Deep Learning Based Prediction Towards Designing A Smart Building Assistant System

Ankur Sarker[⋄], Fan Yao[⋄], Haiying Shen[⋄], Huiying Zhao[†], Haoran Zhu[⋄], Haroon Lone[⋄],
Laura Barnes[⋄], Brad Campbell[⋄], and Mitchel Rosen[⋄]
University of Virginia[⋄] and Beijing Jiaotong University[†]
{as4mz, fy4bc, hs6ms, hz3fr, hl7ck, lb3dp, bradjc, mcr4y}@virginia.edu; 15114197@bjtu.edu.cn

***Abstract*—Nowadays, smart building infrastructures are equipped with hundreds of sensors to monitor building environments and provide smart solutions for occupant comfortability and energy efficiency. Ideally, an automated system can predict and adjust the physical features (e.g., lighting, air quality, temperature, and so on) in a person's office based on his/her personalized preferences and activities. However, since the data is from one person, there may not be sufficient data for machine learning model training, and the data's quality may be low (e.g., with noises). Then, it is a challenge to conduct accurate predictions to provide personalized environment adjustment. To handle this problem, in this paper, we propose a smart building assistance system consisting of different sensor data analysis approaches and a deep neural network (DNN)-based prediction model to make a more accurate prediction despite low-quality sensor data. First, we collected a year-long smart building dataset from four different data sources (i.e., sensors, calendar, weather, and survey). Second, we perform different feature engineering approaches (i.e., concretization, one-hot encoding, and multiple feature combination) on the data as inputs for the prediction models. Third, we identify a support vector regression-based prediction model and propose a hybrid DNN model consisting of several recurrent neural network blocks and a feed-forward DNN block to predict different preferred physical features considering different activities of a person (e.g., meeting, lunch, research activities). Finally, we conduct experimental studies to evaluate the performance of the proposed prediction models compared to other existing machine learning models in terms of accuracy. Our predicted preferred physical features match the occupant's preferred ranges of different physical features during a specific activity. We also open-sourced our code on GitHub.**

***Index Terms*—Activity prediction model, Hybrid deep neural network, Smart building assistance, LSTM**

I. Introduction

Smart-buildings equipped with fully or partially automated systems to control the physical environments are becoming increasingly pervasive. These buildings can potentially provide environmental conditions that meet users' needs and improve the performance, comfort, health, and well-being of the occupants in particular. The physical working environments in the office buildings, such as illumination, air quality, temperature, and humidity, have long been recognized as the key features in the ambient work environments that affect human comfort and health conditions [1]. Optimizing indoor building environments can not only provide benefits to human health but also significantly reduce energy usage and create more environmentally-friendly buildings. Buildings account for about 30% to 40% of the total energy consumptions and carbon dioxide emissions worldwide [2]. For example, buildings use up to 40% of energy in the U.S.A. Reducing the energy consumption of buildings and making them more energy-efficient can significantly reduce the overall carbon dioxide emission, thus helping alleviate global warming. Therefore, occupant comfortability and energy efficiency are key factors that need to be taken into consideration when we design a smart building system that can adjust indoor-environment settings.

Usually, building occupants are not fully aware that they can adjust the physical features of the building. Even if they do, it would be onerous and demanding to adjust the environmental settings according to different factors. Thus, it is ideal that an automated system can adjust the indoor-environment features based on an occupant's preference and activities to provide personalized service. For example, when the occupant is meeting a person in the office, the light needs to be strong, while when (s)he is working on a computer, the light needs to be turned down, and when (s)he is taking a nap, the light needs to be turned off. Since the underlying reasons for the preferred environmental settings are complicated, such as personal preference, activities, time, and so on, machine learning techniques are a practical approach for predicting preferred settings. However, for personalized environment adjustment, since the data is from one person, there may not be sufficient data for model training, and the data's quality may be low (e.g., with noises). Then, it is a challenge to conduct accurate predictions to provide personalized environment adjustment. The goal of this paper is to propose such an automated system that can deal with this challenge.

A significant amount of research efforts have been devoted to smart buildings. One set of existing works [3], [4], focuses on deriving environment physical features based on the feedback from the occupants of a building to increase the usability of the building and the comfortability of the occupants' comfortability. Another set of exiting works [5]–[11] detects occupancy and automatically adjusts the heating, ventilation, and air conditioning (HVAC) systems based on the occupancy detection mechanism. In addition, several works [12]–[14] predict an individual occupant's coarse-grained activities, and some works automatically stop office appliances for energy-saving and maintaining room temperature. Unlike previous works, our work focuses on building an automated system that can accurately predict the indoor-environment features in an office preferred by the office's occupant so it can proactively adjust the indoor-environment features accordingly to provide personalized service.

In this paper, we propose a smart building assistance system that automatically controls an office's environment settings to increase its occupant's comfortability and energy-efficiency. Accurately, the proposed system predicts the physical features preferred by the occupant based on his/her previous activities and the corresponding physical features (s)he set. It can then proactively adjust the physical features so that the occupant's desire for comfortability and energy efficiency is satisfied. In this work, we first deployed sensors in a faculty member's office in smart building and collected data from May 2018 to August 2019. The person can adjust the temperature, humidity, lighting, and air quality based on his/her preference for comfortability, energy efficiency, etc. The dataset consists of different physical features (e.g., lighting, temperature, air quality) of the office environment, calendar events, everyday office activities, public holidays, and weather information. The physical feature settings are the occupant's preferred settings set by him/her. We then utilize the dataset for physical feature prediction to realize the person's proposed system for the office environment. To predict different physical features more accurately, we carefully choose different data analysis techniques and feature selection procedures to exploit the expressiveness of the collected dataset. More specifically, we perform feature discretization, feature combination, and regularization to enhance the expressive nature of the dataset. Then, we propose a support vector regression (SVR) based prediction model and a hybrid deep neural network (DNN) model that leverages a mixture of a feed-forward DNN block and several recurrent neural network (RNN) blocks to achieve higher prediction accuracy. Furthermore, we conduct experimental studies using the collected datasets, and our experimental results show that our proposed prediction methods for the occupant's preferred physical factors achieve higher accuracy. Our predicted preferred physical features also match the occupant's preferred ranges of different physical features during a specific activity. We also open-sourced our code on GitHub [1].

