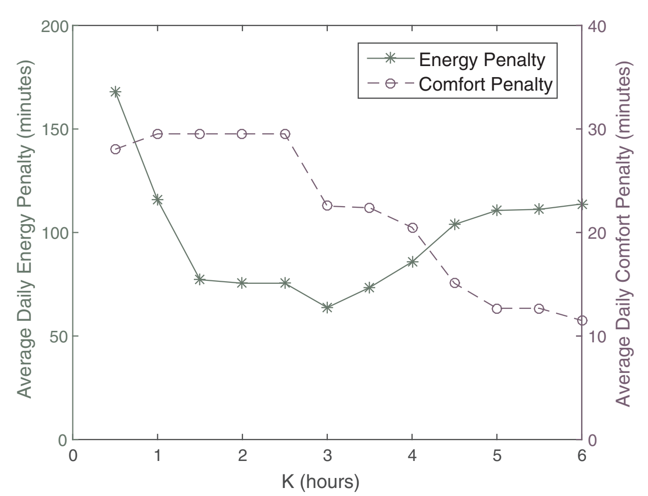


Fig. 17. Smaller values of $\mathscr{K}$ causes more energy wastage and comfort issues because of higher sensing sensitivity in the active zone.

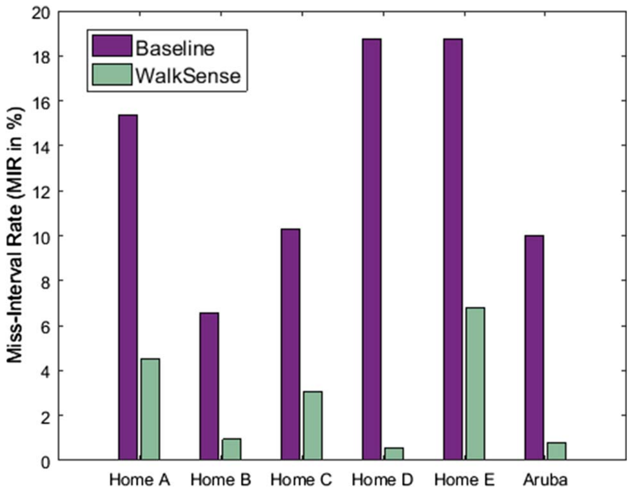


Fig. 18. The offline proposed approach has 3.18% undetected fragmented sleep periods, compared to 13.94% in the baseline.

penalties. Generally, smaller values of $\mathscr{K}$ 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 $\mathscr{K}$ increases, lower comfort penalty is resulted. However, the impact of $\mathscr{K}$ 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 $\mathscr{K}$, which increases the number of undetected sleep periods and energy wastage.

7.6. Sensitivity to intermittent events

Some sleep or away intervals might be partially missed because of intermittent sleep or outside instances such as awaking at midnight to drink water or going back home shortly to pick the items left behind. The online WalkSense deals with this situations by switching to the conditional states in the state machines instead of directly changing the home occupancy states to sleep/away or active. However, the offline Walksense could merge the intermittent inactive periods if they are larger than the defined threshold; otherwise the misdetection of these fragmented events results in lower energy saving. We defined MIR as a metric to analyze the responsiveness of WalkSense against these cases. The overall average MIR in Offline WalkSense is 3%, compared to 14% in the baseline (shown in Fig. 18). The reason for higher MIR value in the baseline is that it merely relies on a period to differentiate sleep and away periods. Therefore, a sleep interval which starts earlier than the silent period or lasts longer than that will cause fragmented sleep intervals.

8. 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. Therefor, a single zone could have 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.

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, if one occupant is sleeping and a second occupant transitions from active to away, the household should transition from active to sleep while the last sensor event belongs to away walkway. Our results indicate that, although such ambiguities are theoretically possible, the timing requirements make them exceedingly rare.

