Zhou, H., Qiao, L., Jiang, Y., Sun, H., and Chen, Q. 2016. "Recognition of air-conditioner operation from indoor air temperature and relative humidity by a data mining approach," *Energy and Buildings*, 111: 233-241.

Recognition of air-conditioner operation from indoor air temperature and relative humidity by a data mining approach

Hao Zhou[1], Lifeng Qiao[2], Yi Jiang[2], Hejiang Sun[1,*], Qingyan Chen[3,1]

[1] *Tianjin Key Laboratory of Indoor Air Environmental Quality Control, School of Environmental Science and Engineering, Tianjin University, Tianjin 300072, China*

[2]*Intelligent Green Solutions Ltd., Beijing 100084, China*

[3]*School of Mechanical Engineering, Purdue University, 585 Purdue Mall, West Lafayette, IN 47907-2088, U.S.A*

Corresponding author: E-mail: sunhe@tju.edu.cn Tel: +862227403620 Fax: +862227403620

Abstract: The variety of occupant behaviors in buildings have led to a significant mismatch between simulated building energy performance and measured one. It is crucial to collect real occupant behaviors in buildings to achieve accurate simulation purpose, although there exists a great challenge due to the cost of monitoring devices and privacy concerns. This study proposed an inexpensive and minimally intrusive method, to recognize behavior information from environment parameters by data mining approach. To validate this method, experiments were conducted in three bedrooms. Two types of classification algorithms were developed to recognize AC operations from the experiment data of indoor air temperature and relative humidity. Two types of recognition rules were generated from algorithm training in one dataset, and tested in the other datasets. Based on the testing results, the performance of the two algorithms were evaluated and compared. The results indicated that the C4.5 decision tree algorithm was not suitable for mining AC operations, while the curve description algorithm had good performance in processing the time-series curves of air temperature and relative humidity. Through this experiment, it is confirmed that AC operations can be recognized from indoor air temperature and relative humidity by data mining approach. The main contribution of this study is that a promising approach was developed, which is inexpensive and minimally intrusive on gathering and interpreting information about occupants' daily behaviors.

Keywords: Occupant behavior; Air-conditioner operations; Data mining; Classification algorithm; C4.5 decision tree; Curve description; Time-series curve

1. Introduction

Most building-energy simulation programs use fixed, statistically-averaged occupant-behavior patterns (such as fixed time schedule of air conditioning and lighting) to predict the energy requirements of buildings in the design phase [1][2]. However, actual energy-related behaviors may differ greatly among occupants. Li et al. [3] conducted a field survey of summer air-conditioner (AC) usage in 25 apartments within a single building in China, and found that the largest electricity consumption was about 15 kW/m^2 while the smallest was almost 0 kW/m^2. As a result, averaged behavior patterns in simulation have led to a mismatch between simulated and measured building energy performance [4]. Reports suggested that the measured energy use could be 2.5 times as much as the simulated one [5]. Thus, it is essential to learn occupant behaviors in buildings to predict the building energy performance more accurately.

The study of occupant behaviors often follows a three-step method, comprised of monitoring, modelling, and simulation [6]. Monitoring is crucial in the three steps since the amount and quality of field data plays an important role in the development and validation of models and simulation. The tasks of monitoring is to collect occupant behaviors and the related environment data at the same time, although collecting behavioral data in daily life is a big challenge [7]. Firstly, different behaviors require different types of monitoring devices, so it is a big expense for data collection in a large scale. Researchers used smart meters to monitor operations to lights, air-conditioners [8], magnetic proximity sensors for window operations [9], and cameras [10], passive infrared (PIR) sensors [11] for occupancy detection. As most of these devices were not necessary to daily life, researchers had to cover the expense by themselves. Secondly, some monitoring method would raise concerns about privacy (such as cameras), and health [12], and thus could not be used in some privacy space like residential buildings. Therefore, inexpensive and noninvasive approaches are necessary for behavioral data collection.