The following lists the major contributions of this paper:

(1) **Smart building data collection.** We have collected and used real data from a faculty office in a smart building such as lighting, cooling, temperature, humidity, office calendar events, occupancy, and historical weather data for our study. Also, we collected the occupant's preferences further verification of the predicted physical features.

(2) **Physical features predictions.** Based on our collected dataset, we used many different machine learning algorithms for prediction. Finally, we identify an SVR model and propose a mixture of feed-forward DNN and RNN models to predict different physical features. Due to low-quality data and insufficient data, the prediction accuracy is low. To improve prediction accuracy, we pre-process the data using different feature engineering approaches (i.e., concretization, one-hot encoding, and multiple feature combinations) on the prediction models' data.

(3) **Experimental studies.** We conducted experimental studies based on the building's indoor environment data and other calendar events and weather-related datasets. The experimental results show the superior performance of the SVR and the DNN models comparing to other machine learning models regarding prediction accuracy.

The rest of the paper is organized as follows. Section II presents the existing literature. Section III presents how we pre-process the dataset and use prediction models. Section V evaluates our proposed models through simulation studies. Finally, Section VI concludes this paper with remarks on future work.

II. Related Work

Researchers have been working for over a decade to make the indoor environments increasingly intelligent in different ways. We can divide the existing works into two different groups, as described below.

Optimizing the physical features of the smart buildings. The aim of improving occupants' satisfaction gave rise to early works in improving the physical features of buildings based on occupants' feedback [3], [4], [15]. Winkler *et al.* [3] developed a voting-based interface to adjust the HVAC system. Through a 40-week user study of 61 university employees across three buildings, the authors showed that the feedback system could increase user satisfaction with the improved thermal condition and reduce energy consumption caused by HVAC systems. Winkler *et al.* [4] presented several occupants' feedback collection methods to design a comfort voting application with the consideration of occupants-environment interaction. García *et al.* [16] proposed a context-aware collaborative learning framework for home management systems, which learns context-aware occupants' desired physical factors and tries to influence occupants to reduce energy usage. These occupants-feedback based methods focus on deriving physical factor settings to satisfy a group of people while we focus on the personalized physical factor prediction to satisfy one person.

Some occupancy-detection based methods have attracted much attention from researchers during recent years that automatically adjust the HVAC systems based on the occupancy detection mechanism. De Silva *et al.* [5] proposed to integrate a lighting system with occupancy detectors and daylight sensors for fully automated operations of adjusting the lighting system. Zou *et al.* [6] presented a wireless occupancy-driven lighting control system to reduce energy consumption while simultaneously preserving the lighting comfort of occupants. Huang *et al.* [7] integrated CO2 and light sensors with a wireless sensor platform and developed an occupancy detection method that can achieve higher accuracy while keeping low cost and non-intrusiveness. Ambient sensor systems [8] use collections of sensors such as temperature, light, door switches, or electricity consumption to infer the presence of people in a room. Das *et al.* [9] presented an occupancy counting solution of the building that only relies on information already available from existing modes of sensing in buildings, including electrical energy demand, water consumption and the number of wireless network-connected devices. Using a large-scale building dataset from a university campus, they showed that these three modalities are reliable indicators of

[1] GitHub link for the source code: https://github.com/as4mz/Smart-building

the accurate number of occupancy estimation using a machine learning-based algorithm such as clustering. Li *et al.* [10] proposed two methods: a new moving-window inhomogeneous Markov model based on change point analysis and an integrated hierarchical probabilistic sampling model to obtain flexible occupancy models of various kinds of temporal (e.g. intra-hour and hourly) and prediction horizons (e.g. hour-ahead and day-ahead). However, all the works discussed above do not consider the individual occupants' environmental preference based on his/her different indoor activities. MMa *et al.* [17] proposed a query selection strategy to combine the potential effect of a labeled activity type with improving activity prediction accuracy. Francis *et al.* [18] implemented a thermal comfort model for occupants that derives thermal comfort using the human body shape. Liu *et al.* [19] proposed a conflict identification system among different appliance services, which suggests a set of remedial actions to the user to resolve the conflict by leveraging the programming abstractions of the system.

Occupant activity prediction. Another set of works [12], [13], [13] predicts occupants' activities using different machine learning approaches (e.g., Bayesian Network (BN), Markov Model (MM), and others). Wilke *et al.* [14] proposed a bottom-up modeling approach together with a set of calibration methodologies to predict residential building occupants' time-dependent activities (e.g., relaxing, sleeping, studying, dancing/party, eating, gardening, and cooking). The authors considered dynamic building simulations where the inputs are the previous time-dependent activities, and output is the probability of the next activity. Nazerfard *et al.* [12] proposed an improved prediction model using BN for the occupants' household activity prediction. Unlike the traditional BN, the proposed model utilizes current features (e.g., activity location, activity time of day, activity day of the week) and next features to classify the immediate next activity (e.g., bathing, sleeping, eating). Wu *et al.* [13] proposed using real-time sensor data with BN for recognizing and predicting multiple human activities (e.g., bathing, working in leaving/bedroom, eating, cooking, watching TV, sleeping) in the building at the same time. The authors utilized the previous sequence of activities as the inputs of the BN network to predict the conditional probability of residents' current activities.

