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

QI HUA
Shanghai Jiao Tong University
HENGYANG ZHAO
University of California, Riverside
HAIBAO 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
XIN LI
Carnegie Mellon University
ESTEBAN TLELO-CUAUTLE
INAOE and CINVESTAV, 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, which can be used for energy reduction and security monitoring for emerging smart buildings. Our work is based on a realistic building simulation program, EnergyPlus, from Department of Energy. EnergyPlus can model the 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 in 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) method and recurrent neural network (RNN) method. Experimental results with SVR method show that 4-feature model provides accurate detection rate giving a 0.638 average error and 0.0532 error ratio, and 5-feature model gives a 0.317 average error and 0.0264 error ratio. This indicates that SVR is a viable option for occupancy detection. In RNN method, Elman's RNN (ELNN) can estimate occupancy information of each room of a building with high accuracy. It has local feedbacks in each layer and for a 5-zones 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.

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

General Terms: Design, Algorithm

This work is supported in part by NSF grant under No. CCF-1017090, in part by NSF Grant under No. CCF-1255899, in part by Semiconductor Research Corporation (SRC) grant under No. 2013-TJ-2417.

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

1. INTRODUCTION

Building takes an instrumental role in energy consumption and smartness of a building has a large impact on inhabitants. According to statistics provided by US Department of Energy, 70% of electricity of all has been consumed by buildings every year. Recent efforts have been poured into the awareness of improving efficiency in quite a few facets, e.g., heating, ventilation, air conditioning (HVAC) system [Erickson et al. 2009][Gao and Whitehouse 2009], lighting[Delaney et al. 2009], IT energy consumption management within buildings[Agarwal et al. 2009][Agarwal et al. 2010], etc. Amongst 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, a few studies [Bias and Cheng 1999] reveal that buildings utilizing programmable thermostats virtually are more likely to consume more energies than ones without using smart devices. 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 system.

Detecting the occupancy (i.e. whether there are 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 keeping human comfort.

Due to the importance of detecting building occupancy, many methods have been proposed in the past using different technologies such as passive infrared sensors [Dodier et al. 2006], wireless camera sensor network [Erickson et al. 2009], and applying sound level, case temperature, carbon-dioxide (CO2) and motion to estimate occupancy number [Ekwevugbe et al. 2013]. Preheat [Scott et al. 2011] built rooms with active radio frequency identification (RFID) and sensors to detect home occupancy. Mozer [Mozer et al. 1997] proposed a neural network method by using the history data from embedded motion sensors and actives RFID to explore occupancy rate. Thermostat [Lu et al. 2010] also devoted a similar approach through the employment of magnetic reed switches and passive infrared sensors to take control of the HVAC system at home. However, those methods are more expensive for deployment as dedicated equipment is required.

Many works have given an approach under the circumstance that the detection requires a comparatively strict requirements for sensors, and obviously the requirements of sensors resulting in transformations of infrastructures may dramatically increase expenditure when it comes to the total cost of the building and system. Besides, overflow data including a vast of different aspects of conditions with respect to inhabitant data looms a latent possibility to burden inhabitants psychological pressure, because a large number of people may not prefer living under the supervision of a great deal of data that is available to someone else. Moreover, it may lead to a threat of leakage of personal data, thus threatening personal privacy.

In this work, we propose a novel approach to detect occupancy under specific conditions by applying machine learning methods, while it does not require plenty of sensors to be installed in a certain building where as it makes detection based on historical data. We generate the mathematical model based on support vector regression (SVR) method and neural network detect occupancy with two sets of features in correspondence with different convenience. Feeding off the features generated from EnergyPlus, the SVR model is able to yield highly accurate results for occupancy detection which is a convenient and new attempt in this field.

The model displays benefits of not requiring highly-precise data set from sensors, therefore, it is able to reduce the equipment expenditure in modern buildings given the fast development of smart building, and two sets of features make it to be conveniently applied under different circumstances. The experiments are performed by using a realistic building simulation program, EnergyPlus, from Department of Energy., which can model the various time-series inputs to a building such as ambient temperature, heating, ventilation, and air-conditioning inputs, power consumption of electronic equipment, lighting and number of occupants in a room sampled in each hour and produce resulting temperature traces of zones (rooms). The next section describes EnergyPlus that is further applied to two machine learning based methods: SVR and Elman's recurrent neural network (ELNN).

2. ENERGYPLUS BASED SIMULATION FOR SMART BUILDINGS

In this section, we review the EnergyPlus software program, which provide accurate input and output traces from buildings for the new thermal modeling algorithm.

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, and weather. EnergyPlus consists of an integrated solution manager that manages the calculation of the heat balance of various surfaces in the building, the heat balance of the air, and the heat balance on the mechanical systems. The solution to each of these three elements is calculated separately and communicated to each other using the manager at each time-step. Due to its modularity, it is easy to establish links to other programs such as Google SketchUp for 3D building display.

An input data file (IDF) and weather file are needed for the EnergyPlus simulation. The IDF includes all the information of the building such as size, structure, position and the HVAC subsystem, etc. 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.

Fig. 1 shows the side view of an office building with 5 rooms and HVAC modeled in EnergyPlus. The heat sources for this building can be HVAC, light, occupants, electric equipment, air filtration, etc. The room temperature is also affected by the weather (ambient temperature and solar effects) and can be controlled by the HVAC system with coil and fan.

Fig. 2 shows the simulated temperature changes and input changes over 15 days from EnergyPlus for an office building with the 5 zones (rooms), as shown in Fig. 1.

