A Deep Reinforcement Learning Based Recommender System for Occupant-Driven Energy Optimization in Commercial Buildings

Peter Wei, Stephen Xia, Runfeng Chen, Jingyi Qian, Chong Li, Xiaofan Jiang

***Abstract*—In this work, we present *recEnergy*, a recommender system for reducing energy consumption in commercial buildings with human-in-the-loop. We formulate the building energy optimization problem as a Markov Decision Process, show how deep reinforcement learning can be used to learn energy saving recommendations, and effectively engage occupants in energy-saving actions. *recEnergy* is a recommender system that learns actions with high energy saving potential, actively distribute recommendations to occupants in a commercial building, and utilize feedback from the occupants to learn better energy saving recommendations. Over a four week user study, four different types of energy saving recommendations were trained and learned. *recEnergy* improves building energy reduction from a baseline saving (passive-only strategy) of 19% to 26%.**

***Index Terms*—Deep Reinforcement Learning, Recommender System, Building Energy Optimization, Energy Savings.**

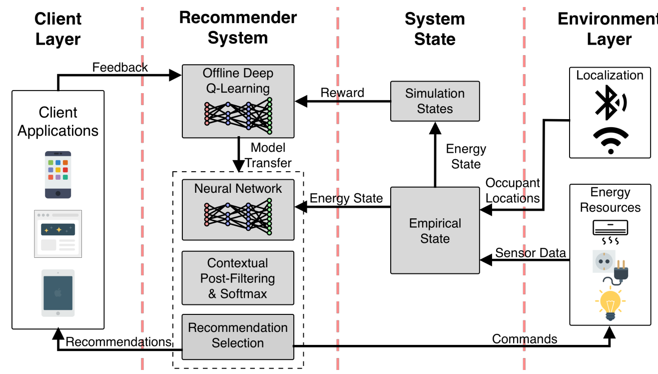

Fig. 1: System architecture of *recEnergy*. The system is composed of four layers: the environment layer, system state layer, recommender system, and client layer.

I. INTRODUCTION

Recent research efforts have made significant progress in reducing commercial building energy consumption through a variety of methods, including optimizing building heating, ventilation, and air conditioning (HVAC), lighting, and personal electric devices [1], [2]. These works focus on reducing energy consuming resources while treating occupants as immovable objects separate from the building energy optimization problem.

However, there is a limit to how much energy can be reduced by passively optimizing around occupants, especially considering most of the energy consumption in a commercial building directly or indirectly services occupants. In this work, we show the potential to further increase the amount of energy savings by shaping occupant behavior and engaging them in energy saving actions.

For occupants to effectively participate in the building energy optimization, they require knowledge of what actions have energy saving potential. There are two obstacles: first, actions that save energy are not always intuitive. The energy saving potential of actions such as reducing electricity usage, or changing the thermostat setpoint is intuitive; however, the energy saving potential of changing location at a certain time is not immediately obvious. Second, some actions that save energy in some situations may actually cause an increase in energy consumption in other situations. For example, arriving at work earlier than usual may result in energy savings on some days of the week, but may also result in excess energy consumption on other days.

Additionally, certain actions may require significant effort on the part of the occupant. Some occupants may be more, or less, willing to perform certain energy saving actions. For example, some occupants may be more inclined to shift their schedule later, while others may have responsibilities early in the day which prevents them from shifting their schedule. Therefore, the energy saving potential of an action must also account for how likely an occupant is willing to perform the recommended action.

To address these challenges, we design a **recommender system**, which **monitors the current building state**, **predicts the actions** that have the best potential for **saving energy**, and **distributes recommendations** to the occupants in a **timely** manner. We formulate the building energy saving problem as a Markov Decision Process and implement the recommender system using deep reinforcement learning, which not only allows us to learn the energy saving potential of actions, but also scales to larger building deployments.

In [3], the authors show that certain feedback mechanisms given to occupants raises awareness and can lead to energy saving behavior. This has been verified in the home and commercial building environments, where previous works such as [4], [5], [6], [7], [8], [9], [10] have shown that occupants

P. Wei, S. Xia, J. Qian, C. Li and X. Jiang are with the Department of Electrical Engineering, Columbia University, New York, NY, 10027, USA, email: {wei.peter, stephen.xia, jq2260, cl3607, xiaofan.jiang}@columbia.edu.

R. Chen is with the Department of Computer Science, Yale University, New Haven, CT, 06520, USA, email: runfeng.chen@yale.edu.

can be influenced to save energy if given feedback and incentives. In this work, we focus on increasing energy saving behavior in two ways: firstly, by improving the quality of recommendations through deep reinforcement learning; and secondly, by incorporating important information such as potential energy saving and feedback from users associated with each action.

The contributions of this work are as follows:

- We introduce *recEnergy*, a scalable system for cooperative optimization of energy consumption through a mixture of personalized energy saving recommendations for building occupants and building management system (BMS) commands. *recEnergy* takes advantage of human-in-the-loop to reduce energy further than passive optimization alone.
- We demonstrate the generalizability of *recEnergy* in a commercial building testbed by learning three different classes of energy saving recommendations, including individual recommendations, cooperative actions, and recommendations for building managers.
- A novel formulation of the building energy optimization problem is proposed. We first devise the problem as a Markov Decision Process, then show how deep reinforcement learning can be used to learn energy saving recommendations. Further, we demonstrate how to adapt different types of recommendations to fit the deep reinforcement learning framework.
- We deployed the system in a commercial setting, and performed a user study to measure real energy savings. Throughout the four-week deployment, our system produced a reduction of 7% in energy consumption with four types of energy saving recommendations.

It is critical to note that in *recEnergy*, occupants are given the *choice* to **accept** or **reject** an energy saving recommendation. This design decision is important, as the recommendations should have minimal impact on the daily lives of occupants. Furthermore, the recommender system provides the advantage of learning these preferences and tailoring future recommendations appropriately.

II. Related Work

Building energy consumption and optimization has been a topic of great interest within the research community in recent years as a means of environment conservation and waste management. As a result, many research papers with regards to reducing energy consumption in residential and commercial buildings have been published.

The first group of works focus on optimizing the design of buildings to reduce energy costs, such as incorporating natural ventilation [11] and optimizing window placement [12]. These "high-performance" buildings were shown to have above standard savings when compared with standard buildings, with some buildings reaching over 50% savings [13]. However, studies have also shown that occupant behavior is a primary factor that determines the final energy consumption in buildings [14], [15], [16]. This implies that the savings potential afforded by high-performance buildings can only be realized if occupants make energy conscious decisions (e.g. use natural lighting and heating when available).

Another group of research studies focuses on evaluating the potential energy savings for various energy consuming resources in buildings. The authors of [17] claim that there is a huge potential to save energy in electrical appliances and lighting through actions such as manually turning off the lights. Additionally, there are works that show a similar large potential to save energy from heating, ventilation, and air conditioning (HVAC) [18]. To realize these savings, these papers assume that people are willing to change their behavior to make energy conscious decisions. Fortunately, a multitude of studies show that people are willing to consider energy saving actions through user feedback and recommendations [4], [5], [6], [7], [8].

The prior work on high-performance buildings and energy savings in building appliances show that occupant behavior has a considerable influence on potential energy savings, but there are few systems that directly attempt to incorporate occupant behavior in optimization of energy consumption in buildings. The authors of [19] propose a system that learns occupant behavior and adapts building operations accordingly (e.g. the system learns when to heat or cool a space depending on past occupancy observations of the space). This system adapts the building to a user's habits, rather than pushing the user to save energy. Additionally, the authors of [20] propose a building recommendation system that recommends building control schemes to building managers based on feedback from other building managers. These works focuses on optimizing the behavior of the building management systems (BMS), rather than influencing energy savings through changes in occupant behavior.