Zhao *et al.* [20] built a four-type office occupant behavior (i.e., occupied computer-based work, occupied non-computer-based work, unoccupied remote work, and unoccupied) prediction model using support vector machine, locally weighted learning methods where inputs are the electricity consumption data of computers, computer monitors, task lights, and other office appliances. Peng *et al.* [21] provided an in-depth analysis of occupants' stochastic behavior (e.g., entering/leaving the office room, working in the desk, and office appliance usages) within an office building. The authors proposed a demand-driven control strategy that automatically responds to occupants' energy-related behavior and automatically stops the office appliances to reduce energy consumption and maintain room temperature for occupants. Zhang *et al.* [22] proposed a convolutional neural network model to perform occupants'

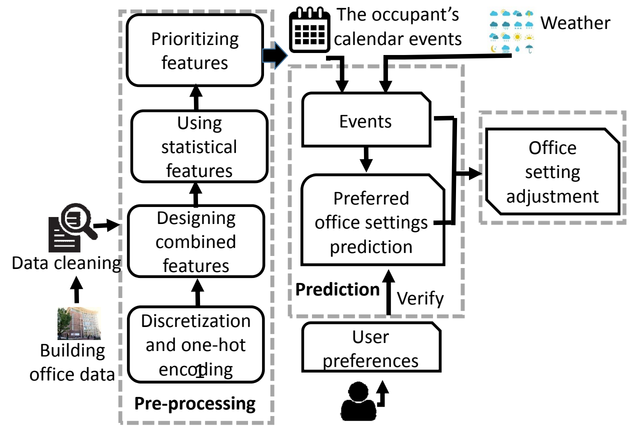


Fig. 1: Overview of the smart building assistant system.

activity recognition tasks inside edge devices.

Unlike these works, we focus on estimating a person's preferred physical factor settings based on his/her preference and activities. In this paper, we propose approaches to predict the physical features in an occupant's office more accurately despite insufficient or low-quality data.

III. Data Collection and System Design

In this section, we first describe the overall architecture of the proposed system (Section III-A). Then, we describe the data and its analysis we used in our system (Section III-B).

A. System Architecture Overview

The proposed smart building assistance system consists of three parts: data collecting and data pre-processing of the physical features, prediction module, and settings module. Fig. 1 shows an overview of the proposed smart building system. First, we collect smart-building sensors data such as lighting, cooling, temperature, humidity, and occupancy, which is time-series sensor data. We pre-process and clean the data in a 15-minutes window basis to combine with the occupants' events (e.g., meetings, classes, lunch hours) and calendar events (i.e., organizational activities). We also collect the user's preferences (e.g., preferred lighting intensity, indoor temperature, window shade, air quality) for the room's physical features regarding comfortability and efficiency-efficiency. We cleaned the data. Then, we employed a few feature engineering approaches for data pre-processing. After we clean and pre-process the data, we combine different input features, the occupant's calendar events (i.e., activities) and weather data to predict the occupant's preferred physical features at each time point. We also utilize the occupant's preferences to verify the predicted preferred physical features. Finally, our model recommends different physical features for the assistant system to adjust the office settings.

B. Data Sources

We have the following data sources from the smart building environment:

(i) *Physical features:* We collected the smart building physical features data, including temperature, humidity, lighting, air quality, door opening/closing, number of people. The physical

TABLE I: Different physical features.

Physical Factors	Term	Description
Temperature ˆ	Temperature	Indication of mean temperature in 15-min time period
Humidity ˆ	Humidity	Indication of mean humidity in 15-min time period
Lighting ˆ	Illumination	Luminous flux per squared meters area; it describes light intensity
Air quality ˆ	CO2	Carbon dioxide concentration
	Concentration	Air substance concentration (i.e., presence of carbon monoxide, lead, nitrogen dioxide, ozone, particulate matter of different size fractions, and sulfur dioxide in the air)
	VOC	Volatile organic compounds (i.e., organic chemicals from paints, varnishes, and wax are emitted as gases)
Door open	Contact	Indication of door opening and closing in 15-min time period
Existing people	PIR	Passive infrared sensor value (i.e., motion object detection); it indicates if there are persons in the room
Number of occupants	Ground truth from the person	How many people in the room

ˆ These are physical environment factors.

TABLE II: Adjustment ways for physical factors.

Physical Factors	Control Systems	Adjustment Ways
Temperature	Heating; Air Conditioning	Turn on/off; Turn up/down
Humidity	Humidifier; Air Conditioning	Turn on/off; Turn up/down
Lighting	Lamp	Turn up/down
Air quality	Window; Door; Air Conditioning	Close/open; Turn on/off; Turn up/down

feature data was collected from a faculty room in a three-semester long period from April 2018–May 2019. Table I shows the different physical features collected from the faculty room. The temperature, humidity, and lighting features are the most obvious ones that directly affect the occupant's comfortability. Corresponding sensors measure them every 15 minutes. The air quality feature has a direct impact on the occupants' physical healths. For example, the concentration of the air is associated with the conditions of the respiratory systems. The higher the concentration, the more unhealthy the air is. Other physical features are for smart building assisting purposes. For example, the number of occupants is negatively correlated to the desired indoor temperature, as the increase in the number of people leads to the rise in room temperature. The air quality is represented by CO2, concentration, and violative organic compounds (VOC). The terms have different meanings as presented in the third column of Table I. We used the contact sensor for monitoring door opening and the infrared sensor for detecting moving objects. The occupant indicates the event/activity-wise number of people inside the room. Later, it is confirmed by the readings from the motion sensor. Furthermore, Table II shows the ways to adjust different factors in the faculty room. For instance, occupants can adjust the lighting condition gradually from dark to fully bright.

(ii) *Calendar events:* We also collected the occupant's professional calendar event data to understand different activities throughout the time. We combined the data with different department activities and game day events to understand the patterns of different events the occupant needed to attend and the relationship between these frequent events and the physical features of the room. Table III shows the different activities we pulled from the calendar data. We can see that there are nine types of main activities from the table, and most of the events happen inside the office room.

TABLE III: Office activities.

Index	Activity
1	Calling
2	Office hour
3	Somebody visiting
4	Preparing for classes
5	Lunch time
6	Individual meeting
7.1	Weekly team meeting
7.2	Project meeting
7.3	Speaker
7.4	Business trip
7.5	Lunch or dinner with visitor
7.6	Faculty meeting
7.7	Teaching
7.8	Night time
8	Research activity
9	Not real event

*These activities are outside of office.

(iii) *Weather:* We also collected external weather information from publicly available historical weather information API [23], in order to take the impacts of external weather on the desired indoor environment into consideration. For example, when the outdoor temperature falls below a certain threshold, the indoor temperature should be adjusted to a higher degree. The weather data includes visibility (in miles), air pressure (in Hg), humidity (38%), dew point (in °F), temperature, wind values, and weather condition every six hours. The weather condition includes broken cloud, clear, fog, rain, thunderstorm, snow, overcast, mostly cloudy, sunny, freezing, and we use an index to represent it.