Finally, the current study is limited to homes with 1 or 2 occupants since the participants were concerned with reporting their sleep and away times for ground truth. Manual reports are annoying for long-term studies and camera recording causes privacy issues. This concern is even more challenging for homes with larger number of occupants. Another limitation in the experimental setup is that the proposed method is not applied to any specific HVAC control systems, so the energy and comfort analyses are limited to the general penalty metrics defined in Section 5.2. The main reason for this decision is that the practical energy and comfort savings highly depend on external parameters such as the properties of the home (e.g. size or number of rooms) and HVAC systems (e.g. thermal lag or setback temperature). Since the proposed method is applicable to any types of HVAC control systems, the defined Energy and Comfort Penalty metrics are considered sufficient to represent the performance of the proposed method.

9. Conclusions

In this paper, we present walkway sensing as a new principle for using motion sensors and show that we can convert it to an accurate occupancy detection approach to differentiate different zone occupancy states as sleep, away, and active. We also show that walkway sensing is cost-effective by requiring small number of sensors and non-intrusive by not requiring user involvement. We build our detection model called WalkSense upon this principle, which operates in two modes of offline and online. The inferred occupancy information in the offline WalkSense can be used for efficiently configuring self-programming thermostats, while online WalkSense could be used in smart thermostats to control the heating and cooling in real time. We implemented the proposed approach using two types of sensors: Off-the-shelf motion sensors, where we need to deal with the clock offsets; and Back-2-Back custom sensor package, which provides the crossing direction information for more accurate occupancy detection. 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.

Acknowledgement

We would like to thank our anonymous participants for helping us in evaluating our system with in-situ experiments. This work was supported by National Science Foundation (NSF) award 1305362.

References

[1] Environmental protection agency. summary of research findings from the programmable thermostat market. Washington, DC: Office of headquarters. 2004.
[2] Gruber J, Huerta F, Matatagui P, Prodanović M. Advanced building energy management based on a two-stage receding horizon optimization. Appl Energy 2015;160:194–205.
[3] Strbac G. Demand side management: benefits and challenges. Energy policy 2008;36(12):4419–26.
[4] Lu J, Sookoor T, Srinivasan V, Gao G, Whitehouse K. The smart thermostat: using occupancy sensors to save energy in homes. In: Proceedings of the 8th ACM conference on embedded networked sensor systems; 2010. p. 211–24.
[5] Goyal S, Ingley HA, Barooah P. Occupancy-based zone-climate control for energy-efficient buildings: complexity vs. performance. Appl Energy 2013;106:209–21.
[6] Gao G, Whitehouse K. The self-programming thermostat: optimizing setback schedules based on home occupancy patterns. Proceedings of the first ACM workshop on embedded sensing systems for energy-efficiency in buildings. ACM; 2009. p. 67–72.
[7] Pisello AL, Asdrubali F. Human-based energy retrofits in residential buildings: a cost-effective alternative to traditional physical strategies. Appl Energy 2014;133:224–35.

