

# Prediction of Indoor Air Temperature Based on Deep Learning

Jing Jin, Shaolong Shu,\* and Feng Lin

School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

(Received January 10, 2019; accepted May 17, 2019)

**Keywords:** prediction, smart home, deep learning, thermal comfort, indoor air temperature

Currently, to achieve the optimal thermal comfort for a given occupant, the optimal indoor air temperature should be set by the occupant himself/herself. Then the air conditioner can be used to control the actual indoor air temperature so that it converges to the optimal temperature. In this study, we develop a method for predicting the optimal indoor air temperature; thereby, the air conditioner can be adjusted automatically without the involvement of the occupant. We first apply the predicted mean vote (PMV) model to describe the relationship between the indoor air temperature and the occupant's thermal comfort. We then adopt the deep learning method to obtain two deep neural network models, which are used to predict the optimal indoor air temperature. One is the regression model and the other is the classification model. We test these models and the results show that the mean average error is about 0.1 °C, which satisfies the accuracy requirements of practical systems.

## 1. Introduction

In recent years, with the development of information technologies such as Internet of Things, cloud computing, and artificial intelligence, the smart home industry has made rapid advances and has a promising future.<sup>(1)</sup> Strategy Analytics reported that the smart home market is expected to reach \$62 billion by 2020.<sup>(2)</sup>

At present, in smart home systems, many appliances such as electric curtains are intelligent enough to be controlled remotely by occupants via mobile apps. Although people can be freed from the daily switching actions to control appliances with the help of these mobile apps, they still need to make decisions on the actions of the appliances. Hence, the research on using a machine to learn the occupants' habits and preferences through collected data and then to make decisions for the occupants becomes very meaningful.

It is reported that human beings spend 80% of their time in their houses. A comfortable indoor environment can greatly improve the quality of life and make people healthier. One important way to improve indoor comfort is to adjust the indoor air temperature via an air conditioner to obtain optimal thermal comfort. One important issue is how to determine the optimal indoor air temperature. There exist some works on this topic. Zhang developed a thermal comfort model with the BP neural network, which can be used to predict indoor thermal

---

\*Corresponding author: e-mail: shushaolong@tongji.edu.cn  
<https://doi.org/10.18494/SAM.2019.2290>

comfort.<sup>(3)</sup> She further proposed an improved particle swarm optimization (PSO) algorithm to optimize the thermal comfort model. Cao also used a neural network to construct a thermal comfort model and proposed a fuzzy controller.<sup>(4)</sup> With this thermal comfort model, he finished the simulation work to verify the effectiveness of the fuzzy controller, using the predicted mean vote (PMV) to evaluate the performance of the closed-loop control systems.

Thermal comfort is a subjective evaluation index. In the same environment, different people have different comfort experiences. Currently, there has been no work on thermal comfort for personalized demands of a given occupant in a smart home. Therefore, in this work, we consider this problem and propose a deep-learning-based method to establish a thermal comfort model for a given individual occupant. The model is used to predict the indoor air temperature expected by the occupant and then the air conditioner is regulated to achieve the optimal indoor air temperature.

## 2. Problem Statement

The existing results show that thermal comfort is mainly determined by the following factors: environmental factors, including air temperature, air relative humidity, air speed, and mean radiant temperature, and personal factors, including metabolic rate and clothing insulation. As we all know, the PMV was developed by Povl Ole Fanger to model the average thermal comfort of a large group of people on the seven-point scale of thermal sensation, as shown in Fig. 1.<sup>(5)</sup> The PMV model is often used to describe the relationship between these factors and the thermal comfort as

$$PMV = f(M, t_a, t_r, I_{cl}, v_a, \varphi), \quad (1)$$

where  $M$  denotes the metabolic rate,  $t_a$  denotes the air temperature,  $t_r$  denotes the mean radiant temperature,  $I_{cl}$  denotes the clothing insulation,  $v_a$  denotes the air speed, and  $\varphi$  denotes the relative humidity.

Among these factors, the air temperature  $t_a$  is the principal factor.<sup>(6)</sup> Therefore, in this paper, we optimize the thermal comfort for a given occupant by adjusting the indoor air temperature via an air conditioner. That is, the absolute value of individual thermal comfort is minimal. When we apply the PMV to model a given occupant's thermal comfort, the model of the given

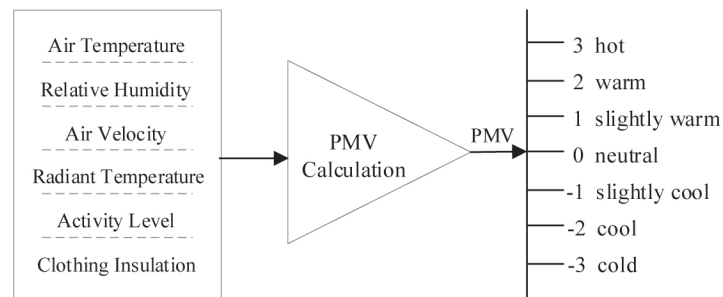


Fig. 1. PMV model.

occupant's thermal comfort should be slightly different from the PMV model and is denoted as

$$PMV_i = f_i(M, t_a, t_r, I_{cl}, v_a, \varphi). \quad (2)$$

The problem that we want to solve is how to determine the optimal indoor air temperature  $t_a^*$  to ensure that the given occupant acquires the optimal thermal comfort. The problem is formally stated as follows.