(iv) *Preference:* We interviewed the occupant of the office room monthly to get to know his/her personal preferences during different activities retrieved from the calendar event. Additionally, we asked about his/her preferred range of different physical features during a specific activity. This data is used to check if the predicted preferred physical features match his/her personal preferences from the interview.

C. Data Preprocessing

After data collection, we pre-processed our datasets and combined them. This process was accomplished in Python with the help of some machine learning packages such as pandas and sklearn. Table IV shows the range of physical features along with three groups. It also shows the units of different physical features. We first converted the raw data into a data frame with size 23808×22 where 23808 is the number of instances for each physical feature within a 15-min time window, and 22 is the number of features. The 22 features in the **input list** include Date, Time (e.g., 0:00-

TABLE IV: The range of physical factors.

Range	Low	Mid: Low	Mid	Mid: High	High
Temperature/°C	<18	[18,22)	[22,25)	[25, 30)	>30
Humidity/%	<28	[28,38)	[38,48)	[48,58)	>58
Lighting/lx	<200	[200,300)	[300,400)	[400,500)	>500
CO2/ppm	<500	[500,600)	[600,700)	[700,800)	>800
Concentration/ ppm	<650	[650,780)	[780,910)	[910,1040)	>1040
VOC/ ppb	<405.8	[405.8,532.6)	[532.6,659.4)	[659.4,913)	>913

0:15, 0:15-0:30), Time Slot (in 96 windows in a day), Day Status (holiday or not), Weekday, the Number of Occupants, Location (inside or outside of the office room), Activity, CO2, Concentration, Contact, Humidity, Illumination, PIR, Temperature, VOC, Weather Condition, Outdoor Highest Temperature (°F) in the Day, Outdoor Lowest Temperature (°F) in the Day, Outdoor Air Pressure (in Hg), Outdoor Wind (mph), and Outdoor Humidity (%).
Then, we considered the distribution of each feature separately to rule out abnormal observation values.

We use the above 22 features as inputs and the outputs include the physical features listed in Table I. From our initial observations, we find that different physical feature values exhibit periodicity. Therefore, we utilize different values of physical features over time as time-series features. First, we arrange the dataset in ascending order with respect to timeslots. Let $x^{(t)}$ be the feature vector obtained after one-hot encoding, feature combination and statistical extraction for the t-th sample (the length of $x^{(t)}$ is 305). Then, we fix a time step window k and concatenate every k adjacent $x^{(t)}, t = T - k + 1, \cdots, T$, to form the complete input for predicting $y^{(T)}$. k is one month time period in our experiments. Briefly, to predict preferred physical features at time slot T, we use data one month before time stamp T.

We use 70% data for training and 30% data for testing for both regression and classification machine learning techniques. The regression machine learning techniques we used include linear regression [24], Least Absolute Shrinkage and Selection Operator (LASSO) [25], Support Vector Regression (SVR) [26], Gradient Boosted Regression Trees (GBRT) [27]. The classification machine learning techniques we used include Softmax regression [28], support vector machine (SVM) [29], deep neural network (DNN) [30], long short-term memory (LSTM) [31]. For classification, we discretize the numerical values into five intervals as show in Table IV. However, the prediction accuracy is low; the MAPE is in the range of [0.25, 0.56] and the accuracy is in the range of [0.69,0.92]. Please see Table VII and Table IX in Section V for more details. In order to improve machine learning prediction accuracy, we employ feature engineering approaches. For feature engineering and selection, we performed the following tasks successively.

- (i) *Discretization and one-hot encoding:* We treat categorial features and numerical features differently. For each categorical feature (e.g., activity, PIR, time slot, weekday and weather condition), we use one-hot coding to generate multiple binary features to replace the original feature. One-hot encoding is a method to split a feature of n values to n features which takes 0 and 1 values. For example, if "weekday" is Monday, then the binary code is 10000. For each numerical feature, we segment its value into ten bins and generate its one-hot representation. We still keep the original numerical feature in case that some tree-based models (e.g., SVR, GBRT) may utilize the expressiveness of numerical features.
- (ii) *Designing combined features:* To enhance the data's expressive power, we design some useful combined features (e.g., Cartesian product of two categorical features) based on common knowledge. For example, the combination of outdoor humidity and outdoor temperature is a more reliable indicator for the indoor temperature setting than either of these two features separately. The combined feature can effectively capture the extreme conditions (e.g., when both the outdoor humidity level and outdoor temperature are incredibly high, the indoor temperature should be set to a low level). A sample vector x consists of 22 x_i features initially. Now, the combined features are added in the sample vector x as new feature columns. The combined features we created include:
  - 8 features (from Date to CO2 in the input list excluding Time Slot) × Time Slot
  - 2 features (from Time to Time Slot in the input list) × 7 activities (in the activity list)
  - 5 features (from Weekday to Activity in the input list) × all 16 activities (in the activity list)
  - 1 feature (CO2) × Weather Condition
  - 5 features (from Contact to Temperature in the input list) × all 16 activities
- (iii) *Using statistical features:* In addition, we extract some statistical values (i.e., minimum, maximum, median) to expand the input feature space. In particular, for each feature x_i in a sample vector x, we calculate the minimum, maximum, median, mean, and standard variation values for all the samples which share the same timeslot with x_i but in different days (e.g., in one month). These statistical values are added in x as the new feature columns.
- (iv) *Prioritizing features:* $\mathbb{L}_1$ regularization [32] is the process of normalizing the features in order to prevent overfitting using the L_1 norm. We apply the $\mathbb{L}_1$ regularization to prioritize different features because it exhibits better performance than other feature selection approaches (e.g., sequential forward/backward selection). However, we feed the whole feature set into deep learning models without explicitly selecting particular features since deep learning models can select features by themselves throughout the training process based on the importance of different features.

IV. Prediction Models for Physical Features

After the data pre-processing, the prediction accuracy is enhanced significantly. We will present the details in Section V. For the physical feature prediction, we tested different machine learning techniques as indicated above and identified support vector regression (SVR) to achieve higher accuracy. To improve the accuracy of classification models, We also propose a DNN model leveraging a mixture of LSTM blocks and a feed-forward DNN model where inputs are the time series data of different physical features. The detailed descriptions of the proposed hybrid DNN model are explained below.

