Toward Personalization of User Preferences in Partially Observable Smart Home Environments

Shashi Suman, Francois Rivest, and Ali Etemad, Senior Member, IEEE

Abstract—The technologies used in smart homes have recently improved to learn the user preferences from feedback in order to enhance the user convenience and quality of experience. Most smart homes learn a uniform model to represent the thermal preferences of users, which generally fails when the pool of occupants includes people with different sensitivities to temperature, for instance, due to age and physiological factors. Thus, a smart home with a single optimal policy may fail to provide comfort when a new user with a different preference is integrated into the home. In this article, we propose a Bayesian reinforcement learning framework that can approximate the current occupant state in a partially observable smart home environment using its thermal preference and, then, identify the occupant as a new user or someone who is already known to the system. Our proposed framework can be used to identify users based on the temperature and humidity preferences of the occupant when performing different activities to enable personalization and improve comfort. We then compare the proposed framework with a baseline long short-term memory learner that takes the thermal preference of the user from the sequence of actions that it takes. We perform these experiments with up to five simulated human models each based on hierarchical reinforcement learning. The results show that our framework can approximate the belief state of the current user just by its temperature and humidity preferences across different activities with a high degree of accuracy.

Impact Statement—Smart home systems have impacted the lives of occupants by improving their daily experience by reducing the amount of manual intervention needed to optimize the ambient conditions. However, in an attempt to provide comfort, smart home agents often generalize a single model across multiple users, which in reality may not be optimal for everyone. For instance, prior works have shown that thermal preference varies between groups of users. As a result, using a single model will not generalize well across populations of occupants, which can create inconvenience, especially for vulnerable groups such as children or the elderly. With our framework, we challenge this issue to improve the smart home towards personalizing the user's comfort by recognizing them based on their activity-specific thermal preference, without any privacy invasive means such as facial recognition, fingerprints, or other biometrics. This improves the model to allow for personalization of experience without any involvement of personal information of the user.

Index Terms—Artificial intelligence–human interaction, Bayesian reinforcement learning, hierarchical reinforcement learning (HRL), smart home, user personalization.

I. INTRODUCTION

THE adoption of smart home systems (SHSs) has increased significantly in recent years. With the development of the Internet of Things, it has become much easier to control and automate various devices, such as thermostats, speakers, voice assistants [2], [18], [21], and many other devices with the overall aim of improving the experience of occupants in the home. As a result of advances in automation, machine learning, and biometrics, such devices are now able to learn user preference patterns and, thus, improve security, safety, and adaptability and personalization toward the occupants.

On the topic of improving the experience of occupants, SHSs are also responsible for monitoring and controlling comfort factors, such as temperature, humidity, and lighting, in order to maximize occupant comfort. These parameters can vary among different people within a group based on their physical characteristics such as gender, age, and clothing [30]. They can even vary for the same person at different times based on their activities, state of health [5], and others. This shows the importance of personalization within SHSs to provide a tailored experience to the occupants, thereby reducing manual intervention. Having a personalized setting attracts consumers’ interest and helps in collecting and analyzing data to improve the adaptability of the SHS to a higher degree. Another way to improve this is with a cross-exchange of suggestions and feedbacks [28] between the SHS and the occupant. By following the directions of the user [20], SHSs can perceive the user state indirectly, thus helping the SHS to build a hierarchical decision tree that is exclusive to each user. To achieve this, however, the SHSs need access to parameters that are exclusive to the occupant in order to learn the behavior of the user.

As argued above, the ambient parameters that are often controlled by smart home agents can vary for different users and even for the same user when performing different activities. While smart home agents can generally learn the behavior of individual users based on user identity information, activities, and preferences, such parameters may not always be observable to the smart home agent when a new user is introduced to the environment. In some cases, smart homes do not have the
ability to identify occupants due to the absence of ubiquitous biometric technologies [3], [4] as well as privacy issues, which can be encountered if sensing systems like cameras were used to identify and personalize occupant preferences.

Previously, in [26], we simulated a series of experiments with a hierarchical reinforcement learning (HRL)-based human model capable of learning activities and setting its thermal preferences in a smart home environment. The SHS, in turn, aimed to learn the preferences of the human to improve the thermal comfort of the occupant (human model). Nonetheless, in [26], the SHS was designed to have full observability about the user. This was done in order to enable the SHS to adapt to users with different intrinsic reward functions. In this article, we tackle the problem of user personalization in a scenario where the user identity is not observable to the smart home. To compensate for the partial information available about the user, we build into our smart home agent the ability to learn to recognize the user through its activity and thermal preferences in order to improve its policy. To accomplish the task of learning a policy with uncertainty, our agent maintains a belief over the human model who is pursuing its activity in the environment. Our experiments show that the SHS can learn to accurately identify the current user in the environment based on its thermal preferences given the current activity among a set of possible activities. Beyond identifying the occupant in the environment, the SHS also learns the optimal thermal preference for each activity for the identified user, which can be used for further personalization.

In summary, our contributions in this article are as follows.

1) We tackle the scenario of user personalization where user identity is not fully observable to the smart home system and instead rely on limited ambient parameters, such as temperature and humidity (TH) preferences.
2) We introduce a Bayesian model of the smart home that can approximate the user state using the thermal preference of the user for a given activity.
3) We test our model on scenarios where we have up to five human models pursuing different activities in a home. We demonstrate that our model can successfully recognize the user by means of their activity-specific thermal preference with a high accuracy while also optimizing their comfort.

The rest of this article is organized as follows. In Section II, we present a summary of the related work on user thermal personalization. Next, in Section III, we describe the architecture of the smart home agent that can learn to recognize the current user and to set the TH according to its preferences. In Section IV, we implement a long short-term memory (LSTM) learner as a baseline to which we compare the performance of our reinforcement learning (RL)-based model. Next, in Section V, we simulate the environment with varying number of human occupants and evaluate the performance of our model in detail. Finally, Section VI concludes this article.

II. RELATED WORK

In this section, we review several important prior works that have explored the personalization of user’s preferences toward achieving thermal comfort. It has been previously shown that when multiple users with different thermal preferences are present in a smart home, the SHS needs to have a personalized profile for each occupant. For instance, in [12], boosted trees were used to learn occupant’s personalized thermal responses using skin temperature and its surrounding, achieving a median accuracy of 84%. Similarly, in [13], occupants’ heating and cooling behaviors were used to design personalized comfort models, obtaining a prediction accuracy of 73%. It was shown in [11] that human occupants have different thermal perceptions in indoor environments based on heat exchange through the skin. A random forest (RF) was used and a median accuracy of 70.8% was achieved for classifying thermal preferences with humans in the loop using air temperature. In [32], a support vector machine (SVM) model was used to learn and classify the individual occupant’s thermal sensation indoors, obtaining an accuracy of up to 86%.