**Problem 1:** For a given occupant  $i$  and the given metabolic rate  $\bar{M}$ , mean radiant temperature  $\bar{t}_r$ , clothing insulation  $\bar{I}_{cl}$ , air speed  $\bar{v}_a$ , and relative humidity  $\bar{\varphi}$ , find the optimal indoor air temperature  $t_a^*$ , such that the occupant's individual thermal comfort is optimal as follows.

$$\left\{ \begin{array}{l} X = [M, t_a, t_r, I_{cl}, v_a, \varphi] \\ \text{s.t. } M = \bar{M} \\ \quad t_r = \bar{t}_r \\ \quad I_{cl} = \bar{I}_{cl} \\ \quad v_a = \bar{v}_a \\ \quad \varphi = \bar{\varphi} \\ \min |f_i(X)| \end{array} \right. \quad (3)$$

In practical systems, some factors such as metabolic rate and clothing insulation are difficult to determine and thus solving the problem becomes difficult. However, we know that people have different metabolic rates that may fluctuate depending on their activity level and environmental conditions.<sup>(7)</sup> Clothing insulation is mainly determined by the individual clothing situation related to the season, weather, and outdoor temperature. The mean radiant temperature approximately equals the current air temperature in houses.<sup>(8)</sup> In an indoor environment, the air speed is mainly determined by the wind speed of the air conditioner. With such knowledge, we can use the individual activity level  $A_c$ , season  $S_e$ , weather  $W_e$ , outdoor temperature  $t_o$ , current indoor air temperature  $t_i$ , indoor humidity  $H_i$ , and air conditioner wind speed  $W_s$  to indirectly reflect the effects of those factors on thermal comfort as

$$PMV_i = g_i(A_c, S_e, W_e, t_o, t_i, H_i, W_s, t_a). \quad (4)$$

Problem 1 is then translated to the following optimization problem.

**Problem 2:** For a given occupant  $i$  and the individual activity level  $\bar{A}_c$ , season  $\bar{S}_e$ , weather  $\bar{W}_e$ , outdoor temperature  $\bar{t}_o$ , current indoor air temperature  $\bar{t}_i$ , indoor humidity  $\bar{H}_i$ , and air conditioner wind speed  $\bar{W}_s$ , find the optimal indoor air temperature  $t_a^*$  to make the individual thermal comfort optimal as follows.

$$\left\{ \begin{array}{l} X = [A_c, S_e, W_e, t_o, t_i, H_i, W_s, t_a] \\ \text{s.t. } A_c = \overline{A_c} \\ S_e = \overline{S_e} \\ W_e = \overline{W_e} \\ t_o = \overline{t_o} \\ t_i = \overline{t_i} \\ H_i = \overline{H_i} \\ W_s = \overline{W_s} \\ \min |g_i(X)| \end{array} \right. \quad (5)$$

Problem 2 is still difficult to solve because it is difficult to obtain the formulaic description of the function  $g_i(\bullet)$ .

### 3. Deep Neural Network Model

In order to solve Problem 2, we construct an individual thermal comfort model with deep neural networks. The input of the model is the individual activity level, season, weather, outdoor temperature, current indoor air temperature, indoor humidity, and air conditioner wind speed. The output is the optimal indoor air temperature.

#### 3.1 Data collection

We adopt the supervised learning method to train the neural network model.<sup>(9)</sup> One key task is to collect the labeled data obtained from an office used as a benchmark. The overall layout of the office and the sensor distribution are shown in Fig. 2. The office is divided into a study area and a rest area. In the office, there are human motion sensors, temperature sensors, humidity sensors, a smoke alarm, an illuminance sensor, a door/window sensor, and a video sensor.