In addition, a number of studies have implemented modern techniques such as reinforcement learning for incorporating occupant behavior into energy optimization. Reinforcement learning has the advantage of being adaptable to dynamic environments; if the initial model of energy consumption is inaccurate, one can provide empirical data to reduce the error between the initial model and the actual behavior. [21] presented a deep reinforcement learning framework to gradually learn optimal control strategies of energy resources; they further implemented the control logic in a real radiant heating system. [22] utilized Q-learning to learn an occupant behavior model with the goal of reducing uncertainty in energy simulations. Finally, a number of works [23], [24], [25], [26] have also utilized reinforcement learning to learn HVAC control strategies although the control strategies are often tested in simulations such as EnergyPlus [27]. All of these methods utilize reinforcement learning to optimize **building** parameters, but do not incorporate **occupant** actions in collaboratively reducing building energy usage.

Finally, there is one recent work that provides recommendations for occupants to optimize energy consumption in commercial buildings [10]. In this work, the authors utilize Q-learning to generate recommendations, demonstrate simulations of energy saving recommendations presented to occupants, and the simulated energy savings. The novelties of this work over [10] is three-fold. Firstly, *recEnergy* implements a deep reinforcement learning framework that is able to incorporate four different types of energy saving recommendations. Secondly, *recEnergy* utilizes deep Q-learning, which allows scalability to a much greater number of states than basic Q-learning. Finally, this work performs a more comprehensive evaluation of the system with a real user study.

III. Problem Formulation

We propose to include occupants as an integral component of energy optimization in commercial buildings. By optimizing the building energy consumption over the occupants and the building's energy consuming resources, greater energy savings can be realized. However, most occupants are unaware of what actions to take in order to help reduce energy consumption. A recommender system that provides personalized energy saving recommendations can equip occupants with this knowledge.

Designing a recommender system for saving energy in commercial buildings is difficult for a number of reasons. First, the effects of an action on the energy consumption are difficult to measure. There are a large number of variables that contribute to energy consumption and combining all of the variables into a model can be impractical. Second, the building environment does not remain static; rather, the locations of occupants, the weather outside, the time, and other factors are transient. Finally, acceptance probabilities of recommendations given to an occupant are dependent on the individual person and are nearly impossible to estimate.

A combination of deep learning and reinforcement learning provides a framework that addresses many of these difficulties. Model-free reinforcement learning is beneficial for the ability to maximize long-term return without an explicit model of the environment, including some human factors. As an example, reinforcement learning has been widely adopted in commercial products. Our recommender system is built on top of that rich body of knowledge, aiming to provide high quality and relevant recommendations to our users. Deep learning greatly improves the scalability of reinforcement learning, especially for large scale deployments.

A. Deep Reinforcement Learning

We consider the building energy saving problem in the context of reinforcement learning. At each time step, the agent, or recommender system, uses a *policy* to choose an *action* a from a set of possible actions. This action is sent to the actor(s) (occupant and/or building manager), which is then accepted or rejected. A *reward*, or energy saved, is measured and used by the recommender system for future learning. (As an action may have impact many time steps into the future, it is important for the learned value to account for future rewards.)

The building environment is constantly changing over time, due partially to agent recommendations but also to external factors such as occupant location changes and environmental factors, e.g, outside temperature. Thus, it is intuitive to understand the problem as a finite Markov Decision Process (MDP). In this representation, an action changes the state of the building environment. The possible actions are generated from four types of recommendations, which are described in Section III-B. Each action has a probability of occurrence at the current state, as well as a reward (r), representing the energy savings of the action.

The recommender system's goal is to choose recommendations that maximize the total energy saved over the long term given the current building state s. Standard reinforcement learning techniques seek to maximize *return* at time t, which is defined as $R_t = \sum_{i=t}^{T} \gamma^i r_i$, with $\gamma \in (0,1)$ as the "discount factor", T representing the end time (in commercial buildings, can be the end of the day), and r represents the immediate reward. One widely-used model-free reinforcement learning method to maximize the return is called Q-learning, where the agent seeks an action-value function $Q(s,a)$ which represents the return of taking an action a at a given state s. Often times, if the state/action space is too large, e.g., a huge amount of finite states/actions or continuous state/action space, $Q(s,a)$ is too complex to be stored in a data structure [28]. In this case, a function approximator can be used to estimate Q(s,a). In 2015, a ground-breaking paper showed that the optimal action-value function can be obtained with a deep convolutional neural network, when invoking the methods of experience replay and fixed Q-targets [29].

B. Recommendation Types

Different types of recommendations can save energy in different ways. Some energy savings can be realized by actions from only the occupant; some require action from both the occupant and the building; and finally, some are realized by the building with no action required from the occupants. Our system includes four types of recommendations that encompass these three classes of energy savings recommendations.

In this work, the main energy saving mechanism we utilize is by "relaxing" the service requirements. The mechanism works as follows: HVAC consumption can be reduced by allowing the setpoint temperature to deviate by a certain amount, such as ± 2 degrees; lighting can often be completely turned off. Plug load energy consumption can also be reduced depending on the operation of the load. Works such as [30], [1], [2], [31] demonstrate energy savings bewteen $10-40\%$ as a result of reduced service.

1) Move Recommendation: Move recommendations encourage an occupant to change to a different location. Energy consumption in a space can be reduced if the space becomes unoccupied because of the reduced service requirements of shared energy consuming resources such as HVAC and lighting.

2) Schedule Change: Schedule change recommendations encourage an occupant to change the period of time (not the duration) spent within a space. The mechanism for reducing energy consumption is similar to that of the move recommendation. For example, if an occupant tends to arrive earlier than other occupants, the building must begin servicing the occupied space earlier. However, if an occupant shifts his schedule to arrive later, the building can begin to service the space later, thus reducing energy consumption.

Action Type	Example	Actor	Number of Actions
Move	Occupant 1, move to space A	Occupant/Building	$\|P\| \times \|S\|$
Shift Schedule	Occupant 1, come to lab now to save energy	Occupant/Building	$\|P\|$
Reduce	Occupant 2, reduce power consumption on plugmeter 2	Occupant	$\|D\|$
Coerce	Building managers, reduce setpoint of space 1 by 2 degrees	Building	$\|S\|$

TABLE I: Action types, examples of each action type, the actors performing the action, and the number of possible actions of each type. P, D, and S denote the sets of occupants, personal energy resources, and spaces respectively, and $|\cdot|$ is the cardinality of the set.

3) Personal Resources: Recommending an occupant to reduce the power of energy consuming resources can lead to immediate energy savings. Reduction of personal computers, idle lab equipment, and other electronic devices can also reduce energy consumption over time.

4) Coerce Recommendation: In addition to the three types of recommendations described previously, we propose a new category of recommendations for building managers, which reduces service in spaces meeting certain criteria, even with the presence of occupants. The motivation for this recommendation is two-fold. First, if the occupancy in the room is small relative to the size of the space (e.g. two people in a hundred-person auditorium), then maintaining regular service levels can consume excessive amounts of energy. The recommendation would thus suggest to the building manager to reduce service to the space. Second, reducing the service in a space would also reduce the comfort level, thus physically encouraging occupants to vacate the space. The degree to which this is used, if at all, depends on the policy, typically set by the organization. During this process, safety, comfort and productivity issues of the occupants will be considered.

C. State Space

The state s of the building can be represented as a set of features. Ideally, the set of features should account for all factors that may impact the amount of energy savings of a recommendation; however, in practice it is difficult to monitor all of these features. For this system, we chose features that impact the energy savings for each recommendation and are easily measurable. The features we chose for each type of recommendation are:

- Move Recommendation: the location of occupants and the possible energy saved in each space.
- Schedule Change: the location of occupants and the possible energy saved in each space.
- Personal Resources: the current energy consumption of each resource.
- Coerce Recommendation: the occupancy of each space and the possible energy saved in each space.

