Learning-Based Occupancy Behavior Detection for Smart Buildings

Hengyang Zhao*, Zhongdong Qi*, Shujuan Wang†, Kambiz Vafai†,
Hai Wang§, Haibao Chen¶ and Sheldon X.-D. Tan*

* Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521
† Department of Mechanical Engineering, University of California, Riverside, CA 92521
§ School of Microelectronic and Solid-State Electronics,
University of Electronic Science and Technology of China, Chengdu, China 610054
¶ Department of Micro/Nano-electronics, Shanghai Jiao Tong University, Shanghai, China 200240

***Abstract*—In this article, we propose a novel method to detect the occupancy behavior of a building through the temperature and/or possible heat source information, which can be used for energy reduction, 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). The new approach is based on a learning based approach in which a recurrent neutral network (RNN) is trained to detect the number of people in a room based on the room temperature and other information such as ambient temperature, and other related heat sources. We applied the Elman's recurrent neural network (ELNN), which has local feedbacks in each layer. We use an empirical formula to calculate the RNN layer number and layer size to configure RNN architecture to avoid overfitting and under-fitting problems. Experimental results from a case study of a 5-zone building show that ELNN can lead to very accurate 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 when we consider ambient, room temperatures and HVAC powers as detectable information. Without knowing HVAC powers, estimation error can still be 0.044 on average, and only 0.71% estimated points have errors greater than 0.5.**

I. Introduction

Detecting the occupancy (i.e. whether there are residents) in a building or a room has many applications ranging from enegy reduction to security monitoring. For instance, occupancy detection is critical for energy and comfort management system in a smart building [16]. Using the occupancy information, HVAC and lighting can be automatically controlled to reduce energy consumption while keeping human comfort. Previous research shows that energy can be saved by 28% by automatically sensing occupancy and turning off HVAC when the building is not occupied [15].

Due to the importance of detecting building occupancy, many methods have been proposed in the past using different technologies such as using passive infrared sensors [4], using wireless camera sensor network [12], and applying sound level, case temperature, carbon-dioxide (CO2) and motion to estimate occupancy number [5]. . However, those methods are more expensive for deployment as dedicated equipment are required.

In this article, we propose a novel method to detect the occupany behavior of a building room just using temperature information from thernal sensors and other available heat source information, which are more convenient for depolyment as many existing buildings already has those built-in thermal sensors and information already. 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 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 new approach is based on a machine learning approach in which a recurrent neutral network is trained to detec the number of people in a room based on the room temperature, ambient temperature, and other related heat sources. We apply the Elman's recurrent neural network, which has local feedbacks in each layer. We use a simple formula to calculate the RNN layer number, layer size to configure RNN architecture to avoid overfitting and underfitting problems.

Experimental results from a case study of a 5-room office show that ELNN can lead to very accurate occupancy behavior prediction. Using only ambient and room temperatures as input, ELNN can deliver average estimation error 0.044 in terms of number of people; among all the estimated occupancy, only 0.71% of estimated points have errors greater than 0.5. If we also include HVAC cooling/heating powers, which are not collectible by equiped temperature sensors, ELNN can give more precise evaluation on occupancy with average error 0.0056 (number of people) and maximum error 0.288.

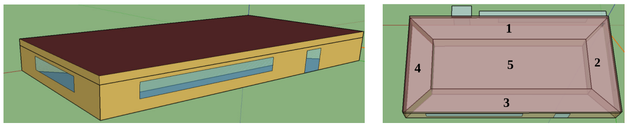
II. Review of EnergyPlus for energy simulation of building

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 [3] 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 specifics of the structure, to heat sources and sinks within the building, and weather. EnergyPlus consists of an integrated solution manager which 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 are calculated separately and communicated to each other using the manager at each time-step. Due to its modularity, it's easy to establish links to other program links such as Google SketchUp [1] 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 will be generated in the spreadsheet file after running the simulation.

978-1-4799-5341-7/16/$31.00 ©2016 IEEE



(a) (b)

Fig. 1: The 5-zone office building (a) side view (b) top view.

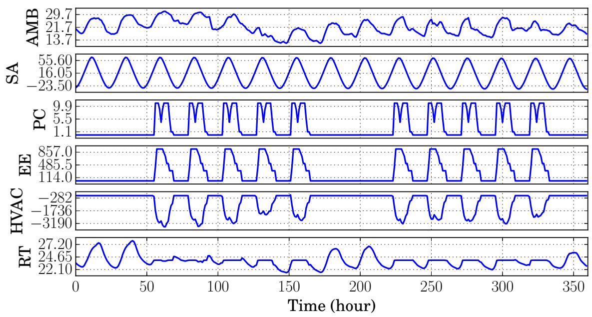


Fig. 2: Selected EnergyPlus input and simulated temperature output data sample in 15 days. (AMB: AMBient temperature; SA: Solar Angle; PC: People Count (occupancy); EE: Electrical Equipment power; HVAC: HVAC system cooling/heating power; RT: Room Temperature)

Fig. 1 shows the side view and the top view of an office building with 5 rooms and HVAC modeled in the 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). The room temperature 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 a office building with the 5 zones (rooms) as shown in Fig. 1. EnergyPlus can assign different schedule 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 [2], [14]. As a result, nonlinear modeling is preferred for accurate temperature control and management.

