Thermal-Sensor-Based Occupancy Detection for Smart Buildings Using Machine-Learning Methods

HENGYANG ZHAO, University of California, Riverside
QI HUA, HAI-BAO CHEN, and YAOYAO YE, Shanghai Jiao Tong University
HAI WANG, University of Electronic Science and Technology of China
SHELDON X.-D. TAN, University of California, Riverside
ESTEBAN TLELO-CUAUTLE, INAOE, Mexico

In this article, we propose a novel approach to detect the occupancy behavior of a building through the temperature and/or possible heat source information. The new method can be used for energy reduction and security monitoring for emerging smart buildings. Our work is based on a building simulation program, EnergyPlus, from the Department of Energy. EnergyPlus can model various time-series inputs to a building such as ambient temperature; heating, ventilation, and air-conditioning (HVAC) inputs; power consumption of electronic equipment; lighting; and number of occupants in a room, sampled each hour, and produce resulting temperature traces of zones (rooms). Two machine-learning-based approaches for detecting human occupancy of a smart building are applied herein, namely support vector regression (SVR) and recurrent neural network (RNN). Experimental results with SVR show that the four-feature model provides accurate detection rates, giving a 0.638 average error and 5.32% error rate, and the five-feature model delivers a 0.317 average error and 2.64% error rate. This indicates that SVR is a viable option for occupancy detection. In the RNN method, Elman's RNN can estimate occupancy information of each room of a building with high accuracy. It has local feedback in each layer and, for a five-zone building, it is very accurate for occupancy behavior estimation. The error level, in terms of number of people, can be as low as 0.0056 on average and 0.288 at maximum, considering ambient, room temperatures, and HVAC powers as detectable information. Without knowing HVAC powers, the estimation error can still be 0.044 on average, and only 0.71% estimated points have errors greater than 0.5. Our article further shows that both methods deliver similar accuracy in the occupancy detection. But the SVR model is more stable for adding or removing features of the system, while the RNN method can deliver more accuracy when the features used in the model do not change a lot.

Categories and Subject Descriptors: J.6 [**Computer-Aided Engineering**]: Computer-Aided Design

General Terms: Design, Algorithm

Additional Key Words and Phrases: Smart building, support vector regression, neural network, indoor temperature, occupancy detection

54

This work was supported in part by UC MEXUS-CONACYT Collaborative Research Grant CN16-161 and in part by the Nature Science Foundation of China (NSFC) under 61604095.
Authors' addresses: H. Zhao and S. X.-D. Tan, Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521, USA; emails: {hzhao, stan}@ece.ucr.edu: Q. Hua, H.-B. Chen (corresponding author), and Y. Ye, Department of Micro/Nano Electronics, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; emails: huaqihappy@gmail.com, {haibaochen, yeyaoyao}@sjtu.edu.cn; H. Wang, State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu, China; email: wanghai@uestc.edu.cn; E. Tlelo-Cuautle, National Institute of Astrophysics, Optics and Electronics, Puebla 72840, Mexico; email: etlelo@inaoep.mx.

https://doi.org/10.1145/3200904

ACM Reference format:
Hengyang Zhao, Qi Hua, Hai-Bao Chen, Yaoyao Ye, Hai Wang, Sheldon X.-D. Tan, and Esteban Tlelo-Cuautle. 2018. Thermal-Sensor-Based Occupancy Detection for Smart Buildings Using Machine-Learning Methods. *ACM Trans. Des. Autom. Electron. Syst.* 23, 4, Article 54 (June 2018), 21 pages.
https://doi.org/10.1145/3200904

1 INTRODUCTION

Building takes an instrumental role in energy consumption, and smartness of a building has a large impact on its inhabitants. According to statistics provided by the U.S. Department of Energy, more than 70% of electricity has been consumed by buildings every year (Department of Energy 2016). Recent efforts have been poured into the awareness of improving efficiency in quite a few facets, e.g., heating, ventilation, air conditioning (HVAC) systems (Erickson et al. 2009; Gao and Whitehouse 2009), lighting (Delaney et al. 2009), information technology, energy-consumption management within buildings (Agarwal et al. 2009, 2010), and so on. Among the overall energy usage of various aspects of buildings, the efficiency of HVAC systems has a tremendous impact on energy consumption (Hobby et al. 2012). On the contrary, some buildings utilizing schedule-based programmable thermostats may consume more energy than ones without using smart devices, when actual occupancy deviates from the programmed schedule (Bias and Cheng 1999). Automatic thermostat control systems have been developed in different approaches (Thomas et al. 2012; Lu 2012), and plenty of techniques are applied in the course of building the systems.

Detecting the occupancy (i.e., existence or number of residents) in a building or a room has applications ranging from energy reduction to security monitoring. For instance, occupancy detection is critical for energy and comfort management system in a smart building (Nguyen and Aiello 2013). Using the occupancy information, HVAC and lighting can be automatically controlled to reduce energy consumption while human comfort is maintained. Energy-consumption optimization can achieve higher quality with knowledge of real-time building occupancy (Majumdar et al. 2014). Previous research shows that energy can be saved by 28% by automatically sensing occupancy and turning off HVAC when a room or building is not occupied (Lu et al. 2010a).

Due to the importance of detecting building occupancy, many methods have been proposed in the past. The most widely used method is by means of motion detection (using different techniques such as infrared, RF, sounds, vibrations, and magnetism), which can detect whether there is a person or not. But motion detection in general cannot tell how many persons are in a room. Other methods include passive infrared sensors (Dodier et al. 2006), wireless camera sensor networks (Erickson et al. 2009), and applying sound level, case temperature, carbon-dioxide (CO_2), and motion to estimate occupancy number (Ekwevugbe et al. 2013). Preheat (Scott et al. 2011)-built rooms use active radio frequency identification (RFID) and sensors to detect home occupancy. Mozer (Mozer et al. 1996) proposed a neural network method by using the history data from embedded motion sensors and active RFID to explore occupancy rate. Thermostat (Lu et al. 2010b) also devoted a similar approach through the employment of magnetic reed switches and passive infrared sensors to take control of the HVAC system in residential buildings. However, those methods are more expensive for deployment as dedicated equipment is required. For the work (Dong and Lam 2014), it uses CO_2, motion, and acoustics to detect occupancy and achieves 92% accuracy. The work (Majumdar et al. 2014) uses CO_2 and motion sensors to build a probability model using history data of 3 months, which is used in energy-comfort optimization. A multiple sensor network method is proposed in Yang et al. (2016) for occupancy detection.

One viable and cost-efficient approach for the occupancy detection is to leverage existing temperature sensors or temperature sensor networks already deployed in many residual and commercial buildings. As a human being will lead to small disruptions or perturbations of temperature in a room, temperature sensor information can be analyzed to detect the occupancy and even the number of persons without additional cost. In this article, we propose a novel approach to detect occupancy based on the temperature sensor information under specific conditions by applying machine-learning methods, while it does not require many sensors to be installed in a certain building. We generate the mathematical models based on support vector regression (SVR) and RNN to detect occupancy with two sets of features for different application situations. Different from other methods, we build a machine-learning model for occupancy detection by carefully selecting some features, including solar angle, indoor temperature, outdoor temperature, working time, and light energy. Comparing to the work (Dong and Lam 2014), our RNN method can achieve higher than 99% accuracy using two or more hidden recurrent layers. Artificial neural network (ANN) is applied in occupation detection works (Lam et al. 2009) in a feed-forward manner, not capturing periodical patterns as using an RNN. The above works did not take historical data into account, which is important in patterned (e.g., weekly, seasonal) occupancy prediction. Other machine-learning techniques, such as support vector machines (SVM) and hidden Markov models, are also evaluated and 75% estimation accuracy was achieved.

We remark that SVR and RNN methods are widely used in non-linear regression applications because of their good interpolation performance. We started with SVR, as it is a widely used traditional machine-learning algorithm. Then we experimented with the RNN network for the same problem, as an RNN is able to efficiently capture frequency-domain characteristics in its recurrent neurons, modeling a non-linear dynamic system underlying, which is an advantage of being aware of daily, weekly, and seasonal occupation patterns. As a result, we are able to compare the traditional machine-learning algorithm with recent deep-learning techniques for our problem. The comparison indeed sheds some interesting light on this problem for the two methods as discussed later. It should be noted that other machine-learning methods such as logistic regression, multivariate adaptive regression splines, and long short-term memory can be used for occupancy detection for smart buildings as well.