Experiments in [1] showed that, on average, the thermal sensation of humans has a standard deviation of 3 °C. This can also be confirmed from [24], where the thermal sensation of a fixed number of users showed that different users perceive the same surroundings in a different way; thus, the difference in the thermal perception among the users can be used to personalize their preferences. The parameters used in [24] were electrodermal activity (EDA), humidity, operational skin temperature, and heart rate, which were used to classify the users using an SVM. Similarly, in [15], the thermal prediction error of users was reduced by 50% using an SVM.

While the aforementioned works use sensors attached to the body to obtain personalized data from the users, factors such as age and gender can also play a role in thermal perception and, thus, personalization. In [7], age and outside temperature were included in the model of [1] to obtain personalized comfort levels on a fixed number of occupants, obtaining a prediction accuracy of 76.7% using an SVM. Similarly, in [30], the parameter of gender was included to estimate the comfort of individuals and concluded that difference in clothing and gender were mostly responsible for explaining difference in preferences. To implement a personalized preference model, Liu [17] used RF models and obtained a mean accuracy of 75% to predict personalized thermal parameters.

The mentioned research works converge on a common aim of suggesting that a “one-model-fits-all” strategy does not work well with all individuals. This notion was further explored in [31], where it was found that the preferences of many users did not fit a basic thermal model of [1] using k-nearest neighbor. After including these users, the thermal comfort among the occupants improved significantly while the thermal model’s [1] accuracy decreased. Similarly, in [17], it was discovered that nearly 30% of users showed discomfort when placed within the same thermal conditions. These studies imply that a personalized model is necessary to learn the optimal comfort of each user in a home. These studies have, however, focused more on a fixed number of individuals with specific sensors to measure information, such as skin temperature, heart rate, cardiac rhythm, and others, which may not always be available to the SHS.

In contrast to the majority of the aforementioned studies, in this article we focus on proposing a system that can recognize the users by their preferences in a situation where the only available information is the TH adjustments made by them.
III. METHODS

In this section, we extend the RL-based SHS system from our previous work [26] with the ability to estimate who is the current occupant using current TH and the human model’s activity and actions only. The assumption for this problem is that there is only one human model in the environment at a given time. In this section, we first describe the thermal comfort model and our approach to synthesizing data with simulations, followed by the general framework for the problem, and finally the description of our proposed solution.

A. Human Thermal Comfort Model and Simulations

In order to implement the human thermal comfort, we use the model presented in [1] that uses thermal data from a large number of participants. The model approximates human thermal comfort with a mean vote. The range of the vote, also called predicted mean vote (PMV), lies between -3 and 3, where -3 denotes very cold feeling and 3 denotes very hot feeling. Here, -0.5 to 0.5 is the most comfortable range. With a discomfortability rate of 5%, these data also account for the deviation in the thermal preference of occupants. The data account for the parameters that affect the thermal feelings of a user, such as temperature, humidity, clothing, activity, etc. Similarly, to simulate human decision-making ability using RL, we implement the methodology in [9] that uses HRL to simulate how humans learn to switch between different activities by anticipating rewards from current and external activities. The complete model description can be found in [26].

B. Partially Observable Markov Decision Process

When there is uncertainty about the current state of the environment (in this case, who the current occupant is, or what its preference are), a common framework for RL is the partially observable Markov decision process (POMDP) [10], [22], [23]. A POMDP is defined by the tuple \( \langle S, A, \Omega, T, R, O, \gamma, b_0 \rangle \), where \( S \) is the set of discretized states, \( A \) is the set of actions, and \( \Omega \) is the set of observations. Moreover, \( T(s_t, a_t, s_{t+1}) : [S \times A \times S] \rightarrow [0, 1] \) is the transition function that defines the probability of ending in state \( s_{t+1} \) after taking action \( a_t \) in state \( s_t \), \( R \) is the reward function that specifies the immediate reward received after taking action \( a_t \) in state \( s_t \), and \( O(a_t, o_{t+1}, s_{t+1}) : [A \times \Omega \times S] \rightarrow [0, 1] \) is the observation function that defines the probability of observing \( o_{t+1} \) after taking action \( a_t \) and ending in state \( s_{t+1} \). Finally, \( \gamma \) is the discount factor and \( b_0 \) is the initial belief state that defines a probability over the observable state of the environment.

To calculate the belief about the current state, the POMDP uses the Bayes theorem, which is defined as

\[
P(A|B) = \frac{P(B|A)P(A)}{P(B)}
\]

where \( A \) and \( B \) are the events, \( P(A) \) is the prior, \( P(B) \) is the evidence, \( P(A|B) \) is the posterior probability of event \( A \) given that event \( B \) has occurred, and \( P(B|A) \) is the probability of event \( B \) given that event \( A \) has occurred (likelihood). Here, we set \( A \) as a possible human model \( H_i \) in the home and \( B \) as the next TH and activity \( o_{t+1} \) observed by the smart home. Note that, here, the state is the concatenation of the observation \( o_{t+1} \) and the current occupant \( H_i \), \( s_t = \{a_t, H_i\} \), but the belief only needs to be over the occupants (since the TH and activity are fully observable). Accordingly, \( P(B|A) \) is the probability of observing \( B \) given that the human occupant \( H_i \) was the most recent human model in the environment at the current time step. Finally, \( P(A) \) is the prior probability of the human occupant \( H_i \) given the current activity \( H_t \) and the occupation of the environment \( H_t \) derived from the transition function \( T \), while \( P(t) \) is the marginal probability of the current observation received from the environment. Given our problem setup, the smart home agent takes an action \( a_t \) at each time step, after which it receives an observation \( o_{t+1} \) (consisting of the current activity of the occupant, temperature, and humidity) from the environment. With this observation, a POMDP agent would update its belief over each possible occupant by the following equation:

\[
\begin{align*}
P(A|B) &= \frac{P(B|A)P(A)}{P(B)} \\
&= \frac{P(B|A)P(A)}{ \sum_{H_i \in H} P(A)} \\
&= \frac{P(B|A)P(A)}{ \sum_{H_i \in H} \sum_{H_j \in H} T([a_t, H_j], a_t, s_{t+1})b_i(H_j) \frac{Pr(o_{t+1}|b_t, a_t)}{P(B)}}
\end{align*}
\]

(2)

where \( b_i(H_j) \) is the previous belief (probability) that the occupant is human model \( H_i \), \( b_{t+1}(H_i) \) is the updated belief for the next time step, and \( Pr(o_{t+1}|b_t, a_t) \) is the probability of observing \( o_{t+1} \) when action \( a_t \) is taken with the belief vector \( b_t \) over all the possible human models (or occupants) \( H \). The denominator is given by

\[
Pr(o_{t+1}|b_t, a_t) = \sum_{H_k \in H} O(a_t, o_{t+1}, [a_{t+1}, H_k]) \\
\times \sum_{H_j \in H} T([a_t, H_j], a_t, s_{t+1})b_i(H_j).
\]

(3)

It should be noted that in the context of our problem, the transition function \( T \) and the observation function \( O \) are unknown. Therefore, in the next sections, we will describe our solution around this issue.

