Deep Learning for Building Occupancy Estimation Using Environmental Sensors

Zhenghua Chen, Chaoyang Jiang, Mustafa K. Masood, Yeng Chai Soh, Min Wu and Xiaoli Li

Abstract Building Energy efficiency has gained more and more attention in last few years. Occupancy level is a key factor for achieving building energy efficiency, which directly affects energy-related control systems in buildings. Among varieties of sensors for occupancy estimation, environmental sensors have unique properties of non-intrusion and low-cost. In general, occupancy estimation using environmental sensors contains feature engineering and learning. The traditional feature extraction requires to manually extract significant features without any guidelines. This handcrafted feature extraction process requires strong domain knowledge and will inevitably miss useful and implicit features. To solve these problems, this chapter presents a Convolutional Deep Bi-directional Long Short-Term Memory (CDBLSTM) method that consists of a convolutional neural network with stacked architecture to automatically learn local sequential features from raw environmental sensor data from scratch. Then, the LSTM network is used to encode temporal dependencies of these local features, and the Bi-directional structure is employed to consider the past and future contexts simultaneously during feature learning. We conduct real experiments

Z. Chen (✉) · M. Wu · X. Li
Institute for Infocomm Research (I2R), A*STAR, Singapore, Singapore
e-mail: Chen_Zhenghua@i2r.a-star.edu.sg

M. Wu
e-mail: wumin@i2r.a-star.edu.sg

X. Li
e-mail: xlli@i2r.a-star.edu.sg

C. Jiang
School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China
e-mail: cjiang@bit.edu.cn

M. K. Masood
Department of Civil Engineering, Aalto University, Espoo, Finland
e-mail: mkm_imn@hotmail.com

Y. C. Soh
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore
e-mail: eycsoh@ntu.edu.sg

W. Pedrycz and S.-M. Chen (eds.), *Deep Learning: Algorithms and Applications*, Studies in Computational Intelligence 865,
https://doi.org/10.1007/978-3-030-31760-7_11

to compare the CDBLSTM and some state-of-the-art approaches for building occupancy estimation. The results indicate that the CDBLSTM approach outperforms all the state-of-the-arts.

Keywords Deep learning · Building occupancy estimation · Environmental sensors · CDBLSTM

1 Introduction

To maintain the thermal comfort of indoor environments, around 40% of the energy has been consumed in building sectors [28]. Thus, a lot of attention has been paid on building energy efficiency and sustainable development. To achieve that, a crucial factor is the building occupancy information, also known as occupant number or range in buildings. It can be used for building climate and adaptive light control [28, 36]. Balaji et al. saved 17.8% of energy for HVAC systems relied on actual occupancy levels in a designed experiment [1]. A light control system developed in [24] has reported a reduction of 35–75% of energy consumption for building light control systems. However, to obtain an accurate and robust occupancy estimation system is a challenging mission and remain unsolved.

Occupancy estimation can be done by the use of different sensors. For instance, Liu et al. present a detection of the absence and presence of occupants via PIR sensors [27]. It will be more meaningful to obtain the actual occupant number or range indoors. In order to fulfill that, the methods relied on RFID and wearable devices were presented in [1, 25]. However, these approaches require users to wear specific devices, which is intrusive and inconvenient. Accurate occupancy estimation can be achieved by using cameras [42]. However, camera based solutions often suffer from the problems of insufficient illumination and high computational load. Besides, they also have the issue of privacy concerns. Some other methodologies rely on occupants' involvement, such as using chair sensors [23] and applicants power usage data [22]. However, occupants that do not involved will not be able to be detected.

Recently, environmental sensors are widely adopted for occupancy estimation, because they are low-cost and non-intrusive for users [21, 29, 40, 41]. Due to the complex relationship between environmental sensor measurements and occupancy levels, physical modeling is with limited performance. An alternative way is to model the complex relationship by using machine learning techniques which work well on function approximation. Since, environmental sensor data are with large noise and not representative for different occupancy levels, the machine learning models trained with raw sensory data may have limited performance. The common operation is to perform feature engineering which intends to extract more informative representations for different occupancy levels [26]. However, the traditional manual feature engineering does not have a guideline on which features should be extracted for occupancy inference. In addition, it requires strong domain knowledge and will inevitably miss implicit and useful features. To solve this problem, this chapter presents a Convolutional Deep Bi-directional Long Short-Term Memory (CDBLSTM) that consists of a convolutional neural network with a stacked structure to learn useful representations (features) automatically from scratch [11]. The convolutional network is able to learn some sequential local features from raw environmental sensor data. Since the environmental sensory data is a typical time series, temporal dependencies are of great importance for accurate and robust occupancy inference. To model the temporal dependencies in data, we adopt a BLSTM network whose inputs are the sequential local features learned by the convolutional neural network. We have compared the CDBLSTM approach with some state-of-the-arts in existing literature by using real evaluation.

2 Literature Review

Many advanced algorithms have been presented for occupancy inferences in buildings using environmental sensor data. The authors in [13] presented an occupancy estimation system for an open office room by using sensor networks that are able to collect data of CO_2, CO, acoustics, PM2.5, motion, illumination, temperature and humidity. Some statistical features, e.g., moving average of 20-min and 1st order difference, were manually extracted. Next, the most important features were chosen via the popular information gain theory. Finally, data-driven methods including Support Vector Machine (SVM), Artificial Neural Network (ANN) and Hidden Markov Model (HMM) were utilized for occupancy estimation. They made a conclusion that the most significant sensors are CO_2 and acoustic, and the HMM achieves the best performance for occupancy estimation.

The authors in [30] employed environmental sensors of temperature, CO_2, humidity, and pressure, to estimate occupancy for a tutorial room. They extracted some similar features used in [13]. An ELM-based wrapper algorithm was developed for feature selection and occupancy inference.

In [38], the authors investigated various sensors including sound, motion, temperature, door state, CO_2, humidity, passive infrared and light to infer occupancy in both multi-occupant and single-occupant offices via some widely used machine learning algorithms. Instead of extracting more useful features, they used raw sensor data as features. Here, the authors applied many informative sensors to guarantee a satisfactory performance of their proposed method. The contribution of different sensors (features) were tested by using the theory of information gain. Eventually, light level, door state and CO_2 are shown to be the most important parameters. For different algorithms, the decision tree (DT) approach has the best performance.

Candanedo et al. developed an occupancy detection system with sensors of humidity, CO_2, temperature and light levels [3]. They also used the raw sensor data as features in this work, and utilized some statistical models identify the two states of absence and presence of occupants. Different combinations of features with distinct statistical approaches were tried, and then the best sensors and models can be selected. At last, they made a conclusion which claims that a satisfactory performance is able to be fulfilled when properly selecting sensors and learning methods.

Since occupancy dynamics has the Markov property [4, 7, 8], the HMM model has achieved great success for building occupancy detection and estimation [13]. But, the traditional HMM often suffers from some limitations, such as the use of mixture of Gaussian model to estimate emission probabilities and the fixed transition probability matrix. To solve these issues, the authors in [12] presented an IHMM-MLR for environmental sensor based occupancy inference. Firstly, inhomogeneous transition probability matrices for capturing occupancy dynamics at distinct time steps were developed. Then, multinomial logistic regression to produce the emission probabilities with environmental sensor data was designed. Two schemes, i.e., online and offline, were formulated to infer occupancy in distinct situations.