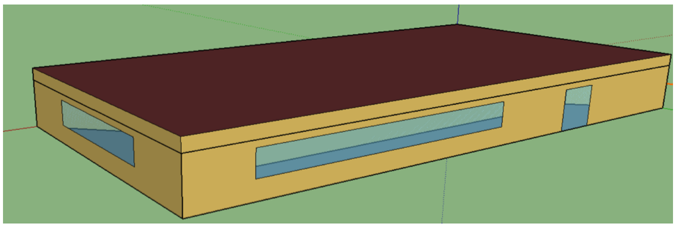


Fig. 1. The 5-zone office building side view

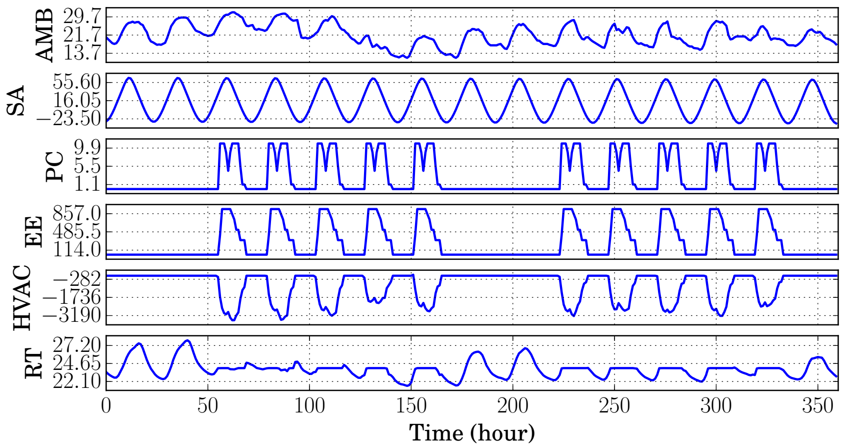


Fig. 2. Selected EnergyPlus input and simulated temperature output data sample in 15 days. (AMB: AMBient temperature; SA: Solar Angle; PC: People Count (occupancy); EE: Electrical Equipment power; HVAC: HVAC system cooling/heating power; RT: Room Temperature)

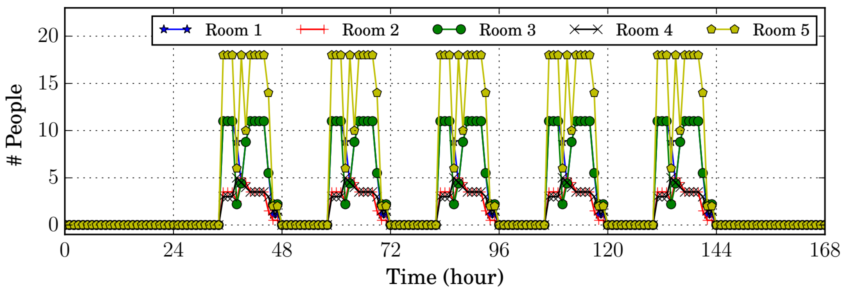


Fig. 3. Occupancy information of 5 rooms during one week.

EnergyPlus can assign different schedules for each room while simulating the thermal model. Fig. 3 shows a typical working schedule of the 5 rooms of the 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 like support vector regression (SVR) and recurrent neural network (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

$$f(x) = w \cdot \Gamma(x) + \lambda$$

In the equation above, $w \in R^n$, $\lambda \in R$, 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

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

where G_i is a loss function

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

Here C is a constant, and $k(x_i, x)$ is known as a kernel function. Through mathematical deduction [Wu et al. 2004], the $\varepsilon-$insensitive loss function can be minimized as

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

subject to $\sum_{i=1}^{l}(\alpha - \alpha_i^*) = 0,\ \ (\alpha_i - \alpha_i^*) \in [0, C]$. Here 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 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

$$\lambda = y_i - (w, x_i) - \varepsilon \quad for\ \alpha_i \in (0, C)$$
$$\lambda = y_i - (w, x_i) + \varepsilon \quad for\ \alpha_i^* \in (0, C)$$

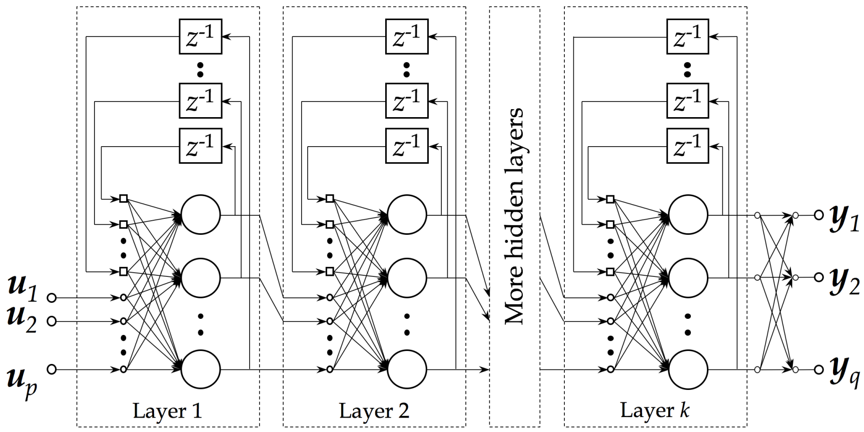


Fig. 4. Architecture of Elman Recurrent Neural Network

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

3.2 Review of Recurrent Neural Network

Learning based techniques such as neural networks, which is 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].

A recurrent neural network (RNN) is constructed by introducing internal status holders to a memory-less network so that it can deal with time-series data. The internal status holders store outputs of designated neurons and usually function as feedbacks into other neurons. The application of feedback enables RNNs to acquire time-dependent state representations, making them suitable devices for applications like time-dependent non-linear prediction, plant control, etc. [Haykin and Network 2004]. There are many RNN structures proposed by varying the form of the recurrent feedbacks [Elman 1990; Haykin and Network 2004; Puskorius et al. 1996].