A. Support Vector Regression

Support vector regression (SVR) is a variation of support vector machine (SVM) used as a regression method. It tries to maintain all the main features that characterize the algorithm by finding the maximal margin. SVR uses the same principles as SVM does for classification. A margin of tolerance (i.e., epsilon) is set in approximation to the SVM. The main purpose of SVR is the same as SVM, that is, to minimize error by individualizing the hyperplane which maximizes the margin considering that part of the error is tolerable. To train a standard SVR for data $(x^i, y^i)_{i=1}^N$, where x^i is the i^{th} sample of the input features and y^i is corresponding class label, we need to solve the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\omega\|^2, \\ \text{subject to} \quad & y_i - \omega x_i - b \leq \varepsilon \\ & \omega x_i + b - y_i \leq \varepsilon, \end{aligned}$$

where ε is a free parameter that serves as a threshold; all predictions have to be within an ε range of the true predictions, b is the offset of the hyperplan from the origin and ω is a normal vector to the hyperplan. Slack variables are usually added into the above problem to allow for errors and to allow approximation when the above problem is infeasible.

B. Proposed Hybrid DNN Model

Figure 2 shows the structure of our proposed hybrid deep learning model. The DNN in the figure is a feed-forward neural network. In the following, we introduce the feed-forward neural network and LSTM first, and then present the details of our proposed hybrid deep learning model.

1) Feed-forward neural network: A feed-forward neural network represents the function $f(x;\theta)$ where x is the input vector and θ is a set of parameters. Specifically, f is a composition of functions. The smallest unit of a neural network is a neuron. It maps the weighted sum $\sum_{i=1}^k x_i w_i$ to an activation value via the function $f_{act}(x^T w)$ where x is the vector of inputs for the neuron and w is a vector of parameters denoted as weights. A layer of the network is a set of neurons that usually use the same activation function. In this case, a layer i can be represented as function $f^{(i)}(x; W^{(i)}) = f_{act}(W^{(i)^T} x)$ where x is the input vector of the layer, $W \subset \mathcal{R}^{k\times l}$ is the matrix that contains the weights of l neurons (i.e., each column of W represents the weights of a neuron). Putting all together, such a feed-forward neural network represents a composition of the layer functions with the parameters $\theta = \{W^{(1)}, W^{(2)}, ..., W^{(n)}\}$. The first layer is called input layer and is only there to receive the input with the identity function $f^{(0)}(x) = x$. The last layer $f^{(n)}$ is called output layer. All other layers are called hidden layers.

2) Long short term memory network: Recurrent neural network (RNN) is a powerful model for time series data modeling. Specifically, given an input temporal sequence $x = (x_1, x_2, ..., x_T)$, where in our case x_i is the i-the input instance of a physical feature, the hidden states of a recurrent layer $h = (h_1, h_2, ..., h_T)$ and the output $y = (y_1, y_2, ..., y_T)$ of a RNN model can be obtained as:

$$h_t = \theta_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

$$y_t = \theta_y(W_{ho}h_t + b_o) \tag{2}$$

where W_{x_h}, W_{h_h}, and W_{h_o} are connection weight matrices for input layer, hidden layer, and output layer; b_h and b_o are bias values, and θ_h and θ_y are activation functions.

LSTM, a version of RNN, is with forget gates and peephole connections. The critical point of LSTM is the cell state c that capacitates RNNs to memorize by removing or adding information to it. This manipulation is mainly regulated by three modules, namely the input gate (i), forget gate (f), output gate (o). LSTM proceeds by the following functions:

$$i(t) = \sigma((W_{xi}x^{(t)} + W_{hi}h^{(t-1)} + W_{ci}c^{(t-1)} + b_i), \tag{3}$$

$$f(t) = \sigma(W_{xf}x^{(t)} + W_{hf}h^{(t-1)} + W_{cf}c^{(t-1)} + b_f), \tag{4}$$

$$c(t) = f^{(t)} \cdot c^{(t-1)} + i^{(t)} \cdot (W_{xc}x^{(t)} + W_{hc}h^{(t-1)} + b_c), \tag{5}$$

$$o(t) = \sigma(W_{xo}x^{(t)} + W_{ho}h^{(t-1)} + W_{co}c^{(t)} + b_o), \tag{6}$$

$$h(t) = \phi(c^{(t)}) \cdot o^{(t)}; \tag{7}$$

in which σ is an element-wise application of the logistic sigmoid function, ϕ is an element-wise application of the *tanh* function, $\cdot$ denotes element-wise multiplication, and h_t is the hidden state vector of the LSTM unit at time t.

3) Proposed hybrid DNN model: To utilize both the time-series nature and the structured form of input data, we propose a hybrid neural network that consists of several standard LSTM blocks and a feed-forward 3-layer neural network block on top of the LSTM blocks. It leverages the time-series data prediction power of LSTM and the high-level feature-selection power of the feed-forward neural network architecture. The architecture of the proposed model is shown in Figure 2. There are $k = 30$ LSTM blocks to utilize previous k time stamps of the inputs. The input of the LSTM blocks at different timestamp t is the concatenation of $X^{(t)}$ and $Y^{(t)}$ where $t = T-k, T-k+1, \cdots, T-1$. $X^{(t)}$ is the feature vector with 305 dimensions and $Y^{(t)}$ is the true 5-class one-hot label vector, both at time slot t. The 5 classes are defined in Table IV. The input of the feed-forward neural network is the concatenation of $X^{(T)}$ and $h(T)$, where $h(T)$ is the output of the LSTM block at current time T. The output of the hybrid DNN model is $\hat{Y}^{(T)}$, a 5-class softmax probability vector. The loss function is given by the cross-entropy between true label $Y^{(T)}$ and the network output $\hat{Y}^{(T)}(\theta, X)$ as given below:

$$Loss(\theta, X) = -\sum_{T=1}^{N}\sum_{i=1}^{5} Y_i^{(T)} \log(\hat{Y}_i^{(T)}(\theta, X^{(T)})) \tag{8}$$

where N is the number of training instances and θ represents the weights of the network.