D. Action Space

The action space represents the set of all possible actions for a building state. The actions are selected to best represent the four types of recommendations described previously, and include actions that may or may not have been explored in the past. Table I displays the different action types, an example for each action type, which actors are involved in the action type (occupant, building, or both), and how many possible actions are included in the recommender system for the action type. The possible actions are concatenated into a vector that acts as the ”label” for the training data to the neural network.

E. Rewards

Associated with each action is a potential *reward*, or amount of saved energy. There are two considerations for calculating the reward as saved energy: recommendation acceptance, and recommendation duration. First, if an occupant rejects a recommendation, then the action has saved no energy (the building state does not change). A recommendation that is rejected many times will have a lower return than a recommendation that saves the same amount of energy, but has a higher acceptance rate. Second, energy savings is computed as the integration of power reduction over time. A recommendation that reduces the power consumption by 100 W will save 100 Wh of energy over one hour, and 200 Wh of energy over two hours. Changes in the building environment can interrupt the energy saving effects of a recommendation; for example, an occupant may choose to move to a different space without being recommended by the system. In depth descriptions for calculating rewards of the different recommendation types are given in Section IV-C.

F. Recommender System Example

Here we provide a small example to illustrate the advantage of reinforcement learning. Consider two occupants, A and B, who commonly occupy space S. If space S is unoccupied, the shared power consumption (HVAC and light) can be reduced by 100 W. During exploration, occupant A is given a move recommendation to move to a different space. The immediate energy saved is 0 Wh, as space S is still occupied by occupant B. After occupant A has vacated the space, two recommendations are given: a coerce recommendation, recommending the building manager to reduce HVAC service to space S due to low occupancy, and a move recommendation, recommending occupant B to vacate space S. If occupant B accepts the recommendation, the building reduces the HVAC and light services by 100 W in space S, and a reward is propagated back to occupant B (the return for occupant A will include a discount of the reward given to occupant B in future calculations). On the other hand, if occupant B rejects the recommendation, the coerce recommendation can be carried out. An illustration of this scenario is shown in Figure 2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974848, IEEE Internet of Things Journal

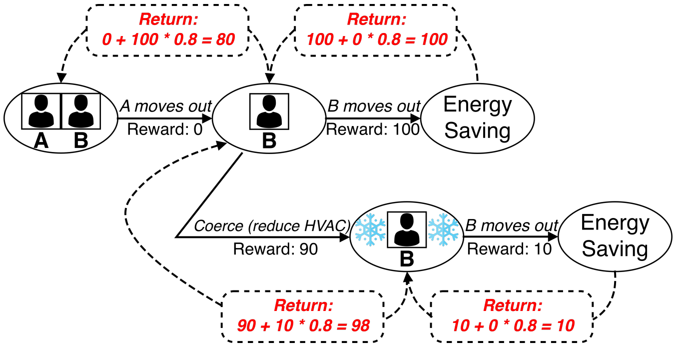

Fig. 2: Recommender system example.

IV. Architecture and Implementation

Figure 1 shows the architecture of *recEnergy*. The system is composed of four layers: the environment layer which measures the building environment; the system state, which maintains the current building state and necessary simulation states; the recommender system, which learns energy saving potential for four different types of recommendations; and the client applications, which receive recommendations and allow users to provide feedback on the relevance of recommendations. The blocks in gray are implemented in a cloud server; the blocks in white are physical devices or sensors.

The benefit of this four layer system is that different implementations of the layers can be interchanged without affecting the overall behavior of the system. For example, the client layer communicates to the recommender system via APIs by receiving energy saving recommendations and allowing users to accept or reject the recommendations. Regardless of the implementation of the client, if the feedback and recommendation mechanisms are available, the client fits seamlessly within the system.

A. Environment

The environment layer consists of two subsystems for measuring the building environment. This layer measures the occupant location and energy consumption of energy resources, and sends this data to the system state layer. The environment layer and system state layer interface through APIs; thus, if other features are required for different types of recommendations, a new monitoring subsystem can be added and connected seamlessly to the system state layer.

1) Localization: Indoor localization is a popular method for determining the location of occupants within a building. In this system, we utilize Bluetooth localization due to the technology's low cost, ease of deployment, and ability to provide coarse-grained location information. 42 BluVision iBeek beacons are deployed throughout the testbed and emit signals at a frequency of 2 Hz. Each occupant's mobile device polls the Bluetooth beacon signal strength values; these values are sent to the server, where a fingerprint-based approach is used to determine the location.

2) Energy Monitoring: In our testbed, we monitor three types of energy consuming resources: HVAC, lights, and plugmeters. To monitor the majority of HVAC resources, we utilize the BACNet protocol, and calculate the energy consumption as described in [32]. Some HVAC resources are not monitored on the BACNet; these resources are monitored using a custom sensing node consisting of a wind velocity sensor and temperature sensor. Lights are monitored using a TSL2561 luminosity sensor and a Huzzah Feather Board. Each occupant participating in our user study is given a plugmeter to track their energy consumption, which sends data to the server through a Samsung smarthub.

B. System State

The system state layer consists of two components: an empirical state, which maintains the current building state, and a simulation state, described in Section IV-C, which calculates variations of the building state given different parameters. The system state layer is primarily responsible for aggregating information gathered by the environment layer, such as location of occupants, energy consumption of resources, and weather conditions. The system state layer produces a number of simulation states representing next states after potential energy saving actions are taken, and passes this along to the recommender system layer.

1) Empirical State: The empirical state is the real-time state of the system. This state is composed of a number of parameters, including the locations of the occupants of the building, configuration of the spaces in a building, and the current energy consumption of appliances and devices. Similar to the tripartite data structure in [9], only the locations of occupants and energy consumption of appliances change in the system.

The empirical state is important for two reasons. Firstly, the recommender system directly utilizes a vector of values representing the empirical state to generate recommendations to the client. An example of the state variables vector passed to the recommender system is shown in Figure 4. Secondly, the empirical state is the baseline state on which the rewards of different actions can be simulated.

C. Simulation States

Measuring the amount of energy saved is difficult. If the building is functioning without external intervention, then the effect of a recommendation is unknown. On the other hand, if a recommendation has been given and the building has responded, the normal building operation is unknown. Ideally, energy savings of a recommendation could be measured by having two identical buildings, performing the recommendation in one of the buildings, and calculating the energy expenditure difference between the two buildings. However, this is rarely possible in practice; thus, simulation is vital to approximating the amount of energy saved.

The goal of the simulation is to produce simulation states that are the result of a single action, or a chain of actions, on the system state. To generate the simulation states, the simulation must have a model of the action's effects on the building's energy consumption. Each recommendation type affects the building environment in different ways, and thus different models may be necessary.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974848, IEEE Internet of Things Journal

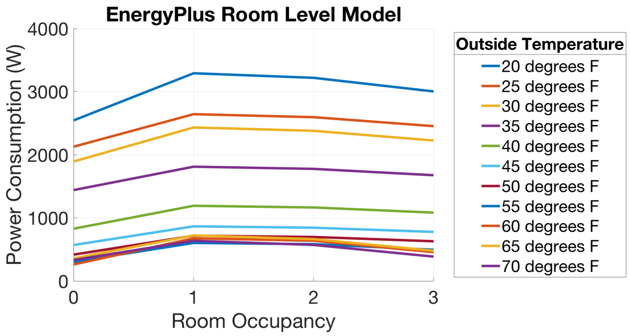

Fig. 3: Power consumption curves for a simulated space. Power varies over outside temperature and occupancy. Values are simulated through EnergyPlus, based on the physical parameters of our test bed.