Chen et al. presented another system to enhance the performance for occupancy estimation by considering occupancy properties [6]. They performed a fusion of traditional machine learning algorithms with a well-developed occupancy model which is able to show occupancy properties. The sensors they utilized include CO_2, humidity, pressure and temperature, which is widely available. The algorithms include ELM, SVM, ANN, KNN, CART and LDA. They formulated a Bayes filter to fuse the occupancy model and six data-driven algorithms for the estimation of occupancy. A detailed survey for occupancy estimation can be found in [5].

Here, we leverage on the environmental sensors including temperature, CO_2, pressure and humidity that are popular in normal HVAC systems [14] instead of applying specific sensors, such as acoustic level [13, 38], motion [19, 38] and light level [3]. Without applying the noisy sensor data as features or using some handcrafted statistical features, we attempt to automatically extract some useful local sequential features by using the convolutional neural network with stacked structure. Then, the BLSTM network is able to encode temporal dependencies for sequential local features during high-level feature learning. We have made a comprehensive comparison with some state-of-the-arts by using actual experiments.

3 Methodology

We firstly demonstrate an overview of the CDBLSTM for environmental sensor based occupancy inference. Then, we introduce the key components in CDBLSTM, i.e., the convolutional neural network, the DBLSTM, and the classification layers. Finally, the introduction of the training process of the CDBLSTM approach will be covered.

3.1 Overview

For environmental sensor based occupancy estimation, the key part is to learn discriminative representations (features) from raw data for distinct occupancy levels. Figure 1 presents the CDBLSTM framework for environmental sensor based occupancy

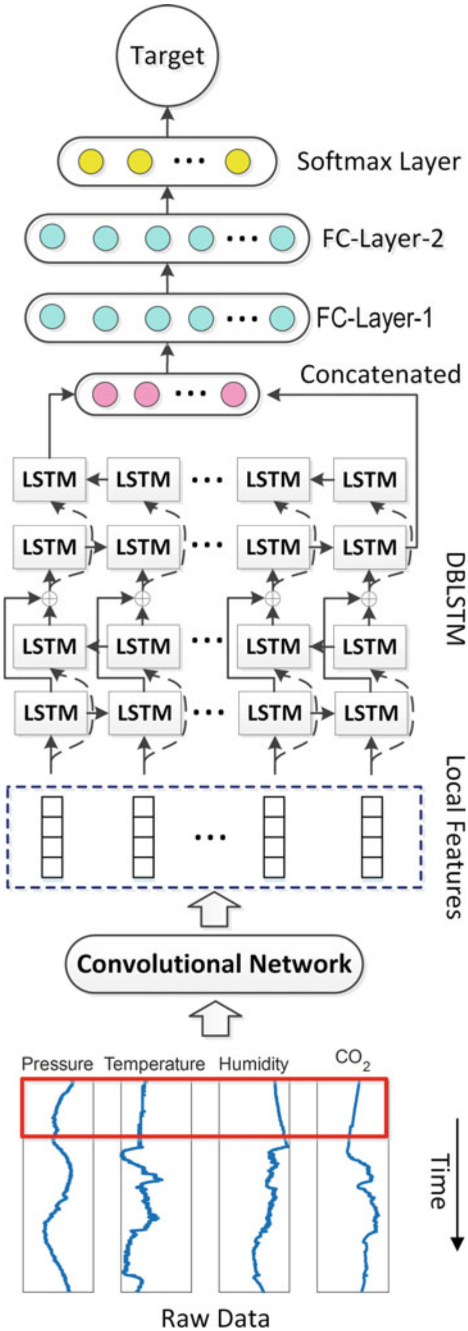


Fig. 1 Framework of the CDBLSTM approach [11]

inference. Raw input is a sliding window of environmental sensor data. Then, a convolutional network with multiple filters is applied for learning features of local sliding windows known as local feature learning, which is of great importance for distinguishing data from different occupancy levels. Next, the DBLSTM is leveraged to encode temporal dependencies of local sequential features in forward and backward directions. Finally, the learned high-level features from the DBLSTM are fed into fully connected and softmax layers for the classification of different occupancy levels.

3.2 Convolutional Operation

We implement convolutional neural network on environmental sensor data to produce sequential local features. Generally, it contains a convolutional layer, together with a pooling layer. Figure 2 shows the convolutional and pooling operations on environmental sensor data. The functionality of the convolutional operation is to use a sliding window over the raw time-series data to get sequential local features. And then, the pooling operation is to reduce feature dimension of the sequential local features. The detailed implementation of the two operations will be presented below.

Convolutional Layer: Suppose that the n input samples are $\{\mathbf{X}_i\}$, $i = 1, 2, \ldots, n$, and each input sample $\mathbf{X}_i \in \mathbb{R}^{r \times d}$ is a sliding window environment sensor data, where r is the length of sequence and d is the number of sensors. It can also be represented as $\mathbf{X}_i = [\mathbf{x}_1, \ldots, \mathbf{x}_r]$. The definition of the convolution operation is to multiply a filter vector $\mathbf{v} \in \mathbb{R}^{md \times 1}$ with a slice of the input $\mathbf{x}_{i:i+m-1} \in \mathbb{R}^{md \times 1}$ which is shown as follows

$$\mathbf{x}_{i:i+m-1} = \mathbf{x}_i \oplus \mathbf{x}_{i+1} \oplus \cdots \oplus \mathbf{x}_{i+m-1} \tag{1}$$

where m denotes the windows size and $\oplus$ is the concatenation operation. Next, an activation function is performed over the multiplied results, shown as

$$c_i = g\left(\mathbf{v}^\top \mathbf{x}_{i:i+m-1} + b\right) \tag{2}$$

where $g(\cdot)$ is the activation function, b is the bias term and $\top$ is the transpose operation. The widely used ReLU activation function [31] is adopted. By sliding the filter from the beginning of the input sequence to its end, we can produce a feature map, shown as follows:

$$\mathbf{c}^j = [c_1, c_2, \ldots, c_{r-m+1}] \tag{3}$$

where $j = 1, 2, \ldots, k$, and k is the number of filters.

Pooling Layer: The pooling operation is to reduce feature dimension, leading to more discriminative features [15]. In this work, we adopt the widely used max-pooling

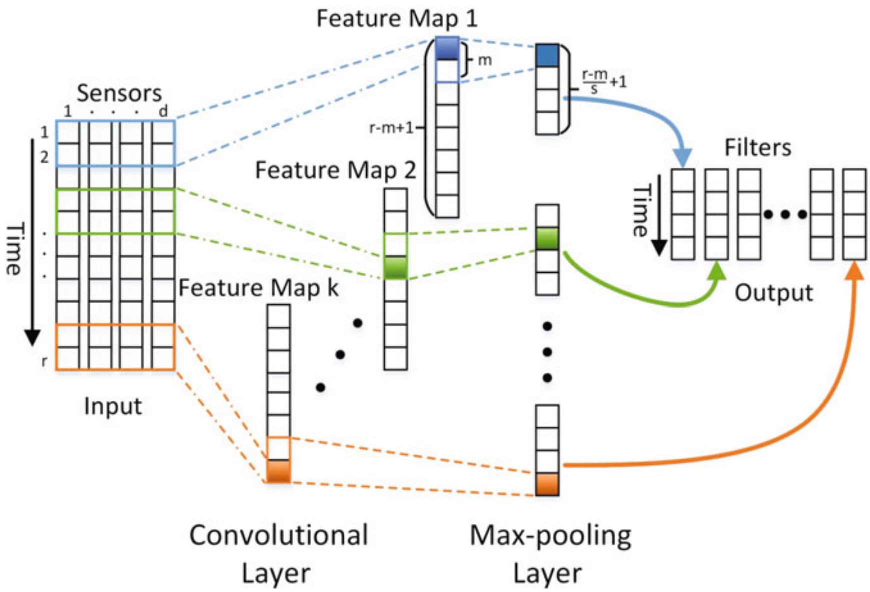


Fig. 2 Convolutional network structure

which conducts an operation of maximum on s consecutive components of feature map $\mathbf{c}^j$. After pooling operation, the features will be