Generally, occupant behaviors have direct or indirect relationship with the indoor environmental parameters

[13]. For example, turning on/off the air conditioner will directly change the room air temperature and relative humidity; occupant's presentation in the room will directly raise the CO_2 concentration [14], and indirectly increase the noise level by talking [15] or watching TV. In data analytics field, engineers applied data mining algorithms to discover the hidden relationship between patterns and data [16]. This study applied this thinking to occupant behavior learning, and proposed to recognize behavior information from environment parameters by data mining approach. Because this method collected only indoor environment parameters, it will not raise the privacy concern. In addition, wireless monitoring devices for indoor environment are becoming a part of daily life [17] due to people's more attention to the indoor environment quality[18][19], so it will be much easier and cheaper to gather indoor environment data for long-term and in a large scale. Therefore, the key task of this method is to develop appropriate data mining algorithms for behavior recognition from environment data. This paper used air-conditioner operations as an example, to present the development of data mining algorithms through experiments in 3 bedrooms. The objective of this study is to identify the applicability of data mining approach in behavioral collection of AC operations, and to provide a pilot study for the development of algorithm for other behaviors related to light and windows operations, and occupancy, etc.

2. Methodologies

2.1 Experiment setup

The experiment was conducted in three bedrooms during the summer of 2014. During the experiment, all the rooms used split-type air conditioners for cooling. In order to collect indoor environment data as well as the ground truth information of AC operations, the team developed a wireless data collection system, as shown in Fig. 1. With this system, air temperature and relative humidity data was collected by the sensing terminal, transmitted by ZigBee to a WiFi gateway, and then sent to a Central Web server through TCP/IP network for storage, and finally displayed and downloaded on a computer, as is shown in Fig.1 (a). Ground truth information of AC operations was collected through an app installed on occupants' phones, and transmitted to the Central Web server through 2G/3G/WiFi, as is shown in Fig. 1 (b).

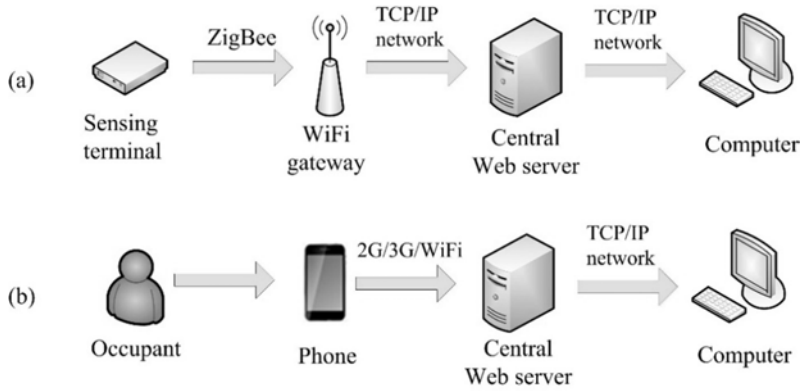


Fig. 1. Architecture of the data collection system: (a) environment data, and (b) "ground truth" behavioral data of AC operations

Note that, "AC operations" herein refers to the actions that occupant manually turned on/off AC. Besides, the actions that the ACs automatically turned off after the occupants had set the AC running time in advance are also included. During the experiment, the AC compressors periodically switched on and off to regulate the room air temperature around the set-point within a very small range of no more than ±0.5K. Such automatic, periodic switching on and off is meaningful for calculation of energy consumption, but it is excluded from the scope of AC operations discussed in this study. Therefore, during the experiment, the participants were asked only to select "Turn on" or "Turn Off" and temperature set-point on their phones when they turned on/off AC. In case that they forgot to record their operations, we checked the environment data as well as behavior records every week. If we found that the parameters presented similar characteristics with those under some operation but there was no records, we would confirm with the occupants whether they had forgotten to log the actions.

Details about the data collection during this experiment is shown in Table 1, where the three bedrooms are labeled as Rooms A, B, and C.