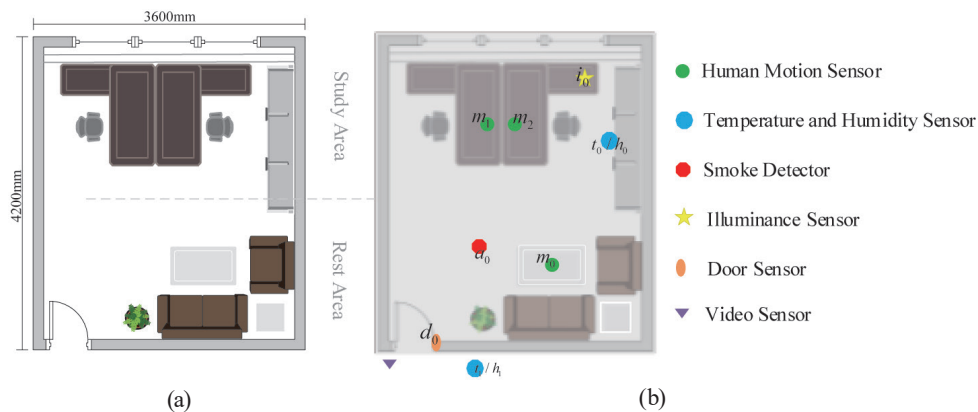


Fig. 2. (Color online) (a) Room layout and (b) sensor map of the benchmark office.

The outdoor temperature is measured using temperature sensor  $t_1$  and the current indoor air temperature is measured using temperature sensor  $t_0$ . The relative humidity in the office is measured using humidity sensor  $h_0$ . The illuminance is measured using illuminance sensor  $i_0$ . The indoor illuminance under different conditions is shown in Table 1. The activity level of the occupant can be determined with reference  $i_0$ . When  $i_0$  is less than 10 lux, the user is in the state of rest. When  $i_0$  is more than 15 lux, the user is in the state of work.

According to the climatic characteristics of Shanghai, spring is from March to May, summer is from June to September, autumn is from October to November, and winter is from December to February. The weather data is obtained from the Internet. The weather has the following four conditions: sunny, cloudy, rainy, and snowy. The air conditioner wind speed data is obtained using a Broadlink Smart Remote Controller with four levels, namely, auto, low, middle, and high.

The output of the model is the optimal indoor air temperature. After learning, the temperature determined by the model will converge to that expected by the occupant so that the model can set the temperature input of the air conditioner automatically.

### 3.2 Data processing

The following are the input variables in the neural network: continuous variables, including outdoor temperature, current indoor air temperature, and indoor humidity, and discrete variables, including season, weather, and individual activity level. For the continuous variables, we use the Z-score standardization method to deal with the sampling data.

$$z = \frac{x - \mu}{\sigma} \quad (6)$$

Here,  $\mu$  and  $\sigma$  are the mean and variance of the original data, respectively,  $x$  is the original data, and  $z$  is the data after normalization.

For these discrete variables, we use the one-hot encoding method to process the sampling data. The results are summarized in Table 2.

There are two different methods to process the output data. The first method is to normalize the output by Z-score standardization. Then, how to formulate an individual thermal comfort model becomes a regression problem. Considering that we set the air conditioner temperature in steps of 1 °C, we can also use the one-hot encoding method to process the output data, as shown in Table 3. Then, how to formulate an individual thermal comfort model becomes a

Table 1  
Indoor illuminance under different conditions.

Curtain	Light source	Ceiling lights	Indoor illuminance (lux)	Meet lighting needs at work
Open	Sunshine	All off	>70	Yes
	None	All off	7	No
Closed	Ceiling light	One on one off	9	No
	Ceiling light	All on	19	Yes

Table 2  
One-hot encoding of discrete input variables.

Variable	One-hot encoding			
Season	Spring 1000	Summer 0100	Autumn 0010	Winter 0001
Weather	Sunny 1000	Windy 0100	Rainy 0010	Snowy 0001
Air conditioner wind speed	Auto 1000	Low 0100	Mid 0010	High 0001
User status	Work 10		Rest 01	

Table 3  
One-hot encoding of discrete output variables.

Variable	One-hot encoding			
	18 °C	19 °C	20 °C	21 °C
	100000000000	010000000000	001000000000	000100000000
Output temperature	22 °C	23 °C	24 °C	25 °C
	000010000000	000001000000	000000100000	000000010000
	26 °C	27 °C	28 °C	29 °C
	000000001000	000000000100	000000000010	000000000001

classification problem. Using the two different processing methods, we will construct the corresponding neural network model.

### 3.3 Model building

In this study, on the basis of the backpropagation algorithm, we adopt the deep neural network<sup>(10,11)</sup> to construct the individual thermal comfort model for predicting the optimal indoor air temperature. A neuron is the basic unit of a neural network as

$$a = f\left(\sum_{i=1}^n \omega_i x_i + b\right), \quad (7)$$

where  $\mathbf{x}$  [ $x_1, x_2, \dots, x_n$ ] is the input vector,  $\boldsymbol{\omega}$  [ $\omega_1, \omega_2, \dots, \omega_n$ ] is the weight vector,  $b$  is the bias, and  $a$  is the output of the neuron.