C. Partially Observable Smart Home System

Here, we describe a modified SHS that implements a Bayesian model to learn multiple user’s personalized TH preferences when the user information is limited. Accordingly, we define a partially observable smart home system (POSHS) as a Bayesian model [22], [23] of the SHS, where the state of the environment is not fully observable, as discussed in the previous section. We will further assume that the model of the occupant does not change during an episode (a set of three activities). We will then use the Bayes theorem and the learnt occupants’ TH preferences to infer the current human model in the environment, thus identifying it.

We depict an overview of POSHS in Fig. 1, where it is divided into two components [panels (a) and (b)]. During the episode [see Fig. 1(a)], at each time step, the agent estimates a probability
distribution over the TH preferences of the current occupant based on the TH and activity observations. This distribution is combined with a pool of previously learned TH preference distributions using the Bayes theorem to estimate the belief state over the possible current occupants. The belief state is then used to weight the $Q$-table of each possible occupant in order to select the optimal action for the current belief. At the end of the episode [see Fig. 1(b)], the agent compares the final estimated TH preference distribution to the pool of learned preferences. If a match is found, then the episode distribution parameters are used to update that occupant’s TH preference distribution. If a match is not found, we assume that this is a new user. Therefore, the estimated probability TH preference distribution is added to the pool along with a new $Q$-table. Finally, the $Q$-tables are updated with the episode data. In the next subsections, we describe these elements (TH preference distribution, belief estimation, distribution similarity, and $Q$ update) in depth.

D. TH Preference Distribution

Since it is impossible to fully define $T$ and $O$ for an unknown set of occupants, we need a different approach to estimate the SHS belief over the set of possible occupants during the episode. To make the problem more tractable, we first assume that the occupants are the same through the whole episode. This implies that the transitional probabilities in an episode have the following property:

$$T([o_t, H_i], a_t; [o_{t+1}, H_i]) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

Accordingly, given a potential occupant $H_i$, we need to define a probability distribution over the TH preference for each activity $P(B|A)$ in (2). Using the Bayes rule, this will provide a way to estimate the probability of a specific occupant, given the TH observation $P(A|B)$ in (2). The TH preference probability distribution $P(TH|H_i)$ of a given occupant $H_i$ over temperature or humidity for each activity is described by a Gaussian distribution given as

$$P(k|H_i) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{1}{2} \left( \frac{k - \mu_{H_i}}{\sigma_{H_i}} \right)^2 \right) \quad (5)$$

where $k$ is an observed scalar value of temperature ($T$) or humidity ($H$) and $\mu_{H_i}$ and $\sigma_{H_i}$ are the mean and standard deviations of the preference distribution for occupant $H_i$. The above equation is defined individually for both temperature and humidity and
for each activity. This Gaussian distribution allows the system to support individual preference variabilities represented by \( \mu \) and \( \sigma \) as the mean and standard deviation of the comfortable thermal values. It should be noted that each human model has three distributions representing each given activity, respectively. For convenience, we will be using the notation “TH” for the equations implying that they apply to both temperature and humidity.

Given a complete episode, the SHS samples each observation when the human agent is not making any changes to the TH and continues with a given activity. Accordingly, the parameters of the TH preference distribution for the occupant of that episode can be estimated using best point estimators given by

\[
\mu_c = \frac{1}{C} \sum_{t=0}^{T} TH_t
\]

and

\[
\sigma_c = \sqrt{\frac{1}{C} \sum_{t=0}^{T} (TH_t - \mu_c)^2}
\]

where \( TH_t \) is the observed temperature or humidity at time step \( t \) for the given activity, \( C \) is the number of valid observations, and \( T \) is the length of the episode. Therefore, a complete TH preference probability distribution for a model can be coded using a 12-D vector (2 parameters \( \times 2 \) TH \( \times 3 \)).

### E. Belief Estimation

Here, we describe the mathematical formulation of the belief estimation of the smart home about the human occupant at time step \( t \) given observation \( TH_t \). For simplicity, let us assume that we have two human agents in a given smart home environment denoted by \( H_a \) and \( H_b \). Restructuring (2) for human models \( H_a \) and \( H_b \) based on the Bayes theorem, we obtain

\[
\frac{P(TH_t|H_a)}{P(TH_t|H_b)} = \frac{P(H_a|TH_t)P(H_a)}{P(H_b|TH_t)P(H_b)}.
\]

Solving the above equation for \( H_a \) and \( H_b \), we obtain

\[
P(H_a|TH_t) = \frac{P(TH_t|H_a)P(H_a)}{P(TH_t|H_a)P(H_a) + P(TH_t|H_b)P(H_b)}
\]

and

\[
P(H_b|TH_t) = \frac{P(TH_t|H_b)P(H_b)}{P(TH_t|H_a)P(H_a) + P(TH_t|H_b)P(H_b)}.
\]

Given \( N \) different human models, we can define the probability of the \( i \)th human model \( H_i \) at the \( t \)th time step as

\[
P(H_i|TH_t) = \frac{P(TH_t|H_i)P(H_i)}{\sum_{i=0}^{N} P(TH_t|H_i)P(H_i)}.
\]

Dividing \( P(H_a|TH_t) \) by \( P(H_b|TH_t) \) from (9) and (10) and then substituting \( P(TH_t|H) \) with (5), we obtain

\[
\frac{P(H_a|TH_t)}{P(H_b|TH_t)} = \frac{\frac{1}{\sigma_{H_a}\sqrt{2\pi}} \exp \left(-\frac{1}{2} \left(\frac{TH_t - \mu_{H_a}}{\sigma_{H_a}}\right)^2\right)}{\frac{1}{\sigma_{H_b}\sqrt{2\pi}} \exp \left(-\frac{1}{2} \left(\frac{TH_t - \mu_{H_b}}{\sigma_{H_b}}\right)^2\right)}
\]

\[
= \frac{\exp \left(\frac{1}{2} \left(\frac{\mu_{H_a} - \mu_{H_b}}{\sigma_{H_a}}\right)^2 - \frac{\mu_{H_a} - \mu_{H_b}}{\sigma_{H_a}}\right)}{\exp \left(\frac{1}{2} \left(\frac{\mu_{H_b} - \mu_{H_a}}{\sigma_{H_b}}\right)^2 - \frac{\mu_{H_b} - \mu_{H_a}}{\sigma_{H_b}}\right)}
\]

\[
= \frac{\sigma_{H_b}}{\sigma_{H_a}} \exp \left(\frac{1}{2} \left(\frac{TH_t - \mu_{H_a}}{\sigma_{H_a}}\right)^2 - \frac{\mu_{H_a} - \mu_{H_b}}{\sigma_{H_a}}\right)
\]

\[
\times P(H_a).
\]

Assuming \( \sigma_{H_a} \approx \sigma_{H_b} = \sigma \), we obtain

\[
\frac{P(H_a|TH_t)}{P(H_b|TH_t)} = \exp \left(\frac{1}{2\sigma^2} \left(\frac{2TH_t - \mu_{H_a} - \mu_{H_b}}{\mu_{H_a} - \mu_{H_b}}\right)^2\right)
\]

\[
\times P(H_a).
\]

At the beginning of the training episode, the POSHS agent has no information about the user; thus, it maintains an equal initial belief for each possible human occupant. Accordingly, the prior probability of each model can be defined as \( P(H_a) = P(H_b) \).