$$\mathbf{z}^j = [z_1, z_2, ..., z_{\frac{r-m}{s}+1}] \tag{4}$$

where $z_i = \max(c_{is-s}, c_{is-s+1}, ..., c_{is-1})$. Hence, the pooling operation will generate compressed feature map $\mathbf{z}^j$, $j \in 1, 2, \ldots, k$. Eventually, the output of the convolutional neural network will have a feature dimension of $\left(\frac{r-m}{s} + 1\right) \times k$.

In general, assume the number of samples n, the input data has a dimension of $n \times r \times d$. The output of the convolutional neural network has a size of $n \times \left(\frac{r-m}{s} + 1\right) \times k$. It can be found that the length of the input data is compressed from r to $\left(\frac{r-m}{s} + 1\right)$. In addition, the data dimension changes from d (number of sensors) to k (number of filters), where k is much larger than d. This means that the data becomes more informative. In other word, the convolutional neural network can be treated as a local feature learned which is able to get more informative representations and preserve the temporal information from raw environmental sensor data.

3.3 Deep Bi-directional LSTM

Recurrent Neural Network (RNN) is widely used for the modeling of time series data thanks to its strong sequential modeling capacity. However, the conventional RNN often has the problem of gradient vanishing or exploding during training. This dramatically influence the performance of RNN on modeling long-term dependencies in time-series data [2]. To solve this issue, the authors in [17] proposed a new architecture, named LSTM, which attempts to use some gates to control the information for preserving or discarding, such that it is able to capture long-term dependencies of the sequence. The LSTM network has been successfully employed in a number of important and challenging tasks, e.g., activity recognition [9, 10] and natural language processing [34]. The conventional LSTM only considers the sequential information in one direction, that is the forward direction. This is not adequate for sequential modeling of environmental sensor data. The future information may also be useful. To consider both the future and past contexts for occupancy inference, we adopt the BLSTM which contains a forward layer and a backward layer to process sequential data in the forward and backward directions.

Recently, deep structures have achieved great success in representation learning [16]. The Deep Bi-directional LSTM (DBLSTM) which stacked multiple BLSTM layers is adopted in this study to encode the temporal dependencies and learn high-level features from the sequential local features extracted by the convolutional neural network. In addition to that, the DBLSTM is able to make the inputs to propagate through time and space (layers), simultaneously, such that, the model parameters are able to distribute over layers instead of enlarging memory size of the network. This will result a more efficient non-linear operation of the data and is also the ultimate purpose for stacking multiple layers in deep learning [16]. Figure 3 illustrates a hidden layer l at time step $t-1$, t and $t+1$ of the DBLSTM network, where the arrows pointing to the left and right denote the backward and forward operations respectively. Here, the forward operation from time step $t-1$ to t is to capture the past information, and the backward operation from time step $t+1$ to t is to model the future information. We use one hidden layer l at time step t as an example to show the detailed operation of the DBLSTM network. Assume that h_{l-1}^{t} is the hidden state, C_l^{t-1} is the memory cell state, w_l^f, w_l^i, w_l^C and w_l^o are the weights, b_l^f, b_l^i, b_l^C and b_l^o are the biases, and $\sigma(\cdot)$ denotes the sigmoid activation function. The forward process shown as $\rightarrow$ and the backward process shown as $\leftarrow$ can be formulated as follows:

$$
\begin{aligned}
\overrightarrow{f}_l^t &= \sigma\left(\overrightarrow{w}_l^f[\overrightarrow{h}_l^{t-1}, \overrightarrow{h}_{l-1}^t] + \overrightarrow{b}_l^f\right) \\
\overrightarrow{i}_l^t &= \sigma\left(\overrightarrow{w}_l^i[\overrightarrow{h}_l^{t-1}, \overrightarrow{h}_{l-1}^t] + \overrightarrow{b}_l^i\right) \\
\overrightarrow{\tilde{C}}_l^t &= \tanh\left(\overrightarrow{w}_l^C[\overrightarrow{h}_l^{t-1}, \overrightarrow{h}_{l-1}^t] + \overrightarrow{b}_l^C\right) \\
\overrightarrow{C}_l^t &= \overrightarrow{f}_l^t * \overrightarrow{C}_l^{t-1} + \overrightarrow{i}_l^t * \overrightarrow{\tilde{C}}_l^t \\
\overrightarrow{o}_l^t &= \sigma\left(\overrightarrow{w}_l^o[\overrightarrow{h}_l^{t-1}, \overrightarrow{h}_{l-1}^t] + \overrightarrow{b}_l^o\right) \\
\overrightarrow{h}_l^t &= \overrightarrow{o}_l^t * \tanh\left(\overrightarrow{C}_l^t\right)
\end{aligned}
\tag{5}
$$