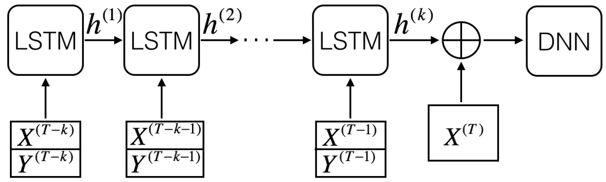


Fig. 2: The proposed hybrid deep learning model.

Hyperparameters. The model is implemented using *Keras* with input feature space dimension being 305, input time-series length k being 30, and the output length of LSTM block being 5, correct. Therefore, the number of neurons in each layer of the DNN is 305, 30, 10, and 5, respectively. In the proposed hybrid DNN model, we use the ReLU activation function. We also use Adam optimizer instead of other optimizers (e.g., Adamax, RMSprop, and Adagrad) to train our model where we set the learning rate as 10^{-3} in the optimizer. The number of epochs is set to 200. We use the cross-validation to test the trained model and reduce the overfitting of the trained model.

V. Performance Evaluation

In this section, we first present the experimental settings we used to evaluate the proposed and comparison approaches. Then, we present the experimental evaluations for the proposed approach and comparison approaches.

A. Experimental Settings

For the experiment, we predicted the six indoor physical factors including temperature, humidity, lighting, CO2, concentration, and VOC based on the sensor data. The followings summarize the experimental settings of different approaches.

1) Inputs: The inputs are the 22 features presented in Section III-C. After applying data pre-processing and feature engineering methods as described in Section III-B, we obtained an input sequence $X^{(t)}, t = 1, 2, \cdots, N$ where $N = 23808$. Each $X^{(t)}$ is a vector with dimension of 305 and contains all the information for a specific time slot.

2) Outputs: The outputs are six physical features including temperature, humidity, lighting, CO2, concentration, and VOC. For each factor i, we set a numerical objective $y^{(i)}$ using its original value and a categorical value $Y^{(i)}$ by discretizing its numerical into five intervals (see Table IV for more details).

3) Comparison methods: In the case of regression, we used four different methods: linear regression, LASSO, SVR, and GBRT. In the case of classification, we used five different methods: softmax regression, SVM, DNN, LSTM, and our proposed hybrid DNN model.

All the comparative methods were implemented using sklearn or Keras. The hyper parameters of the comparative models were all well-tuned to obtain the best possible performances for our prediction. The optimization solver was Adam with learning rate of 10^{-3}, and the termination criteria is 200-epoch iteration on the training set. Major hyperparameters are presented in Table V. The parameter settings for neuron network models are already introduced in Section IV. Note

TABLE V: The hyper parameters in model.

Model	API	Hyper-parameter Settings	Info.
Linear Regression	sklearn	Default	Standard LR without regularizations
LASSO	sklearn	$\alpha = 0.002$	α: Constant that multiplies the $\mathbb{L}_1$ term
SVR	sklearn	C=10	C: Penalty parameter of the error term
GBRT	sklearn	n=50	n: The number of boosting stages to perform
SVM	sklearn	C=20	C: Penalty parameter of the error term
Softmax Regression	Keras	Default	Implemented as 1-layer neural network
DNN	Keras	Default	3-layer neural network with size $30 \times 10 \times 5$
LSTM	Keras	Default	See IV-B3
Hybrid DNN	Keras	Default	See IV-B3

TABLE VI: The comfortable range for physical factors.

Factors	Notation	Lowest value $f_i^{(L)}$	Highest value $f_i^{(H)}$
Temperature/°C	f_1	15	25
Humidity/%	f_2	40	80
CO2/ppm	f_3	0	600
Concentration/ ppm	f_4	700	900
VOC/ ppb	f_5	300	1000
Illumination/lx	f_6	300	1000

that in Table V, "Default" means that we applied the default hyper-parameter setting implementation in these packages.

4) Evaluation metrics: For the comparative analysis, we mainly applied the following different evaluation metrics: mean absolute percentage error (MAPE), and accuracy.

1) We utilized MAPE to evaluate the regression models.
$$MAPE = \frac{1}{N}\Sigma_{i=1}^{N} \mid \frac{y_i - p_i}{y_i} \mid, \tag{9}$$
where N is the total number of predictions, y_i is the actual value, and p_i is the predicted value. We found that MAPE is a better metric compared with root mean square error (RMSE) because it alleviates the bias introduced by the magnitude of different objectives.
2) We used accuracy to evaluate the classification models. The accuracy represents the fraction of truly predicted physical factor level among all time slots as follows:
$$Accuracy = \frac{\sum_{i=1}^{N} \mathbb{I}(Y_i = P_i)}{N}, \tag{10}$$
where Y_i is the actual level and P_i is the predicted level. $\mathbb{I}(Y_i = P_i) = 1$ if and only if $Y_i = P_i$. Note that these two metrics are both calculated in 5-fold validation.

B. Experimental Evaluations

In this section, we describe the experimental evaluations in terms of prediction accuracy, respectively.

First, Table VII and Table IX show the prediction accuracy of different algorithms without using the data pre-processing introduced in Section III-C. Second, Table VIII and Table X

TABLE VII: The MAPE of predicted physical factors (without data pre-processing).

Objectives	Linear regression	LASSO	SVR	GBRT
Temperature/°C	0.35	0.38	0.31	0.38
Humidity/%	0.27	0.31	0.25	0.32
CO2/ppm	0.41	0.46	0.35	0.43
Concentration/ppm	0.41	0.38	0.28	0.45
VOC/ ppb	0.56	0.49	0.41	0.55
Illumination/lx	0.30	0.29	0.25	0.34

TABLE VIII: The MAPE of predicted physical factors (with data pre-processing).

Objectives	Linear regression	LASSO	SVR	GBRT
Temperature/°C	0.06	0.05	0.03	0.07
Humidity/%	0.13	0.13	0.07	0.18
CO2/ppm	0.06	0.05	0.04	0.06
Concentration/ppm	0.06	0.06	0.05	0.07
VOC/ ppb	0.22	0.21	0.17	0.26
Illumination/lx	0.08	0.08	0.10	0.11

show the prediction accuracy of different algorithms using the data pre-processing. Comparing Table VII and Table VIII, we see that the data pre-processing we introduced can improve the accuracy by 8.02%–272.10% times.