Hence, we can rewrite (13) as

\[
P(H_a|TH_t) = \frac{1}{\exp \left(\frac{1}{2\sigma^2} \left(\frac{2TH_t - \mu_{H_a} - \mu_{H_b}}{\mu_{H_a} - \mu_{H_b}}\right)^2\right) + 1}
\]

(15)

and

\[
P(H_b|TH_t) = \exp \left(\frac{1}{2\sigma^2} \left(\frac{2TH_t - \mu_{H_a} - \mu_{H_b}}{\mu_{H_a} - \mu_{H_b}}\right)^2\right) + 1
\]

(16)

With the above equation, we derive the initial posterior for each human agent at each time step for two human occupants. At the next time step, we perform the Bayesian update, where the posterior \( P(H_t|HT_{t+1}) \) at time step \( t+1 \) becomes the prior \( P(H_t) \) at time step \( t+1 \).

Accordingly, we can take (8) and divide both the numerator and the denominator by \( P(TH_t|H_a) \) and, then, replace the ratio in the second term of the denominator by (12). This yields

\[
P(H_a|TH_t) = \frac{P(TH_t|H_a)P(H_a)}{P(TH_t|H_a)P(H_a) + P(TH_t|H_b)P(H_b)}
\]

\[
= \frac{P(H_a)}{P(H_a) + P(TH_t|H_b)P(H_b)}
\]

(17)

where

\[
C(H_b, H_a) = \exp \left(\frac{(2TH_t - \mu_{H_a} - \mu_{H_b})(\mu_{H_b} - \mu_{H_a})}{2\sigma^2}\right).
\]

Similarly, for Model \( H_b \), the posterior can be computed as

\[
P(H_b|TH_t) = \frac{P(H_b)}{C(H_a, H_b)P(H_a) + P(H_b)}.
\]
Expanding the above equations for $N$ human models, we can get our posterior for the $i$th human model as
\begin{equation}
P(\mathcal{H}_i|\mathcal{P}) = \frac{P(\mathcal{H}_i)}{\sum_j^N C(\mathcal{H}_j, \mathcal{H}_i)P(\mathcal{H}_j)}. \tag{20}
\end{equation}

F. Distribution Similarity

Upon the completion of an episode, the SHS agent needs to know if the current occupant belongs to the pool of previously observed occupants or if it is a new occupant in the environment. To do so, the SHS agent follows the procedure in Fig. 1(b). Here, the best point estimators $[\mu_c$ and $\sigma_c$ from (6) and (7)] for the current occupant TH distribution are computed. The resulting TH preference distribution is then compared with the distributions in the pool of previously observed occupants using the Jensen–Shannon divergence (JSD) measure [6]. We select this particular divergence measurement since it has shown success in similar RL-related applications in prior works [25]. Moreover, this metric provides benefits over other divergence metrics; in particular, 1) when variations between distributions are small, the divergence remains smooth, and 2) the divergence between distributions is symmetric, i.e., $JSD(P_0, P_1) = JSD(P_1, P_0)$.

The JSD between $n$ probability distributions is formulated as
\begin{equation}
JSD = H \left[ \sum_{i=1}^n w_i p_i \right] - \sum_{i=1}^n w_i H(p_i) \tag{21}
\end{equation}
where $H$ is the Shannon entropy [16] and $w_i$ are the weights selected for each probability distribution $p_i$. For our problem, we compare only two distributions at an instance, i.e., the current distribution of the occupant in question and each of the distributions from the pool of observed occupants. As a result, we set $n = 2$, and thus, $w_1 = w_2 = 0.5$. Next, we define a new term $\gamma_{\text{JSD}}$ as the threshold such that
\begin{equation}
JSD \geq \gamma_{\text{JSD}} \iff \text{different human model} \tag{22}
\end{equation}
where JSD is the measured Jensen–Shannon divergence between the two TH distributions. If the divergence between the distributions is greater than $\gamma_{\text{JSD}}$, the current occupant’s distribution is added to the pool and the SHS agent creates a new $Q$-table. Otherwise, the distribution with the smallest divergence (distribution with highest likelihood) is updated using a moving average with the current estimated parameters to improve the long-term estimation of the occupant’s preference distribution.

G. Q Update

Here, we describe the computation of the $Q$-values of state–action pairs for a given occupant, followed by optimally selecting the best action using the belief vector over the user space. During each episode, as shown in Fig. 1(a), the SHS agent updates the TH distribution estimation parameters. It then calculates the belief vector using Bayes rule after which the $Q$-tables of the observed occupants are weighted using their belief vector to get a summarized $Q_{\text{net}}$ value. The SHS agent then chooses the optimal action from the $Q_{\text{net}}$ values of all the possible actions, which is then executed in the environment.

**Algorithm 1: Jensen–Shannon Divergence.**

1: function JSD Distribution $d_a$, Distribution $d_b$,
2: Amplification Factor $A_f$
3: let $d_c = \frac{d_a + d_b}{2}$ \hfill $\triangleright$ Calculate Shannon entropy for $d_a$
4: let $H_a = d_a \times \log \frac{d_a}{\pi_c}$
5: let $H_b = d_b \times \log \frac{d_b}{\pi_c}$ \hfill $\triangleright$ Calculate Shannon entropy for $d_b$
6: return $A_f \ast \frac{1}{2}$ \hfill $\sum_{H_a + H_b}$

The observation $o_t$, computed belief $b_t$, action taken $a_t$, and the reward received $r_{t+1}$ from the environment are all stored in the memory. To calculate the weighted $Q$ update, each weight is decided by the belief of the specific human model for a given state from the belief vector. This update is given by the following equation:
\begin{equation}
Q_{\text{net}}(o_t, a_t) = \sum_{H} b_t(H)Q_H(o_t, a_t) \tag{23}
\end{equation}
where $o_t$ is the observation, $a_t$ is the chosen action, the weight $b_t(H) = P(H|\mathcal{P})$ is the conditional probability given the current $TH$, and $Q_{\text{net}}$ is the weighted sum of all the $Q$-values of all the occupants for a given observation.