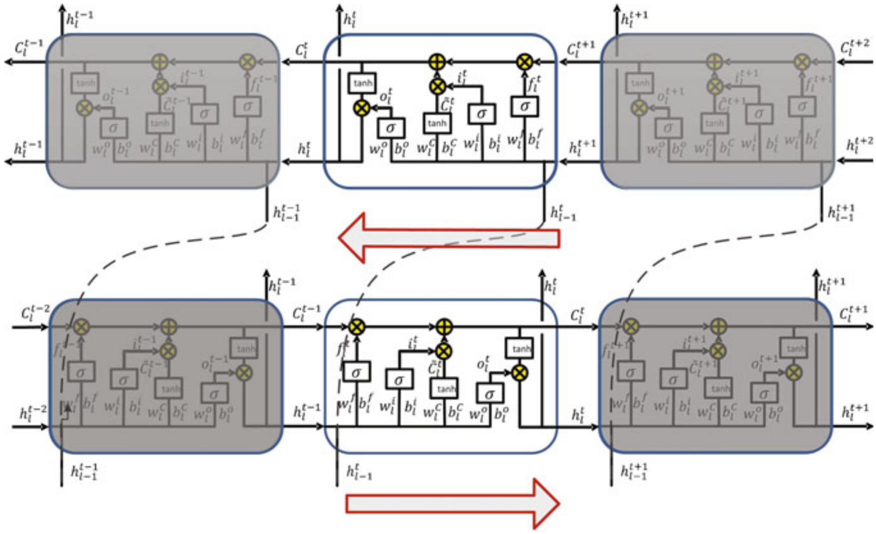


Fig. 3 Structure of DBLSTM

$$
\begin{aligned}
\overleftarrow{f}_l^t &= \sigma\left(\overleftarrow{w}_l^f[\overleftarrow{h}_l^{t+1}, \overleftarrow{h}_{l-1}^t] + \overleftarrow{b}_l^f\right) \\
\overleftarrow{i}_l^t &= \sigma\left(\overleftarrow{w}_l^i[\overleftarrow{h}_l^{t+1}, \overleftarrow{h}_{l-1}^t] + \overleftarrow{b}_l^i\right) \\
\overleftarrow{\tilde{C}}_l^t &= \tanh\left(\overleftarrow{w}_l^C[\overleftarrow{h}_l^{t+1}, \overleftarrow{h}_{l-1}^t] + \overleftarrow{b}_l^C\right) \\
\overleftarrow{C}_l^t &= \overleftarrow{f}_l^t * \overleftarrow{C}_l^{t+1} + \overleftarrow{i}_l^t * \overleftarrow{\tilde{C}}_l^t \\
\overleftarrow{o}_l^t &= \sigma\left(\overleftarrow{w}_l^o[\overleftarrow{h}_l^{t+1}, \overleftarrow{h}_{l-1}^t] + \overleftarrow{b}_l^o\right) \\
\overleftarrow{h}_l^t &= \overleftarrow{o}_l^t * \tanh\left(\overleftarrow{C}_l^t\right)
\end{aligned}
\tag{6}
$$

The final output of the l-th hidden layer at time t of the DBLSTM network is a concatenation of the forward and backward layers, which can be expressed as

$$h_l^t = \overrightarrow{h}_l^t \oplus \overleftarrow{h}_l^t \tag{7}$$

where $\overrightarrow{h}_l^t$ can update the current hidden state by using the past information, that is the time from 1 to $t-1$, and $\overleftarrow{h}_l^t$ can update the current hidden state by using the future information, that is the time from $t+1$ to r.

3.4 Occupancy Inference Layers

The outputs of the DBLSTM network are high-level features which will be fed into some fully connected layers to get more abstract representations. The expression of the fully connected layers can be shown as:

$$\mathbf{o}^i = g\left(\alpha_i \mu^i + \beta_i\right) \tag{8}$$

where μ^i and $\mathbf{o}^i$ are the input and output of the i-th fully connected layer respectively, α_i and β_i are the weights and bias respectively, and $g(\cdot)$ is the activation function. We choose the activation function of ReLU in this study. Suppose that we have stacked c fully connected layers, the output of the last fully connected layer, known as $\mathbf{o}^{c-1}$, is the final representation of the input data. The final feature representations are fed into a softmax classification layer to obtain the occupancy.

3.5 Training Process of the CDBLSTM

With the outputs of the CDBLSTM and the true labels (occupancy ranges), the errors can be calculated over all the training data, and then error gradients will be derived and back-propagated to adjust model parameters for the training of CDBLSTM [37]. More precisely, given training data with the true occupancy levels, the network outputs can be calculated. Then, the cross-entropy losses can be derived based on the network outputs and true occupancy levels. Next, we can get the error gradients to back-propagate for the adjustment of model parameters via some gradient based optimization algorithms. In this study, we adopt the popular optimization method of RMSprop [35]. Precisely, given θ_t the parameter for optimization, and $L(\theta_t)$ the loss function, the parameter update of θ_{t+1} by using the optimization method of RMSprop can be calculated as:

$$g_{t+1} = \gamma g_t + (1-\gamma)\nabla L(\theta_t)^2 \tag{9}$$

$$\theta_{t+1} = \theta_t - \frac{\eta \nabla L(\theta_t)}{\sqrt{g_{t+1}} + \epsilon} \tag{10}$$

where g_t is a moving average of the squared gradient at time step t, and the learning rate η, the parameter γ and the decaying rate ϵ are chosen to be 0.001, 0.9 and 0, respectively.

In order to alleviate the overfitting problem, we use the technique of dropout. By using dropout, we will randomly mask parts of the hidden nodes with probability p during training. Figure 4 illustrate the operation of dropout. During model training, a thinned architecture will be preserved and trained each time. Given a network containing n nodes with a dropout probability of p equaling to 0.5, the network could be treated as an ensemble of 2^n thinned networks. Due to the shared structure

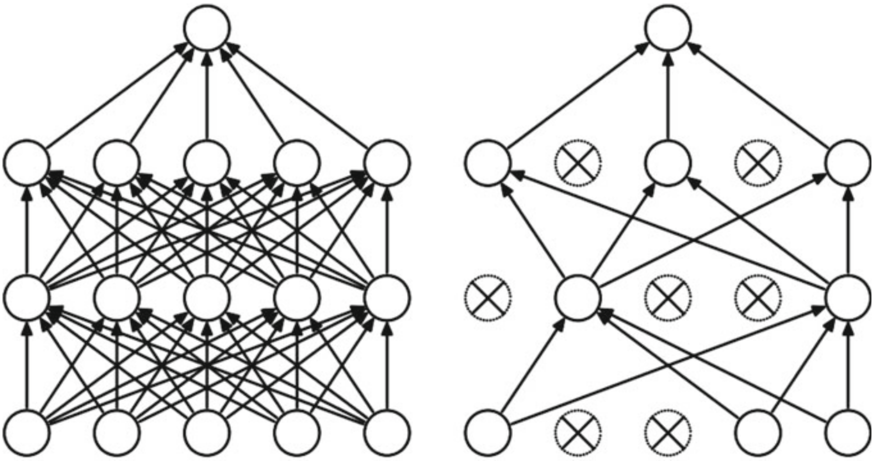


Fig. 4 The operation of dropout. **Left**: the network without dropout; **Right**: the network after dropout. Crossed nodes have been dropped during model training [33]

of these thinned networks, the number of parameters will remain the same. During testing, the dropout will be switched off and all the network nodes will take effect for model outputs, which is similar to an ensemble of some distinct thinned networks. In other words, the dropout is used to enlarge training data size. In each training iteration, random masking will also create some variants into data, which will make the trained network more robust. The dropout technique has been shown to be effective for preventing Overfitting [33]. Therefore, in this study, we leverage on one dropout layer between the DBLSTM and the first fully-connected layer and another dropout layer between the two fully connected layers, where the masking probabilities are chosen to be 0.5 and 0.3 respectively.

4 Evaluation Results

In this section, we firstly introduce the data acquisition process. Then, evaluation setup and experimental results are presented. After that, the generalization performance of the CDBLSTM is analyzed by randomly selecting the data for training and testing. Finally, to further demonstrate the performance of CDBLSTM for building occupancy inference using environmental sensors, we demonstrate additional results of the CDBLSTM using data collected from another environment, i.e., a tutorial room.

4.1 Data Collection

The sensor data of CO_2, temperature, air pressure and humidity have been collected from a research lab at a university campus. The lab has an office area which contains 24 cubicles and 11 open seats. Generally, nine postgraduate students and eleven research staffs will work at the office area. Besides, the lab also has six PCs for undergraduate students on their final year projects and five PCs for other students. It is well known that identifying the exact occupancy (number) is very challenging and may require to use some high-cost sensors in a crowded space. Here, instead of estimating the exact occupancy, we divide the exact occupancy into ranges of zero, low, medium and high. These occupancy ranges are enough for common building control and scheduling systems [18]. To make the four ranges balanced, which will maximize the impact of state changes, we define the low occupancy as 1–6 subjects, the medium occupancy as 7–14 subjects, and the high occupancy as larger than 14 subjects.