In the neuron, we call  $f(\mu)$  the activation function.<sup>(12)</sup> Activation functions are used to introduce nonlinearity to neural networks. The popular activation functions include the sigmoid function [ $f(\mu) = 1/(1 + e^{-\mu})$ ], hyperbolic tangent (tanh) function [ $f(\mu) = \tanh(\mu)$ ], and rectified linear unit (ReLU) function [ $f(\mu) = \max(\mu, 0)$ ].

Neural networks are typically organized in layers. Each layer is made up of a number of interconnected neurons. Patterns are presented to the network via the input layer, which communicates to one or more hidden layers where the actual processing is done via a system of weighted connections. The hidden layers link to an output layer that outputs an answer.

The classification neural network model built in this study is shown in Fig. 3. The input is

a 17-dimensional feature vector. The output is a 12-dimensional feature vector representing the one-hot code of the temperature range from 18 to 29 °C as shown in Table 3. The number of hidden layers and the number of neurons in each layer will be determined by the grid search method.<sup>(13)</sup> The feature vector of the input layer of the regression model is the same as that of the classification model. The output of the regression model is a one-dimensional vector representing the expected temperature of the occupant, which is the optimal indoor air temperature. Similarly, the number of hidden layers and the number of neurons in each layer will be determined by the grid search method.

We select one part of the labeled sampling data as a training set to train the deep neural network. Training is the process of adjusting the weights and offsets of the neural network. During the training process, the cost function  $C(\omega, b)$  is used to evaluate the error between the true value and the output value of the neural network. The goal of training is to minimize  $C(\omega, b)$ , which is implemented by the gradient descent method to update the weight and bias iteratively as

$$\omega - \eta \frac{\partial C(\omega, b)}{\partial \omega} \rightarrow \omega', \quad b - \eta \frac{\partial C(\omega, b)}{\partial b} \rightarrow b', \quad (8)$$

where  $\eta$  is the learning rate. The learning rate determines how quickly or how slowly to update the parameters. When the training set is very large, we use the minibatch method<sup>(14)</sup> to accelerate the training process. The minibatch gradient descent is a variation of the gradient descent that splits the training set into small batches to calculate model error and update model coefficients.

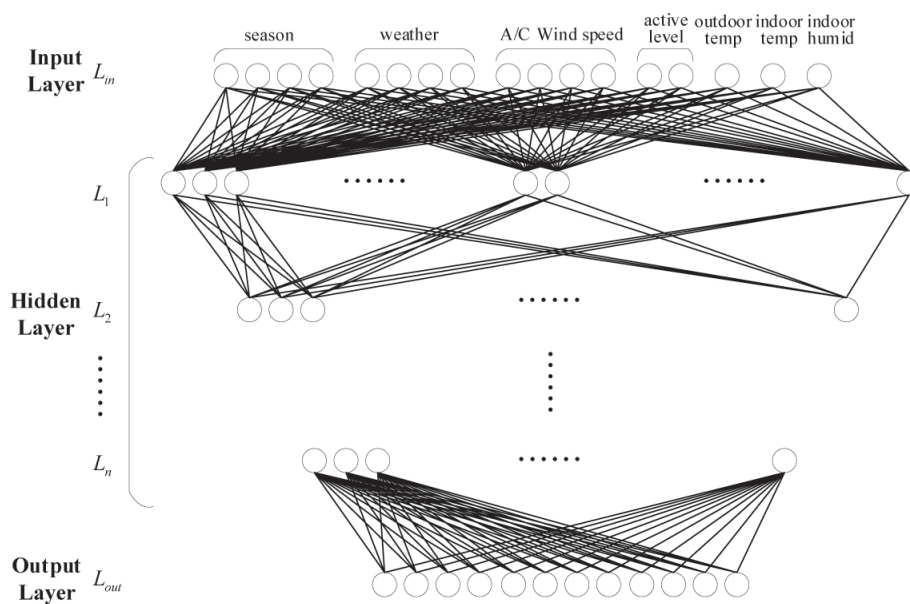


Fig. 3. Classification neural network model.

## 4. Experiment Analysis

We collected 12 days' worth of data from August 3, 2018 to August 14, 2018 at a 1 min sampling interval. The total number of samples is 7305. We randomly divide the data into training set  $S$  (90% of the collected data) and test set  $T$  (10% of the collected data). Before we train a neural network, we need to determine some hyperparameters including the number of layers ( $L$ ) of the neural network, the number of neurons ( $j$ ) in each hidden layer, the size of a minibatch, the number of epochs, the activation function of neurons, and the cost function. Hence, we adopt the method shown in Fig. 4 to complete the experiment.