Accordingly, we now update the $Q$-table for a given occupant $H$. Let $b_t$ be the belief vector at time step $t$ in the episode and $b_t(H)$ be the belief component of $b_t$ for human $H$. Moreover, let $o_t$ be the received observation at time step $t$ in the episode, $\alpha = 0.05$ be the learning rate, $\gamma = 0.98$ be the discount factor, and $r_{t+1}$ be the reward received after action $a_t$ from observation $o_t$ at time step $t$. Accordingly, the $Q_H$ update is given by
\begin{equation}
Q_H(o_t, a_t) = (1 - \alpha)Q_H(o_t, a_t) + \alpha [\gamma r_{t+1} + b_t(H)] + \gamma \max_{a'} Q_H(o_{t+1}, a') \tag{24}
\end{equation}
We obtain $\alpha = 0.05$ and $\gamma = 0.98$ with some preliminary experiments such that the model converges. With the updated $Q_{\text{net}}$ value, the agent can select the optimal action. For a given observation $o_t$, actions are selected using a decaying $\epsilon$-Greedy policy keeping a balance between exploration and exploitation. The policy is given by
\begin{equation}
\pi(o_t) = \begin{cases}
\arg \max_{\alpha \in A} \{Q_{\text{net}}(o_t, \alpha)\} \text{ with probability } 1 - \epsilon \\
\alpha \in A \text{ with probability } \epsilon/|A| 
\end{cases}
\tag{25}
\end{equation}
where $\epsilon$ is a threshold value less than 1 that decays exponentially down to 0.005, and $A$ is the set of all possible actions.

H. Pseudocodes

In order to better facilitate reproduction of our work, in this subsection, we present the pseudocodes and detailed algorithms for computing the JSD, action selection and belief estimation, $Q$-table update, and POSHS in Algorithms 1–4.
Algorithm 2: GetAction.
1: function getAction(o_t)
2:     let N = number of human models
3:     let T_H, t ← o_t
4:     calculate belief $P(H|T_H)$ = $P(T_H|H)P(H)$
5:     $Q_{net}(o_t, a_t) = P(H_a)Q_a(o_t, a_t) + \sum_{o \neq 0} P(T_H|H)P(H)$
6:     return $\arg \max_a Q(o_t, a)$

Algorithm 3: Update.
1: function updateMemory, newNode
2:     let N = number of human models
3:     for H in all human models do
4:         let $b = P(H)$
5:         if newNode then
6:             $Q_H ≤ 0$
7:         Endif
8:     for $o_t, a_t, o_{t+1}, r_{t+1}, t_{t+1}$ in memory do
9:         $Q_H(o_t, a_t) = (1 - \alpha)Q_H(o_t, a_t) + \alpha r_{t+1} + b + \gamma \max_a Q_{H}(o_{t+1}, a')$
10:     Endfor
11: Endfor

IV. Baseline

We compare our model (POSHS) against two baselines: 1) a recurrent neural network (RNN) and 2) a transformer attention layer. In this section, we describe the details of these two models.

Classical RNNs, however, often suffer from issues such as vanishing gradient, making it difficult for the model to remember long-term information. As a result, we use an LSTM networks as our baseline. LSTM is a type of RNN that can hold a sequence of information in its memory and learn temporal relations. To do this, LSTM uses long-term sequences in memory referred to as “cell state.” The output from the previous step is referred to as “hidden state.” Information in an LSTM is controlled via three gates: 1) forget gate, which decides the information that needs to be rejected or accepted; 2) input gate, which is used to control the information into the cell state; and 3) output gate, which generates the output for each time step.

In a POMDP environment, the deep Q network (DQN) fails to learn the optimal policy because it assumes complete information about the state which we lack in our problem. Unlike DQNs, LSTM networks are capable of remembering long sequences of TH preferences and encoding the observation onto a latent space vector, making it possible to approximate the underlying hidden state [10], [33]. Thus, unlike POSHS where the agent needs to recognize the hidden state of the human occupant using Bayes rule explicitly, LSTMs can implicitly learn the underlying states using its recurrent memory.

To use LSTM as a baseline, we design two networks, namely, train and target. Both the networks have the same architecture and are initialized with the same weights and biases. The inputs are sequences of TH, activity, and human action of the occupant. The input layer of the model is a 1-D convolutional layer with 32 filters, a kernel size of 1, a stride of 1, and a rectified linear unit (ReLU) activation function. The output of this layer is fed to an LSTM layer with 175 cells followed by a tanh activation function. In the end, the output of this layer is fed to a dense layer of five neurons and ReLU activation to output the $Q$-values for each possible action (total of five actions: increase/decrease $T$.
and $H$, and no action). We set the learning rate as 0.0013 obtained empirically to keep the networks stable. Like the POSHS model, the $\epsilon$ decays exponentially down to 0.005. We do not decrease it completely to zero to explore all the possible states. The training procedure for this baseline is described in detail in Algorithm 5 with its architecture shown in Fig. 2.

In this algorithm, in contrast to POSHS where we update the $Q$-values directly, we update the parameters of the network with which we predict the next $Q$-value. The $Q$-values of the current state from the train network are updated using the $Q$-values of the next state from the target network. Using the same network (train) for predicting the target $Q$-value would cause instability in the target $Q$-value. The weights for the train and target networks are represented by $\theta$ and $\theta^-$, respectively. Target weights $\theta^-$ are updated after every $C$ iterations with $\theta$, where $C$ is a hyperparameter set to 7 empirically. We use mean squared error (MSE) as our loss function and ADAM [14] optimizer. We define our loss function similar to the update in table-based Q-learning, which is the difference between the target and current $Q$-values. Accordingly, the loss function is given by

$$
\mathcal{L}(s_t, a_t|\theta) = (r_t + \gamma \max_a Q(s_{t+1}, a|\theta^-) - Q(s_t, a_t|\theta))^2.
$$

(26)

In order to recognize the user, the LSTM model uses the embedding of the sequence of TH preferences. However, for this embedding, the input sequence contains preferences of the occupant when it is not making any changes in TH, which means that we only select TH preferences when the occupant is either executing the continue or leave action in a given activity. The embedding is obtained from the hidden state of the last cell from the LSTM layer of the architecture (see Fig. 2).