As we can see from Table VIII, the proposed approach SVR performs better than all other methods. The MAPEs for all the other five physical factors reaches a level under 10% except for VOC. For VOC, the prediction performances are relatively poor for all methods. It is due to the fact that the VOC factor is less correlated with the collected indicators.

Similarly, from Table X, we can see that our proposed hybrid DNN model achieves the best performance of classification. It is due to the fact that the proposed hybrid DNN model exploits both the temporal structure of the data by LSTM and the high-level feature-selection power of neural network architecture. With the proposed Hybrid DNN model, the accuracy for each physical factor is at least 85%. The improvements of the proposed Hybrid DNN model over LSTM is around 1%, which is still important to accurately predict different physical features and increase the comfortability and energy-efficiency. All of our predicted preferred physical features match the occupant's preferred ranges of different physical features during a certain activity.

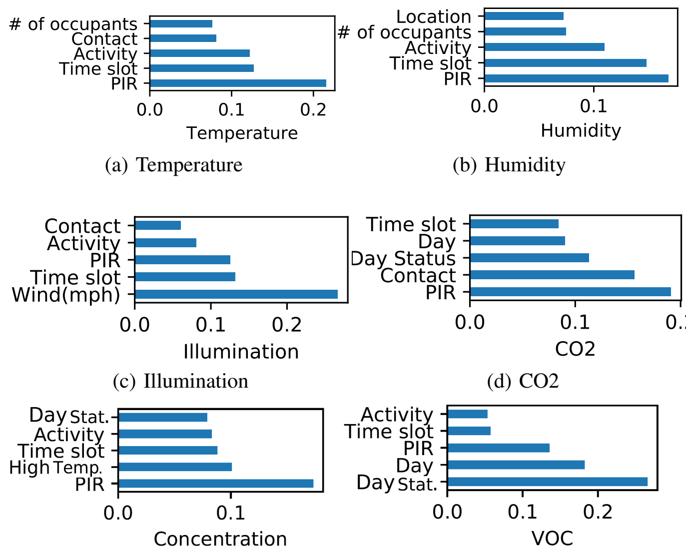
Figure 3 shows the top five most important features in predicting each physical feature. The importance of a feature is measured by calculating the decrease in the model's prediction error after discarding that feature from the input. A feature is important if changing its value decreases the model error. If so, the model significantly relies on that feature for prediction.

TABLE IX: The prediction accuracy of physical factors (without data preprocessing).

Objectives	Softmax regression	SVM	DNN	LSTM	Hybrid Model
Temperature/°C	0.70	0.72	0.75	0.78	0.79
Humidity/%	0.69	0.72	0.78	0.79	0.78
CO2/ppm	0.80	0.80	0.86	0.87	0.88
Concentration/ppm	0.76	0.77	0.80	0.79	0.80
VOC/ ppb	0.75	0.75	0.88	0.90	0.90
Illumination/lx	0.88	0.89	0.91	0.92	0.92

TABLE X: The prediction accuracy of physical factors (with data pre-processing).

Objectives	Softmax regression	SVM	DNN	LSTM	Hybrid Model
Temperature/°C	0.76	0.79	0.84	0.84	0.85
Humidity/%	0.76	0.78	0.82	0.84	0.85
CO2/ppm	0.89	0.88	0.91	0.92	0.92
Concentration/ppm	0.85	0.85	0.86	0.84	0.86
VOC/ ppb	0.80	0.82	0.92	0.97	0.97
Illumination/lx	0.97	0.97	0.98	0.98	0.99



(a) Temperature (b) Humidity

(c) Illumination (d) CO2

(e) Concentration (f) VOC

Fig. 3: Top five relevant features for each physical feature.

From the figure, we can see that for every different target feature, different features play key roles in the prediction process. This is primarily because each target physical feature depends on a different set of input features. Specifically, CO2 mainly depends on the number of people in the office instead of what people are doing in the office while temperature has a close relationship with the activity that people conduct in the office.

Despite the differences in these top five lists, they generally match the intuitive sense in daily life. For example, in the prediction of CO2 concentration in the office, PIR and contact have the largest feature importance. This matches the intuition as CO2 from humans exhalation is the primary source of CO2 in an indoor environment. Thus, the more people in the office (and the higher PIR consequently), the higher the CO2 concentration. Moreover, since the office door controls the air exchange between the office and outside environment, contact, an indication of the door opening and closing in a time period, has the second-highest feature importance. Day status, day and time slot also have major impacts on the number of people in the office. For instance, intuitively, there are more people in the office on weekdays than on weekends. These features indirectly influence the CO2 concentration in the office and thus have relatively high feature importance.

Table XI shows the prediction time of different machine learning models. We can see that the SVR model's prediction time is 1.6s, which is reasonable considering the number of different input features. We can also see that the prediction time of the hybrid DNN model is around 0.32s, which is

TABLE XI: The prediction time of different models.

Models	Time	Models	Time	Models	Time
Linear regression	2.5×10^{-4}s	GBRT	1.2×10^{-3}s	DNN	1.7×10^{-2}s
LASSO	1.5×10^{-3}s	Softmax regression	2.0×10^{-4}s	LSTM	3.3×10^{-1}s
SVR	1.6×10^{0}s	SVM	5.9×10^{0}s	Hybrid DNN	3.2×10^{-1}s

approximately similar to the LSTM model and slightly higher than the DNN model. Both SVR and hybrid DNN models can quickly output the prediction results for timely environment setting adjustment.

From the above figures and discussion, we can conclude that the proposed automatic smart building assistance is able to predict the activity-wise different physical features throughout the year to increase the comfortability of the building occupants.