The new approach is based on machine-learning approaches in which thermal-related features, including room temperature, ambient temperature, and other related heat sources, are used for detecting the number of people in a room. The experiments are conducted by using a building simulation program, EnergyPlus (Crawley et al. 2001) from the Department of Energy, which can model the various time-series inputs to a building such as ambient temperature, HVAC inputs, power consumption of electronic equipment, lighting, and number of occupants in a room, sampled each hour, and produce resulting temperature traces of zones (rooms). Experimental results with the SVR method show that the four-feature model provides accurate detection rates, giving a 0.638 average error and 5.32% error rate, and the five-feature model gives a 0.317 average error and 2.64% error rate. This indicates that SVR is a viable option for occupancy detection. In the RNN method, we apply Elman's recurrent neural network (ELNN), which has local feedback in each hidden layer. We use a simple formula to calculate layer numbers and sizes to configure RNN to avoid overfitting and underfitting problems. The error level, in terms of number of people, can be as low as 0.0056 on average and 0.288 at maximum considering ambient room temperatures and HVAC powers as detectable information. Without knowing HVAC powers, the estimation error can still be 0.044 on average, and only 0.71% estimated points have errors greater than 0.5. Our article further shows both methods can deliver similar accuracy in the occupancy detection. But the SVR model is more stable for adding or removing features of the system, while the RNN method can deliver better accuracy when the features used in the model do not change a lot. Compared to first-principles methods, SVR and RNN methods work in a black-box way, which reduces the efforts of explicitly building models of occupancy detection and deployment of sensor network, while good detection accuracy can still be achieved, as shown in the results. Our major contributions are listed as follows:

— We use thermal system consistency to detect occupancy using other thermal-related parameters acquired by existing sensors, which dramatically reduces the deployment effort, such as elimination of the requirement of infrared sensors.
— We apply machine-learning techniques, such as deep RNNs, to explore the possibility and accuracy of occupancy detection. This does not require explicit application-specific or building-specific modeling.
— Two machine-learning methods are thoroughly compared in terms of accuracy, using different feature sets, which highlights the importance of certain significant features, such as HVAC power consumption.
— We use dynamic machine-learning models (e.g., RNN instead of non-recurrent alternatives) such that periodical occupancy patterns can be taken into account.

The rest of this article is organized as follows. Section 2 reviews the EnergyPlus program used for generating realistic building data. Section 3 reviews the two machine-learning methods, SVR and ELNN, used in this article. Section 4 introduces the two methods for the given problems of occupancy detections based on the thermal sensor information. Then the experimental results based on the two methods, discussions, and comparison between the two methods are presented in Section 5. Section 6 concludes this article.

2 ENERGYPLUS-BASED SIMULATION FOR SMART BUILDINGS

In this section, we review the EnergyPlus software program, which provides accurate input and output traces from buildings for validation of the new thermal modeling algorithms.

The EnergyPlus software package is a suite of algorithms that calculate the energy required to operate a building and its resulting thermal behavior based on numerous considerations ranging from the specifications of the structure, to heat sources and sinks within the building. EnergyPlus consists of an integrated solution manager that manages the calculation of the heat balance of various surfaces in a simulated building, its mechanical systems, and the air. The solutions to each of these three elements are calculated separately, and loosely coupled to each other using the manager at each timestep. Due to its modularity, it is easy to establish links to other programs such as Google SketchUp for 3D modeling and visualization. EnergyPlus has been proven as a successful energy simulation program with widely accepted accuracy for modern buildings (Yang et al. 2016). Among all its applications, for instance, it has been used as a kernel of high-level building energy and control systems test bed (Wetter 2011), and a key validation tool of an energy saving model (Mardaljevic et al. 2009). Instead of conducting real building experiments, using EnergyPlus to test and simulate the proposed models is therefore valid, with a stateof-the-art simulation accuracy, which was a major pain point in the past. On the other hand, using EnergyPlus allows the authors to efficiently choose from a variety of buildings to test on, and to easily configure energy inputs.

An input data file (IDF) and a weather file are needed for the EnergyPlus simulation. The IDF includes all the information of the building such as size, structure, position, the HVAC subsystem, and so on. The IDF editor in EnergyPlus can be used to change parameters of the building, the schedule of the HVAC subsystem, and also the output information. The selected output information is generated in the spreadsheet file after running the simulation.

Figure 1 shows the side view of an office building with five rooms and HVAC modeled in EnergyPlus. The heat sources for this building can be HVAC, light, occupants, electric equipment, air filtration, and so on. The room temperature is also affected by the weather (ambient temperature and solar factors) and can be controlled by the HVAC system with coils and fans.

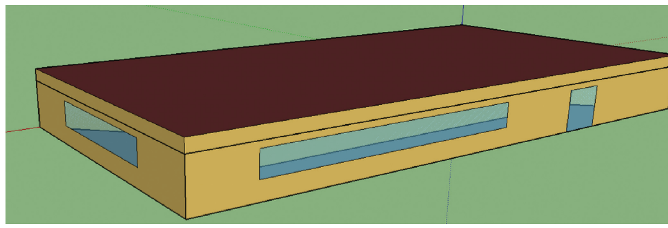


Fig. 1. Side view of a five-zone office building.

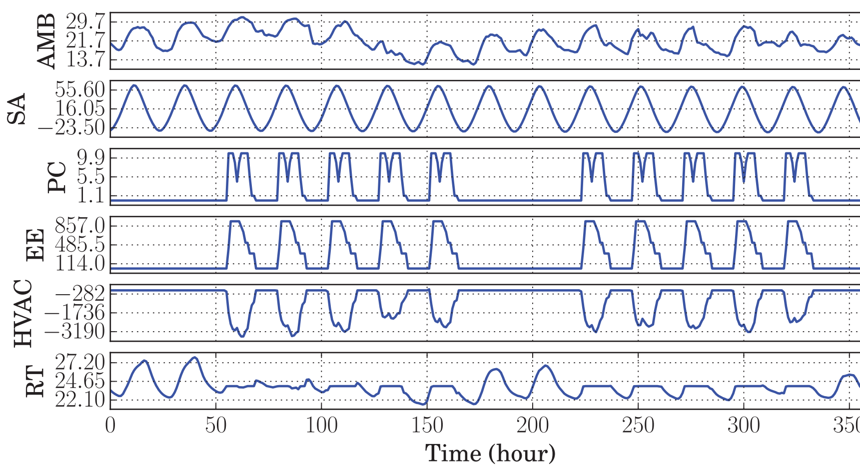


Fig. 2. Selected EnergyPlus input and simulated temperature output data sample in 15 days. (AMB, ambient temperature; SA, solar angle; PC, people count (occupancy); EE, electrical equipment power; HVAC, HVAC system cooling/heating power; RT, room temperature).

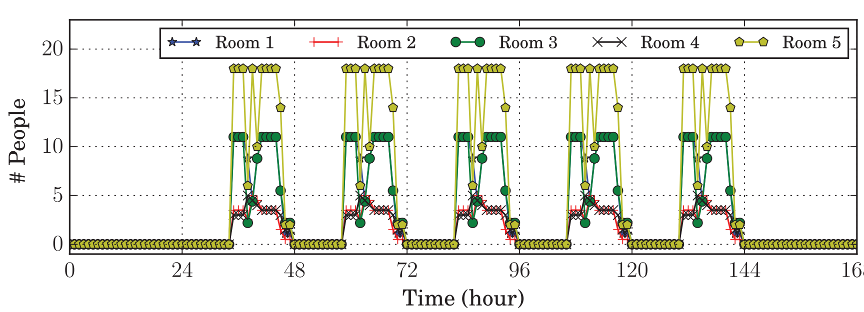


Fig. 3. Occupancy information of five rooms during 1 week.

Figure 2 shows the simulated temperature changes and input changes over 15 days from EnergyPlus for an office building with the five zones (rooms), as shown in Figure 1. EnergyPlus can assign different schedules for each room while simulating the thermal model. Figure 3 shows a typical working schedule of the five-room office building.

We want to stress that, fundamentally, thermal behavior of building systems is typically nonlinear (at least weakly nonlinear) due to the temperature-dependent properties of the building materials and thermal radiation effects. As a result, nonlinear modeling is preferred for accurate temperature control and management.

3 REVIEW OF MACHINE-LEARNING METHODS

This section briefly introduces some basic concepts of machine-learning methods: SVR and RNN. Some specific tweaks in applying those methods in the model are also illustrated herein.