1) Definitions: Starting from the empirical state, a number of simulated states can be defined. We are mostly interested in three common types of energy consuming resources: HVAC, lighting, and plug loads. For each resource d_i, we assume a *normal* operation energy consumption at time t and with external parameters θ, $P_{d_i}(t;\theta)$, and a *relaxed* operation energy consumption $\hat{P}_{d_i}(t;\theta)$. In this application, relaxed operation refers to the energy consumption after an action has been taken. For HVAC, relaxed operation often refers to a relaxed setpoint temperature; for lighting, relaxed operation refers to the off state; for plug loads, relaxed operation refers to a low usage or no usage.

We also define the number of people affected by a resource as $O(d_i)$. This definition is important for this application, as many of the recommendations utilize occupancy as the primary factor for normal or relaxed operation of energy resources. We borrow the concept of "human-centric zoning" from [9] to help define affected occupants. The number of people affected is then the sum of the occupancy of each zone serviced by the energy resource.

2) EnergyPlus Power Model: Lighting and plug load power consumption is often unaffected by external parameters such as the outside temperature or numerical occupancy; for example, the lighting state is defined solely by the presence of occupants, while plug load consumption is often dependent only on whether the load is in use. HVAC, however, is highly dependent on outside temperature, and even the numerical occupancy. This makes it difficult to approximate the energy consumption for different parameters without a proper model.

To address this issue, we created empirical HVAC consumption models for each space. We simulate power consumption traces by utilizing *EnergyPlus* [27]. In EnergyPlus, we reconstructed each space and simulated the power consumption in different environmental conditions, θ. The environmental conditions we considered included occupancy and temperature, since these conditions vary most frequently throughout the course of a workday. We assume that the setpoint temperature can be relaxed by two degrees if the space is unoccupied, as in Section III-B. Examples of simulated power consumption curves used in our system are shown in Figure 3.

3) Move Recommendation: As described in Section III-B, HVAC and light service can be reduced when a space is unoccupied. Under this policy, the energy consumption of an energy consuming resource d_i can be calculated as:

$$\mathbb{E}_{d_i} = \int_{t=t_1}^{t_2} \mathbb{1}_{(O(d_i)=0)}\hat{P}_{d_i}(t;\theta) + \mathbb{1}_{(O(d_i)>0)}P_{d_i}(t;\theta)dt, \quad (1)$$

where $\mathbb{1}_{(\cdot)}$ is the indicator function, and t_2 is the end time of the recommendation, or the time when the recommendation stops yielding energy savings. In other words, from time t_1 to t_2, the energy resource operation is $relaxed$ if there are no occupants present, and $normal$ if there are occupants present.

To calculate the energy savings of a move recommendation, we can consider the power consumption of all energy resources with and without the move recommendation. Note that most of the energy resources will continue current operation; however, there are two sets of resources which change operation. One set, D_o, is the set of resources such that $O(d_i) > 0$ before the move recommendation and $O(d_i) = 0$ after. The second set, D_d, is the set of resources such that $O(d_i) = 0$ before the move recommendation, and $O(d_i) > 0$ after. Then, the energy savings of a move recommendation ($\mathbb{E}_R$) is:

$$\mathbb{E}_R = \int_{t=t_1}^{t_2} \sum_{d_i \in D_o} (P_{d_i}(t;\theta) - \hat{P}_{d_i}(t;\theta)) - \sum_{d_j \in D_d} (P_{d_j}(t;\theta) - \hat{P}_{d_j}(t;\theta))dt. \quad (2)$$

From Equation 2, the saved energy of a move recommendation depends on two power consumption traces for each resource: power consumption at normal operation, and power consumption at relaxed operation. At any time, one of the two power consumption traces can be measured from building sensors; the other must be simulated.

4) Shift Schedule: A shift schedule recommendation has two parts to consider: when the occupant arrives and when the occupant leaves. The recommendation should have as little effect as possible on the occupant's work, so the duration the occupant spends is assumed to be constant. A recommendation to shift schedule earlier will ideally save more energy when the occupant leaves than the extra energy consumption when the occupant arrives. Likewise, a recommendation to shift schedule later should save more energy when the occupant arrives than the additional energy consumption when the occupant leaves. The energy savings can be calculated as:

$$\mathbb{E}_R = \int_{t=t_1}^{t_1+\Delta t} \sum_{d_i \in D_a} (P_{d_i}(t;\theta) - \hat{P}_{d_i}(t;\theta))dt - \int_{t=t_2}^{t_2+\Delta t} \sum_{d_j \in D_d} (P_{d_j}(t;\theta) - \hat{P}_{d_j}(t;\theta))dt,$$

where D_a is the set of resources influencing the arrival space of the occupant, D_d is the set of resources influencing the departure space of the occupant, and Δt is the schedule shift.

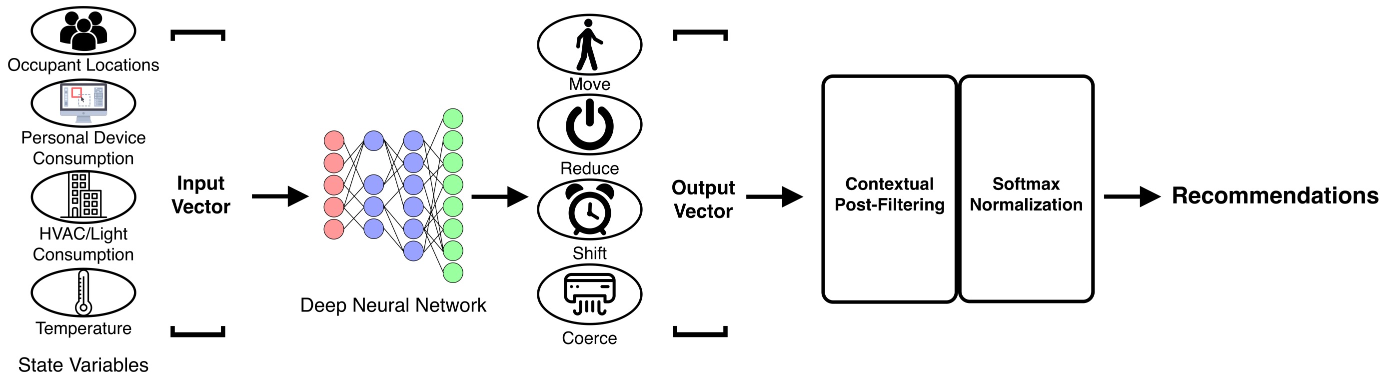

Fig. 4: Recommender system structure. The DNN model takes in a vector of state variables, and calculates a output return vector for all actions. The actions are further filtered before generating the recommendations.

5) Coerce Recommendation: The reward for the coerce recommendation of an energy consuming resource can be calculated as the integral of the power reduction:

$$\mathbb{E}_{d_i} = \int_{t=t_1}^{t_2} P_{d_i}(t;\theta) - \hat{P}_{d_i}(t;\theta) dt. \tag{3}$$

The end time, t_2, occurs when the occupancy changes. If the occupancy increases, then the service returns to normal operating setpoint temperature; if the occupancy reduces to zero, the energy savings will instead be credited to the vacating occupant.

6) Personal Resources: The amount of energy saved can be estimated by measuring the decrease in power consumption, and the amount of time the reduced power level is maintained. Similar to Equation 3, the energy saved is calculated as the integral of the power level reduction over the duration of lower power consumption.