[8] Kwok SS, Lee EW. A study of the importance of occupancy to building cooling load in prediction by intelligent approach. Energy Convers Manage 2011;52(7):2555–64.
[9] Li N, Yang Z, Becerik-Gerber B, Tang C, Chen N. Why is the reliability of building simulation limited as a tool for evaluating energy conservation measures? Appl Energy 2015;159:196–205.
[10] Krishnan NC, Cook DJ. Activity recognition on streaming sensor data. Pervasive Mob Comput 2014;10:138–54.
[11] Scott J, Bernheim Brush A, Krumm J, Meyers B, Hazas M, Hodges S, et al. Preheat: controlling home heating using occupancy prediction. Proceedings of the 13th international conference on ubiquitous computing. ACM; 2011. p. 281–90.
[12] Agarwal Y, Balaji B, Gupta R, Lyles J, Wei M, Weng T. Occupancy-driven energy management for smart building automation. Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building. ACM; 2010. p. 1–6.
[13] Kleiminger W, Beckel C, Staake T, Santini S. Occupancy detection from electricity consumption data. Proceedings of the 5th ACM workshop on embedded systems for energy-efficient buildings. ACM; 2013. p. 1–8.
[14] Pisharoty D, Yang R, Newman M, Whitehouse K. Thermocoach: reducing home energy consumption with personalized thermostat recommendations. Proceedings of the 2th ACM international conference on embedded systems for energy-efficient buildings. ACM; 2015.
[15] Chen D, Barker S, Subbaswamy A, Irwin D, Shenoy P. Non-intrusive occupancy monitoring using smart meters. Proceedings of the 5th ACM workshop on embedded systems for energy-efficient buildings. ACM; 2013. p. 1–8.
[16] Cook DJ. Learning setting-generalized activity models for smart spaces. IEEE Intell Syst 2010;2010(99):1.
[17] E.P. Agency. Summary of research findings from the programmable thermostat market. Washington, DC: Office of Headquarters; 2004.
[18] Erickson VL, Carreira-Perpiñán MÁ, Cerpa AE. Observe: occupancy-based system for efficient reduction of HVAC energy. 10th international conference on information processing in sensor networks (IPSN). IEEE; 2011. p. 258–69.
[19] Prothermostats. Verdant v8-bb-7s line voltage heat only thermostat-programmable. < http://www.prothermostats.com/product.php?pverdant_v8-bb7s&product173694 > .
[20] Peco. Smart energy management for classrooms and portables. < http://www.pecomanufacturing.com/products/datasheets/school_salessheet_rev_00_web.pdf > .
[21] Tominaga S, Shimosaka M, Fukui R, Sato T. A unified framework for modeling and predicting going-out behavior. Pervasive computing. Springer; 2012. p. 73–90.
[22] Scellato S, Musolesi M, Mascolo C, Latora V, Campbell AT. Nextplace: a spatio-temporal prediction framework for pervasive systems. Pervasive computing. Springer; 2011. p. 152–69.
[23] Farrahi K, Gatica-Perez D. Discovering routines from large-scale human locations using probabilistic topic models. ACM Trans Intell Syst Technol (TIST) 2011;2(1):3.
[24] Oldewurtel F, Parisio A, Jones CN, Gyalistras D, Gwerder M, Stauch V, et al. Use of model predictive control and weather forecasts for energy efficient building climate control. Energy Build 2012;45:15–27.
[25] Širokỳ J, Oldewurtel F, Cigler J, Prívara S. Experimental analysis of model predictive control for an energy efficient building heating system. Appl Energy 2011;88(9):3079–87.
[26] Aswani A, Master N, Taneja J, Culler D, Tomlin C. Reducing transient and steady state electricity consumption in HVAC using learning-based model-predictive control. Proc IEEE 2012;100(1):240–53.
[27] Hayes TL, Riley T, Pavel M, Kaye J, et al. Estimation of rest-activity patterns using motion sensors. Engineering in medicine and biology society (EMBC). IEEE; 2010. p. 2147–50.
[28] Soltanaghaei E, Whitehouse K. WalkSense: classifying home occupancy states using walkway sensing. BuildSys@ SenSys. 2016. p. 167–76.
[29] Li N, Calis G, Becerik-Gerber B. Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations. Automat Constr 2012;24:89–99.
[30] Manzoor F, Linton D, Loughlin M. Occupancy monitoring using passive RFID technology for efficient building lighting control. 2012 fourth international EURASIP workshop on RFID technology (EURASIP RFID). IEEE; 2012. p. 83–8.