As shown in Fig. 4, we first find the optimal hyperparameters, including the number of layers ( $L$ ) of the neural network, the number of neurons ( $j$ ) in each hidden layer, the minibatch size, and the number of epochs by the grid search method. With the optimal hyperparameters, we then retrain the model with training set  $S$  to obtain a neural network model. Finally, we test the model with test set  $T$  to verify its effectiveness.

### 4.1 Regression model

Before we perform the experiment, we should determine the activation function of neurons and the cost function. The vanishing gradient problem is one of the major problems for both the sigmoid function and the tanh function. However, the ReLU activation function can overcome this difficulty. Hence, we choose the ReLU activation function for neurons in the input layer and hidden layers of the regression neural network. The regression model does not need to classify the prediction result, so the output layer does not set the activation function and directly outputs the value.

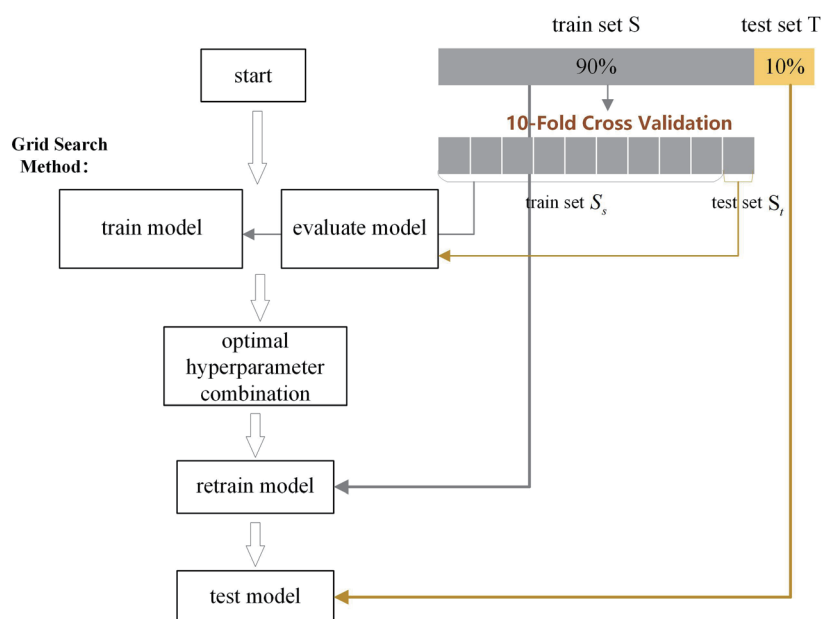


Fig. 4. (Color online) Flow chart of experiment.



We choose the mean square error (MSE) as the cost function.

$$L = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (9)$$

Here,  $y_i$  denotes the true value, that is, the temperature value for the air conditioner set by the given occupant and can be obtained using the Broadlink Smart Remote Controller.  $\hat{y}_i$  denotes the predicted value of the neural network.

Even though minibatch gradient descent takes the best of both batch gradient descent and stochastic gradient descent, it does not guarantee good convergence and it is difficult to choose a proper learning rate.<sup>(15)</sup> There are many improved approaches such as Momentum,<sup>(16)</sup> Adagrad,<sup>(17)</sup> Adadelata,<sup>(18)</sup> RMSprop,<sup>(19)</sup> and Adaptive Moment Estimation (Adam).<sup>(20)</sup> Among these approaches, Adam computes adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients, similarly to Adadelata and RMSprop, Adam also keeps an exponentially decaying average of past gradients similar to momentum. Furthermore, Adam has a relatively high convergence speed. In this study, we choose the Adam algorithm to update the weights and offsets.

We use the grid search method to search for the optimal solution of the number of hidden layers of the model, the number of neurons in each hidden layer, the minibatch size, and the number of epochs. The parameter list is shown in Table 4. The grid search method is used to systematically traverse a variety of parameter combinations and determine the best parameters by cross-validation.<sup>(21)</sup> When the sampling data is insufficient, to make full use of the data set to test the algorithm performance, we divide the training set  $S$  into  $k$  packets, one of which is used as a test set each time, and the remaining  $k - 1$  packets are used as the training set. Here, we set  $k = 10$ .

The results show that when the number of neurons in the hidden layers is [120, 100, 80, 60, 40, 20], the number of epochs is 200, and the batch size is 20, the model is optimal.

The results of the tenfold cross-validation of the models with the optimal hyperparameters are shown in Fig. 5. The blue curve is the loss value (i.e., MSE) of the model on training set  $S_s$ , and the green curve is the loss value on test set  $S_t$ .

The evaluation results (MSEs) of the ten models are shown in Table 5, and the average of the evaluation results is 0.0933.

## 4.2 Classification model

Similarly, we choose the ReLU activation function for neurons in the input layer and hidden layers of the classification neural network. Because the output unit of the classifier requires the

Table 4  
Parameter list of grid search method.