III. Review of recurrent neural networks

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 consist of many layers recently have dramatically improve the state-of-the-art in speech recognition and image recognition [13].

Memoryless feed-forward neural networks does not work for the time domain models, which depend on the history of the inputs of the models instead of current inputs. Recurrent neural networks (RNN) has been proposed [19] to build time domain models.

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

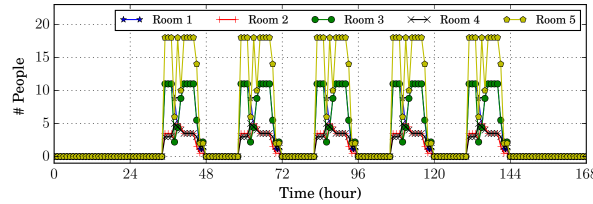


Fig. 3: Occupancy information of 5 rooms during one week

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. [8]. There are many RNN structures proposed by varying the form of the recurrent feedbacks [6], [8], [17]. In his work, we will focus on applying the Elman recurrent network architecture (ELNN) [6], which applies local recurrent feedback on each layer of neurons, which shows good performance for many time-series based learning (like voice recognition).

A. Review of recurrent neural network training

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 [9], and the Quasi-Newton method on the cost function $\boldsymbol{J}$. In practice, algorithms with lower computational cost has been developed. Back-propagation algorithm is a widely-used algorithm and has been well studied [10]. 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 [18], which is more adaptive approach, and a further improvement method: RPropMinus [11], 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) [20] unfolds every network activation of a continuous sequence. Back-propagation through structure (BPTS) [7] delivers more computational efficiency on arbitrary structured networks.

IV. Recurrent Neural Network Based Occupancy Estimation

In this section, we apply the Elman recurrent neural network architecture 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 estimation. 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.

EnergyPlus takes outdoor thermal factors (such as ambient temperatures and solar factors), people occupancy and HVAC

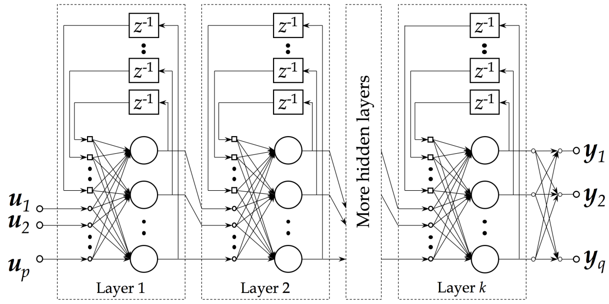


Fig. 4: Architecture of Elman Recurrent Neural Network

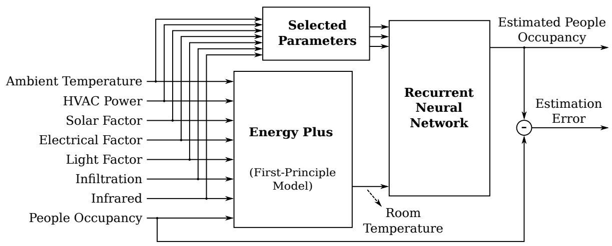


Fig. 5: Data configuration of Elman recurrent neuron network

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 the Elman recurrent neural network. Given the simulated data provided by EnergyPlus, as shown in Fig.5, we feed selected channels of ambient factors and other power data, along with room temperatures, into the Elman recurrent neural network 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 the Elman network 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 HVAC cooling/heating powers. This will be referred as configuration II from now on.

	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 I: 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 II: Validation error statistics of three Elman architectures using two different input configurations

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.

V. Experimental results and analysis

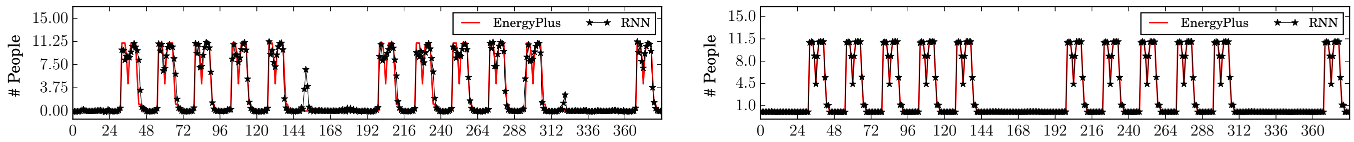
In this section, we evaluate the performance (mainly accuracy) of the proposed occupancy estimation method on a dataset of one year using the building example shown in Fig. 1. We picked a 15-day data subset starting from the 3rd Sunday in August to be plotted in Fig. 6, Fig. 7. We report the validation 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.