We apply the Elman recurrent neural network architecture as shown in Fig. 4 to the occupancy of each room in a certain smart building. We describe the structure Elman architecture, how the gold-referencing data is computed, and the detailed works on training the networks. We construct Elman recurrent neural network architecture (as shown in Fig. 4) to build the black-box model for occupancy detection. In our work 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, which shows good performance for many time-series based learning (like voice recognition).

In 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 method such as gradient decent, Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [Heath 2010], and the Quasi-Newton method on the cost function. In practice, algorithms with lower computational cost has 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 more adaptive approach, and a further improvement method: RPropMinus [Igel and 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 recurrent neural networks. 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

Fig. 5. The office model: top view.

In the machine learning model we built for occupancy detection, we carefully selected 5 features each of which possesses some unique information hidden inside.

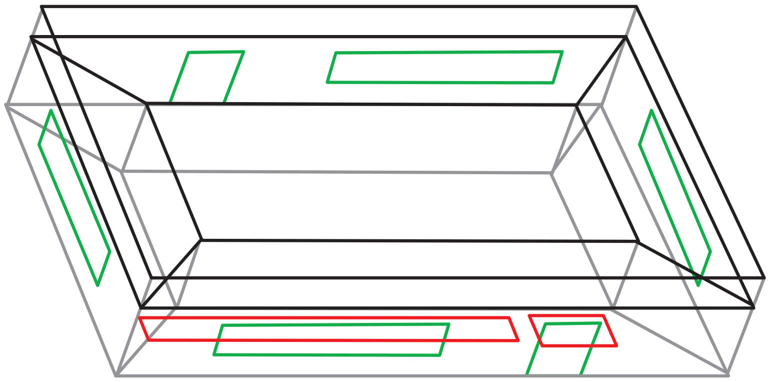


Fig. 6. The office model: side view.

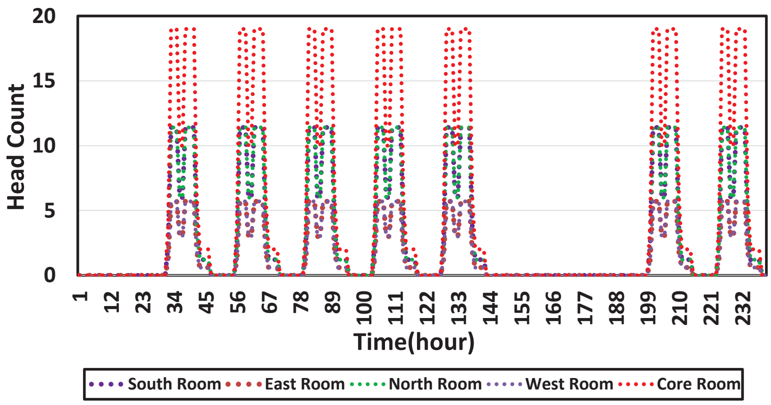


Fig. 7. Occupation information of 5 rooms for 10 days.

The features are solar angle, indoor temperatures, outdoor temperatures, working time, and lights energy. Solar angle is believed to have periodic information which varies across the entire year, outdoor temperature is an apparent factor that impacts the indoor temperature, working time denotes whether regular working schedule is executed, and lights energy gives out a radiation metric that causes rise of the temperature.

Fig. 5 and Fig. 6 show the side view and the top view of the building which contains five rooms and HVAC system by using the software EnergyPlus. This building can be influenced by heat sources produced from occupants, electric equipment, air filtration, etc. 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

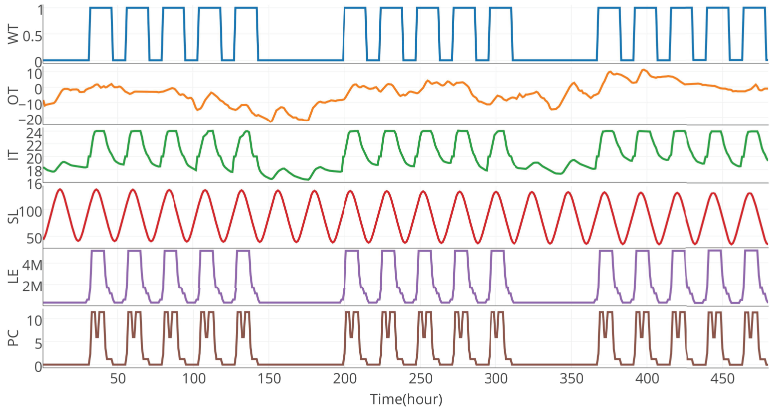


Fig. 8. Selected EnergyPlus input and simulated temperature output data sample in 20 days. (WT: Work Time; OT: Outdoor Temperature; IT: Indoor Temperature; SL: Solar Angle; LE: Lights Energy; PC: People Count.)

centre of the sun disc

$$\cos\theta_s = \sin\phi\sin\delta + \cos\phi\cos\delta\cos h$$

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 affect the number of working people in a certain office. It is also convenient to achieve 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 data set. Here the working time has a strong correlation with the employee common schedule, which is a relatively data set 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 are off-the-rack data set collected from weather forecast which also are the inputs for EnergyPlus to generate indoor temperatures data as an output. Herein we use the outdoor temperature data set of the location of the building simulated by EnergyPlus, so as our simulated building model goes through the exact same weather conditioning the genuine building has gone through. And outdoor temperatures 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 turns out to be one of the key features used in the detection model, to be more accurately, all of the factors including employee occupancy constitute the list of elements that results in the fluctuation of temperature inside the building. In essence, the approach to achieve the detection for a number of employee in a certain room is based on the contribution of heat emitted from different number of employees in a certain room. Owing to the fact that different number of employees gives out different amount of heat to impact the indoor temperature, makes the approach viable to detect the occupancy combing other factors that affect the indoor temperature.