Parameters	List			
No. of neurons	[80, 60, 35, 12]	[100, 80, 60, 40, 20]	[120, 100, 80, 60, 40, 20]	[140, 120, 100, 80, 60, 40, 20]
No. of epochs	100	130	160	200
Batch size	5	12	20	30

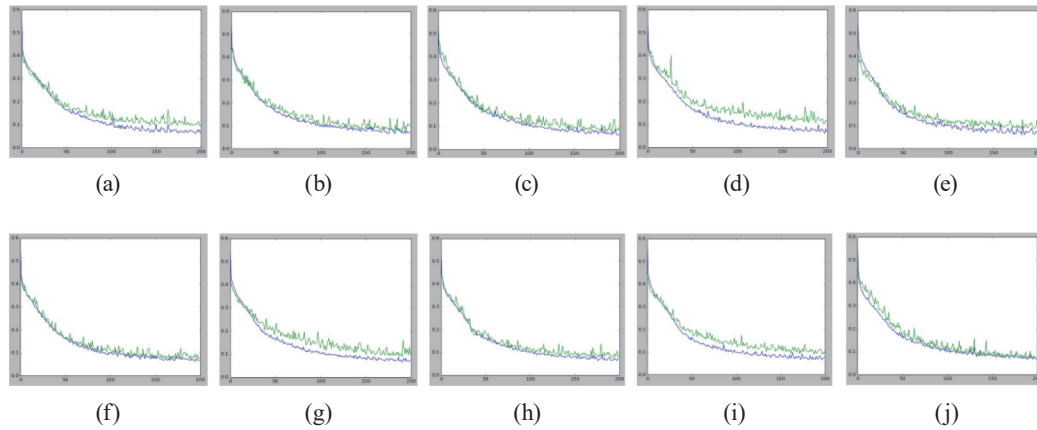


Fig. 5. (Color online) Tenfold cross-validation evaluation results of regression model.

Table 5  
Tenfold cross-validation evaluation results of regression model.

Model	a	b	c	d	e	f	g	h	i	j
Score	0.1002	0.1076	0.0813	0.1127	0.0962	0.0761	0.0998	0.0886	0.1051	0.0655

softmax activation function to perform numerical processing, we choose the softmax activation function for neurons in the output layer as

$$S_i = \frac{e^{V_i}}{\sum_i^C e^{V_i}}, \quad (10)$$

where  $V_i$  denotes the output of the prestige unit of the classifier,  $i$  denotes the category index,  $C$  denotes the total number of categories, and  $S_i$  denotes the ratio of the index of the current element to the exponential sum of all elements. Corresponding to the softmax activation function, we choose the categorical cross entropy as the cost function of the classification model as

$$L = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m t_{i,j} \log(p_{i,j}), \quad (11)$$

where  $n$  is the number of samples,  $m$  is the number of categories that is equal to 12,  $t_i = [t_{i,1}, t_{i,2}, \dots, t_{i,m}]$  is a 12-dimensional one-hot code that represents a true temperature set by the occupant, and  $p_{i,j}$  denotes the prediction value of the  $j$ th output of the neural network.

The results show that when the number of neurons in the hidden layers is [120, 100, 80, 60, 40, 20], the number of epochs is 200, and the batch size is 20, the model is optimal. Different from the regression model, we use accuracy to evaluate the performance of the classification model as

$$accuracy = \frac{TP}{T}, \quad (12)$$

where  $T$  is the number of all the samples and  $TP$  is the number of samples whose output is equal to the output of the neural network.

The results of the tenfold cross-validation of the models with the optimal hyperparameters are shown in Fig. 6. The blue curve is the accuracy of the model on training set  $S_s$ , and the green curve is the accuracy on test set  $S_t$ .

The evaluation results (accuracies) of the ten models are shown in Table 6, and the average of the evaluation results is 0.8751.

### 4.3 Model prediction

With the optimal hyperparameters obtained as described in the above subsections, we train the regression neural network model for 264 s and the classification neural model for 315 s with training set  $S$ , and obtain the thermal comfort model. We use test set  $T$  to evaluate its performance.