D. Recommender System

The information flow of the recommender system is shown in Figure 4. The current empirical building state is collected and fed into the input layer of the neural network. The neural network computes the return of each action. Contextual post-filtering and softmax normalization are used to remove invalid recommendations, encourage exploration of novel recommendations, and to select recommendations for each occupant. Finally, the recommendations are sent to the client layer (mobile devices and wearables).

The neural network is composed of four layers in total: one input layer, two hidden layers and an output layer. The input layer contains 75 nodes, the output layer contains 210 nodes, and each hidden layer has 100 nodes.

1) Training Data: Ideally, all of our training data should come from the testing data set we build while the users are using our system. Unfortunately, this would take too long to acquire a reasonable amount of data; this issue is known as the cold start problem in recommender systems. To solve this problem, we utilized data augmentation to create a larger training data set based on our user study data. The core idea is that given a recommendation, certain state variables can be changed without affecting the probability of an occupant accepting the recommendation. Figure 5 shows a example of how data augmentation is performed. In this example, there are four rooms, each with its own HVAC control. State 1 represents the current building state, and the person in room 1 receives a recommendation to move to room 2. With regards to this recommendation, room 3 and room 4's conditions should have no effect on his or her behavior, or the action's energy savings. Hence, we can augment the data set with similar states with more people in rooms 3 and 4, as an example. By augmenting the data set, we can ensure a broader coverage of the state space.

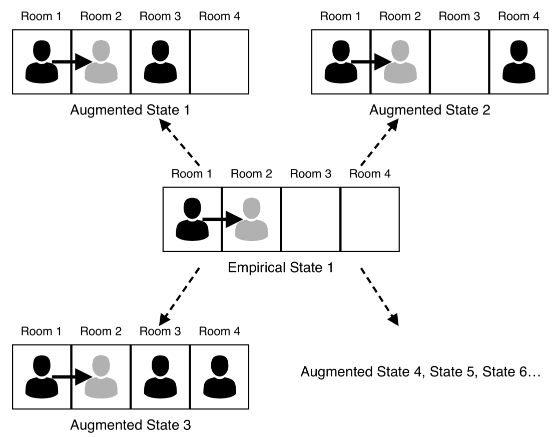

Fig. 5: Data augmentation example.

During the user study, the recommender system was regularly retrained with new data generated from the user study. Acceptance and rejection probabilities were included in the generated data, which helps to improve the quality of the data. The inclusion of both energy savings as well as acceptance and rejection probabilities leads to overall higher energy savings.

2) Recommendation Selection: After the offline training, the neural network is transferred to the server. The current building state is fed to the neural network, which outputs the return for each action. Before sending recommendations to the occupants, we perform contextual post-filtering and softmax normalization to further improve the quality of recommendations.

Contextual post-filtering is used to screen non-sensical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974848, IEEE Internet of Things Journal

recommendations depending on context. For example, a recommendation suggesting an occupant to move to an location which the occupant does not have access to should not be pushed to the occupant. Ideally, this module is not required as the DNN model learns which recommendations are unlikely to be accepted; however, due to the limited amount of gathered recommendation acceptance data, we chose to manually post-filter recommendations to decrease the amount of training data required.

When selecting a recommendation to send to an occupant, the primary tradeoff is between exploration and exploitation. On the one hand, the recommender system should *exploit* recommendations that have a high expected return; on the other hand, the recommender system should *explore* unknown recommendations which may result in even higher returns. To address this exploration vs. exploitation tradeoff, we utilize softmax normalization, as shown in Equation 4. The output values $q(a)$ for each recommendation from the DNN are given a probability $P(a)$ based on the softmax function. The temperature, τ, initially has a high value to encourage exploration; over time, the temperature decreases, which encourages exploitation. Using this method, recommendations are selected based on a probability distribution such that recommendations with high expected return are chosen more often, and recommendations with low expected return are chosen less often.

$$P(a) = \frac{\exp \frac{q(a)}{\tau}}{\sum_{i=1}^{n} \exp \frac{q(i)}{\tau}} \tag{4}$$

After contextual post-filtering and softmax normalization, recommendations are distributed to the occupants. If a recommendation has been pushed to an occupant recently, a backoff time is implemented to prevent the same recommendation from being pushed again within this time frame.

3) Recommendation Updates: The current state of the building is updated constantly as data is received from the sensors; for example, the HVAC and lighting sensors send data to the server at rates between every 15 seconds to every 5 minutes, and the locations of occupants is sent from mobile devices every 15 to 60 seconds.

While we did not explicitly study the optimal rate to send recommendations to users, there are a number of considerations which informed the recommendations update rates in *recEnergy*. Firstly, shift schedule recommendations are intended for arrival and departure changes the following day. Thus, these recommendations are only updated once per day.

Secondly, move and reduce recommendations are intended to save energy from the current building state. These recommendations may become irrelevant if too much time passes; thus, these recommendations should be provided as real-time as possible. However, small changes in the building state can affect the recommendations for a user, leading to rapid changes to a user's recommendation feed that may adversely affect the user experience. Thus, we allow move and reduce recommendations to persist for 5 minutes before changing these recommendations. In future works, studying different update rates can help optimize for user experience and maintain recommendation relevance.

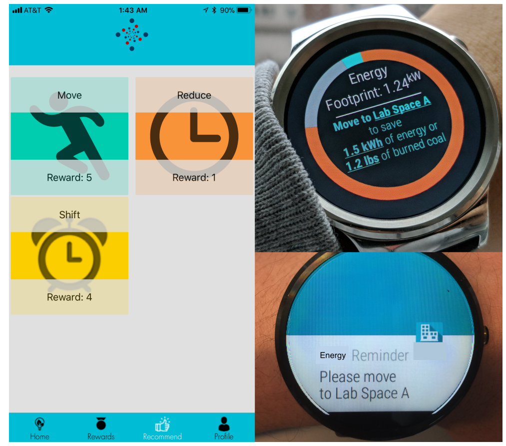

Fig. 6: iOS application recommendation display and wearOS recommendation notification.

E. Client Applications

1) Mobile Application: The mobile application we developed is based on a real-time energy monitoring application of a previous work [9]. The added recommendations page displays an occupant's current recommendations, as shown in Figure 6. A reward based on the recommendation's calculated return is also displayed together with the recommendation. After the occupant clicks on the block to either accept or reject the recommendation, the feedback is sent to the server and the block is deleted from the screen. New recommendations are pushed to occupants periodically.

2) Wearables: An Android Wear application was developed to reliably notify occupants of new recommendations. The application displays a notification whenever new recommendations are pushed to the occupant. The notification can be tapped on the wearable device to activate the paired mobile application.

V. Evaluation

A. Deployment Setup

To evaluate *recEnergy*, we deployed the system in a commercial building and conducted a user study lasting four weeks. The deployment spanned two floors in a campus building, and included a diverse set of spaces including cubicles, offices, conference rooms, and public spaces such as hallways. 99 BACNet endpoints, 15 shared ceiling lights, and 11 plugmeters, each connected to multiple personal appliances via power-strips, were monitored. Some energy consuming resources cannot be controlled by occupants in the building, or through the building management system without permissions. We envision that future buildings and energy saving systems will be capable of controlling all energy consuming resources. Thus, we chose to include recommendations that involve these energy consuming resources, and simulated energy savings when needed.

B. Room Level Models

To ensure an accurate estimate of the simulated energy states in Section IV-C, we developed room level models dependent on outdoor temperature, room parameters, and occupancy through EnergyPlus, as described in Section IV-C2. Figure 7 and Figure 8 show an example of real-time power and energy measurements for a space in our deployment. In both figures, the simulated power and energy consumption for normal operation and relaxed operation are shown as yellow and purple dotted lines, respectively. The realtime measurements are shown in blue, and the simulated measurements are shown in orange.