VI. Conclusion

In this paper, we proposed a smart building assistance system to increase the comfortability of an office's occupant and save more energy at the same time. First, we collected and analyzed one-year-long data from a smart building office from different sources (i.e., sensors, calendar, weather, and survey). Second, we combined all the datasets and performed feature engineering (i.e., discretization, one-hot encoding, and multiple features combination) for further usage in machine learning models. Third, we proposed a hybrid DNN model composed of several LSTM blocks and a feed-forward DNN block to predict different physical features for different activities. Fourth, we compared the proposed hybrid DNN model with other existing models in terms of accuracy and MAPE. We expect future research to focus on the following tasks: (1) Study the interpretability of our automatic control system to make each adaptation of factors understandable to humans and provide valid reasons for the changes. (2) Integrate human knowledge or customized settings in the system. (3) Improve the robustness of our system. Improving the robustness of the system and avoiding malicious manipulations of indoor factors might be essential concerns in some safety-critical scenarios such as government agencies and military forces.

Acknowledgment

This research was supported in part by U.S. NSF grants NSF-1827674, CCF-1822965, and Microsoft Research Faculty Fellowship 8300751.

References

[1] J. G. Allen, P. MacNaughton, J. G. C. Laurent, S. S. Flanigan, E. S. Eitland, and J. D. Spengler, "Green buildings and health," *Current Environmental Health Reports*, vol. 2, no. 3, 2015.

[2] B. Dean, J. Dulac, K. Petrichenko, and P. Graham, "Towards zero-emission efficient and resilient buildings.: Global status report," 2016.

[3] D. A. Winkler, A. Beltran, N. P. Esfahani, P. P. Maglio, and A. E. Cerpa, "Forces: Feedback and control for occupants to refine comfort and energy savings," in *Proc. of UbiComp*, 2016.

[4] ——, "Forces: feedback and control for occupants to refine comfort and energy savings," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1188–1199.

[5] U. De Silva, A. Lertsinsrubtavee, A. Sathiaseelan, and K. Kanchanasut, "Named data networking based smart home lighting," in *Proc. of SIGCOMM*, 2016.

[6] H. Zou, Y. Zhou, H. Jiang, S.-C. Chien, L. Xie, and C. J. Spanos, "Winlight: A wifi-based occupancy-driven lighting control system for smart building," *Energy and Buildings*, vol. 158, pp. 924–938, 2018.

[7] Q. Huang and C. Mao, "Occupancy estimation in smart building using hybrid co2/light wireless sensor network," *Journal of Applied Sciences and Arts*, vol. 1, no. 2, p. 5, 2017.

[8] Z. Yang and B. Becerik-Gerber, "Cross-space building occupancy modeling by contextual information based learning," in *Proc. of BuildSys*, 2015.

[9] A. K. Das, P. H. Pathak, J. Jee, C.-N. Chuah, and P. Mohapatra, "Non-intrusive multi-modal estimation of building occupancy," in *Proc. of SenSys*, 2017.

[10] Z. Li and B. Dong, "Short term predictions of occupancy in commercial buildingsłperformance analysis for stochastic models and machine learning approaches," *Energy and Buildings*, vol. 158, pp. 268–281, 2018.

[11] H. Wang, J. Gong, Y. Zhuang, H. Shen, and J. Lach, "Healthedge: Task scheduling for edge computing with health emergency and human behavior consideration in smart homes," in *Proc. of Big Data*, 2017.

[12] E. Nazerfard and D. J. Cook, "Crafft: an activity prediction model based on bayesian networks," *JAIHC*, vol. 6, 2015.

[13] Z.-H. Wu, A. Liu, P.-C. Zhou, and Y. F. Su, "A bayesian network based method for activity prediction in a smart home system," in *Proc. of SMC*, 2016.

[14] U. Wilke, F. Haldi, J.-L. Scartezzini, and D. Robinson, "A bottom-up stochastic model to predict building occupants' time-dependent activities," *Building and Environment*, vol. 60, pp. 254–264, 2013.

[15] V. L. Erickson and A. E. Cerpa, "Thermovote: participatory sensing for efficient building hvac conditioning," in *Proc. of SenSys*, 2012.

[16] Ó. García, J. Prieto, R. Alonso, and J. Corchado, "A framework to improve energy efficient behaviour at home through activity and context monitoring," *Sensors*, vol. 17, no. 8, p. 1749, 2017.

[17] J. Ma, D. Hong, and H. Wang, "Selective sampling for sensor type classification in buildings," in *Proc. of IPSN*, 2020.

[18] J. Francis, M. Quintana, N. von Frankenberg, S. Munir, and M. Bergés, "Occutherm: Occupant thermal comfort inference using body shape information," in *Proc. of BuildSys*, 2019.

[19] R. Liu, Z. Wang, L. Garcia, and M. Srivastava, "Remediot: Remedial actions for internet-of-things conflicts," in *Proc. of BuildSys*, 2019.

[20] J. Zhao, R. Yun, B. Lasternas, H. Wang, K. P. Lam, A. Aziz, and V. Loftness, "Occupant behavior and schedule prediction based on office appliance energy consumption data mining," in *CISBAT 2013 Conference-Clean Technology for Smart Cities and Buildings*, 2013, pp. 549–554.

[21] Y. Peng, A. Rysanek, Z. Nagy, and A. Schlter, "Using machine learning techniques for occupancy-prediction-based cooling control in office buildings," *Applied Energy*, vol. 211, pp. 1343 – 1358, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261917317129

[22] S. Zhang, W. Li, Y. Wu, P. Watson, and A. Zomaya, "Enabling edge intelligence for activity recognition in smart homes," in *Proc. of MASS*, 2018.

[23] "Past weather in charlottesville, virginia, usa yesterday and last 2 weeks," https://www.timeanddate.com/weather/usa/charlottesville/historic?month=%27+month+%27&year=%27+yearraw_html.

[24] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*, 2012, vol. 821.

[25] C. Hans, "Bayesian lasso regression," *Biometrika*, vol. 96, no. 4, 2009.

[26] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. of ANIPS*, 1997.

[27] C. Persson, P. Bacher, T. Shiga, and H. Madsen, "Multi-site solar power forecasting using gradient boosted regression trees," *Solar Energy*, vol. 150, 2017.

[28] M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue, and R. Guan, "Text classification based on deep belief network and softmax regression," *Neural Computing and Applications*, vol. 29, no. 1, 2018.

[29] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, 1999.

[30] T. L. Fine, *Feedforward neural network methodology*, 2006.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, 1997.

[32] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *Proc. of ECML*, 2007.