The training and validation error statistics are shown in Table I and Table II 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-integrer numbers as the estimated number of people in a room is a average number in a period.

In Table I and Table II, 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.

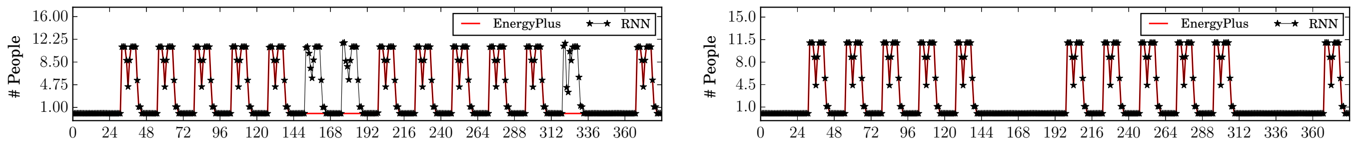
In data configuration I, one-layer network suffers from under-fitting problem (about 20% data points have errors greater than 0.5). This is because the network needs more internal status to have the capability to estimate the people



(a) Conf. I: Ambient factors and room temperatures

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

Fig. 6: Occupancy estimation accuracy using one-layer recurrent neural network with input configuration I and II



(a) Conf. I: Ambient factors and room temperatures

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

Fig. 7: Occupancy estimation accuracy using two-layer recurrent neural network with input configuration I and II

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.

VI. Conclusion

In this article, we propose a recurrent neural network based method to detect the occupancy behavior of a building through the temperature and/or possible heat source information. We use a realistic building simulation program EnergyPlus to collect training and validation datasets. Ambient factors, room temperature, and/or HVAC power are selected as features to train the Elman recurrent neural network. 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 network.

References

[1] "Google SketchUp," http://www.sketchup.com.

[2] T. Bergman, A. Lavine, F. P. Incropera, and D. P. DeWitt, Fundamentals of Heat and Mass Transfer, 7th ed. New York: John Wiley & Sons, 2011.

[3] 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, "EnergyPlus: creating a new-generation building energy simulation program," Energy and Buildings, vol. 33, no. 4, pp. 319–331, Apr. 2001.

[4] R. H. Dodier, G. P. Henze, D. K. Tiller, and X. Guo, "Building occupancy detection through sensor belief networks," Energy and buildings, vol. 38, no. 9, pp. 1033–1043, 2006.

[5] T. Ekwevugbe, N. Brown, V. Pakka, and D. Fan, "Real-time building occupancy sensing using neural-network based sensor network," in Digital Ecosystems and Technologies (DEST), 2013 7th IEEE International Conference on, July 2013, pp. 114–119.

[6] J. L. Elman, "Finding structure in time," Cognitive science, vol. 14, no. 2, pp. 179–211, 1990.

[7] C. Goller and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure," in Neural Networks, 1996., IEEE International Conference on, vol. 1. IEEE, 1996, pp. 347–352.

[8] S. Haykin and N. Network, "A comprehensive foundation," Neural Networks, vol. 2, no. 2004, 2004.

[9] M. T. Heath, Scientific Computing: An Introductory Survey. McGraw-Hill, 1997.

[10] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in Neural Networks, 1989. IJCNN., International Joint Conference on. IEEE, 1989, pp. 593–605.

[11] C. Igel and M. Hüsken, "Empirical evaluation of the improved rprop learning algorithms," Neurocomputing, vol. 50, pp. 105–123, 2003.

[12] A. Kamthe, L. Jiang, M. Dudys, and A. Cerpa, "SCOPES: smart cameras object position estimation system," in Wireless Sensor Networks, 6th European Conference, EWSN 2009, Cork, Ireland, February 11-13, 2009. Proceedings, 2009, pp. 279–295. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00224-3_18

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.

[14] Z. Liu, S. X.-D. Tan, H. Wang, A. Gupta, and S. Swarup, "Compact nonlinear thermal modeling of packaged integrated systems," in Proc. Asia South Pacific Design Automation Conf. (ASPDAC), January 2013, pp. 157–162.

[15] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, "The smart thermostat: Using occupancy sensors to save energy in homes," in Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 211–224. [Online]. Available: http://doi.acm.org/10.1145/1869983.1870005

[16] T. A. Nguyen and M. Aiello, "Energy intelligent buildings based on user activity: A survey," Energy and Buildings, vol. 56, pp. 244–257, 2013.

[17] G. V. Puskorius, L. Feldkamp, L. Davis et al., "Dynamic neural network methods applied to on-vehicle idle speed control," Proceedings of the IEEE, vol. 84, no. 10, pp. 1407–1420, 1996.

[18] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in Neural Networks, 1993., IEEE International Conference on. IEEE, 1993, pp. 586–591.

[19] R. Rojas, Neural networks: a systematic introduction. Springer Science & Business Media, 2013.

[20] P. J. Werbos, "Backpropagation through time: what it does and how to do it," Proceedings of the IEEE, vol. 78, no. 10, pp. 1550–1560, 1990.