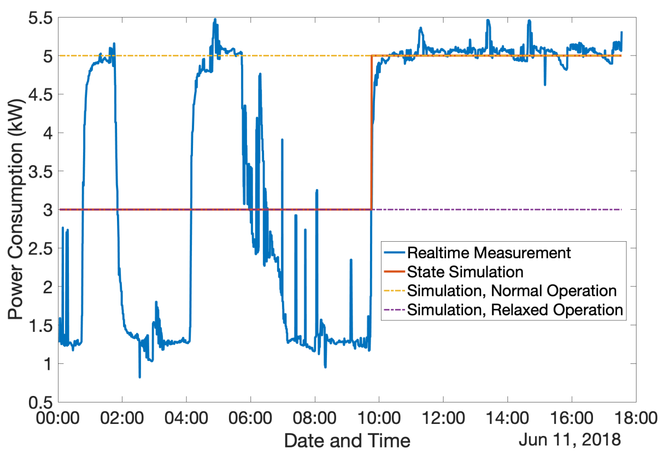

Fig. 7: Direct comparison of simulated power measurements and realtime power measurement data.

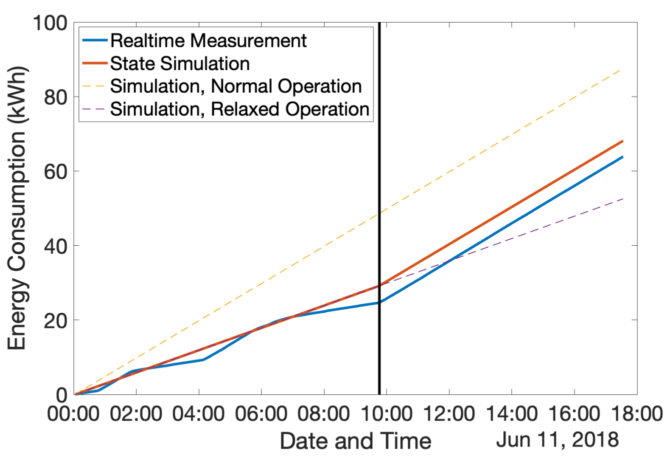

Fig. 8: Comparison of monitored energy data with simulated normal operation, simulated relaxed operation, and simulation state.

In this example, the room is unoccupied and in relaxed operation until 09:53 (indicated by the black vertical line in Figure 8). At this time, an occupant enters the room, causing the room to switch to normal operation. The realtime power consumption (blue) increases to 5000 W, causing the energy consumption slope to increase. The simulated power consumption (orange) also increases to 5000 W and the energy consumption slope increases to match the realtime consumption. This shows that our simulation states provide a reasonable approximation of the realtime energy consumption.

C. User Study

A user study was conducted to evaluate the recommender system. For this study, we deployed mobile applications to ten users, and regularly sent energy saving recommendations through the mobile user interface. The experiment kept track of 26 personal devices of the users, and contained 17 spaces that the users can occupy. The study was conducted in two periods: a control period and a recommender period. Each period lasted for a two-week duration. In the control period, the recommender system presented pseudo-random energy saving recommendations; in the recommender period, the recommender system presented the users with recommendations learned by the recommender system using the feedback from the control period.

1) Accepted Recommendations: One important metric for personalized recommendations is how likely an occupant is to accept the recommendation. Although a recommendation may theoretically save a large amount of energy, in practice there is no energy savings if the occupant does not accept the recommendation. The difference in the acceptance rate between the control period and recommender period is shown in Figure 9. As shown in the figure, the acceptance rate is much higher during the recommender period than during the control period. This dramatic increase is largely due to the improved usefulness and relevancy of the recommendations after our recommender system is deployed.

Most notably, we noted that recommendations accepted during the control phase were recommended significantly more during the recommender phase. One reason is that rejected recommendations are labeled with an energy saving of 0W to represent that the recommendation has not resulted in any energy savings. As a result, the acceptance rate of each recommendation plays an important role in the expected energy saving value; a recommendation with a higher acceptance rate will be valued higher than a recommendation with a similar energy saving value and lower acceptance rate. Thus, it is clear that our recommender system is a success in terms of generating higher quality recommendations.

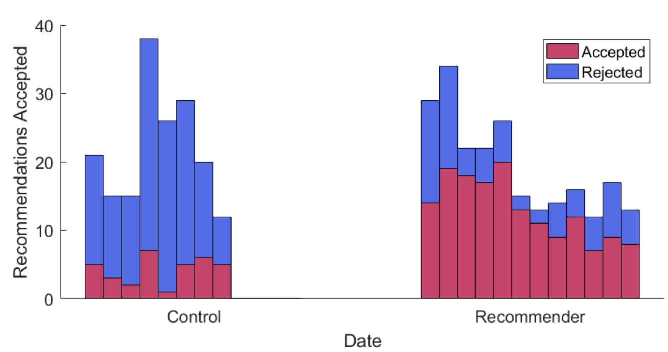

Fig. 9: Acceptance rate of recommendations in the control and recommender periods of the user study.

2) Recommendation Energy Savings: The key metric for the recommender system is the amount of energy savings achieved. As a baseline, we utilized a passive optimization strategy similar to [30], [1]. This strategy consists of relaxing

	Move	Shift Earlier	Shift Later	Reduce
Control	15%	32%	0%	53%
Recommender	73%	75%	15%	67%

TABLE II: Acceptance rate of different type of energy saving recommendations.

the setpoint temperature of spaces which are unoccupied, without influencing the occupants.

In addition, the energy saved by each accepted recommendation, and the potential saved energy by each rejected recommendation, were recorded. The system energy savings is the energy saved by the recommender system and the passive optimization strategy. The potential energy savings is calculated by including the potential saved energy (result if all recommendations were accepted). The results for the control and recommender phases are displayed in Figure 10.

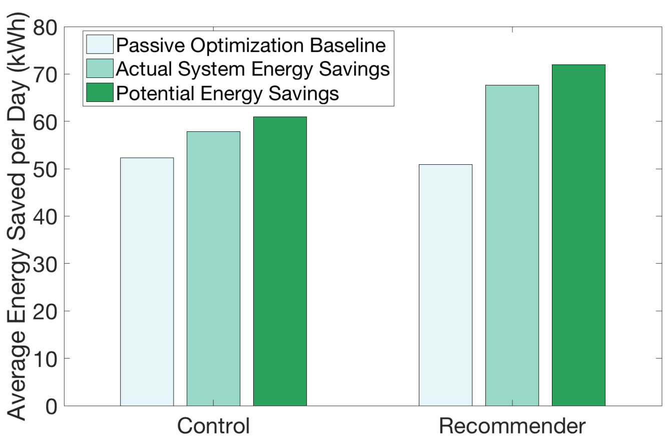

Fig. 10: Comparison of energy savings from passive energy optimization, as well as actual and potential energy savings from the *recEnergy* deployment.

The recommender phase produced an increase in energy savings over the control phase; this is likely due to the improvement in recommendation quality, leading to more accepted recommendations as in Section V-C1. Additionally, the energy saved due to recommendations increased energy savings of the passive optimization baseline from 19% to 26%.

The total and average energy savings for each type of recommendation are shown in Table III. Reduction of personal resources resulted in the least amount of energy savings, while move recommendations accounted for the highest total energy savings. Additionally, the average energy savings of coerce recommendations were highest among all recommendations. While these initial results are promising, further research may discover new energy saving recommendations that can result in even greater energy savings.

D. Scalability

The recommender system should be scalable to accommodate larger deployment sizes and greater numbers of occupants, spaces, and energy consuming resources. To demonstrate the scalability of *recEnergy*, we tested the recommender system using larger deployment sizes, as shown in Table IV. The neural network computations consume the largest fraction of time; however, even at large deployment sizes, the entire recommendation pipeline finishes within 10 seconds, showing that the system can scale to larger deployments.