Table 1. Data collection details

Item	Details		
Environment data	Indoor air temperature and relative humidity		
Sampling frequency	5 minutes		
Behavioral data	AC operation ("Turn on," "Turn off"), temperature set point, and time stamps		
Room	A	B	C
Measurement period	Jun. 29 – Aug. 30, 2014	Jul. 6 – Aug. 15, 2014	Jul.28 – Aug. 20, 2014

2.2 Recognition algorithms

Recognizing AC operations from indoor environment data is actually to distinguish periods when AC is running from periods when AC is off, so it can be considered as a classification problem. Classification algorithms used in previous studies includes C4.5 decision tree [21], naive Bayes classifier [22], neural networks [23], and support vector machine [24]. Because C4.5 decision tree classifier (C4.5) is simple and has good performance, it becomes the standard against which new algorithms are judged [25]. Therefore, this study chose the C4.5 as one of the recognition algorithms to be developed and compared.

As C4.5 is a statistical-type classifier [26], it investigates each sampling point as an individual, regardless of the time dependence among individuals. However, most environmental parameters have good time continuity, so they are often displayed as time-series curves. Therefore, a contextual curve classifier, the curve description algorithm (CD) [27], was introduced to conduction behavioral recognition in this study.

2.2.1 C4.5 decision tree algorithm

C4.5 uses a decision tree to display its recognition rules [28]. The tree is comprised of a root node layer, leaf node layers, an outcome layer, and branches, as is shown in Fig. 2, where root node and leaf nodes are features; outcomes are possible classes; and branches are used to connect one node to another. Branches along the path from root to each outcome generate the recognition rules in the "if" clause as follows: "If Branch 1, and Branch 2… and Branch n, then possible outcome."

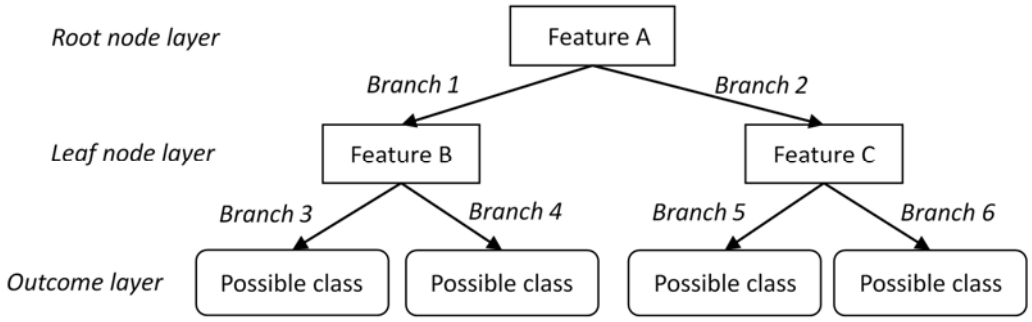


Fig. 2. Decision tree built in C4.5

In the tree, a feature is an individual measurable property of a phenomenon being observed. For example, change rate can be a feature of any time-dependent parameter. Choosing informative, discriminating and independent features is a crucial step for effective classification, namely, for the tree growth in C4.5. C4.5 uses the Information Gain Ratio (IGR) to quantize the importance of features, and then select. According to information theory [29], IGR is calculated by the following equations.

Assuming that *S* is a dataset of |*S*| samples with m classes, the information entropy of *S* is defined as:

$$H(S) = \sum_{i=1}^{m} -p(C_i)\log_2 p(C_i) \tag{1}$$

where $p(C_i)$ is the proportion of samples in Class C_i. When Feature A splits the Dataset *S* into n subsets (branches), the information gain from Feature A is:

$$Gain(S,A) = H(S) - I(S,A) = H(S) - \sum_{j=1}^{n} \frac{|S_j|}{|S|} \bullet H(S_j) \tag{2}$$

The intrinsic information for this split is defined as:

$$IntI(S,A) = -\sum_{j=1}^{n} \frac{|S_j|}{|S|} \bullet \log_2\left(\frac{|S_j|}{|S|}\right) \tag{3}$$

where $I(S, A)$ is the conditional entropy of S given Feature A, and S_j is the jth subset of S. Then the information gain ratio can be calculated as:

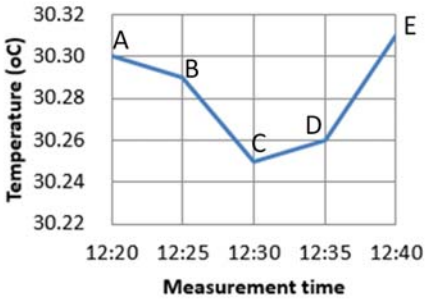
$$IGR(S,A) = Gain(S,A) / IntI(S,A) \tag{4}$$

The *IGR* calculation above is also used for threshold optimization of features in C4.5. For example, if Feature A has only one threshold x, S can be split into 2 subsets, where A<=x in Subset 1 and A>x in Subset 1. Different values and numbers of thresholds will lead to different *IGR*s, so *IGR* of Feature A herein refers to the maximum *IGR* with optimal thresholds. Often, only one threshold (binary-split) is chosen for each feature so as to make the tree uncomplicated.

Then, the one with maximum *IGR* among all the possible features is chosen as the root node, and splits the dataset into two subsets. Repeat recursively the feature selection and split in new subsets. The growth do not stop until all the data for each class has been distributed into the same subset, as shown in Fig. 2. However, a "pruning" step is often executed after a full tree has been built in practice to reduce the size of the tree [30] and simplify the rules.

2.2.2 Curve description algorithm (CD)

CD uses curve templates to generate its recognition rules. A curve template is comprised of a typical curve shape and a feature vector combination, as is shown in Fig.3. If any curve segment has similar curve shape with the curve template of a specific class, and also matches with the feature vector combination, this segment will be classified into this class. Here, the typical curve shape presents the variation of a parameter under the specific class. For example, the typical curve shape in Fig.3 (a) can be seen as the variation of temperature under a process of air-conditioner status changing from ON (A-C) to OFF (C-E). The feature vector combination is used to describe the numeric characteristics of each node on the typical curve.



$$\begin{cases} F_A = (3, f_{1A}, f_{2A} \cdots f_{nA}) \\ F_B = (1, f_{1B}, f_{2B} \cdots f_{mB}) \\ F_C = (4, f_{1C}, f_{2C} \cdots f_{kC}) \\ F_D = (0, f_{1D}, f_{2D} \cdots f_{pD}) \\ F_E = (0, f_{1E}, f_{2E} \cdots f_{qE}) \end{cases}$$

(a) (b)

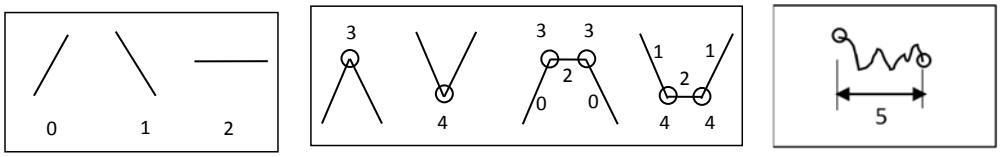
Fig. 3. An example of curve template: (a) typical curve shape, and (b) feature vector combination

A feature vector F_X in the feature vector combination is defined as:

$$F_X = (M_X, f_{1X}, f_{2X}, \ldots, f_{nX}) \tag{5}$$

where M_X is the morpheme of Point or Segment X, and f_{1X} to f_{nX} are features used to describe the numerical characteristics of X.

Morpheme M_X is a mathematical symbol used to describe the local shape characteristics of X. Fig. 4 shows its definition. Three basic morphemes that are numbered 0, 1 and 2 describe the rising, falling, and level directions, respectively, as is shown in Fig.4 (a). Two extensional morphemes that are numbered 3 and 4 describe the peak and valley points as shown in Fig.4 (b). An additional wildcard morpheme (numbered 5) is defined to present those curve segments whose characteristics are ignored for the analysis, as is shown in Fig. 4(c). Assuming that x_i is the value at the ith sampling point, that x_{i+1} is the value at the next point and x_{i-1} at the last point, Morpheme 0-4 can be defined as: 0, $x_i - x_{i-1} > 0$; 1, $x_i - x_{i-1} < 0$; 2, $x_i - x_{i-1} = 0$; 3, $x_i - x_{i-1} > 0$ and $x_i - x_{i+1} > 0$; and 4, $x_i - x_{i-1} < 0$ and $x_i - x_{i+1} < 0$. However, if the curve shape is 0-2-0, two Morpheme 3 are defined; if the curve shape is 1-2-1, two Morpheme 4 are defined, as is shown in Fig. 4(b).