We measure pressure level by leveraging on Lutron MHB-382SD sensor, and CO_2, temperature, and relative humidity by using the CL11 sensor from Rotronic. The sampling frequency is one sample per minute for both sensors. During data collection, we firstly stored the data in the sensor internal memory and then transmitted to a PC by using a USB cable. Note that, the area is air-conditioned by the conventional Variable Air Volume and Active Chilled Beam systems, and is ventilated by Air Handling Unit (AHU) that will constantly provide fresh air.

Table 1 shows the accuracy and resolution of the sensors. During experiments, we attach the sensors on supporters with a height of 1.1 m from the ground. Figure 5 illustrates the layout of the apace which has a size of $20\,\text{m} \times 9.3\,\text{m} \times 2.6\,\text{m}$. We apply two pairs of sensors in this space. Here, the placements of sensors are intuitively selected considering occupant density. To get ground truth occupancy, we deploy three IP cameras at each door to record occupant movements. Then, the true occupancy is counted manually with the help of motion detection software which is able to take pictures when occupants move. The entire space contains three doors. The main door (placement of camera 1) connects the space with the office area for administrative staffs. Another door which locates at camera 2 in Fig. 5 opens to a lab space. And the third door is always closed. Note that, all windows are closed, due to the operation of air-conditioning and ventilation systems.

Totally, we collected 31 days of data in workdays, where the first 26 days of data are utilized for model training and the rest 5 days of data is utilized for model testing. Since building control systems are with slow response, a resolution of 15-min is enough for occupancy estimation [39]. But the original sensor data and occupancy have a resolution of 1 min, we firstly transfer them into a 15-min resolution by using the simple averaging. Note that, the number of occupants are an integer value, so that a rounding operation is conducted after the use of averaging on original occupancy.

Table 1 The accuracy and resolution the sensors

Sensor	Environmental parameter	Resolution	Accuracy
Rotronic CL11	CO_2	1 ppm	±5% of the measured value
	Temperature	0.05 °C	±0.3 °K
	Humidity	0.1% RH	<2.5% RH
Lutron MHB-382SD	Pressure	0.1 hPa	±2 hPa

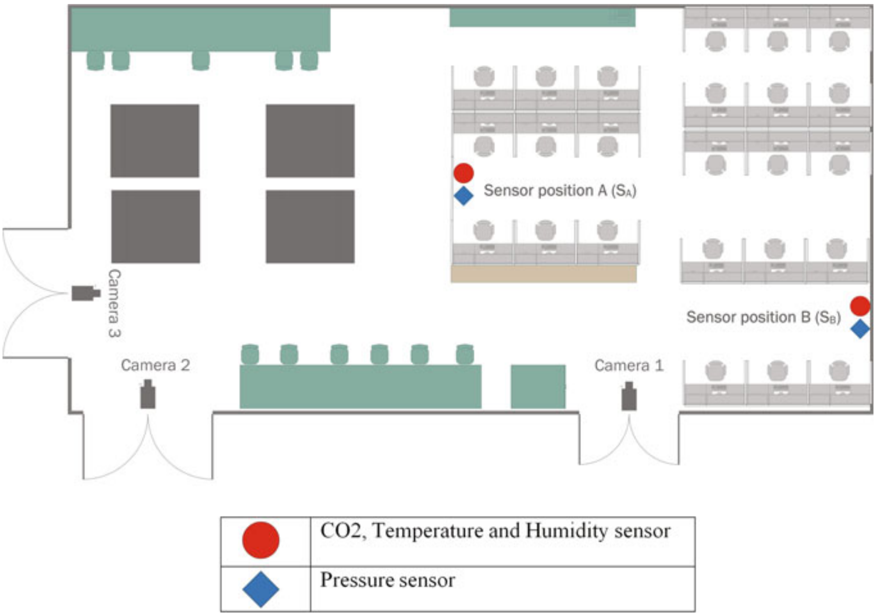


Fig. 5 Layout of the research lab

4.2 Evaluation Setup

To evaluate the performance of CDBLSTM, a comparison has been made between the CDBLSTM and some state-of-the-arts including the HMM approach with the information gain based feature selection of some statistical handcrafted features (Dong's method) [13], the DT with raw data for features (Yang's method) [38], the ELM with the wrapper based feature selection of some statistical handcrafted features (Masood's method) [30], and the LDA with raw data for features (Candanedo's method) [3].

The DBLSTM without the convolutional network for local sequential feature extraction is also implemented for comparison. Since we choose the resolution to be 15-min and the sampling frequency of sensors is 1-min, the length of the input sequence r is 15. With 2 pairs of sensors shown in Fig. 5, the total number of sensors d is 8. Hence, the input is with a dimension of 15×8 for environmental sensor based occupancy estimation. We use cross-validation with the training data to choose proper hyperparameters for all the approaches. Specifically, the DBLSTM consists of three BLSTM layers with hidden nodes of 24, 75 and 100. Then, two fully connected layers with hidden nodes of 150 and 100 are adopted. For the CDBLSTM approach, the window size, the pooling size and the number of filters are chosen to be 3, 2, 100, respectively. The CDBLSTM contains three BLSTM layers with hidden size to be 100, 150 and 200. The two fully-connected layers have 200 and 300 hidden nodes. The implementation of the deep algorithms, i.e., CDBLSTM and DBLSTM, is under Keras. The other shallow algorithms are performed using Matlab.

Here, occupancy estimation is regarded as a typical classification problem. Hence, the criterion of classification accuracy can be adopted for model performance evaluation. Besides, we use another widely used evaluation criterion of Normalized Root Mean Square Error (NRMSE) which will show the range of classification errors [38]. As we all know, the absence and presence are of great significance for building control systems, especially the light control system [32], the detection accuracy of the two states is also analyzed.

4.3 Evaluation Results

The evaluation results for different methodologies under the defined three evaluation criteria are shown in Table 2. Candanedo's and Yang's approaches which applied the raw data as features performs the worst. Note that Candanedo et al. [3] and Yang et al. [38] used many sensors in their works to guarantee the satisfactory performance, which is not practical due to the high cost and the inconvenience caused by constant maintenance. Masood's and Dong's approaches performs better than Candanedo's and Yang's approaches, due to the use of statistical features instead of raw data for features. These results clearly show that feature extraction is compulsory and useful, especially with limited sensors. Since Masood's and Dong's methods used

Table 2 The Evaluation results of different methods under the three evaluation criteria. P/A represents Presence/Absence

Criterion	Dong's [13]	Yang's [38]	Masood's [30]	Candanedo's [3]	DBLSTM	CDBLSTM
Classification accuracy (%)	71.46	66.67	72.31	70.21	74.38	**76.04**
NRMSE	0.1912	0.2509	0.2322	0.2297	0.1574	**0.1169**
Detection accuracy of P/A (%)	93.13	90.21	92.38	88.54	95.21	**95.42**

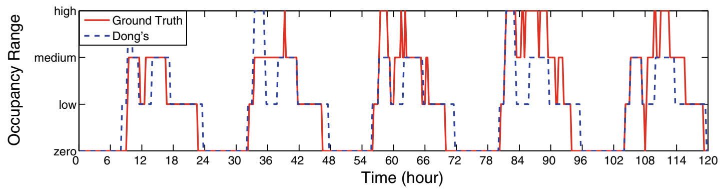
manually extracted features which will inevitably miss useful and implicit features, the performances of these methods are also limited for environmental sensor based human activity recognition.