Recommendation	Energy Saved	Average	% Total
Passive Energy Savings	458.5 kWh	N/A	21%
Move Recommendation	**91.9 kWh**	1.70 kWh	4.2%
Coerce Recommendation	44.6 kWh	**2.79 kWh**	2.0%
Shift Schedule	12.9 kWh	0.76 kWh	0.6%
Reduce Personal Energy	0.58 kWh	0.02 kWh	0.02%

TABLE III: Total energy saved, average energy saved, and percentage of total energy for each energy saving recommendation type during the recommender phase. Passive energy savings is included as a baseline.

People	Devices	Spaces	State	DNN	Recs
40	80	20	17.2ms	304.8ms	45.1ms
160	320	80	34.4ms	546.1ms	66.2ms
400	800	200	62.9ms	2264.2ms	84.5ms
480	960	240	71.8ms	3983.4ms	101.4ms
640	1280	320	76.3ms	8989.8ms	119.9ms

TABLE IV: Timing measurements for generating the DNN input vector from the state variables (State), running the DNN (DNN), and generating recommendations through contextual post-filtering and softmax normalization (Recs) for varying deployment sizes.

VI. Future Directions

This work demonstrates the potential for reduced energy consumption as a result of a deep reinforcement learning based recommender system. However, there are a number of future research directions which can improve *recEnergy*. Firstly, *recEnergy* can be improved with a model that can accommodate changes to the occupants, energy resources, and recommendations. Secondly, the recommendations can be improved by incorporating user costs to increase recommendation quality and providing rewards to increase user engagement. Finally, different recommendations can be included to increase recommendation diversity, giving users additional options for saving energy. We describe our plans in these three directions below.

A. Model Adaptability

The current model used in *recEnergy* relies on a fixed set of occupants, energy resources and recommendations. In many scenarios, the occupants and energy resources may vary (e.g. visitors, or newly installed energy resources), and new recommendations may become available. To accommodate for these changes, a new model is required which can account for an unknown number of occupants, energy resources and recommendations. For example, one possibility for addressing a variable number of occupants is to modify the state space to include an occupancy percentage for each location, rather than the exact location of each occupant; this would allow for new occupants to be integrated into the state space. An adaptable model would enable deployment of *recEnergy* into environments with transient sets of occupants, energy resources and recommendations.

B. System

Another important extension of this work is the investigation of the "cost" of a recommendation to the occupant. The recommendations explored in this work are dependent on the actions of the occupant, and thus requires effort on behalf of the occupant. In the future, metrics such as comfort and productivity can be measured as associated costs of recommendation actions. Investigation of these costs can improve understanding of which recommendations are more likely to be accepted.

To further incentivize occupants of *recEnergy*, a monetary or points reward system can be incorporated in the recommender system. Because energy savings do not directly compensate the users, a reward system will provide tangible incentives to increase the percentage of accepted recommendations and the energy saved. Furthermore, the rewards can be given directly on the mobile application. One potential strategy is to adjust the rewards proportional to the expected amount of energy saved, to encourage users to accept recommendations with higher energy saving potential.

C. Recommendations

While this work explores a sample of possible energy saving recommendations, other recommendations are possible. For instance, group recommendations (recommendations given to a group of users) might result in more consistent energy savings. Recommending a small group of associated occupants to change location, resulting in an unoccupied space, may be more effective than sending a move recommendation to each individual occupant.

The academic community has explored a number of energy saving actions which can be tailored into recommendations. Examples of these actions include manipulation of shade as a control of natural light [33], and manipulation of windows for natural ventilation [34], [35]. These actions could be coupled with a reduction in lighting or ventilation to reduce energy consumption.

The addition of different recommendations is important for *diversity* [36] in the recommender system. By providing a more diverse set of recommendations, the recommender system may have a higher chance of showing the user is willing to perform, thus increasing the potential energy savings.

VII. Conclusion

In this paper, we demonstrate the potential of a deep reinforcement learning based recommender system to reduce energy consumption in a commercial building. We formulate the problem as a Markov Decision Process, and defined four different types of energy saving actions: move recommendations, shift schedule recommendations, coerce recommendations, and reduce personal resources recommendations. Over a four week user study, *recEnergy* learned expected energy savings for these recommendation types, distributed recommendations to ten subjects through their mobile devices, and utilized occupant feedback to retrain the deep neural network and improve recommendations both in energy saved and in occupant acceptance rate. By using passive energy saving strategies as a ground truth, we found that *recEnergy* improves building energy reduction from a baseline saving (passive-only strategy) of 19% to 26%.

Acknowledgment

This research was partially supported by the National Science Foundation under Grant Numbers CNS-1704899 and CNS-1815274. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed implied, of Columbia University, NSF, or the U.S. Government or any of its agencies.

References

[1] B. Balaji, H. Teraoka, R. Gupta, and Y. Agarwal, "Zonepac: Zonal power estimation and control via hvac metering and occupant feedback," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. ACM, 2013, pp. 1–8.

[2] V. L. Erickson, M. Á. Carreira-Perpiñán, and A. E. Cerpa, "Observe: Occupancy-based system for efficient reduction of hvac energy," in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*. IEEE, 2011, pp. 258–269.

[3] S. Darby *et al.*, "The effectiveness of feedback on energy consumption," *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, no. 2006, p. 26, 2006.

[4] B. Xia, H. Ming, K.-Y. Lee, Y. Li, Y. Zhou, S. Bansal, S. Shakkottai, and L. Xie, "Energycoupon: A case study on incentive-based demand response in smart grid," in *Proceedings of the Eighth International Conference on Future Energy Systems*. ACM, 2017, pp. 80–90.

[5] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?" *Energy Efficiency*, vol. 1, no. 1, pp. 79–104, Feb 2008.

[6] E. Hunter, "Motivating Household Energy Conservation Using Feedback and Social Nudges: A Field Experiment," Master's thesis, 2016.

[7] J. E. Petersen, V. Shunturov, K. Janda, G. Platt, and K. Weinberger, "Dormitory residents reduce electricity consumption when exposed to real-time visual feedback and incentives," *International Journal of Sustainability in Higher Education*, vol. 8, pp. 16–33, 2007.

[8] J. K. Day and D. E. Gunderson, "Understanding high performance buildings: The link between occupant knowledge of passive design systems, corresponding behaviors, occupant comfort and environmental satisfaction," *Building and Environment*, vol. 84, pp. 114–124, 2015.

[9] P. Wei, X. Chen, J. Vega, S. Xia, R. Chandrasekaran, and X. Jiang, "eprints: a real-time and scalable system for fair apportionment and tracking of personal energy footprints in commercial buildings," in *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM, 2017, p. 6.

[10] P. Wei, S. Xia, and X. Jiang, "Energy saving recommendations and user location modeling in commercial buildings," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. ACM, 2018, pp. 3–11.

[11] P. A. Torcellini, R. Judkoff, and D. B. Crawley, "High-performance buildings," *ASHRAE Journal*, vol. 46, no. 9, pp. S4–S11, 09 2004.

[12] J. Carmody, S. Selkowitz, E. S. Lee, D. Arasteh, and T. Willmert, *Window Systems for High-Performance Buildings*. W. W. Norton & Company, Inc., 2004.

[13] P. A. Torcellini, M. Deru, B. Griffith, N. Long, S. Pless, R. Judkoff, and D. Crawley, "Lessons learned from field evaluation of six high-performance buildings," in *ACEEE Summer Study on Energy Efficiency in Buildings*. ACEEE, 08 2004, pp. 3–325 – 3–337.