3.1 Review of Support Vector Regression

The elemental idea of the regression is to seek out a function that can accurately detect future values and the generic SVR estimating function is formed as

$$f(x) = w \cdot \Gamma(x) + \lambda. \tag{1}$$

In the equation above, $w \in \mathbb{R}^n$ is a hyperplane direction, $\lambda \in R$, $x \in R^n$ is derived from the feature space, and Γ stands for a nonlinear transformation from R^n to a high dimensional space. The transformation grants the power for a feature to be transferred into more complex dimension. Our objective is to find a value of w and λ such that the value of x can be resolved via minimization of the regression risk:

$$R_{reg}(f) = C \sum_{i=0}^{l} G_i + \frac{1}{2} \|w\|^2, \tag{2}$$

where C is a constant, l denotes the size of the training data, and G_i is a loss function:

$$G_i = \begin{cases} |f(x_i) - y_i| - \varepsilon, & |f(x_i) - y_i| \geq \varepsilon \\ 0, & otherwise \end{cases}. \tag{3}$$

Here, $x_i \in \mathbb{R}^n$ is the given training data and y_i is the corresponding target value. Through mathematical deduction (Wu et al. 2004), the ε-insensitive loss function can be minimized as

$$\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i^* - \alpha_i) \left(\alpha_j^* - \alpha_j\right) k(x_i, x_j) - \sum_{i=1}^{l} \alpha_i^* (y_i - \varepsilon) - \alpha_i (y_i + \varepsilon), \tag{4}$$

subject to $\sum_{i=1}^{l} (\alpha - \alpha_i^*) = 0,\ (\alpha_i - \alpha_i^*) \in [0, C]$, and $k(x_i, x)$ is a kernel function. α_i and α_i^* are Lagrange multipliers, which denote solutions to the quadratic problem. The constant C decides penalties to estimation errors: When C becomes larger, the penalties to errors become higher, thus the regression is trained to reduce the error with lower generalization. On the contrast, a small C assigns lower penalties to errors, which results in a higher generalization model. If C becomes infinitely large, SVR would not bear any errors and generates a complex model, whereas the model would tolerate a huge number of errors if C is set to zero. The value of C can be calculated by the grid search algorithm (Hsu et al. 2003), which is performed over confidence penalty weight values to find a confidence penalty weight to work best. The value of w in accordance with the Lagrange multipliers is already acquired before we find the value of variable λ. Using KKT conditions λ, it can be calculated as follows:

$$\begin{aligned} \lambda &= y_i - (w, x_i) - \varepsilon \quad for\ \alpha_i \in (0, C), \\ \lambda &= y_i - (w, x_i) + \varepsilon \quad for\ \alpha_i^* \in (0, C). \end{aligned} \tag{5}$$

Putting it together enables us to apply SVR without knowing the concrete transformation. By adjusting parameters in the SVR model, it is capable of accurately conducting detection on office occupancy.

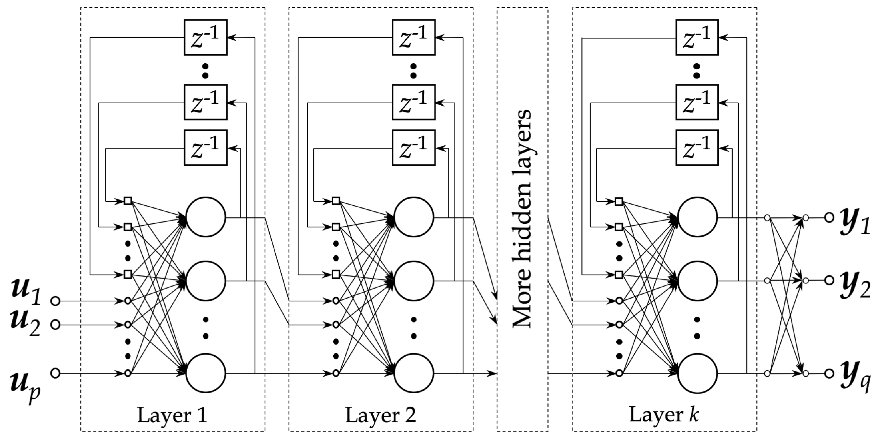


Fig. 4. Architecture of ELNN.

3.2 Review of Recurrent Neural Network

Learning-based techniques, such as neural networks composed of multiple processing layers, can learn representations of data with multiple levels of abstraction. Deep-learning techniques with many layers recently have dramatically improved the state of the art in speech recognition and image recognition (Schmidhuber 2014).

An RNN is constructed by introducing internal status holders to a memoryless network so that it can deal with time-series data. The internal status holders store outputs of designated neurons and usually function as feedback into other neurons. The application of feedback enables RNNs to acquire time-dependent state representations, making them suitable devises for applications like time-dependent non-linear prediction, plant control, and so on (Haykin 2007). There are many RNN structures proposed by varying the form of the recurrent feedback (Elman 1990; Haykin 2007; Puskorius et al. 1996).

The ELNN shown in Figure 4 is used to detect the occupancy behavior in a building. We describe the structure Elman architecture, how the gold-referencing data is computed, and the detailed works on training the networks. We construct ELNN architecture (as shown in Figure 4) to build the black-box model for occupancy detection. In our article, the size (number of neurons) of hidden layers are assigned according to empirical equation $N_{1,\ldots,k-1} = \frac{1}{5}p + 5$ and $N_k = 2q$, where N_i is the size of ith layer, p and q are, respectively, the number of network inputs and outputs. We will focus on applying the Elman recurrent network architecture (ELNN) (Elman 1990), which applies local recurrent feedback on each layer of neurons, and shows good performance for time-series-based learning such as voice recognition.

In the theoretical aspect, training a neural network is equivalent to the optimization problem to minimize cost function. Therefore, the neural network training problem can be solved by applying existing optimization methods such as gradient decent, Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Heath 2002), and the Quasi-Newton method on the cost function. In practice, algorithms with lower computational costs have been developed. Back-propagation algorithm is a widely used algorithm and has been well studied (Hecht-Nielsen 1988). It collects errors in weighting matrices in a backward propagation, after the errors of output vectors have been observed in each epoch. Based on the back-propagation algorithm, many improvements have been developed such as the resilient back-propagation (RProp) method (Riedmiller and Braun 1993), which is a more adaptive approach, and a further improvement method: RPropMinus (Igel and

Fig. 5. The office model: top view.

Hsken 2003), which has an overall better performance in reducing average error in late training phase. The back-propagation algorithm family has also been extended to train RNNs. Back-propagation through time (BPTT) (Werbos 1990) unfolds every network activation of a continuous sequence. Back-propagation through structure delivers more computational efficiency on arbitrary structured networks.

4 PROPOSED OCCUPANCY ESTIMATION APPROACHES

In this section, we apply the SVR and RNN methods on occupancy estimation in a smart building that contains five zones. The whole smart building is simulated by using the energy simulation tool EnergyPlus. We will first discuss the principles based on the features used for detection and then conduct the data configuration used in the model for occupancy detection.

4.1 Feature Selection

In the machine-learning model we built for occupancy detection, we carefully selected five features, each of which possesses some unique information hidden inside. The features are solar angle, indoor temperatures, outdoor temperatures, working time, and light energy. Solar angle is periodic and repeats every year. Outdoor temperature is an important thermal factor that impacts the indoor temperature. Working time denotes whether a regular working schedule is executed, and light energy gives out a radiation metric that causes rise of the temperature.

Figures 5 and 6 show the perspective view and the top view of the building, which contains five rooms and an HVAC system, by using the software EnergyPlus. This building is influenced by heat sources produced from occupants, electric equipment, air filtration, and so on. The weather (ambient temperature and solar effects) affects the room temperature as well. Through the HVAC system with coil and fan, room temperature can be administered properly to ensure that a comfortable temperature in the environment can be produced in the room.

In this room, the solar angle θ_s is defined as the angle between the zenith and the centre of the sun disc:

$$\cos\theta_s = \sin\phi\sin\delta + \cos\phi\cos\delta\cos h, \tag{6}$$

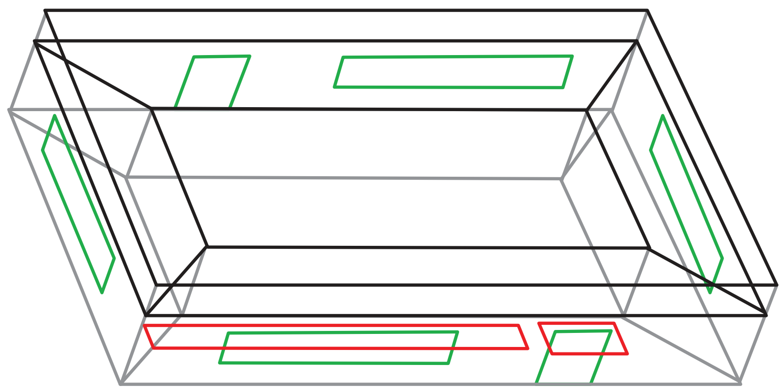


Fig. 6. The office model: perspective view.

where h is the hour angle in the local solar time, δ is the current declination of the sum, and ϕ is the local latitude. This equation enables us to compute the feature solar angle and all the variables are correlated to the location of the building.