Before training, the initialized weight of the LSTM layer is same irrespective of the number of occupants. This helps the model to ensure that different embeddings are obtained for different TH sequences. The embedded information of TH sequence is collected at the end of episode when the model converges. At the end of episode, the obtained embedding is compared with each embedding stored in the pool. To compare the embedding, we use the Kolmogorov–Smirnov test (KST) [19]. This test quantifies the distance between two empirical distributions and returns the statistical distance and its $p$-value with which we can accept or reject the null hypothesis. For our problem, the null hypothesis is that two embeddings are same, which means that both the observed users are the same. The equation to compute the KST for two empirical distributions $F_a$ and $F_b$ (of the embeddings of TH sequence) is given as

$$
D, p = \max_{1 \leq i \leq N} (F_a(y_i) - F_b(y_i))
$$

(27)

where $D$ is the statistical distance, $p$ is the $p$-value, $F(y)$ is the cumulative density function, and $N$ is the number of samples. If $p$ is less than the significance level $\alpha$, then we can reject our null hypothesis, which states that the two user embeddings are same. For our problem, after some preliminary tests, we set the significance level $\alpha = 0.18$. We set the value higher than the general value of 0.05 because the TH preference of occupants will always have some overlap. Thus, discovering a new user is defined as

$$
p \leq \alpha \iff \text{Different Human Model.}
$$

(28)
Using the above equations, we compare the similarity between the current occupant TH embedding and the embeddings of previously observed occupants. If the similarity is lower than \( \alpha \), we add the embedding in the pool as a new occupant. In order to identify the current user in the environment, we use the following equation:

\[
H_i = \arg \min_i D_i
\]  

(29)

where \( D_i \) is the statistical distance between the current user’s embedding and the \( i \)th embedding from the pool of previously observed users.

For the transformer attention [29] baseline model, we follow a similar architecture to the LSTM model having a train and target network with the same number of inputs and \( Q \)-values as the action outputs. The input layer of this model is a linear layer followed by a transformer encoder layer with a three-head attention unit. This is followed by a ReLU activation function. Next, all the encoder layers are stacked and followed by a normalization layer. The normalized output is then passed to the fully connected layer that outputs the \( Q \)-values. Parameters like the learning rate and epsilon for this model were kept the same as that of the LSTM model. The loss between the current and the target \( Q \)-values is computed using the MSE, which is then propagated backward through the train network. This loss is computed between the train and target network’s \( Q \)-values. These values are then optimized using the ADAM optimizer. All the actions are taken using \( Q \)-values obtained from the train network. In order to avoid a race condition, after every \( C \) steps, the weights of the target network are updated with the train network weights. Here, \( C \) is a hyperparameter set to 7, which is found empirically to give a stable train network.

During training, it is possible for the model to recognize the user. We do this by using the encoder layers’ stable outputs learned from the sequence provided, as shown in Fig. 3. Similar to the LSTM model, we compare the obtained embeddings using KST as per (27). After obtaining the KST distance, the identity is computed using (29). For this model, we set the significance value \( \alpha = 0.11 \) obtained empirically to optimize the performance.

V. EXPERIMENTS AND RESULTS

In this section, we present our results using the proposed POSHS and the baseline models: 1) LSTM and 2) transformer attention layer. In the first experiment, we evaluate the POSHS performance by its accuracy in approximating the occupant’s distribution accurately in the environment. In the second experiment, we evaluate the impact of the SHS on the occupant by measuring the time steps required by the occupant to set TH with and without the SHS. Here, we refer time steps as the number of steps the human model takes to set TH to an optimal comfort level. We further compare those results to the LSTM baseline.

A. Experiment A

In this experiment, we measure the performance of the POSHS in identifying occupants while learning the thermal preferences of each user. We train each human model separately for 350 episodes such that each model can complete each activity while also setting the optimal TH. Next, we train the POSHS with the trained occupant for 150 episodes. During the training of the POSHS, a human model is chosen randomly before each episode. Finally, each human model is run with the POSHS for 50 episodes to evaluate the final performance of the system.

First, we only experiment with two human models, namely, \( H_a \) and \( H_b \). We set the metabolism indices for \( H_a \) to \([1, 1.2, 1.4]\) based on [1] and [27]. For \( H_b \), slightly different indices are required. We, therefore, add 0.05 to each of the indices for \( H_a \) to achieve \([1.15, 1.25, 1.45]\) for \( H_b \) for the PMV [1] range of \( d_{0.25} \). We use \( \tau_{\text{JSD}} = 0.13 \), which empirically showed the best results in preliminary tests. Fig. 4 shows the accuracy of identifying the current model correctly during the training of POSHS. As discussed earlier, the occupant identification is performed by comparing the TH preference distribution estimated by the POSHS from the episode to the ones stored in the pool of distributions using the JSD and selecting the best match.

![Fig. 3. Transformer attention baseline architecture: the input is a fixed length sequence with 5-D features, which is then passed to a multihead attention layer. The output is fed to the normalization layer, which is connected with a fully connected layer outputting five \( Q \)-values for five actions to be taken by the SHS.](image-url)
increase the indices of $\mathcal{H}_a$ by small fixed amounts to get the metabolism indices for the new human models $\mathcal{H}_c$, $\mathcal{H}_d$, and $\mathcal{H}_e$. The final values are $[1.15, 1.22, 1.35]$, $[1.15, 1.25, 1.4]$, and $[1.05, 1.3, 1.45]$, respectively. The final accuracies as well as its test F1 scores are reported in Table I. We observe that with only two human models, the POSHS is able to perform identification with high accuracy based on the thermal preferences. We also observe that with the integration of more human models, the accuracy drops due to the increased overlap between TH preferences. Nonetheless, strong results (68%) above chance level (20%) are still obtained for higher number of human models, for instance 5.

B. Experiment B

Here, we evaluate how both the approaches, the POSHS and the LSTM model, perform in terms of rewards and time steps required for the human occupant to set TH as we increase the number of human models in the home. Note that both POSHS and LSTM must form an internal representation of the underlying hidden state. The POSHS does it by explicitly estimating the current occupant’s TH preference, while the LSTM builds an internal representation based on the sequence of observed actions (TH and activities).

Similar to Experiment A, we train the LSTM with the trained human occupant for 175 episodes with a learning rate of 0.0013. During the training, a human model is chosen randomly at the beginning of the episode. We repeat the experiment for up to five human models. For testing purposes, we test the LSTM model for 50 episodes to evaluate the final performance. Therefore, both POSHS and LSTM were tested under the same training and testing conditions.

The performance in terms of the occupant’s rewards for both the SHS models is shown in Table II. There is a slight increase in the observed mean reward for the occupants with the SHS in comparison to mean reward without the SHS, suggesting that the human occupants spend less time changing the TH in the presence of the SHS model.

We also compare the user recognition ability of the POSHS and the LSTM model of SHS. The POSHS recognizes the current user by selecting the TH distribution that has the highest belief. Similarly, the LSTM-based model recognizes the user by comparing the embedding of its hidden state with the embeddings of previously observed users using KST, as described in Section IV. Table I shows the identification accuracy by the POSHS and the LSTM model. We observe that for two to three human models in the environment, the baseline LSTM is only slightly behind POSHS. However, with more human models, the difference increases.

We then measure the actual number of time steps required by the occupants to set their preferences with the SHS models compared to the time steps required by the occupants without the SHS. This is shown in Fig. 7, where we present the results for two, three, four, and five human occupants in the SHS. It is observed that the number of extra time steps taken by the occupants is close for POSHS and LSTM for up to three occupants.