(a) (b) (c)

Fig. 4. Definition of morpheme: (a) basic morphemes, (b) extensional morphemes, and (c) wildcard morpheme

Features f_{1X} to f_{nX} are also selected and sorted through *IGR* comparison. However, CD investigates features extracted among morphemes rather than from points. According to the definition, a morpheme can be a curve segment comprised of numerous sampling points. Therefore, features selected in CD are often different from those selected in C4.5.

2.3 Development of classification algorithms

The development process of classification algorithms in this study is comprised of 4 steps, pre-processing, training of algorithms, testing of the rules, and performance evaluation and comparison, as is shown in Fig.5.

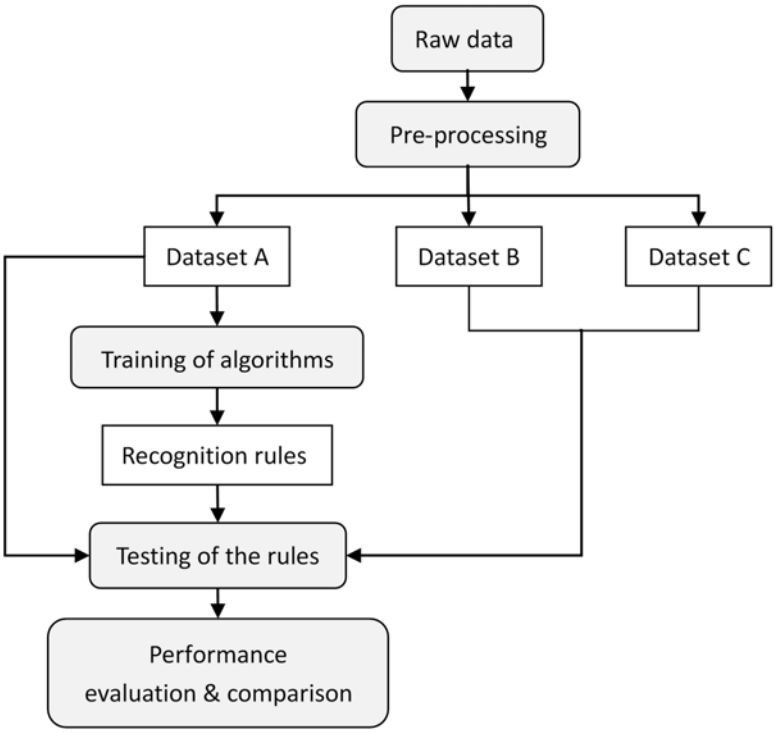


Fig. 5. Development process of classification algorithms

The task of pre-processing is to clean data and integrate different types of data. First, environmental samples with missing data were abandoned. Then, all the environmental samples were labeled as “AC on” (indicating that the AC was running) or “AC off” (indicating that the AC was turned off) according to the ground truth behavioral data. Table 2 shows the final data sizes for the three rooms after pre-processing, where an “AC usage” instance is defined as one continuous set of processes in which the AC state changed from “off” to “on” and then to “off” again.

Table 2. Data sizes for the three rooms

	Dataset A (Room A)	Dataset B (Room B)	Dataset C (Room C)
Total number of samples	17295	9325	6366
Number of samples labeled “AC on”/”AC off”	7292/10003	783/8542	307/6059
Number of “AC usage” instances	64	18	11

Dataset A was used to train the algorithms. The Objective of algorithm training is to identify the relationship

between AC operations and environmental parameters, and generate recognition rules. Because both C4.5 and CD are supervised algorithms, they may suffer from the problem of over-fitting. In order to minimize this problem, this study deployed the *k* fold cross-validation technique [31] during the training. The training process of C4.5 was conducted in the Weka program [32], and CD in a Python [33] program developed by the team.