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

As listed above, five features are applied in the machine learning method we apply in our approach. In keeping with different degree to application, different combination of features are used in the methods, which makes the application more flexible.

4.2 Data Configuration

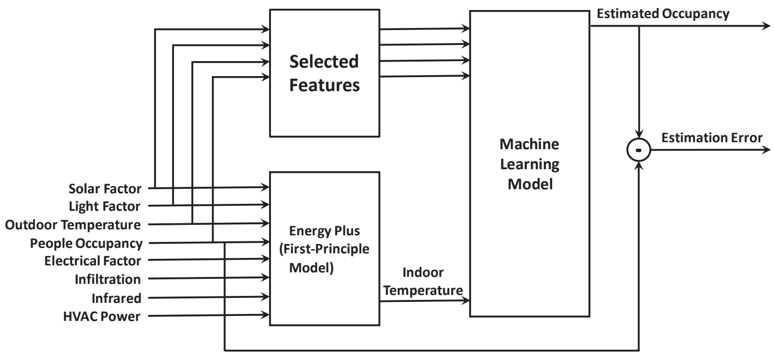


Fig. 9. Data configuration of machine learning model.

Fig. 9 shows how EnergyPlus produces indoor temperature in a certain room specifically. EnergyPlus feeds off factors such as indoor temperature, solar factor, electrical factor, light factor, infiltration, infrared, and people occupancy before it produces indoor temperature as an output using built-in methods and methods in accordance with its inputs. Of all those features taken into the method, indoor temperature is the key feature because it is directly impacted by the increase or drop in the number of employees in a certain office.

In general, the machine learning model takes the selected parameters as its features to train the data 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 a non-linear problem used for detection. And machine methods are increasingly used in all kinds of supervised and unsupervised problems in all aspects.

Fig. 7 shows a vintage working schedule of the five zones of an office building. Our goal of the detection model is to detect accurate number of employees in a room by certain parameters and data collections. Fig. 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 Fig. 6. The schedule 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. It should be noted that there is a nonlinear relationship between the occupancy and the related factors. The machine learning model can be used to solve this nonlinear problem for occupancy detection.

In this model, we provide two sets of model which offers different extent of convenience to detect number of employees in an office. The first set of model is comprised of features such as solar factor, working time, indoor temperature and outdoor temperature, data of 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 features in the second feature set, light factor, to enhance the accuracy of 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 condition.

For separating the whole data set, we split the one-year simulation data into twelve months and three time periods, in which the months 1-3, 5-7, 9-11 are referred to as the training data and the months 4, 8, 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 different number of features in this model.

Table I. Training error statistics 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
Err. ratio	0.311	0.0856	0.188	0.0889	0.181	0.124

5.1.1 *Experiments.* We evaluate the SVR detection model using testing data set. Through a wide range of experiments, we learn radial basis function kernel also known as Gaussian kernel work best in this model. The two most important parameters in SVR model are penalty C and radius ε, where plenty of tests are conducted to obtain a good set of parameter, hence, we experiment different values

Table II. Validation error statistics 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
Err. ratio	0.068	0.0264	0.0578	0.0274	0.0532	0.0325

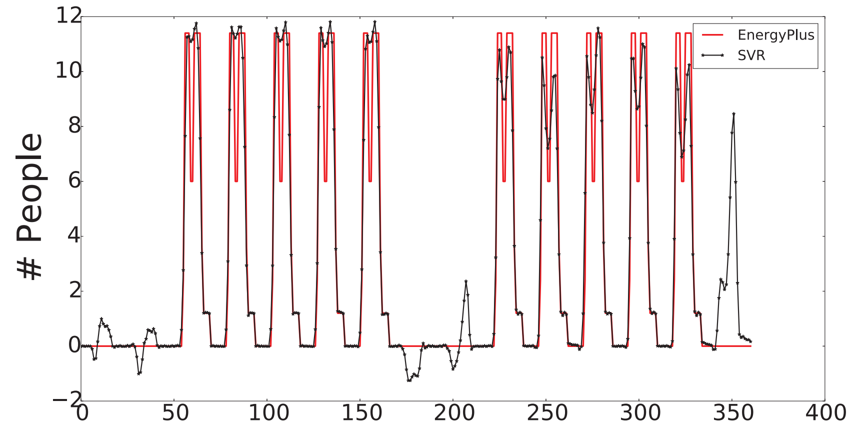
of C and ε hoping to obtain the best. It is widely known that to get an accurate performance in SVR model or any other machine learning methods, the best approach is to enumerate a quantity of combinations of parameters, and conduct experiments to drive the result toward a better trend. During the process of seeking out the best result for the model, the set of parameters is adjusted step by step to obtain a model which has a better accuracy than the previous one. After a batch of experiments for the model are conducted, we pick out the parameters in the model that brings about the best performance. Here we also want to highlight, that most of the time, the parameters working best for a model sometimes can not be proved theoretically, therefore confirmed parameters often are determined by a great number of trials and experiments.

Table I 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 $e_i = \left|O_i^{SVR} - O_i^{EP}\right|$ 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 ratio by $\frac{1}{n}\sum_n e_i$ and $\frac{Average\ error}{full\ occupancy}$, respectively. Also, in this table different values of C are tested to seek out an accurate model for occupancy detection.