The output of the regression model is a decimal value. However, the temperature setting of the air conditioner is an integer. We round off the prediction value of the regression model and take it as the final prediction value. The MSE of the regression model on test set  $T$  is 0.133. The accuracy of the regression model on test set  $T$  is 0.876. The accuracy here only indicates that 87.6% of the predicted values are equal to the true values. In order to evaluate the performance of the model, we calculate the mean absolute error (MAE) of the model as

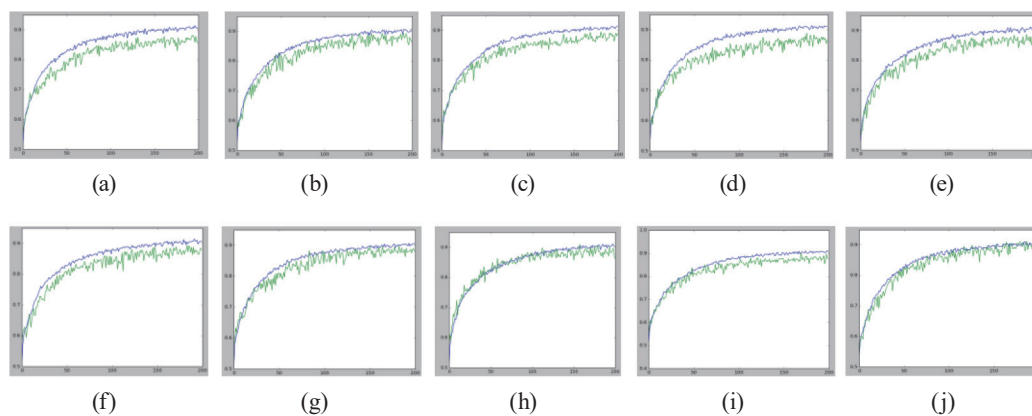


Fig. 6. (Color online) Tenfold cross-validation evaluation results of classification model.

Table 6  
Tenfold cross-validation evaluation results of classification model.

Model	a	b	c	d	e	f	g	h	i	j
Score	0.8556	0.8632	0.8891	0.8663	0.8630	0.8798	0.8798	0.8630	0.8782	0.9132

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (13)$$

where  $n$  denotes the number of samples,  $y_i$  denotes the output value of samples, and  $\hat{y}_i$  denotes the prediction value of the neural network. The MAE of the model on test set  $T$  is 0.127, which means that the average error between the predicted optimal indoor air temperature and the true optimal indoor air temperature is 0.127 °C. This result shows that the model has high prediction accuracy. There are only two samples with an absolute error greater than 1 °C. Therefore the performance of the model is very stable.

From the output of the classification model, we can obtain the prediction value using data in Table 3. The MSE of the classification model on test set  $T$  is 0.138. The accuracy of the classification model on test set  $T$  is 0.882. The MAE of the model on test set  $T$  is 0.124. There are five samples with an absolute error greater than 1 °C. In this work, we extract the sampling data of the last two days, then put them into the regression and classification models. As shown in Figs. 7 and 8, we compare the prediction value with the true value. The blue dots in the figures represent true values, and the red dots represent prediction values. As can be clearly seen from the figures, both models can track well the trend of the occupant's expected indoor air temperature.

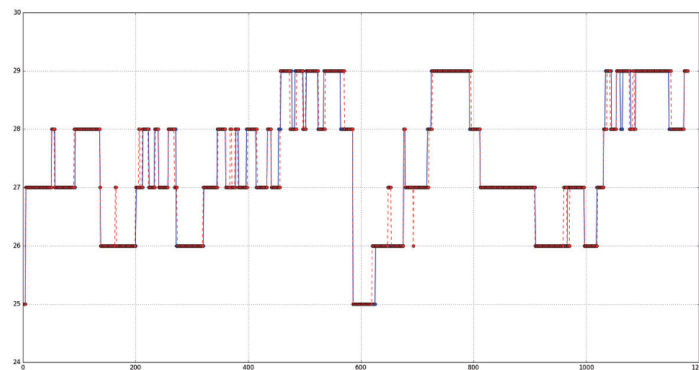


Fig. 7. (Color online) Prediction result of regression model.

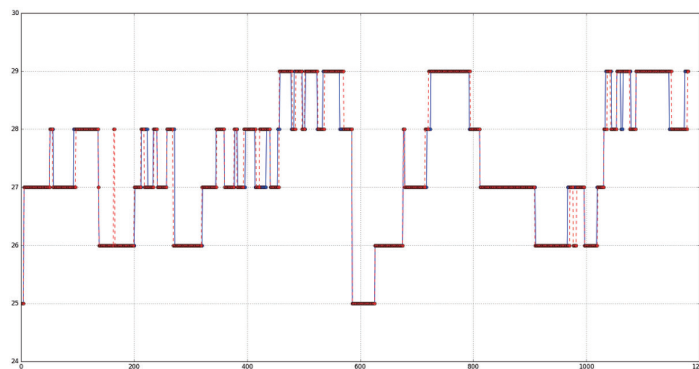


Fig. 8. (Color online) Prediction result of classification model.

## 5. Conclusions

In this study, using the thermal comfort index PMV, we established two deep neural network models to predict the indoor air temperature expected by the occupant. The experimental results show that the MAEs of two models are both less than 0.13 °C. The models obtained can be used to adjust the temperature setting of an air conditioner in real time without human involvement, and finally realize the personalized control of the indoor air temperature.