[14] A. Roetzel, A. Tsangrassoulis, and U. Dietrich, "Impact of building design and occupancy on office comfort and energy performance in different climates," *Building and Environment*, vol. 71, pp. 165–175, 2014.

[15] P. Hoes, J. Hensen, M. Loomans, B. de Vries, and D. Bourgeois, "User behavior in whole building simulation," *Energy and Buildings*, vol. 41, no. 3, pp. 295–302, 2009.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974848, IEEE Internet of Things Journal

[16] K. Anderson, K. Song, S. Lee, H. Lee, and M. Park, "Energy consumption in households while unoccupied: Evidence from dormitories," *Energy and Buildings*, vol. 87, pp. 335–341, 2015.

[17] M.-C. Dubois and A. Blomsterberg, "Energy saving potential and strategies for electric lighting in future north european, low energy office buildings: A literature review," *Energy and Buildings*, vol. 43, no. 10, pp. 2572 – 2582, 2011.

[18] F. Oldewurtel, D. Sturzenegger, and M. Morari, "Importance of occupancy information for building climate control," *Applied Energy*, vol. 101, pp. 521 – 532, 2013, sustainable Development of Energy, Water and Environment Systems.

[19] M. C. Mozer, "The neural network house: An environment hat adapts to its inhabitants," in *Proc. AAAI Spring Symp. Intelligent Environments*. AAAI, 1998, pp. 110–114.

[20] M. LeMay, J. J. Haas, and C. A. Gunter, "Collaborative recommender systems for building automation," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*. IEEE, 2009, pp. 1–10.

[21] Z. Zhang and K. P. Lam, "Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system," in *Proceedings of the 5th Conference on Systems for Built Environments*. ACM, 2018, pp. 148–157.

[22] M. Bonte, F. Thellier, B. Lartigue, and A. Perles, "An occupant behavior model based on artificial intelligence for energy building simulation," in *Proceedings of the 13th International IBPSA Conference BS2013, Chambery, France*, 2014.

[23] K. Dalamagkidis, D. Kolokotsa, K. Kalaitzakis, and G. S. Stavrakakis, "Reinforcement learning for energy conservation and comfort in buildings," *Building and environment*, vol. 42, no. 7, pp. 2686–2698, 2007.

[24] L. Yang, Z. Nagy, P. Goffin, and A. Schlueter, "Reinforcement learning for optimal control of low exergy buildings," *Applied Energy*, vol. 156, pp. 577–586, 2015.

[25] Y. Chen, L. K. Norford, H. W. Samuelson, and A. Malkawi, "Optimal control of hvac and window systems for natural ventilation through reinforcement learning," *Energy and Buildings*, vol. 169, pp. 195–205, 2018.

[26] Z. Zhang, A. Chong, Y. Pan, C. Zhang, S. Lu, and K. P. Lam, "A deep reinforcement learning approach to using whole building energy model for hvac optimal control," in *2018 Building Performance Analysis Conference and SimBuild*, 2018.

[27] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte *et al.*, "Energyplus: creating a new-generation building energy simulation program," *Energy and buildings*, vol. 33, no. 4, pp. 319–331, 2001.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[30] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*. ACM, 2010, pp. 1–6.

[31] T. Hoyt, E. Arens, and H. Zhang, "Extending air temperature setpoints: Simulated energy savings and design considerations for new and retrofit buildings," *Building and Environment*, vol. 88, pp. 89–96, 2015.

[32] Y. Agarwal, R. Gupta, D. Komaki, and T. Weng, "Buildingdepot: an extensible and distributed architecture for building data storage, access and sharing," in *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*. ACM, 2012, pp. 64–71.

[33] H. B. Gunay, W. O'Brien, I. Beausoleil-Morrison, and S. Gilani, "Development and implementation of an adaptive lighting and blinds control algorithm," *Building and Environment*, vol. 113, pp. 185–199, 2017.

[34] T. Schulze and U. Eicker, "Controlled natural ventilation for energy efficient buildings," *Energy and Buildings*, vol. 56, pp. 221–232, 2013.

[35] T. Pasquay, "Natural ventilation in high-rise buildings with double facades, saving or waste of energy," *Energy and buildings*, vol. 36, no. 4, pp. 381–389, 2004.

[36] K. Bradley and B. Smyth, "Improving recommendation diversity," in *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland*. Citeseer, 2001, pp. 85–94.

Peter Wei is currently a Ph.D. candidate in Electrical Engineering at Columbia University, having received his B.S. and M.S. in Electrical and Computer Engineering from Carnegie Mellon University in 2015 and 2016. His current research involves the design of intelligent systems for energy optimization in commercial buildings. He is the recipient of the Presidential Fellowship at Columbia University, the Best Poster Award at ACM BuildSys '16, and the Best Paper Runner-Up Award at ACM BuildSys '17 and ACM BuildSys '19.

Stephen Xia received his B.S. in electrical engineering from Rice University, USA, in 2016, and his M.S. in electrical engineering from Columbia University in 2018. He is currently a Ph.D. candidate in the Department of Electrical Engineering at Columbia University, where his current research focuses on designing intelligent systems for mobile/wearable computing, connected health, and the Internet of Things. He is the recipient of the Runner-Up Best App Award and Best Presentation Award at IEEE VNC '18.

Runfeng Chen is currently pursuing a Masters in Computer Science at Yale University. He previously completed his B.S. in Computer Engineering at Georgia Tech in 2019.

Jingyi Qian graduated with an M.S. in Electrical and Computer Engineering from Columbia University in 2018. He is currently a software engineer at Google.

Chong Li is a Co-Founder of Nakamoto & Turing Labs, Inc. He is Chief Architect and Head of Research at Canonchain Network. He is also an adjunct professor at Columbia University. Prior to that, he has been working with Qualcomm Research. Dr. Li received his Ph.D. from Iowa State University and his B.E. from Harbin Institute of Technology, both in Electrical Engineering. He is a holder of 200+ U.S. patents (granted and pending). Dr. Li has been actively publishing papers in top-ranking journals, including Proceedings of the IEEE, IEEE Trans. on Information Theory, IEEE Communications Magazine, and Automatica. He has also served as reviewer and technical program committee for prestigious journals and conferences in communications and control societies. He is the author of the book "Reinforcement Learning for Cyber-physical Systems". Dr. Li has broad research interests including information theory, machine learning, networked control and communication, PYH/MAC system design for advance telecommunication technologies (5G and beyond). He is an IEEE senior member.

Xiaofan Jiang is an Assistant Professor of Electrical Engineering and Computer Engineering at Columbia University and co-Chair of Smart Cities Center at the Data Science Institute. Jiang received his B.Sc., M.Sc., and Ph.D. in Electrical Engineering and Computer Science from UC Berkeley, in 2004, 2007, and 2010, respectively. He was Director of Analytics and IoT Research at Intel Labs China prior to joining Columbia University in 2015. His research interest lies at the intersection of systems and data, with a focus on intelligent embedded systems and their applications in mobile and wearable computing, intelligent built environments, Internet of Things, and connected health. Jiang led one of the earliest projects on IP-based smart-buildings, bringing about the first IPv6/6LowPAN smart metering network and fine-grain real-time energy analytics. His ACme building energy platform was widely adopted by industry and academia. His city-scale air-quality project was featured on China Central Television and Peoples Daily and was successfully incubated into a startup. He has published in top-tier venues with over 3,500 total citations and was awarded Best Paper at IEEE/ACM IPSN '05, Best Demo at ACM SenSys '11, Best Poster at ACM BuildSys '16, Best Paper-Runner Up at ACM BuildSys '17 and ACM BuildSys '19, and Best Demo at ACM IoTDI 18. He has chaired BuildSys '14, IoT Expo '16, ICCCN HoT '17, and ACM SenSys '19.