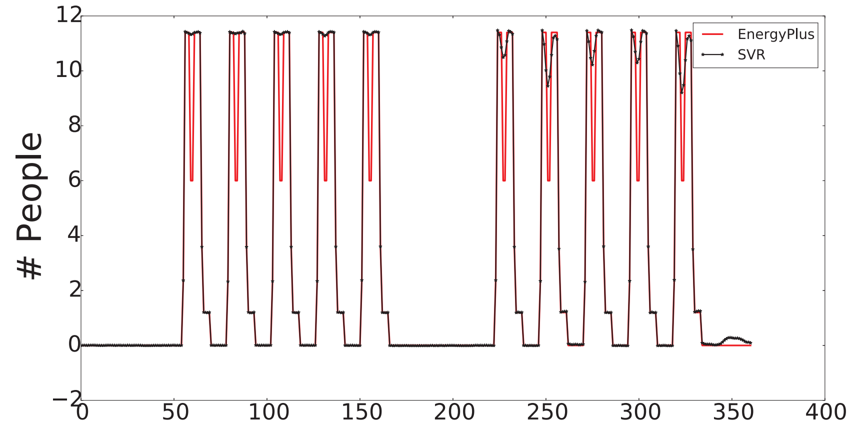
Table II shows the validation error statistics of the proposed SVR model. The reason why the 5-feature model performs even better than the 4-feature model for occupancy detection is that the 4-feature model suffers slightly in under-fitting issue which results in a high bias. It is important to determine the parameters which 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 4 features or 5 features in the model, and offers a vivid comparison in those figures.

We now display the accuracy of applying 4 features or 5 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 data sets by EnergyPlus. If the gap of accuracy between training and testing errors are relatively large, which means that this model is over-fitting on the training data. In accuracy on training data implies that the model has a potential under-fitting. In this situation, we need to adjust parameters



(a) 4 features.

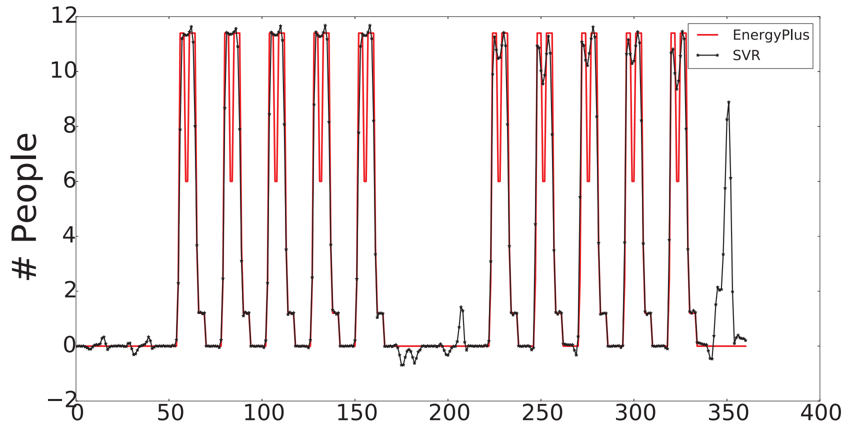


(b) 5 features.

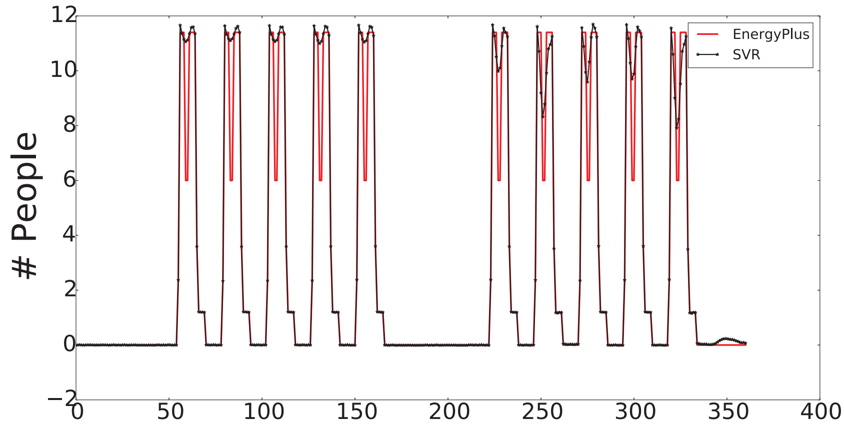
Fig. 10. Occupancy estimation accuracy when C equals 100 using SVR with 4 features and 5 features.

to make the SVR-based model work better.

We randomly pick out a 15-day period from the testing data set and compare it to the genuine value generated by EnergyPlus. Large number of experiments in this proposed SVR model shows that 0.01 is a stable-performing value for ε. Fig. 10, Fig. 11, and Fig. 12 show the simulation results of occupancy detection accuracy of SVR model by using different numbers of feature. 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 5-feature model,



(a) 4 features.

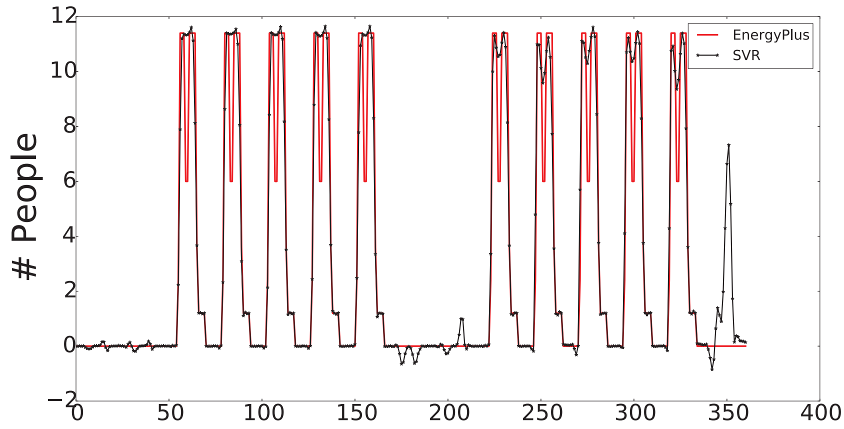


(b) 5 features.