Working time is a feature that determines if the time detected is local working time or not, and it apparently affects the number of working people in a certain office. It is also convenient to achieve the schedule of an ordinary worker in an office within the building and it is a good feature contributing to the detection. When a feature is being considered to be incorporated in the feature pool, we first figure out the convenience and difficulty in acquiring the dataset. Here the working time has a strong correlation with the employee common schedule, which is a relative dataset to acquire. Therefore, the working time is chosen as a feature element in the feature pool, and it is even a basic feature owing to its convenience.

Outdoor temperatures can be easily collected from a public weather forecast. We use the outdoor temperature dataset of the location of the building simulated by EnergyPlus, so our simulated building model goes through the exact same weather conditioning the genuine building has gone through. The outdoor temperature also plays an instrumental feature role in detection, because it directly influences the indoor temperature, which has a non-linear relationship with the number of employees in a certain office.

The indoor temperature is one of the most important features used in the detection model. It is the result of the interaction of all thermal inputs and the necessary part of determining thermal consistency. Indoor temperature is simulated by EnergyPlus as a thermal system output, but is used as a key input in the occupancy detection. The proposed occupancy model essentially calculates the occupancy that meets the indoor temperature, which makes the thermal system consistent.

Light energy is the fifth feature in the feature pool, and it is different from the first four features. The first four features are quite convenient features to acquire, whereas the measurement of light energy is comparatively inconvenient to acquire. However, we want to make the detection more versatile and to be applied in different situations. Despite the inconvenience of the dataset of light energy, it is an important metric related to the number of employees in a specific office as well. Aiming at providing a more accurate detection, the light energy feature is incorporated into the feature pool.

As listed above, five main features are used in the machine-learning methods for occupancy detection. These features cover all major thermal-related aspects to be coupled with the human activity, such that occupancy can hardly be determined theoretically. To investigate the impact

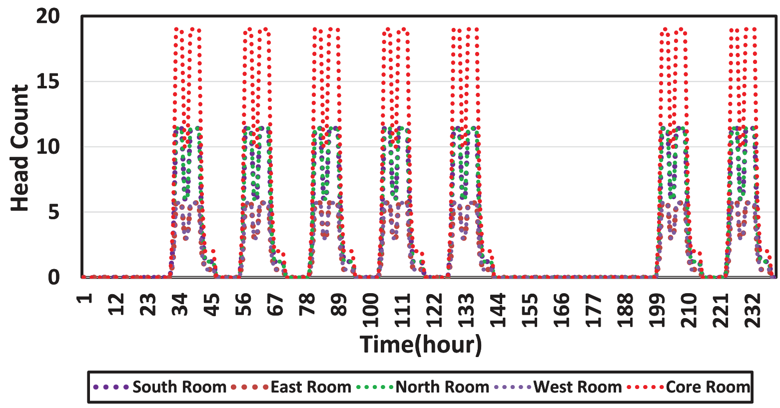


Fig. 7. Occupation information of five rooms for 10 days.

of choosing different sets of features and to improve the flexibility of practical deployment, two different feature sets are used in the methods.

4.2 Data Configuration

Figure 9 shows how EnergyPlus predicts occupancy in a room. The machine-learning model takes indoor temperature as one of its inputs. This requires us to use EnergyPlus to calculate the indoor temperature using the thermal inputs, including people occupancy. Then we take selected features and indoor temperature as training inputs, and real occupancy as the training output, to perform model training.

In general, the machine-learning model takes the selected parameters as its features to train the model and yield results. It is highlighted that the relationship between occupancy and other factors are non-linear related, where SVR and RNN are good choices to solve non-linear problems used for detection. Machine-learning methods are increasingly used in various supervised and unsupervised problems in many applications.

Figure 7 shows a vintage working schedule of the five zones of an office building. Our goal of the detection model is to detect an accurate number of employees in a room by certain parameters and data collections. Figure 8 shows the simulated temperature curves and input curves over 20 days from EnergyPlus for a smart building model with the five separate zones shown in Figure 6. The schedules for each room can be assigned differently by EnergyPlus. The selected features are then combined together to occupancy behavior in the smart building. In general, the SVR model takes the selected features as its parameters to train the data and yield the corresponding results.

In the proposed machine-learning model, two feature sets are provided for occupancy detection in a building. The first set is comprised of features such as solar factor, working time, indoor temperature, and outdoor temperature, which can be acquired through sole mathematic computation and EnergyPlus simulation. It is relatively convenient to obtain all the features required in the first feature set. However, we introduce one more feature in the second feature set, light factor, to enhance the accuracy of the model. Light factor requires the model to learn overall energy that light consumes during a certain quantity of time, which is a comparatively inconvenient feature to obtain, however, it is capable of making the detection more accurate. It is also highlighted here that all of the data we feed in the model is generated from EnergyPlus or obtained through mathematical calculation, thus further experiments are likely to be conducted in real-life data conditions.

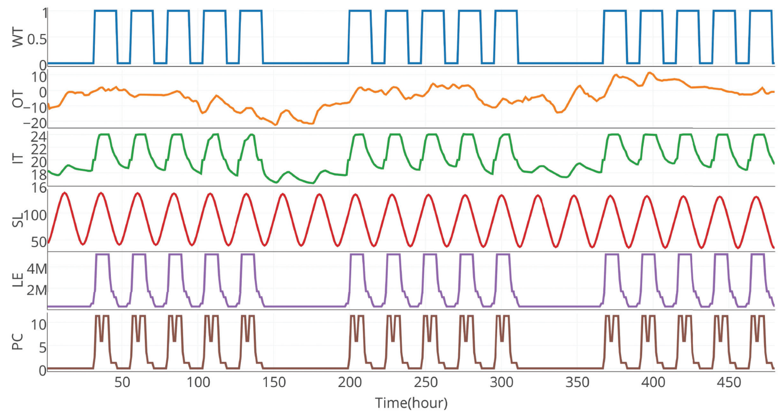


Fig. 8. Selected EnergyPlus input and simulated temperature output data sample in 20 days. (WT, work time; OT, outdoor temperature; IT, indoor temperature; SL, solar angle; LE, light energy; PC, people count.)

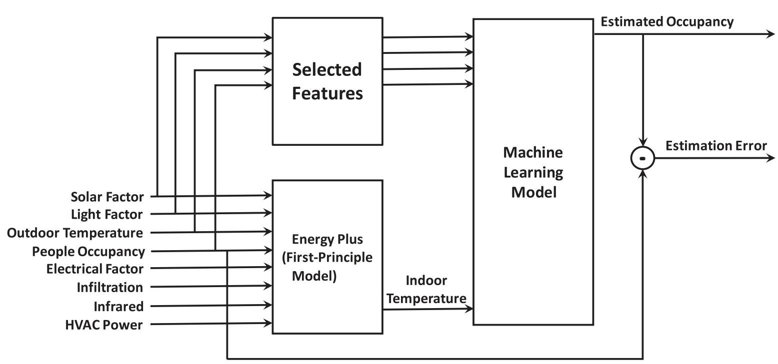


Fig. 9. Data configuration of the machine-learning model.

For separating the whole dataset, we split the 1-year simulation data into 12 months and three time periods, in which the months 1–3, 5–7, and 9–11 are referred to as the training data and the months 4, 8, and 12 are specified for the testing data in the proposed machine-learning model.

5 EXPERIMENTS AND DISCUSSIONS

5.1 SVR-Based Occupancy Detection

In this section, we illustrate how we measure error variation for the proposed SVR model and discuss the effectiveness of applying a different number of features in this model.

5.1.1 Experiments. We evaluate the SVR detection model using a testing dataset. Through a wide range of experiments, we learn that the radial basis function kernel, also known as Gaussian kernel, works best in this model. The two most important parameters in the SVR model are the penalty C and the radius ε. For the confidence penalty, the grid search algorithm (Hsu et al. 2003) is performed for finding a confidence penalty weight to work best. It is widely known that to get an accurate performance in an SVR model or any other machine-learning methods, the best approach is to enumerate a quantity of combinations of parameters, and conduct experiments to drive the result toward a better trend. During the process of seeking out the best result for the model, the set of parameters is adjusted step by step to obtain a model that has a better accuracy than the previous

Table 1. Training Errors of SVR Model Using Different Numbers of Features

	$C = 100$		$C = 500$		$C = 1000$	
#Features	4	5	4	5	4	5
Avg. error	0.721	0.310	0.576	0.326	0.550	0.348
Error rate	31.1%	8.56%	18.8%	8.89%	18.1%	12.4%

Table 2. Validation Errors of SVR Model Using Different Numbers of Features

	$C = 100$		$C = 500$		$C = 1000$	
#Features	4	5	4	5	4	5
Avg. error	0.819	0.317	0.694	0.330	0.638	0.390
Error rate	6.80%	2.64%	5.78%	2.74%	5.32%	3.25%

one. After a batch of experiments for the model are conducted, we pick out the parameters in the model that bring about the best performance. Here we also want to highlight that, most of the time, the best choice of the parameter set is usually difficult to be theoretically proved; therefore, confirmed parameters often are determined by a great number of trials and experiments.