All the three datasets were used to test the rules. Samples in the three datasets were classified with generated recognition rules from C4.5 and CD, respectively. The classification results were then compared with the ground truth behavioral data. Based on the comparison, the study evaluated and compared the performance of the two algorithms, using three evaluation indices: correctly classified percentage (*CCP*), correctly recognized "AC on" percentage (*CRN*), and correctly recognized "AC off" percentage (*CRF*). The three indices can be defined using the confusion matrix in Table 3, and expressed by Equation (6) to (8). *CCP* indicates the total accuracy of the algorithm in identifying the two classes, so it is also called "accuracy" in data mining. Commonly, the larger the *CCP* value, the better the performance of an algorithm. However, the *CCP* could also be very high if the number of samples in one class is much smaller than in the other, such as $x_1+x_2 << x_3+x_4$. Therefore, this study introduced *CRN* and *CRF* to perform an integral evaluation, where *CRN* (also called "sensitivity") reveals the algorithms' ability to recognize "AC on" samples, and *CRF* (also called "specificity") reveals the algorithms' ability to recognize "AC off" samples.

Table 3. Confusion matrix for recognition of AC operations

Confusion matrix		Recognized as	
		AC on	AC off
Ground truth	AC on	x_1	x_2
	AC off	x_3	x_4

$$CCP = \frac{x_1 + x_4}{x_1 + x_2 + x_3 + x_4} \cdot 100\% \quad (6)$$

$$CRN = \frac{x_1}{x_1 + x_2} \cdot 100\% \quad (7)$$

$$CRF = \frac{x_4}{x_3 + x_4} \cdot 100\% \quad (8)$$

where x_1, x_4 are the number of samples correctly recognized as "AC on" and "AC off", respectively, x_2 is the number of "AC on" samples that were incorrectly recognized as "AC off", and x_3 is the number of "AC off" samples that were incorrectly recognized as "AC on".

Note that, this study did not divide the samples into "Turning on AC", "Turning off AC", and "No actions". Because actions of turning on/off AC are finished in a moment, it is hard to accurately capture such actions with a 5-min sampling frequency. In addition, the impacts of one action on the air temperature or relative humidity can last for a few minutes. Such impacts are difficult to be described by only the features of one sampling point at the moment of action. Therefore, we turned instead to recognition of the actions based on changes to the AC state according to the rule: "When the AC state changes, an operation must have been performed on the AC."

3. Results

3.1 Training results of the two algorithms

3.1.1 Training result for C4.5

Because both the absolute values and change rates of environmental parameters changes significantly with AC operations, we investigated these two types of features during the training of C4.5. As listed in Table 4, Abs is the absolute value of parameter, while Bd5, Bd10, and Bd15 are backward difference of 5, 10, and 15 minutes, respectively, presenting different response speeds to AC operations. Through *IGR* calculation and comparison, the most informative backward difference was chosen for both air temperature and relative humidity. Therefore, the final features used to build the C4.5 decision tree were T_Abs (absolute value of temperature); T_Bd15 (Bd15 of temperature); H_Abs (absolute value of humidity); and H_Bd10 (Bd10 of humidity).

Table 4. Features of each parameter

Feature	Definition	Expression
Abs	Absolute value of parameter	abs(i)
Bd5	Backward difference of 5 minutes	[abs(i)-abs(i-1)]/5
Bd10	Backward difference of 10 minutes	[abs(i)-abs(i-2)]/10
Bd15	Backward difference of 15 minutes	[abs(i)-abs(i-3)]/15

Fig. 6 illustrates the generated decision tree. The tree includes a total of 15 nodes, among which eight are outcomes (rectangles). “Number A/Number B” in the outcomes represents the ratio of the number of correctly classified samples to the number of incorrectly classified ones. According to the principle of rule formation, the tree forms eight paths from root to outcomes, and each path can be used independently as a recognition rule. For example, if T_Abs > 29.545 °C and T_Bd15 <= -0.097 °C/min at a sample piont, the outcome is “On,” which means that the sample is recognized as belonging to the “AC on” class.