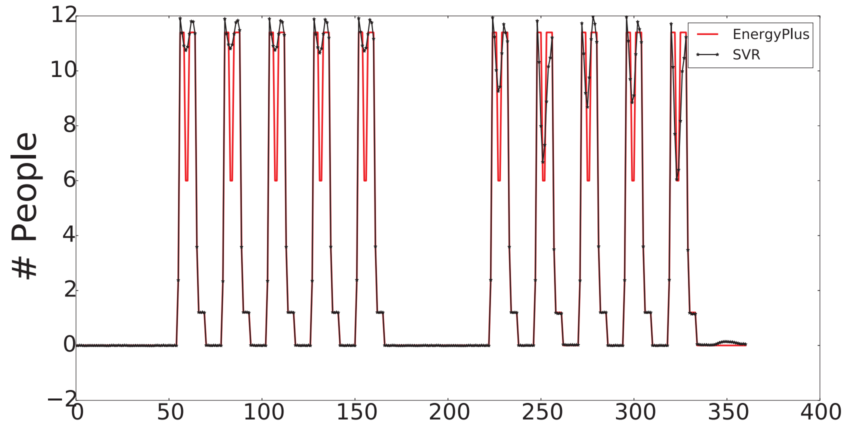
Fig. 11. Occupancy estimation accuracy when C equals 500 using SVR with 4 features and 5 features.

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 convenience in detecting to meet different demand. The first set of features can be relatively easily acquired while the second set requests more efforts. The first set of feature only requires data set that can be obtained from mathematic computation, whereas the second set requires light energy which is a set of statistics that needs 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 model revolving around absorbing the current data set into the SVR mod-



(a) 4 features.



(b) 5 features.

Fig. 12. Occupancy estimation accuracy when C equals 1000 using SVR with 4 features and 5 features.

el, which makes the model work as a dynamic equation that is able to self-improved by newly absorbed data set and remain more effectively according to the current circumstance. Most importantly, it is suggested to choose the most efficient approach based on the facing situation, after all, the model is able to fulfill ordinary demand in accuracy in 4-feature model. Further improvement is likely to happen if one considers specific conditions for other office in detail.

5.2 RNN Based Occupancy Detection

In this section, we apply the Elman's recurrent neural network (ELNN) architecture shown in Fig. 4 to the occupancy of each room in a certain smart building.

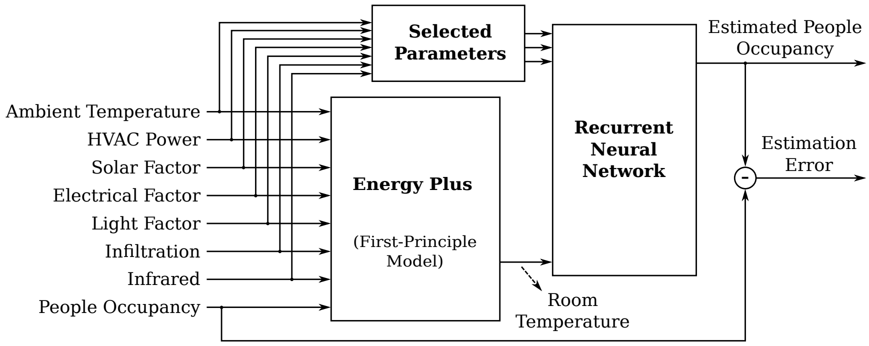


Fig. 13. Data configuration of Elman's recurrent neuron network.

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 it's output. People occupancy is in unit of number of people, which maybe decimal as it represents average people count over a short time span. We treat all the data used and produced by EnergyPlus equally as real-world factors, regardless they were inputs or outputs of EnergyPlus. In the occupancy estimation work, we select data from those real-world factors, feed them into the recurrent neural network, 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 Fig.13, we feed selected channels of ambient factors and 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.

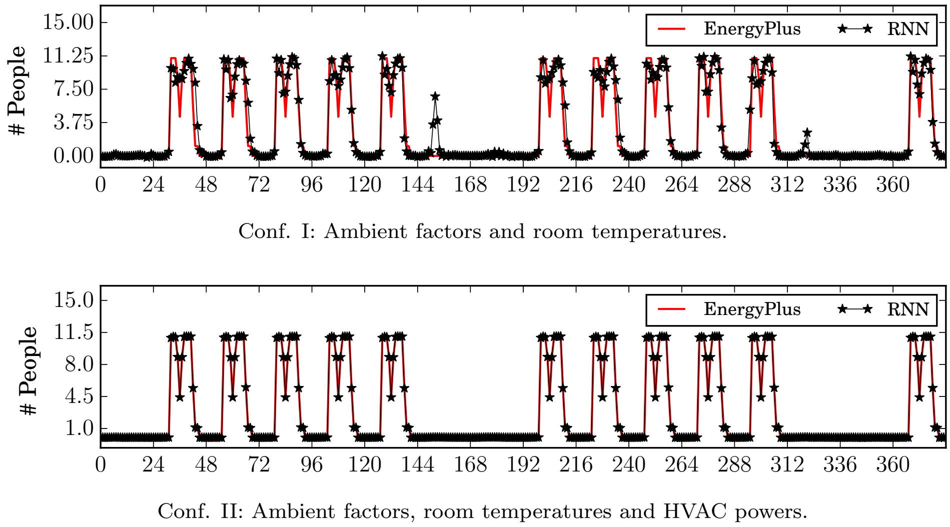
In practical smart-building applications, room temperatures are easy to acquire from the installed sensors. Ambient temperature, solar factors are also relatively easier to be acquired or calculated. While other factors, such as HVAC colling or heating powers, electrical equipment powers and air infiltrations, 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 as configuration I from now on.