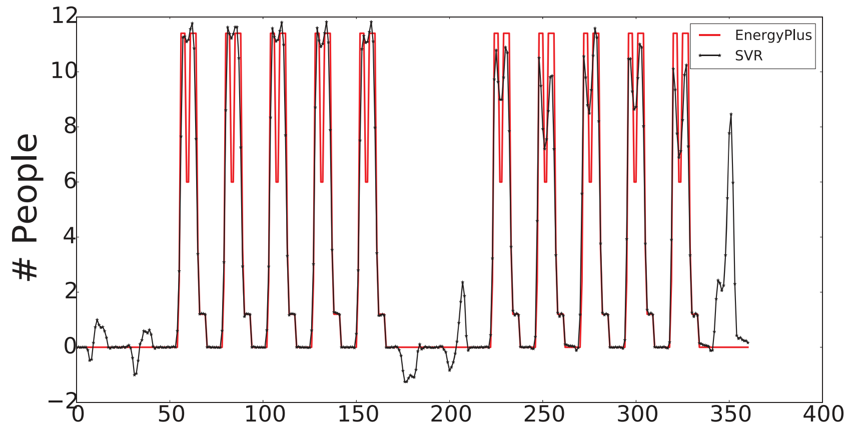
Table 1 shows the training error statistics of the proposed SVR model used for occupancy detection. Some comparisons between two sets of features are apparently displayed from the results shown in this table. Numerical simulation shows that ε being equivalent to 0.01 is a reliable choice for this SVR-based occupancy model. At each sample point, the estimation error e_i is defined as $e_i = |O_i^{\mathrm{SVR}} - O_i^{\mathrm{EP}}|$, where O_i^{SVR} denotes the occupancy value obtained by the proposed SVR model and O_i^{EP} denotes the real value of occupancy generated from EnergyPlus. We calculate the average error and the error rate by $\frac{1}{n}\sum_n e_i$ and $\frac{\text{average error}}{\text{full occupancy}}$, respectively. Also, in this table different values of C are tested to seek out an accurate model for occupancy detection.

Table 2 shows the validation error statistics of the proposed SVR model. The reason why the five-feature model performs better than the four-feature model for occupancy detection is that the four-feature model suffers slightly in the under-fitting issue, which results in a high bias. It is important to determine the parameters that can maintain a balance between under-fitting and over-fitting.

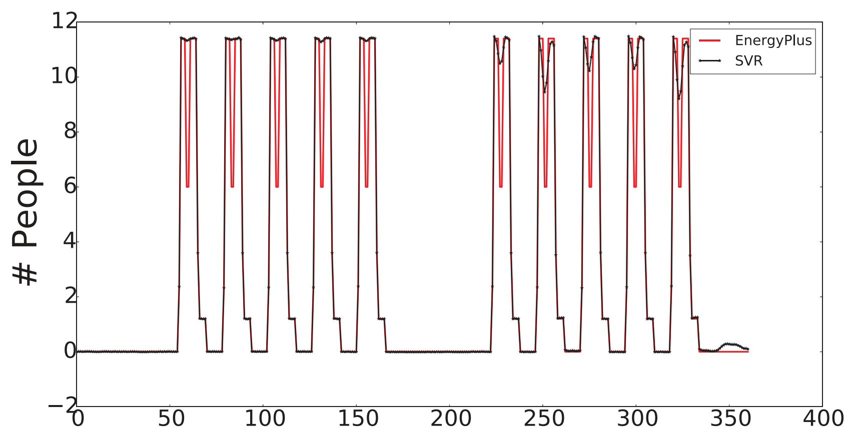
5.1.2 Analysis. In this part of the section, the process of achieving the best model we build is revealed, and several figures of performance for models are displayed to have a direct comparison for the model under different number of figures. The figures mainly display the accuracy when applying four features or five features in the model, and offers a clear comparison in those figures.

We now display the accuracy of applying four features or five features in the proposed SVR model. To obtain the best parameter setting in the SVR model used for occupancy detection, we need to constantly compare the gap between the training and testing errors for the collected datasets by EnergyPlus. If the gap of accuracy between training and testing errors are relatively large, it means that this model is over-fitting on the training data. The accuracy of training data implies that the model has a potential under-fitting. In this situation, we need to adjust parameters to make the SVR-based model work better.

We randomly pick out a 15-day period from the testing dataset and compare it to the genuine value generated by EnergyPlus. A large number of experiments in this proposed SVR model show that 0.01 is a stable-performing value for ε. Figures 10, 11, and 12 show the simulation results of the



(a) 4 features.

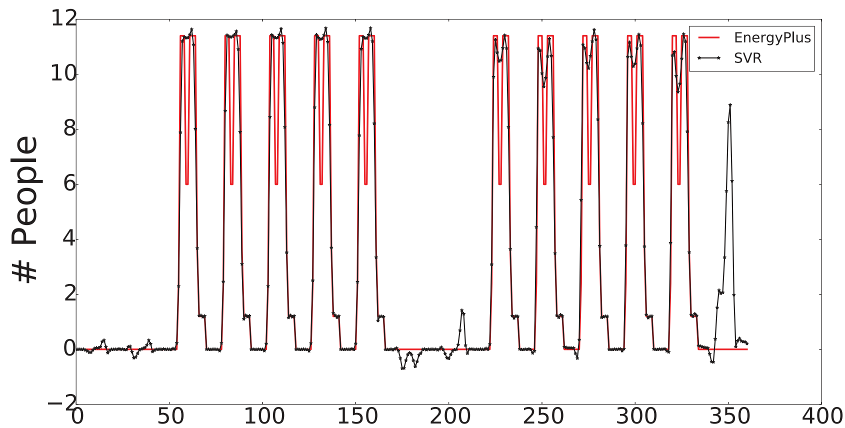


(b) 5 features.

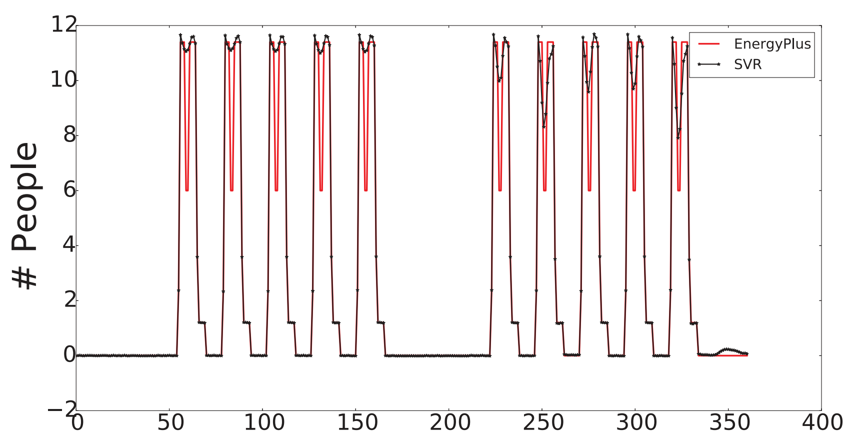
Fig. 10. Occupancy estimation accuracy when C equals 100 using SVR with four features and five features. X-axis spans over 360 hours (15 days), sampled hourly.

occupancy detection accuracy of the SVR model by using different numbers of features. To find a better performance model, we set C to be 100, 500, and 1000, respectively. Simulation results show that the SVR model tends to become more complex when the value of C becomes larger, hence the goal of optimizing the SVR model is to find the value of C that is one better tradeoff between under-fitting and over-fitting. For the five-feature model, it can be seen that the SVR model can obtain better performance for occupancy detection when the value of C varies in the range from 10 to 1000.

The two sets of features provide different conveniences in detecting to meet different demands. The first set of features can be relatively easily acquired while the second set requires more effort. The first set of features only requires a dataset that can be obtained from mathematical computation, whereas the second set requires light energy, which is a set of statistics that needs



(a) 4 features.