[31] Krumm J, Brush AB. Learning time-based presence probabilities. Pervasive computing. Springer; 2011. p. 79–96.
[32] Soltanaghaei E, Kalyanaraman A, Whitehouse K. Peripheral WiFi vision: exploiting multipath reflections for more sensitive human sensing. Proceedings of the 4th international on workshop on physical analytics. ACM; 2017. p. 13–8.
[33] Soltanaghaei E, Kalyanaraman A, Whitehouse K. Poster: occupancy state detection using WiFi signals. Proceedings of the 15th annual international conference on mobile systems, applications, and services. ACM; 2017. p. 161.
[34] Kalyanaraman A, Hong D, Soltanaghaei E, Whitehouse K. Forma track: tracking people based on body shape. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2017;1(3):61. ACM.
[35] Gupta M, Intille SS, Larson K. Adding gps-control to traditional thermostats: an exploration of potential energy savings and design challenges. Pervasive computing. Springer; 2009. p. 95–114.
[36] de Morais WO, Wickstrom N. A lightweight method for detecting sleep-related activities based on load sensing. Serious games and applications for health (SeGAH). IEEE; 2014. p. 1–7.
[37] Adami A, Hayes T, Pavel M. Unobtrusive monitoring of sleep patterns. Engineering in medicine and biology society, vol. 2. IEEE; 2003. p. 1360–3.
[38] Adami AM, Adami AG, Schwarz G, Beattie ZT, Hayes TL. A subject state detection approach to determine rest-activity patterns using load cells. Engineering in medicine and biology society (EMBC). IEEE; 2010. p. 204–7.
[39] Austin D, Beattie ZT, Riley T, Adami AM, Hagen CC, Hayes TL. Unobtrusive classification of sleep and wakefulness using load cells under the bed. Engineering in medicine and biology society (EMBC). IEEE; 2012. p. 5254–7.
[40] Borazio M, Van Laerhoven K. 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; 2012. p. 71–80.
[41] Hikosaka S, Kurashima S, Suzuki S. Verification of sleep/wake detection algorithms for kindergarten support system based on activity recognition. SICE J Contr Measur Syst Integr 2015;8(2):139–43.
[42] Wang S, Burnett J, Chong H. Experimental validation of CO_2-based occupancy detection for demand-controlled ventilation. Indoor Built Environ 1999;8(6):377–91.
[43] Kim Y-S, Heidarinejad M, Dahlhausen M, Srebric J. Building energy model calibration with schedules derived from electricity use data. Appl Energy 2017;190:997–1007.
[44] Zikos S, Tsolakis A, Meskos D, Tryferidis A, Tzovaras D. Conditional random fields-based approach for real-time building occupancy estimation with multi-sensory networks. Automat Constr 2016;68:128–45.
[45] Balaji B, Xu J, Nwokafor A, Gupta R, Agarwal Y. Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. Proceedings of the 11th ACM conference on embedded networked sensor systems. ACM; 2013. p. 17.
[46] Wang W, Chen J, Huang G, Lu Y. Energy efficient HVAC control for an IPS-enabled large space in commercial buildings through dynamic spatial occupancy distribution. Appl Energy 2017.
[47] Utgoff PE. Incremental induction of decision trees. Mach Learn 1989;4(2):161–86.
[48] Bulusu N, Jha S. Wireless sensor network systems: a systems perspective (artech house mems and sensors library). Norwood, MA: Artech House; 2005.
[49] Alipour M, Gheitasi A, Harris DK, Ozbulut OE, Barnes LE. A data-driven approach for automated operational safety evaluation of the National Inventory of Reinforced Concrete Slab Bridges. Transportation research board 95th annual meeting. 2016. 16-6168.
[50] Alipour M, Harris DK, Barnes LE. Pattern recognition in the national bridge inventory for automated screening and the assessment of infrastructure. Structures Congress 2017. 2017. p. 279–91.
[51] Pal M, Mather PM. An assessment of the effectiveness of decision tree methods for land cover classification. Remote Sens Environ 2003;86(4):554–65.
[52] Alipour M, Harris DK, Barnes LE, Ozbulut OE, Carroll J. Load-capacity rating of bridge populations through machine learning: application of decision trees and random forests. J Bridge Eng 2017;22(10):04017076. American Society of Civil Engineers.