(2) Input includes ambient temperature, solar factors, room temperatures and H-VAC cooling/heating powers. This will be referred as configuration II from now on.

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

We evaluate the performance (mainly accuracy) of the proposed occupancy estimation method on a dataset of one year using the building example shown in Fig. 1. We picked a 15-day data subset starting from the 3rd Sunday in August to be plotted in Fig. 14 and Fig. 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 the 5 rooms in this case, having similar error situations.



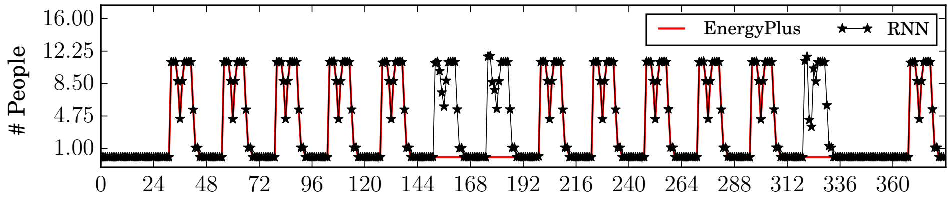
Conf. I: Ambient factors and room temperatures.

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

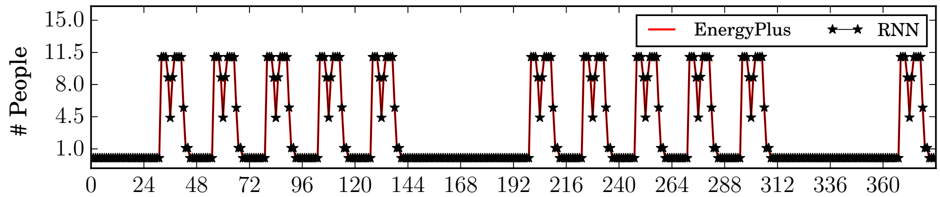
Fig. 14. Occupancy estimation accuracy using one-layer recurrent neural network with input configuration I and II.

5.2.2 *Analysis.* The training and validation error statistics are shown in Table III and Table IV, respectively. In the input configuration I, we use ambient and room temperatures only as the inputs (this is similar to 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 $e_i = \left|p_i^{\mathrm{RNN}} - p_i^{\mathrm{EP}}\right|$, where p_i^{RNN} is people occupancy estimated by the RNN and p_i^{EP} is referencing value used in EnergyPlus. Note that we may have zero people in a room (so the occupancy value $p_i^{EP} = 0$), so no relative errors are used. Also, occupancy values can be non-integer numbers as the estimated number of people in a room is a average number in a period.



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 configuration I and II.

	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 III. Training error statistics 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%

Table IV. Validation error statistics of three Elman architectures using two different input configurations.

In Table III and Table IV, 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 configuration I and II.

In data configuration I, one-layer network suffers from under-fitting problem (about 20% 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 use room and ambient temperature. As we increasing the number of network layers, estimation accuracy improves (error rate 2.4% for 2-layer and 0.71% for 3-

layer). 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 the configuration II, we provides more information (HVAC powers) for the occupancy training process than the configuration I. As a result, the ELNN with only two hidden recurrent layers can already perform quite well (no points having error grater than 0.5 were observed in the one-year data). As network size grows (up to 3), the estimation error grows (0.38%), but stays in acceptable level.

5.3 Comparison Between SVR and RNN

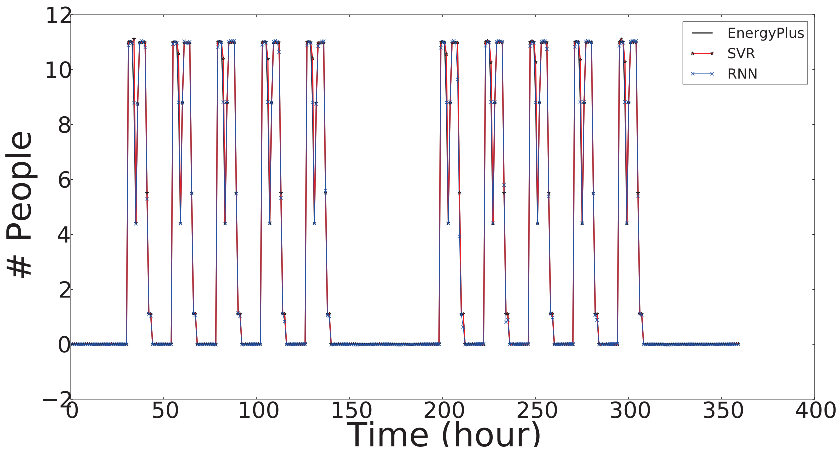


Fig. 16. Comparison between SVR and RNN in occupancy detection.

In this section, we compare the accuracy and characteristics of the occupancy detection result respectively from SVR and RNN. Fig. 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 model it self. From the figure, it is easy to make a summarization that both methods quite fit the original curve which means both model could be put into practice in detecting occupancy for a certain room at a certain time.

In light of the experimental results, there is small fluctuation in Max. error when SVR model is applied. This indicates the maximum error of SVR model will stay at a stable range when the features of the model are slightly altered. In comparison, Max. error of RNN model is relatively sensitive, which doubles, triples, or rapidly diminishes when the features of the model is altered. This phenomenon also applies for the Avg. error, the numeric value of Avg. error swings more for RNN model than it is for SVR model under the situation that the features are changed. This could imply the two different characters of different models, which gives the user different flexibility to apply for those two models. SVR model is used when the features of a model will change from every now and then, in order to keep the detection precision at a stable range. RNN model is applied when features are not changed in a real context, which possess a even higher precision in detecting occupancy compared to SVR.