Fig. 5. Model $\mathcal{H}_a$ belief update for overlapping ($d_{0.5}$) and nonoverlapping ($d_{0.25}$) TH preferences (with Model $\mathcal{H}_b$). In plot (a), the POSHS learns the activity (12-D) specific TH distributions of the occupant, while in plot (b), it learns the episode (4-D) specific TH distribution of the occupant.
TABLE I
POSHS, LSTM, AND TRANSFORMER ATTENTION MEAN ACCURACY (IN %) AND F1 SCORE FOR ACCURATELY APPROXIMATING THE OCCUPANT DISTRIBUTION UP TO FIVE HUMAN MODELS (HIGHER VALUES ARE BETTER)

| Model     | POSHS Accuracy | LSTM Accuracy | Trans. Att. Accuracy |
|-----------|----------------|---------------|----------------------|
| 2 Models  |                |               |                      |
| Model $H_A$ | 0.96           | 0.96          | 0.95                 |
| Model $H_B$ | 1.00           | 0.93          | 0.94                 |
| Mean      | 0.98           | 0.94          | 0.95                 |
| 3 Models  |                |               |                      |
| Model $H_A$ | 0.92           | 0.94          | 0.93                 |
| Model $H_B$ | 0.96           | 0.83          | 0.84                 |
| Model $H_C$ | 0.92           | 0.94          | 0.88                 |
| Mean      | 0.90           | 0.88          | 0.91                 |
| 4 Models  |                |               |                      |
| Model $H_A$ | 0.60           | 0.60          | 0.63                 |
| Model $H_B$ | 0.83           | 0.64          | 0.81                 |
| Model $H_C$ | 0.77           | 0.75          | 0.77                 |
| Mean      | 0.75           | 0.74          | 0.77                 |
| 5 Models  |                |               |                      |
| Model $H_A$ | 0.67           | 0.78          | 0.68                 |
| Model $H_B$ | 0.88           | 0.75          | 0.81                 |
| Model $H_C$ | 0.62           | 0.56          | 0.59                 |
| Mean      | 0.67           | 0.62          | 0.67                 |

TABLE II
MEAN REWARD VALUES FOR UP TO FIVE HUMAN MODELS FOR THE POSHS, LSTM, AND TRANSFORMER ATTENTION BASELINE IN COMPARISON TO WHEN THE SHS IS NOT INTEGRATED (HIGHER VALUES ARE BETTER)

| Model     | Without SHS | POSHS | LSTM     | Trans. Att. |
|-----------|-------------|-------|----------|-------------|
| 2 Models  | 284±1.17    | 291±1.14 | 290±1.62 | 293±2.73    |
| 3 Models  | 283±1.80    | 291±2.81 | 286±2.78 | 291±3.01    |
| 4 Models  | 283±1.92    | 287±3.49 | 284±3.89 | 289±2.71    |
| 5 Models  | 283±2.35    | 286±4.30 | 283±5.70 | 283±6.41    |

However, when compared to Fig. 7(a) and (b) in Fig. 7(c) and (d), the difference between the time steps of LSTM and POSHS increases with more human occupants. Furthermore, referring back to Table II, we observe that the POSHS slightly outperforms the LSTM model as the amount of reward received by the occupant with POSHS is higher, and the difference increases as the number of human models increases.

The LSTM training loss curve is presented in Fig. 6, showing that the LSTM struggles to learn the $Q$-values accurately as the number of occupants increases. This is particularly highlighted with four and five human models, where the LSTM seems to suffer from considerable forgetting [8] (depicted by the peaks in the training loss curve) due to the changing weights of the network required to learn the sequences of different TH preferences.

C. Experiment C

Here, we modify our LSTM baseline by replacing the LSTM layer with a transformer attention layer. Similar to the LSTM model, this new baseline also has to learn the internal representation of the human using its activity and thermal preferences. We train this model for 175 episodes with a learning rate of 0.0013. We perform the experiment with up to five human models. In order to test the model’s performance, we evaluate the trained model with trained human models for 50 episodes. In order to identify the current occupant, we use the encoding obtained from the fully connected layer of our transformer model for each occupant. The KST, as described in Section IV, is then used to identify the similarity between the current encoding and previously observed encodings. Table I shows the identification accuracy of POSHS, LSTM, and transformer attention.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 7 shows the comparison of the time steps required by the occupant to set TH in the presence of POSHS, LSTM, and transformer attention. We observe that for up to three human models, the transformer attention performs slightly better than the LSTM baseline and comparable to the POSHS model. This is due to the ability of attention [29] in processing nonsequential states, which means that the order of activity execution by the human model does not matter unlike for the LSTM model. With four or five human models, the transformer attention baseline outperforms the LSTM and achieves competitive results with respect to the POSHS model. Nonetheless, it should be mentioned that our POSHS model, like the LSTM, only carries out a simple update at each time step, while the transformer attention needs to re-read the entire sequence at each time step, thus requiring quadratic computations with respect to the sequence length. Moreover, our POSHS model has 10 125 parameters (for five human models), while the LSTM network contains 146 256 parameters (weights), and the transformer attention network consists of 193 832 parameters.

Fig. 6(b) shows the loss curves for the transformer attention model, which is slightly better than the LSTM curves where occasional spikes were observed as the number of humans models increased and the order of activities were changed. For two to three human models, the loss decreases sharply unlike for four to five human models, where the converging loss is slightly greater.

VI. Conclusion

In this article, we investigated the scenario where limited information about the occupants is available to the smart home, resulting in often suboptimal actions by the smart home in a given state. We model a smart home to learn to recognize the occupant’s thermal preference distributions using Bayesian modeling, which helps maintain a belief over the hidden user state that the smart home uses to improve its policy. As a baseline for comparison, we design an approach in which the temporal relations between the TH sequences are learned to approximate the hidden state. We integrate both the smart home models with up to five human models and evaluate them based on their ability to identify the underlying state (or occupant’s TH preferences), as well as the overall human performance in terms of human rewards and time steps spent changing the TH with the SHS. Finally, we compare the POSHS with the baseline model that aims to learn its own representation of underlying states of the partially observable environment. Our simulations show good performance with the POSHS when the number of human occupants is low without an increase in time required to set the thermal preferences by the occupant. Similar performance was observed with the baseline. With more human models integrated in the environment, the time-step difference between the baseline and the POSHS increases where the occupants now take more time to set the TH with the baseline compared to the
POSHS model. In the end, with our simulated experiments, we demonstrate that in an environment where the user information is not fully observable, it is possible to approximate the user’s hidden state with multiple occupants without having an impact on the user’s behavior even with suboptimal approximations.