Owing to the deep structures for feature learning and temporal encoding of the DBLSTM approach, it is able to perform better than all the state-of-the-arts under these three evaluation criteria. With the powerful local feature extractor fulfilled by the convolutional network, the CDBLSTM further enhance the performance of DBLSTM. It outperforms all the approaches where the occupancy estimation accuracy, the NRMSE and the detection accuracy are 76.04%, 0.1169 and 95.42%, respectively.

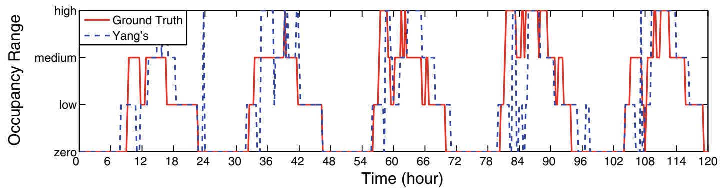
We also illustrate the occupancy estimation results of all the testing days in Fig. 6, where useful insights can be concluded:

– Candanedo's and Yang's approaches perform worse than other approaches, due to the use of raw data as features. With sensor noise and limited number of sensors, the raw sensor data is not representative for different occupancy levels. The more efficient way is to extract some representative features.
– Since Masood's exhaustively searches the best integration of features with the proposed wrapper method, it overfits on the testing data. Similarly, Dong's method also cannot track occupancy profiles well with the handcrafted features. It can be concluded that handcrafted features lack a clear guideline and will inevitably miss useful and implicit features, which limited the system performance.
– One interesting phenomenon is that the estimated occupancy suddenly increases at midnight for Candanedo's, Masood's and Yang's approaches. By checking the data carefully, it should be caused by a sudden increase of CO_2 data. Then, the recorded video was checked, and we find that one subject siting near a pair of sensors usually walks around to prepare for leaving at that time. The optimal locations sensors will be considered as one of our future works [20]. Due to the sequential modeling capacity of HMM and the BLSTM structure, Dong's approach, DBLSTM and CDBLSTM can almost immune to this issue caused by the increase of CO_2 data.
– With the deep structure for feature learning and the BLSTM network for temporal encoding, the DBLSTM and CDBLSTM approaches outperforms all the state-of-the-arts.
– Owing to the convolutional network for local feature extraction, the CDBLSTM further enhances the performance of DBLSTM, and its better performance over all methodologies indicates the effectiveness of using CDBLSTM for building occupancy inference based on environmental sensors.

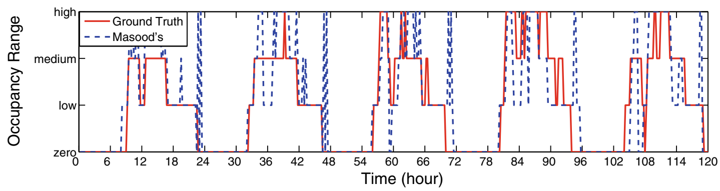
Time complexity is a big concern about deep learning based methods. To show the time complexity of the CDBLSTM, we tested its training and testing time during experiments. Here, the state-of-the-art algorithms all based on manual feature extraction and conventional machine learning algorithms have much smaller training and testing time when compared with CDBLSTM. The CDBLSTM is implemented with a computer which has dual core CPUs of Intel Xeon(R) E5-2697 v2 2.70 GHz and a GPU of NVIDIA Tesla K40c. Its training time is about 16 min and 40 s. Although



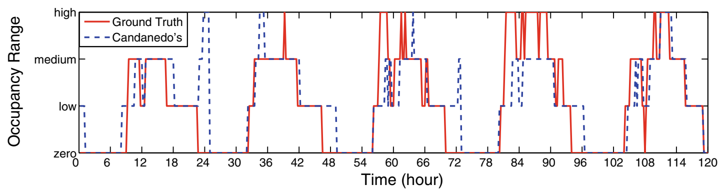
(a) Dong's approach



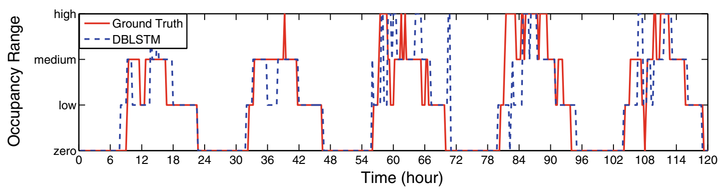
(b) Yang's approach



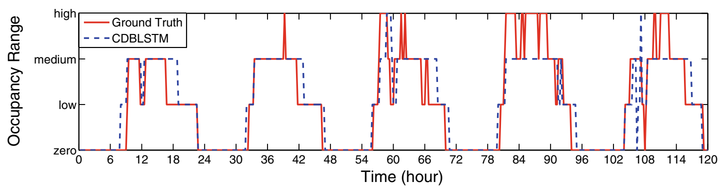
(c) Masood's approach



(d) Candanedo's approach



(e) DBLSTM



(f) CDBLSTM

Fig. 6 The evaluation results of the testing data for all the methodologies [11] this amount of time for training is large, it is still acceptable because the training only requires to be done once in offline. The testing time of the CDBLSTM for all the samples (480 samples) is 0.35 s. This can be neglected for building control systems with a resolution of 15 min. Hence, we can conclude that the CDBLSTM method can be used for real-time occupancy estimation with environmental sensors.

4.4 HyperParameters

Some hyperparameters are crucial for the CDBLSTM approach. Here, the parameters of the masking probabilities of the two dropout layers and the number of hidden layers are investigated. We explored three masking probability levels, including high (0.7), medium (0.5) and low (0.3). Figure 7 demonstrates the occupancy estimation accuracy of the CDBLSTM with different combinations of masking probability. We can find that the CDBLSTM may underfit with a degraded performance when high masking probabilities, such as the combinations of [0.7 0.7], [0.7 0.5], [0.5 0.7] and [0.5 0.5] are used. It is clear that a good selection of this hyperparameter will enhance the performance of CDBLSTM. The number of hidden layers is another key hyperparameter for the model. The estimation performance of the model with distinct number of hidden layers is shown in Fig. 8. When the number of hidden layers increases from 1 to 3, the model performance improves. But, if the number of hidden layers is larger than 4 in this study, the model may overfit, resulting a limited performance.

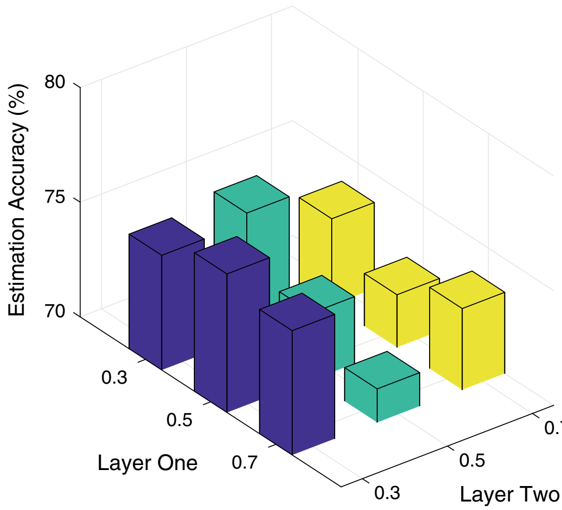
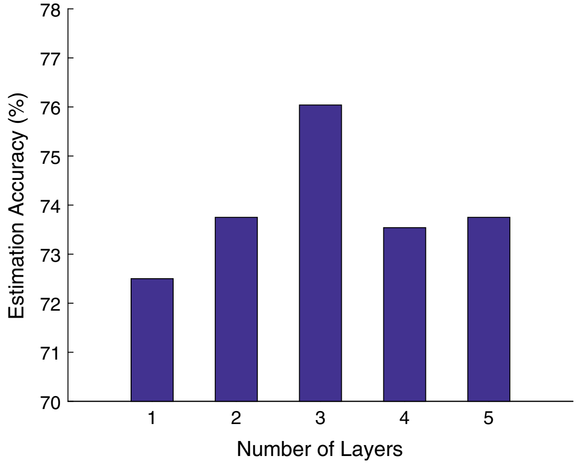