6. CONCLUSION AND FUTURE WORKS

A mathematical model based on SVR and RNN to detect employee occupancy in a smart building, has been introduced. Two machine learning based occupancy detection methods for smart buildings through the thermal sensor temperature information and/or possible heat source information, have been discussed. In all experiments, we used 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 Elman's recurrent neural network (ELNN). In SVR model, two sets of features are offered to feed off the model for different conveniences. The first set of features is comprised of 4 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, 4-feature model has a quite accurate detection rate which gives a 0.638 average error and 0.0532 error ratio. However, 5-feature SVR model giving a 0.317 average error and 0.0264 error ratio has a better performance than 4-feature model, which we consider as moderating the under-fitting issue. This indicates that using 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 even simpler neural networks.

As a conclusion, what influences the model is the number of features and the number of data points, virtually the two different methods SVR and RNN work similarly well for solving this problem. However, further improvement is likely to occur when the model is refined to be able self-improved using current data set in the future.

REFERENCES

Agarwal, Y., Hodges, S., Chandra, R., Scott, J., Bahl, P., and Gupta, R. 2009. Somniloquy: Augmenting network interfaces to reduce pc energy usage. *9*, 365–380.

Agarwal, Y., Savage, S., and Gupta, R. 2010. Sleepserver: A software-only approach for reducing the energy consumption of pcs within enterprise environments. 22–22.

Bias, L. S. and Cheng, C. M. 1999. Electronic programmable thermostat with temporary reset.

Delaney, D. T., O'Hare, G. M., and Ruzzelli, A. G. 2009. Evaluation of energy-efficiency in lighting systems using sensor networks. 61–66.

Dodier, R. H., Henze, G. P., Tiller, D. K., and Guo, X. 2006. Building occupancy detection through sensor belief networks. *Energy and Buildings 38,* 9, 1033–1043.

EKWEVUGBE, T., BROWN, N., PAKKA, V., AND FAN, D. 2013. Real-time building occupancy sensing using neural-network based sensor network. 114–119.

ELMAN, J. L. 1990. Finding structure in time. *Cognitive Science A Multidisciplinary Journal 14,* 2, 179–211.

ERICKSON, VARICK, L., LIN, YIQING, KAMTHE, BRAHME, ROHINI, SURANA, AND AMIT. 2009. Energy efficient building environment control strategies using real-time occupancy measurements.

ERICKSON, V. L., LIN, Y., KAMTHE, A., BRAHME, R., SURANA, A., CERPA, A. E., SOHN, M. D., AND NARAYANAN, S. 2009. Energy efficient building environment control strategies using real-time occupancy measurements. 19–24.

GAO, G. AND WHITEHOUSE, K. 2009. The self-programming thermostat: optimizing setback schedules based on home occupancy patterns. 67–72.

HAYKIN, S. AND NETWORK, N. 2004. A comprehensive foundation. *Neural Networks*.

HEATH, MICHAEL T. /MUNSON, E. M. 2010. *Scientific Computing: An Introductory Survey.* McGraw-Hill Higher Education.

HECHT-NIELSEN, R. 1988. Theory of the backpropagation neural network. *Neural Networks 1,* 1, 65C93.

HOBBY, J. D., SHOSHITAISHVILI, A., AND TUCCI, G. H. 2012. Analysis and methodology to segregate residential electricity consumption in different taxonomies. *IEEE Transactions on Smart Grid 3,* 1, 217–224.

IGEL, C. AND HSKEN, M. 2003. Empirical evaluation of the improved rprop learning algorithms. *Neurocomputing 50,* 1, 105–123.

LU, J., SOOKOOR, T., SRINIVASAN, V., GAO, G., HOLBEN, B., STANKOVIC, J., FIELD, E., AND WHITEHOUSE, K. 2010. The smart thermostat: using occupancy sensors to save energy in homes. 211–224.

LU, N. 2012. An evaluation of the hvac load potential for providing load balancing service. *IEEE Transactions on Smart Grid 3,* 3, 1263–1270.

MOZER, M. C., VIDMAR, L., DODIER, R. H., ET AL. 1997. The neurothermostat: Predictive optimal control of residential heating systems. *Advances in Neural Information Processing Systems*, 953–959.

NGUYEN, T. A. AND AIELLO, M. 2013. Energy intelligent buildings based on user activity: A survey. *Energy and Buildings 56,* 1, 244C257.

PUSKORIUS, G. V., FELDKAMP, L. A., AND DAVIS, L. I. 1996. Dynamic neural network methods applied to on-vehicle idle speed control. *Proceedings of the IEEE 84,* 10, 1407–1420.

RIEDMILLER, M. AND BRAUN, H. 1993. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *IEEE International Conference on Neural Networks.* 586–591.

SCHMIDHUBER, J. 2014. Deep learning in neural networks: An overview. *Neural Networks 61*, 85–117.

SCOTT, J., BERNHEIM BRUSH, A., KRUMM, J., MEYERS, B., HAZAS, M., HODGES, S., AND VILLAR, N. 2011. Preheat: controlling home heating using occupancy prediction. 281–290.

THOMAS, A. G., JAHANGIRI, P., WU, D., CAI, C., ZHAO, H., ALIPRANTIS, D. C., AND TESFATSION, L. 2012. Intelligent residential air-conditioning system with smart-grid functionality. *IEEE Transactions on Smart Grid 3,* 4, 2240–2251.

WERBOS, P. J. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE 78,* 10, 1550–1560.

WU, C.-H., HO, J.-M., AND LEE, D.-T. 2004. Travel-time prediction with support vector regression. *Intelligent Transportation Systems, IEEE Transactions on 5,* 4, 276–281.