## Acknowledgments

The work is supported by the Fundamental Research Funds for the Central Universities, the National Science Foundation of China, under Grants 61673297 and 61773287.

## References

- 1 Y. Wang: *Telecommun. Network Technol.* **3** (2018) 15.
- 2 <http://www.cctime.com/html/2016-10-31/1234607.htm>
- 3 L. Zhang: *Research of Thermal Comfort Predict and Control Algorithms of the Air Conditioning*, Hunan University (2014).
- 4 Y. Cao: *The Optimization of Thermal Comfort Prediction and Control Model for Indoor Thermal Environment*, Shenyang University (2016).
- 5 P. O. Fanger and J. Toftum: *Energy Build.* **34** (2012) 533.
- 6 D. L. Zeng, K. J. Li, Y. K. Gao, J. Zhang, and C. Liu: *Chin. Refrig. Air-Conditioning Ind. Assoc.* **6** (2016).
- 7 J. Toftum: *Thermal Comfort Indices*, *Handbook of Human Factors and Ergonomics Methods* (CRC Press, Boca Raton, 2005).
- 8 C. Tanaya, Y. C. Soh, S. Bose, L. Xie, and H. Li: *IECON 2016 42nd Annu. Conf. IEEE Ind. Electron. Soc.* (2016) 7065–7070.
- 9 R. D. Brandt and F. Lin: *Supervised Learning in Neural Networks Without Feedback Network* (IEEE, 1996) 86–90.
- 10 L. C. Yann, B. Yoshua, and H. Geoffrey: *Nature* **521** (2015) 436.
- 11 D. E. Rumelhart, R. E. David, E. H. Geoffrey, and J. W. Ronald: *Nature* **323** (1986) 533.
- 12 Z. H. Zhou: *Machine Learning* (Tsinghua University Press, Beijing, 2016) p. 23.
- 13 Q. J. Huang, J. L. Mao, and Y. Liu: *2012 IEEE 14th Int. Conf. Communication Technology* (Chengdu, 2012) 1022–1026.
- 14 B. Mao, K. Xu, Y. H. Jin, and X. L. Wang: *Chin. J. Comput.* **12** (2018) 55.
- 15 R. D. Brandt and F. Lin: *Inf. Sci.* **121** (1999) 201.
- 16 N. Qian: *Off. J. Int. Neural Network Soc.* **12** (1999) 145.
- 17 J. Duchi, E. Hazan, and Y. Singer: *J. Mach. Learn. Res.* **12** (2011) 2121.
- 18 M. D. Zeiler: *ADADELTA: An Adaptive Learning Rate Method* (2012).
- 19 R. Gylberth, R. Adnan, S. Yazid, and T. Basaruddin: *2017 Int. Conf. Advanced Computer Science and Information Systems* (2017) 387–394.
- 20 D. P. Kingma and J. Ba: *Int. Conf. Learning Representations* (2015) 1–13.
- 21 J. X. Hu and G. J. Zhang: *Bull. Sci. Technol.* **29** (2013) 115.

## About the Authors



**Jing Jin** received her B.E. degree in measurement and control technology and instrumentation programs from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. She is currently working toward her M.E. degree in control engineering at Tongji University, Shanghai, China. Her current research interests include discrete event systems, deep learning, and their applications in smart homes.



**Shaolong Shu** received his B.E. degree in automation and his Ph.D. degree in control science and engineering from Tongji University, Shanghai, China, in 2003 and 2008, respectively. He has been with the School of Electronics and Information Engineering, Tongji University since 2008, where he is currently a professor. From 2007 to 2008 and from 2014 to 2015, he was a visiting scholar with Wayne State University, Detroit, MI, USA. His current research interests include the modeling and control of discrete event systems and cyber-physical systems and their applications in smart homes, smart buildings, and smart cities.



**Feng Lin** received his B.Eng. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1982, and his M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1984 and 1988, respectively. He was a post-doctoral fellow with Harvard University, Cambridge, MA, USA, from 1987 to 1988. Since 1988, he has been with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA, where he is currently a professor. His current research interests include discrete event systems, hybrid systems, robust control, and their applications in alternative energy, biomedical systems, and automotive control. He has authored a book entitled *Robust Control Design: An Optimal Control Approach*. He was a consultant for GM, Detroit, Ford, Dearborn, MI, USA, Hitachi, Tokyo, Japan, and other automobile companies. Dr. Lin coauthored a paper that received the George Axelby Outstanding Paper Award from the IEEE Control Systems Society. He is also a recipient of the Research Initiation Award from the National Science Foundation and the Outstanding Teaching Award from Wayne State University. He was an associate editor of the *IEEE Transactions on Automatic Control*.