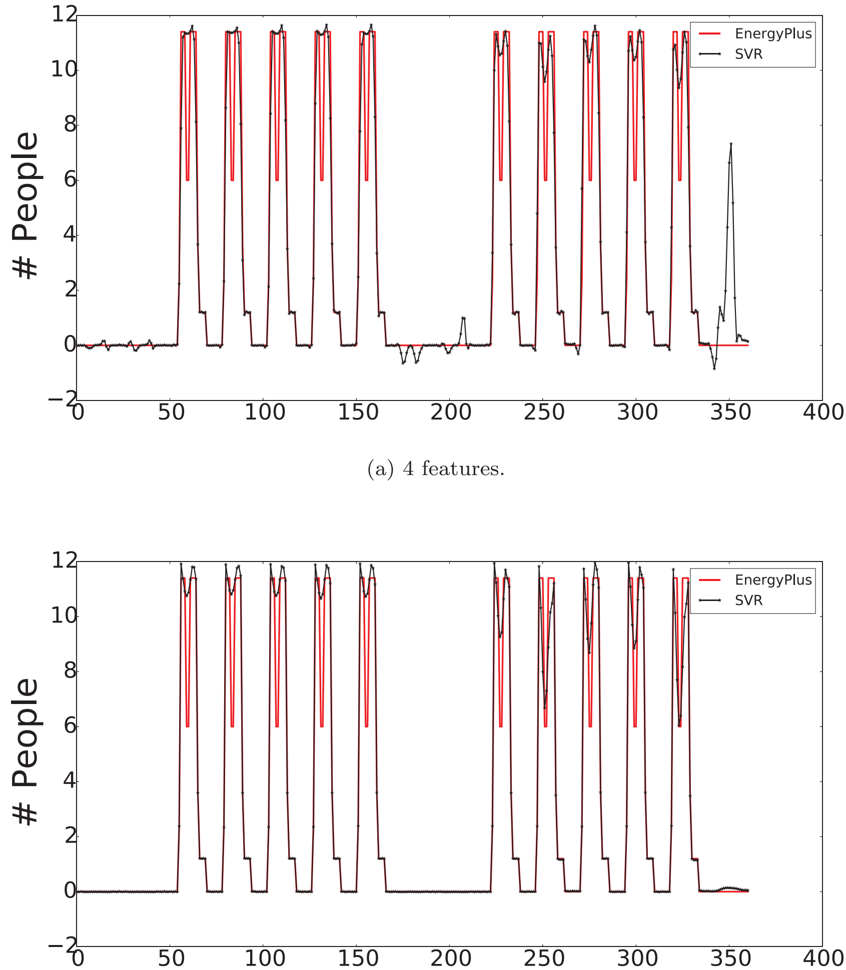
(b) 5 features.

Fig. 11. Occupancy estimation accuracy when C equals 500 using SVR with four features and five features. X-axis spans over 360 hours (15 days), sampled hourly.

more effort to achieve. In terms of practical application, it is suggested to choose the one that meets demand and gives the best convenience. However, further improvement can be considered by building models revolving around absorbing the current dataset into the SVR model, which makes the model work as a dynamic equation that is able to self-improve by a newly absorbed dataset and remain more effective, according to the current circumstance. Most importantly, it is suggested to choose the most efficient approach based on the specific situation. After all, the model is able to fulfill ordinary demand in accuracy of the four-feature model. Further improvement is likely to happen if one considers specific conditions for other offices in detail.

5.2 RNN-Based Occupancy Detection

In this section, the ELNN as shown in Figure 4 is used for occupancy detection in a building.



(a) 4 features.

(b) 5 features.

Fig. 12. Occupancy estimation accuracy when C equals 1000 using SVR with four features and five features. X-axis spans over 360 hours (15 days), sampled hourly.

5.2.1 Experiments. EnergyPlus takes outdoor thermal factors (such as ambient temperatures and solar factors), people occupancy, and HVAC-related powers as input, and produce the temperatures of rooms as its output. People occupancy is in units of number of people, which may have a fractional part as it represents the average people count over a short time span. We treat all the data used and produced by EnergyPlus equally as real-world factors, regardless if they were inputs or outputs of EnergyPlus. In the occupancy estimation work, we select data from those real-world factors, feed them into the RNN, and try to get estimated occupancy from it.

We use EnergyPlus to simulate the room thermal behavior in a year, using various inputs including occupancy information. We collect the inputs and outputs (room temperatures) of EnergyPlus simulation, which is discretized into hourly data points, to train ELNN. Given the simulated data provided by EnergyPlus, as shown in Figure 13, we feed selected channels of ambient factors and

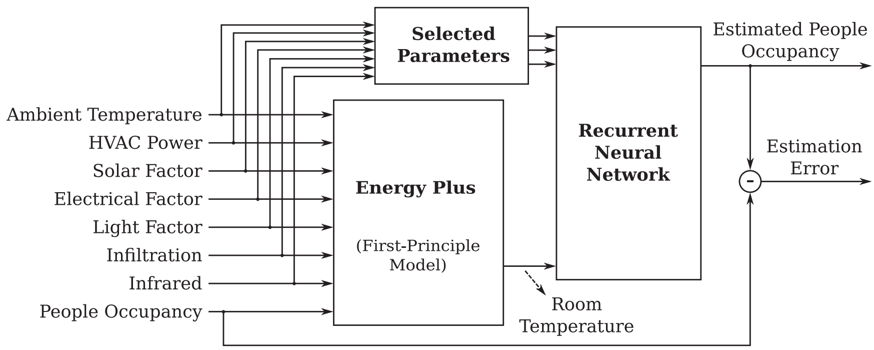


Fig. 13. Data configuration of ELNN.

other power data, along with room temperatures, into ELNN as input. We use estimated and real occupancy to drive the training process. We will configure two different selected datasets: one uses ambient factors and room temperatures only, another dataset uses ambient factors, room temperatures, and HVAC cooling/heating powers. The output of ELNN has multiple channels, which are, respectively, each room's estimated people occupancy.

In practical smart-building applications, room temperatures are easy to acquire from the installed sensors. Ambient temperature and solar factors are also relatively easy to be acquired or calculated. While other factors, such as HVAC cooling or heating powers, electrical equipment powers, and air infiltrations, need more instruments to per-room estimate in real-time. Because of these limitations, we select two different sets of real-world factors as the network input and compare the occupancy evaluation accuracy:

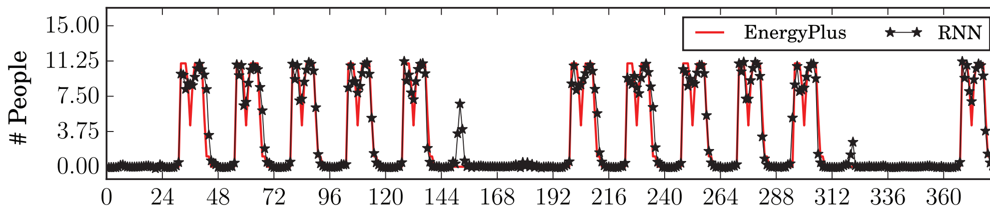
(1) Input includes ambient temperature, solar factors, and room temperatures only. This will be referred to as configuration I.
(2) Input includes ambient temperature, solar factors, room temperatures, and HVAC cooling/heating powers. This will be referred to as configuration II.

With different factors as network inputs, we also configure the RNN with different hidden recurrent layers varying from one to three ($k = 1, 2, 3$), to compare the estimation accuracies. We divide the 1-year simulation data into 12 months. Months 1–3, 5–7, and 9–11 are used for training; months 4, 8, and 12 are used for validating the trained networks.

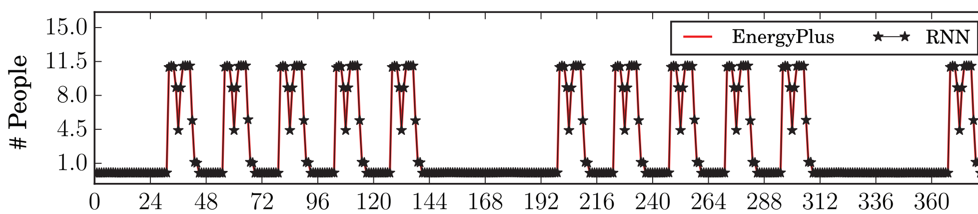
We evaluate the performance (mainly accuracy) of the proposed occupancy estimation method on a dataset of 1 year, using the building example shown in Figure 1. We picked a 15-day data subset starting from the third Sunday in August to be plotted in Figures 14 and 15. We report the validation errors for one-layer and two-layer networks since they are non-trivial and more noticeable. These figures show that the occupancy estimation of room 1, among all five rooms in this case, has similar error situations.