Fig. 7 Occupancy Estimation performance of the CDBLSTM with different combinations of masking probability

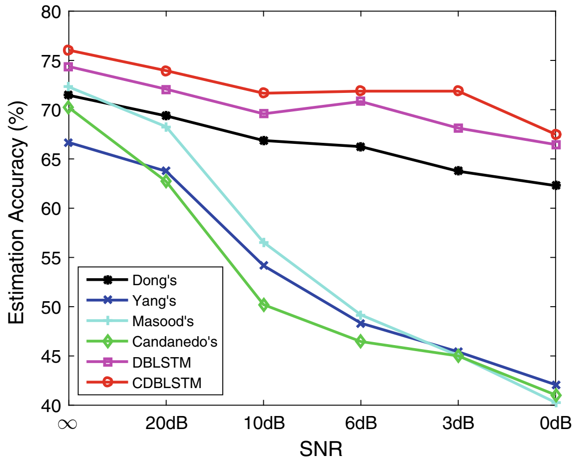
Fig. 8 Estimation performance of CDBLSTM with varying number of hidden layers



4.5 *The Impact of Noise*

The CDBLSTM approach is able to almost immune to some abnormal and noisy data as analyzed in Sect. 4.3, due to its ability to consider temporal dependencies in data. In order to explore the robustness of CDBLSTM on noise data, we manually include some noise into the raw sensor data. Figure 9 presents the performance of all the approaches with different noise levels. Note that the signal to noise ratio (SNR) is ∞ when no noise is added. When the SNR decreases (noisier), the performance of all the approaches degrade accordingly. Due to the capability of modeling temporal dependencies in data, the noise impact on the HMM model (Dong's), DBLSTM

Fig. 9 Estimation performance with varying SNR



and CDBLSTM is smaller, which is consistent with the previous conclusion. The evaluation manifests that the CDBLSTM approach is robust against the noise in data.

4.6 Generalization Performance

In order to verify the generalization performance of the CDBLSTM method, additional experiments are conducted. Specifically, we randomly select five days of data

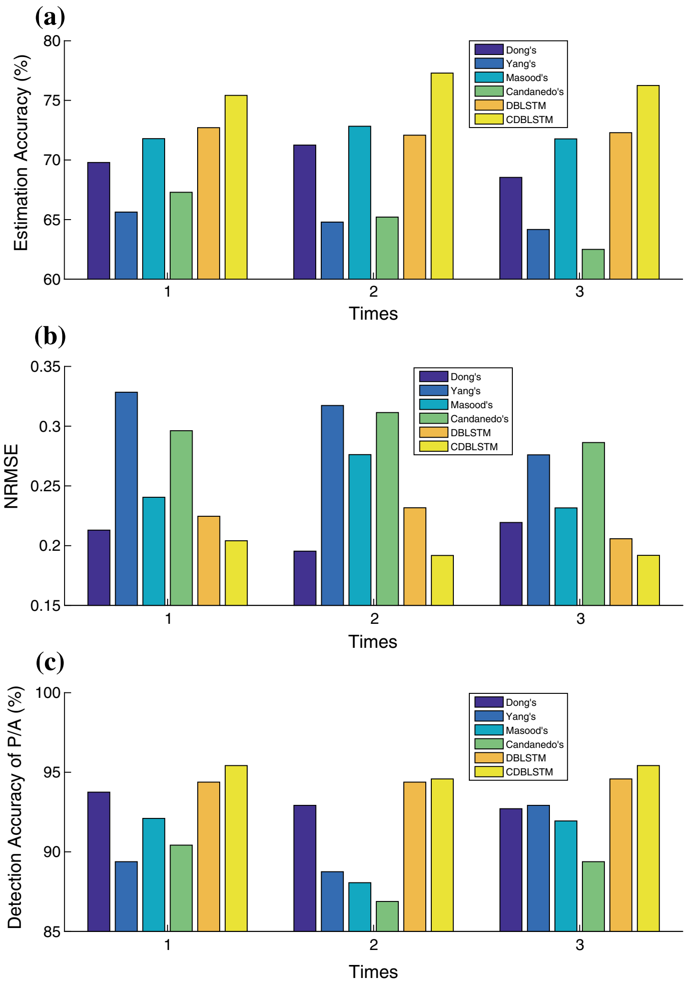


Fig. 10 The evaluation results for the analysis of generalization performance **a** estimation accuracy, **b** NRMSE and **c** detection accuracy of P/A

for model testing and the rest for training. Note that, each day of data have equal probability to be chosen as training or testing, that guarantees the indication of the generalization capability of the CDBLSTM approach. We performed three times for the experiments. Figure 10 shows the final results. It can be found that the DBLSTM approach has a better performance than the state-of-the-arts, and CDBLSTM performs the best under the three evaluation criteria. The conclusions are the same as the previous analysis. This clearly manifests the good generalization performance of the CDBLSTM method for environmental sensor based occupancy detection and estimation.

4.7 Additional Evaluation with Data from Another Environment

To further evaluate the performance of the CDBLSTM, we perform an additional experiment with the data collected from a tutorial room. Totally, we collected fourteen workdays of data for evaluation, where we randomly choose eleven days of data for training and the rest for testing. A more comprehensive illustration of data is presented in [30]. The evaluation results of all the approaches is shown in Table 3. It can be found that all the approaches perform worse in this scenario. The reason is that we only deployed one pair of sensors in this large environment. To enhance the performance, more sensors should be deployed. In this evaluation, we can get the same conclusion. The DBLSTM outperforms all the state-of-the-arts. The CDBLSTM performs the best. This further manifests the effectiveness and robustness of the CDBLSTM approach for environmental sensor based building occupancy estimation.

Table 3 Evaluation results in the tutorial room

Criterion	Dong's [13]	Yang's [38]	Masood's [30]	Candanedo's [3]	DBLSTM	CDBLSTM
Estimation accuracy (%)	57.78	54.44	54.22	55.56	58.89	**65.56**
NRMSE	0.3768	0.3201	0.3214	0.3296	0.2676	**0.2383**
Detection accuracy of P/A (%)	70.00	78.89	85.22	78.89	85.56	**87.78**

5 Conclusion

This chapter introduces a deep learning algorithm, termed Convolutional Deep Bidirectional Long Short-Term Memory (CDBLSTM), for environmental sensor based occupancy inference in buildings. The CDBLSTM consists of a convolutional network for sequential local feature extraction from the raw environmental sensor data and a DBLSTM for temporal coding and feature learning. To verify the performance of CDBLSTM, we perform experiments in a research lab environment and compare with some existing approaches and the DBLSTM method without the convolutional operation. The results indicate that DBLSTM outperforms the state-of-the-arts and CDBLSTM has the best performance, which indicates the merits of the convolutional network and the DBLSTM structure for temporal encoding and feature learning. We also test some hyperparameters of the CDBLSTM with a conclusion that a proper selection of model hyperparameters will boost the performance of CDBLSTM. Then, the impact of noise on model performance is evaluated. The results manifests that the CDBLSTM is able to alleviate the noise effect due to its unique structure. After that, we test the generalization performance of the CDBLSTM by randomly selecting data for training and testing. We can obtain the same conclusion in this scenario. Finally, we perform an additional test in a tutorial room. Similarly, the CDBLSTM achieves a superior performance over all the other methodologies.