For future work, we may include additional occupant-related parameters in our algorithm, which can help in better approximation of the occupant and, thus, improve performance. These parameters may include heart rate, skin temperature, skin conductance, surrounding radiation, and others. Some of the most common biosignals, such as photoplethysmogram, electrocardiogram, EDA, and electroencephalogram, can be obtained by smart wearables that have currently become very popular. This will not only improve our algorithm’s accuracy, but will also provide more data to work with, which can allow for our algorithm to scale to more than five humans. Having multiple occupants at the same time also increases the complexity of the problem. However, by recognizing the users, the methods proposed in previous work, such as [26], can also be readily employed.

REFERENCES

[1] Thermal Environmental Conditions for Human Occupancy, ANSI/ASHRAE Standard 55-2010, 2010.
[2] F. Auffenberg, S. Stein, and A. Rogers, “A personalised thermal comfort model using a Bayesian network,” in Int. Joint Conf. Artif. Intell., 2015, pp. 2547–2553.
[3] K. O. Bailey, J. S. Okolica, and G. L. Peterson, “User identification and authentication using multi-modal behavioral biometrics,” Comput. Secur., vol. 43, pp. 77–89, 2014.
[4] C. Bo, L. Zhang, X.-Y. Li, Q. Huang, and Y. Wang, “SilentSense: Silent user identification via touch and movement behavioral biometrics,” in Proc. 19th Annu. Int. Conf. Mobile Comput. Netw., 2013, pp. 187–190.
[5] R. Chacón-Aguilar, J. M. Osorio-Cámara, I. Sanjurjo-Jiménez, C. González-González, J. López-Carnero, and B. Pérez-Moneo, “COVID-19: Fever syndrome and neurological symptoms in a neonate,” Anal. Pediatr., vol. 92, pp. 373–374, 2020.
[6] D. M. Endres and J. E. Schindelin, “A new metric for probability distributions,” IEEE Trans. Inf. Theory, vol. 49, no. 7, pp. 1858–1860, Jul. 2003.
[7] A. A. Farhan, K. Pattipati, B. Wang, and P. Luh, “Predicting individual thermal comfort using machine learning algorithms,” in Proc. IEEE Int. Conf. Autom. Sci. Eng., 2015, pp. 708–713.
[8] R. M. French, “Catastrophic forgetting in connectionist networks,” Trends Cogn. Sci., vol. 5, no. 4, pp. 126–135, 1999.
[9] C. Gebhardt, A. Oulasvirta, and O. Hihlges, “Hierarchical reinforcement learning as a model of human task interleaving,” 2020, arXiv:2001.02122.
[10] M. Hausknecht and P. Stone, “Deep recurrent Q-learning for partially observable MDPs,” in Proc. AAAI Fall Symp., 2015, pp. 29–37.
[11] W. Jung, F. Jazizadeh, and T. E. Diller, “Heat flux sensing for machine-learning-based personal thermal comfort modeling,” Sensors, vol. 19, no. 17, 2019, Art. no. 3691.
[12] K. Katić, R. Li, and W. Zeiler, “Machine learning algorithms applied to a prediction of personal overall thermal comfort using skin temperatures and occupants’ heating behavior,” Appl. Ergonom., vol. 85, 2020, Art. no. 103078.
[13] J. Kim, Y. Zhou, S. Schiavon, P. Raftery, and G. Brager, “Personal comfort models: Predicting individuals’ thermal preference using occupant heating and cooling behavior and machine learning,” Building Environ., vol. 129, pp. 96–106, 2018.
[14] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.
[15] E. Laftchiev and D. Nikovski, “An IoT system to estimate personal thermal comfort,” in Proc. IEEE 3rd World Forum Internet Things, 2016, pp. 672–677.
[16] J. Lin, “Divergence measures based on the Shannon entropy,” IEEE Trans. Inf. Theory, vol. 37, no. 1, pp. 145–151, Jan. 1991.
[17] S. Liu, “Personal thermal comfort models based on physiological parameters measured by wearable sensors,” 2018.
[18] J. Liu et al., “The smart thermostat: Using occupancy sensors to save energy in homes,” in Proc. 5th ACM Conf. Embedded Netw. Sensor Syst., 2010, pp. 211–224.
[19] F. J. Massey, Jr., “The Kolmogorov-Smirnov test for goodness of fit,” J. Amer. Statist. Assoc., vol. 46, no. 253, pp. 68–78, 1951.
[20] M. Nicolescu and M. Mataric, “Learning and interacting in human-robot domains,” IEEE Trans. Syst., Man, Cybern. A, Syst. Hum., vol. 31, no. 5, pp. 419–430, Sep. 2001.
[21] J. Y. Park, T. Dougherty, H. Fritz, and Z. Nagy, “LightLearn: An adaptive and occupant centered controller for lighting based on reinforcement learning,” Building Environ., vol. 147, pp. 397–414, 2019.
[22] S. Ross, B. Chaib-draa, and J. Pineau, “Bayes-adaptive POMDPs,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2007, pp. 1225–1232.
[23] S. Ross, J. Pineau, B. Chaib-draa, and P. Kreitmann, “A Bayesian approach for learning and planning in partially observable Markov decision processes,” J. Mach. Learn. Res., vol. 12, no. 5, pp. 1729–1770, 2011.
[24] F. Salamone et al., “Application of IoT and machine learning techniques for the assessment of thermal comfort perception,” Energy Procedia, vol. 148, pp. 798–805, 2018.
[25] R. Sturgeon, “Automatic option discovery within non-stationary environments,” Ph.D. dissertation, 2018.
[26] S. Suman, A. Etemad, and F. Rivest, “Potential impacts of smart homes on human behavior: A reinforcement learning approach,” IEEE Trans. Artif. Intell., Nov. 13, 2021, doi: 10.1109/TAL.2021.3127483.
[27] F. Tartarini, S. Schiavon, T. Cheung, and T. Hoyt, “CBE thermal comfort tool: Online tool for thermal comfort calculations and visualizations,” SoftwareX, vol. 12, 2020, Art. no. 100635.
[28] A. L. Thomaz and C. Breazeal, “Teachable robots: Understanding human teaching behavior to build more effective robot learners,” Artif. Intell., vol. 172, no. 6/7, pp. 716–737, 2008.
[29] A. Vaswani, “Attention is all you need,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 6000–6010.
[30] Z. Wang et al., “Individual difference in thermal comfort: A literature review,” Building Environ., vol. 138, pp. 181–193, 2018.
[31] Z. Wang, T. Parkinson, P. Li, B. Lin, and T. Hong, “The squeaky wheel: Machine learning for anomaly detection in subjective thermal comfort votes,” Building Environ., vol. 151, pp. 219–227, 2019.
[32] A. Yousef, A. Y. A. Amer, N. Caballero, and J.-M. Aerts, “Towards online personalized-monitoring of human thermal sensation using machine learning approach,” Appl. Sci., vol. 9, no. 16, 2019, Art. no. 3303.
[33] P. Zhu, X. Li, P. Poupard, and G. Miao, “On improving deep reinforcement learning for POMDPs,” 2017, arXiv:1704.07978.