5.2.2 Analysis. The training and validation error statistics are shown in Tables 3 and 4, respectively. In the input configuration I, we use ambient and room temperatures only as the inputs (this is similar to the situation in which we only know temperature information from thermal sensors); in the input configuration II, we use ambient, room temperatures, and HVAC cooling/heating powers as the inputs (in case we know more information about a building).

At every sample point, estimation error e_i is calculated using $e_i = |p_i^{\text{RNN}} - p_i^{\text{EP}}|$, where p_i^{RNN} is people occupancy estimated by the RNN and p_i^{EP} is the referencing value used in EnergyPlus. Note

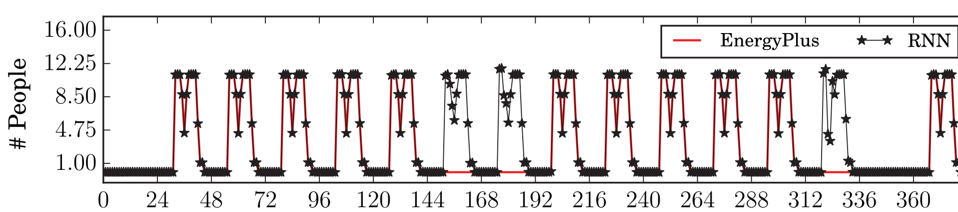


Conf. I: Ambient factors and room temperatures.

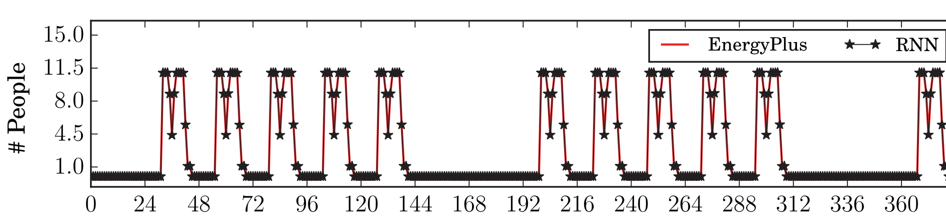


Conf. II: Ambient factors, room temperatures and HVAC powers.

Fig. 14. Occupancy estimation accuracy using one-layer recurrent neural network with input configurations I and II. *X*-axis spans over 384 hours (16 days), sampled hourly.



Conf. I: Ambient factors and room temperatures.



Conf. II: Ambient factors, room temperatures and HVAC powers.

Fig. 15. Occupancy estimation accuracy using two-layer recurrent neural network with input configurations I and II. *X*-axis spans over 384 hours (16 days), sampled hourly.

Table 3. Training Errors of Three Elman Architectures Using Two Different Input Configurations

	1 Hidden Layer		2 Hidden Layers		3 Hidden Layers	
	Conf. I	Conf. II	Conf. I	Conf. II	Conf. I	Conf. II
Avg. error	0.451	0.0149	0.00635	0.00643	0.0308	0.0291
Max. error	12.6	0.544	0.284	0.141	0.807	0.788
Error rate	20%	0.0061%	0.00%	0.00%	0.082%	0.015%

Table 4. Validation Errors of Three Elman Architectures Using Two Different Input Configurations

	1 Hidden Layer		2 Hidden Layers		3 Hidden Layers	
	Conf. I	Conf. II	Conf. I	Conf. II	Conf. I	Conf. II
Avg. error	0.538	0.0175	0.153	0.00560	0.0439	0.0340
Max. error	17.8	2.82	18.1	0.288	11.4	1.66
Error rate	21%	0.11%	2.4%	0.00%	0.71%	0.38%

that we may have zero people in a room (the occupancy value $p_i^{\mathrm{EP}} = 0$), so no relative errors are used. Also, occupancy values can be non-integer numbers as the estimated number of people in a room is the average number in a period.

In Tables 3 and 4, average error is calculated using $\frac{1}{n}\sum_n e_i$; maximum error is calculated using $\max\{e_i\}$; error rate is the number of points, where $e_i > 0.5$. We discuss the estimation accuracy separately about data configurations I and II.

In configuration I, the one-layer network suffers from the under-fitting problem (about 20% of data points have errors greater than 0.5). This is because the network needs more internal status to have the capability to estimate the people occupancy only using room and ambient temperature. As we increase the number of network layers, estimation accuracy improves (error rate 2.4% for two layers and 0.71% for three layers). Experiment results show that the RNN is able to estimate people occupancy only with ambient and room temperatures with a good accuracy (lower than 1%).

In configuration II, we provide more information (HVAC powers) for the occupancy training process than in configuration I. As a result, the ELNN with only two hidden recurrent layers can already perform quite well (no points having error greater than 0.5 were observed in the 1-year data). As network size grows (up to three), the estimation error grows (0.38%), but stays in an acceptable level.

5.3 Comparison Between SVR and RNN

In this section, we compare the accuracy and characteristics of the occupancy detection result, respectively, from SVR and RNN. It should be noted that the same features are used for the two different modelsto make a fare comparison. Figure 16 shows the result in which the black curve stands for the original outcome, the red curve stands for the SVR outcome, and the blue curve stands for the RNN outcome. Those results are generated based on the same features and number of data, which means the different results shown in the figure are only influenced by the models. From the figure, we can see that the results from both methods agree well with the original curves. Furthermore, we observe there is a small fluctuation in maximum error when the SVR model is applied. This indicates the maximum error of the SVR model will stay at a stable range when the features of the model are slightly changed. In comparison, maximum error of the RNN model is relatively more sensitive. The maximum error may double, triple, or rapidly diminish when the features of the model are changed. This phenomenon is also observed in the average error. The numeric value of average error swings more for the RNN model than for the SVR model under the situation that the features are changed. This could imply the two different characteristic behaviors of the two models, which gives the user the flexibility to apply for those two models. The SVR model is used when the features of a model will change from every now and then, to keep the detection precision at a stable range. The RNN model is applied when features are not changed in a real context, which possess even higher precision in detecting occupancy compared to SVR.

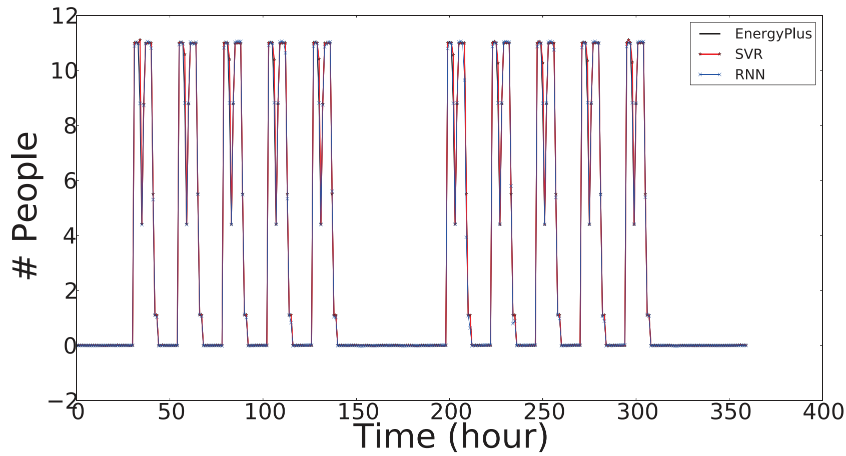


Fig. 16. Comparison between SVR and RNN in occupancy detection.

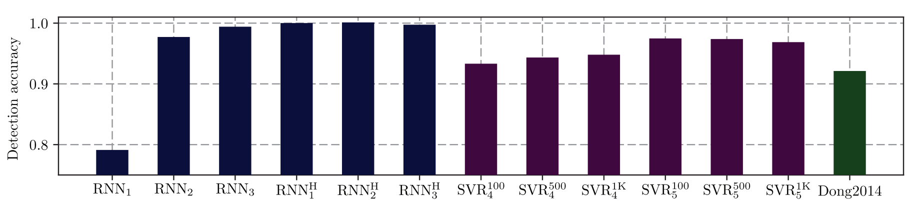


Fig. 17. Occupancy detection accuracy using SVR and RNN. (Explanation for the *X*-axis labels: subscript and superscript of RNN denotes number of hidden layers and awareness of HVAC; subscript and superscript of SVR denotes number of features and *C* value.)

Comparing to 92% detection accuracy in Dong and Lam (2014), as shown in Figure 17, moderately configured SVR and RNN models can achieve occupancy detection accuracy higher than 96%.