References

1. Balaji, B., Xu, J., Nwokafor, A., Gupta, R., Agarwal, Y.: Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, p. 17. ACM (2013)
2. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
3. Candanedo, L.M., Feldheim, V.: Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models. Energy Build. **112**, 28–39 (2016)
4. Chen, Z., Jiang, C.: Building occupancy modeling using generative adversarial network. Energy Build. **174**, 372–379 (2018)
5. Chen, Z., Jiang, C., Xie, L.: Building occupancy estimation and detection: a review. Energy Build. (2018)
6. Chen, Z., Masood, M.K., Soh, Y.C.: A fusion framework for occupancy estimation in office buildings based on environmental sensor data. Energy Build. **133**, 790–798 (2016)
7. Chen, Z., Soh, Y.C.: Modeling building occupancy using a novel inhomogeneous Markov chain approach. In: 2014 IEEE International Conference on Automation Science and Engineering (CASE), pp. 1079–1084. IEEE (2014)
8. Chen, Z., Xu, J., Soh, Y.C.: Modeling regular occupancy in commercial buildings using stochastic models. Energy Build. **103**, 216–223 (2015)
9. Chen, Z., Zhang, L., Cao, Z., Guo, J.: Distilling the knowledge from handcrafted features for human activity recognition. IEEE Trans. Ind. Inform. (2018)
10. Chen, Z., Zhang, L., Jiang, C., Cao, Z., Cui, W.: WiFi CSI based passive human activity recognition using attention based BLSTM. IEEE Trans. Mob. Comput. (2018)

11. Chen, Z., Zhao, R., Zhu, Q., Masood, M.K., Soh, Y.C., Mao, K.: Building occupancy estimation with environmental sensors via CDBLSTM. IEEE Trans. Ind. Electron. **64**(12), 9549–9559 (2017)
12. Chen, Z., Zhu, Q., Masood, M.K., Soh, Y.C.: Environmental sensors-based occupancy estimation in buildings via IHMM-MLR. IEEE Trans. Ind. Electron. **13**(5), 2184–2193 (2017)
13. Dong, B., Andrews, B., Lam, K.P., Höynck, M., Zhang, R., Chiou, Y.S., Benitez, D.: An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network. Energy Build. **42**(7), 1038–1046 (2010)
14. Frodl, R., Tille, T.: A high-precision NDIR gas sensor for automotive applications. IEEE Sens. J. **6**(6), 1697–1705 (2006)
15. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: European Conference on Computer Vision, pp. 392–407. Springer (2014)
16. Hinton, G.E.: Learning multiple layers of representation. Trends Cogn. Sci. **11**(10), 428–434 (2007)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
18. Iglesias, F., Palensky, P.: Profile-based control for central domestic hot water distribution. IEEE Trans. Ind. Inform. **10**(1), 697–705 (2014)
19. Jiang, C., Chen, Z., Png, L.C., Bekiroglu, K., Srinivasan, S., Su, R.: Building occupancy detection from carbon-dioxide and motion sensors. In: International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 931–936 (2018)
20. Jiang, C., Chen, Z., Su, R., Soh, Y.C.: Group greedy method for sensor placement. IEEE Trans. Signal Process. **67**(9), 2249–2262 (2019)
21. Jiang, C., Masood, M.K., Soh, Y.C., Li, H.: Indoor occupancy estimation from carbon dioxide concentration. Energy Build. **131**, 132–141 (2016)
22. Jin, M., Jia, R., Spanos, C.J.: Virtual occupancy sensing: using smart meters to indicate your presence. IEEE Trans. Mob. Comput. **16**(11), 3264–3277 (2017)
23. Labeodan, T., Zeiler, W., Boxem, G., Zhao, Y.: Occupancy measurement in commercial office buildings for demand-driven control applications a survey and detection system evaluation. Energy Build. **93**, 303–314 (2015)
24. Leephakpreeda, T.: Adaptive occupancy-based lighting control via grey prediction. Build. Environ. **40**(7), 881–886 (2005)
25. Li, N., Calis, G., Becerik-Gerber, B.: Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations. Autom. Constr. **24**, 89–99 (2012)
26. Liu, H., Motoda, H.: Feature Extraction, Construction and Selection: A Data Mining Perspective. Springer Science & Business Media (1998)
27. Liu, P., Nguang, S.K., Partridge, A.: Occupancy inference using pyroelectric infrared sensors through hidden Markov models. IEEE Sens. J. **16**(4), 1062–1068 (2016)
28. Mantovani, G., Ferrarini, L.: Temperature control of a commercial building with model predictive control techniques. IEEE Trans. Ind. Electron. **62**(4), 2651–2660 (2015)
29. Masood, M.K., Jiang, C., Soh, Y.C.: A novel feature selection framework with hybrid feature-scaled extreme learning machine (HFS-ELM) for indoor occupancy estimation. Energy Build. **158**, 1139–1151 (2018)
30. Masood, M.K., Soh, Y.C., Chang, V.W.C.: Real-time occupancy estimation using environmental parameters. In: 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2015)
31. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814 (2010)
32. Parise, G., Martirano, L., Cecchini, G.: Design and energetic analysis of an advanced control upgrading existing lighting systems. IEEE Trans. Ind. Appl. **50**(2), 1338–1347 (2014)
33. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

34. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Thirteenth Annual Conference of the International Speech Communication Association (2012)
35. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw. Mach. Learn. **4**(2) (2012)
36. Tran, D., Tan, Y.K.: Sensorless illumination control of a networked led-lighting system using feedforward neural network. IEEE Trans. Ind. Electron. **61**(4), 2113–2121 (2014)
37. Williams, D., Hinton, G.: Learning representations by back-propagating errors. Nature **323**(6088), 533–538 (1986)
38. Yang, Z., Li, N., Becerik-Gerber, B., Orosz, M.: A systematic approach to occupancy modeling in ambient sensor-rich buildings. Simulation **90**(8), 960–977 (2014)
39. Yu, Z., Jia, L., Murphy-Hoye, M.C., Pratt, A., Tong, L.: Modeling and stochastic control for home energy management. IEEE Trans. Smart Grid **4**(4), 2244–2255 (2013)
40. Zhu, Q., Chen, Z., Masood, M.K., Soh, Y.C.: Occupancy estimation with environmental sensing via non-iterative LRF feature learning in time and frequency domains. Energy Build. **141**, 125–133 (2017)
41. Zimmermann, L., Weigel, R., Fischer, G.: Fusion of nonintrusive environmental sensors for occupancy detection in smart homes. IEEE Internet Things J. **5**(4), 2343–2352 (2018)
42. Zou, J., Zhao, Q., Yang, W., Wang, F.: Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation. Energy Build. **152**, 385–398 (2017)