We remark that since SVR has great generalization ability and can guarantee finding the local minimum, SVR can perform well for time-series analysis for occupancy detection. RNN can also be used for a time-series prediction problem as the data configuration for RNN includes timesteps. The most important features related to occupancy behavior in a building are used in the machine-learning methods. These features covers all major thermal-related aspects to be coupled with the human activity. As a result, occupancy behavior can be accurately determined by using the selected machine-learning methods with time-series analysis.

6 CONCLUSION AND FUTURE WORKS

In this article, we propose machine-learning-based methods to detect the occupancy behavior of a building through the temperature and/or possible heat source information. Supporting vector regression and RNN methods, which are developed for smart buildings through the thermal sensor temperature information and/or possible heat source information, have been discussed. In all experiments, we use the realistic building simulation program EnergyPlus to collect training and validation datasets. Ambient factors, room temperature, and/or HVAC power were selected as features to train ELNN. In the SVR model, two sets of features are offered to feed off the model for different conveniences. The first set of features is comprised of four features including solar factor, working time, indoor temperature, and outdoor temperature, which are regarded as easily obtained features; whereas the second set of features adds light energy as the fifth feature. In light of the experimental results, the four-feature model has a quite accurate detection rate, which gives a 0.638 average error and 5.32% error rate. However, the five-feature SVR model giving a 0.317 average error and 2.64% error rate has a better performance than the four-feature model, which we consider as moderating the under-fitting issue. This indicates that using the SVR model is a viable option when it comes to occupancy detection given its convenience in data acquirement. In the recurrent neutral network-based method, the resulting Elman network can estimate occupancy information of each room of a building with high accuracy. Using ambient factors and room temperatures only, the average estimation error is 0.044, and only 0.71% of the estimated points have errors greater than 0.5 in terms of number of people. This indicates that it is possible to precisely estimate the occupancy only using ambient factors and room temperatures. With HVAC powers added, the estimation can be even more accurate with simpler neural networks. Our study further shows both methods can deliver similar accuracy in the occupancy detection. But the SVR model is more stable for adding or removing a feature from the feature pool, while the RNN method can deliver more accuracy when the features used in the model do not change a lot.

REFERENCES

Yuvraj Agarwal, Steve Hodges, Ranveer Chandra, James Scott, Paramvir Bahl, and Rajesh Gupta. 2009. Somniloquy: Augmenting network interfaces to reduce PC energy usage. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI'09)*. 365–380.

Yuvraj Agarwal, Stefan Savage, and Rajesh Gupta. 2010. SleepServer: A software-only approach for reducing the energy consumption of PCs within enterprise environments. In *Proceedings of the 2010 USENIX Annual Technical Conference (USENIXATC'10)*. 22.

Larry Stephen Bias and Chung Ming Cheng. 1999. Electronic programmable thermostat with temporary reset. US Patent No. 5937942, Filed Mar. 17, 1998, Issued Aug. 17, 1999.

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, and J. Glazer. 2001. EnergyPlus: Creating a new-generation building energy simulation program. *Energy Build.* 33, 4 (April 2001), 319–331.

Declan T. Delaney, Gregory M. P. O'Hare, and Antonio G. Ruzzelli. 2009. Evaluation of energy-efficiency in lighting systems using sensor networks. In *Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys'09)*. 61–66.

Department of Energy. 2016. Annual Energy Review. Retrieved from https://www.eia.gov/totalenergy/data/annual/index.php#consumption.

Robert H. Dodier, Gregor P. Henze, Dale K. Tiller, and Xin Guo. 2006. Building occupancy detection through sensor belief networks. *Energy Build.* 38, 9 (2006), 1033–1043.

Bing Dong and Khee Poh Lam. 2014. A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. In *Building Simulation*, Vol. 7. Springer, 89–106.

T. Ekwevugbe, N. Brown, V. Pakka, and D. Fan. 2013. Real-time building occupancy sensing using neural-network based sensor network. In *Proceedings of the 2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST'13)*. 114–119.

Jeffrey L. Elman. 1990. Finding structure in time. *Cogn. Sci. Multidiscip. J.* 14, 2 (1990), 179–211.

Varick L. Erickson, Yiqing Lin, Ankur Kamthe, Rohini Brahme, Amit Surana, Alberto E. Cerpa, Michael D. Sohn, and Satish Narayanan. 2009. Energy efficient building environment control strategies using real-time occupancy measurements. In *Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys'09)*. 19–24.

Ge Gao and Kamin Whitehouse. 2009. The self-programming thermostat: Optimizing setback schedules based on home occupancy patterns. In *Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys'09)*. 67–72.

Simon Haykin. 2007. Neural Networks: A Comprehensive Foundation. Prentice-Hall, Inc. Upper Saddle River, NJ.

Michael T. Heath. 2002. Scientific Computing: An Introductory Survey. McGraw-Hill, New York, NY.

Robert Hecht-Nielsen. 1988. Theory of the backpropagation neural network. *Neural Netw.* 1, 1 (1988), 65–93.

John D. Hobby, Alexander Shoshitaishvili, and Gabriel H. Tucci. 2012. Analysis and methodology to segregate residential electricity consumption in different taxonomies. *IEEE Trans. Smart Grid* 3, 1 (2012), 217–224.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. 2003. *A Practical Guide to Support Vector Classification.* Technical Report, Department of Computer Science, National Taiwan University.

Christian Igel and Michael Hsken. 2003. Empirical evaluation of the improved Rprop learning algorithms. *Neurocomputing* 50, 1 (2003), 105–123.

Khee Poh Lam, Michael Höynck, Bing Dong, Burton Andrews, Yun-Shang Chiou, Rui Zhang, Diego Benitez, Joonho Choi, et al. 2009. Occupancy detection through an extensive environmental sensor network in an open-plan office building. *IBPSA Building Simul.* 145 (2009), 1452–1459.

Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. 2010a. The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10).* ACM, New York, NY, 211–224. DOI: http://dx.doi.org/10.1145/1869983.1870005

Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. 2010b. The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10).* ACM Press, New York, NY, 211–224.

Ning Lu. 2012. An evaluation of the HVAC load potential for providing load balancing service. *IEEE Trans. Smart Grid* 3, 3 (2012), 1263–1270.

Abhinandan Majumdar, Jason L. Setter, Justin R. Dobbs, Brandon M. Hencey, and David H. Albonesi. 2014. Energy-comfort optimization using discomfort history and probabilistic occupancy prediction. In *Proceedings of the 2014 International Green Computing Conference (IGCC'14).* IEEE, Los Alamitos, CA, 1–10.

John Mardaljevic, Lisa Heschong, and Eleanor Lee. 2009. Daylight metrics and energy savings. *Lighting Res. Technol.* 41, 3 (2009), 261–283.

Michael C. Mozer, Lucky Vidmar, Robert H. Dodier. 1996. The neurothermostat: Predictive optimal control of residential heating systems. In *Proceedings of the 9th International Conference on Neural Information Processing Systems (NIPS'96).* MIT Press Cambridge, MA, 953–959.

Tuan Anh Nguyen and Marco Aiello. 2013. Energy intelligent buildings based on user activity: A survey. *Energy Build.* 56, 1 (2013), 244–257.

G. V. Puskorius, L. A. Feldkamp, and L. I. Davis. 1996. Dynamic neural network methods applied to on-vehicle idle speed control. *Proc. IEEE* 84, 10 (1996), 1407–1420.

M. Riedmiller and H. Braun. 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks.* 586–591.

Jurgen Schmidhuber. 2014. Deep learning in neural networks: An overview. *Neural Netw.* 61 (2014), 85–117.

James Scott, A. J. Bernheim Brush, John Krumm, Brian Meyers, Michael Hazas, Stephen Hodges, and Nicolas Villar. 2011. PreHeat: Controlling home heating using occupancy prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp'11).* 281–290.

Auswin George Thomas, Pedram Jahangiri, Di Wu, Chengrui Cai, Huan Zhao, Dionysios C. Aliprantis, and Leigh Tesfatsion. 2012. Intelligent residential air-conditioning system with smart-grid functionality. *IEEE Trans. Smart Grid* 3, 4 (2012), 2240–2251.

Paul J. Werbos. 1990. Backpropagation through time: What it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.

Michael Wetter. 2011. Co-simulation of building energy and control systems with the building controls virtual test bed. *J. Building Perform. Simul.* 4, 3 (2011), 185–203.

Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. 2004. Travel-time prediction with support vector regression. *IEEE Trans. Intell. Transport. Syst.* 5, 4 (2004), 276–281.

Junjing Yang, Mattheos Santamouris, and Siew Eang Lee. 2016. Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings. *Energy Build.* 121 (2016), 344–349.

Received October 2017; revised February 2018; accepted March 2018