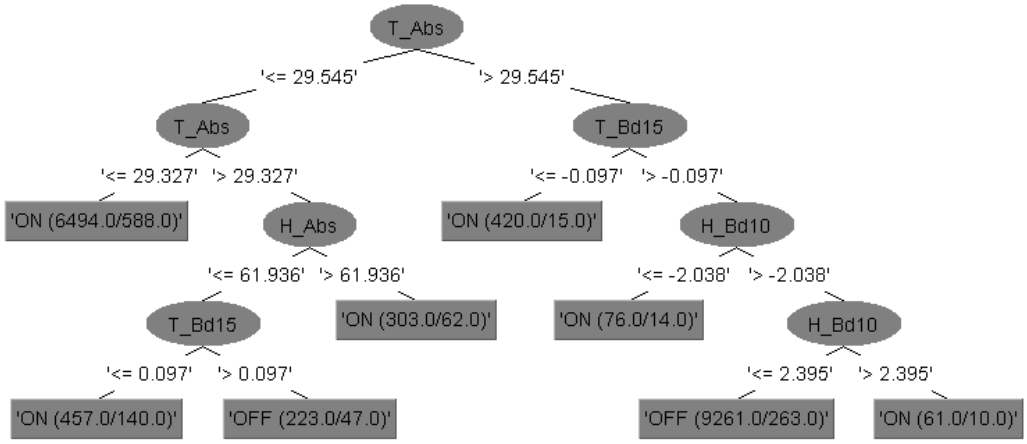


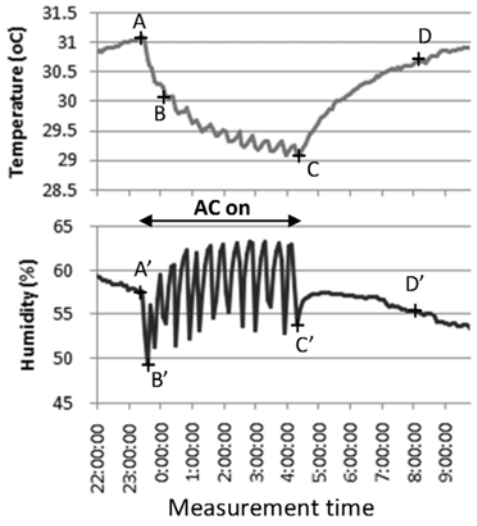
Fig. 6. Decision tree generated from C4.5

Generally, a decision tree does not stop partitioning until all the data for each pattern has been classified into the same subset. A large depth of tree will result in a huge tree with a number of branches. This will complicate the decision rules, and lead to over-fitting. However, if the tree is too short, it will not be able to extract enough characteristics to recognize classes, and lead to under-fitting. Therefore, we set the minimum number of instances per leaf at 200 in this study. This means that there must be at least 200 cases for each behavior category in order to continue splitting, as shown in Fig.8.

3.1.2 Training result for CD

Fig. 7 shows the curve templates generated in CD. Fig. 7(a) is the typical curve shapes of air temperature (A-D) and relative humidity (A’-D’) for an “AC usage” instance, where A-B is the rapid decline period of air temperature after AC is turned on, B-C is the transitional period of room air temperature from declining to the set-point, and C-D is the rising period of temperature after AC is turned off. For the AC without an inverter, Section B-C is the period that the AC automatically switched on and off, which leads to the temperature fluctuations in Fig. 7(a)

Features for each curve were also determined by IGR calculation and comparison, and were shown in in Fig. 7(b). CR_{AB} in Fig. 7(b) is the change rate between A and B, representing the decline speed of the parameters immediately after the AC was turned on. Min_{BC} is the minimum value between B and C, denoting the lowest level that the air temperature or relative humidity should have reached when the AC was running. D_{CD} is the absolute value of the difference between C and D, representing the variation soon after the AC was turned off. AD_{BC} is the average difference between adjacent peaks and valleys from B to C, while ACR_{BC} is the average change rate between adjacent peaks and valleys. AD_{BC} and ACR_{BC} were used to describe the fluctuations of air temperature and relative humidity when the air-conditioner was running. Note that Vectors $F_{A'}$ and $F_{C'}$ have no attributes, indicating that there are no restrictions to the relative humidity at Codes A’ and C’.



(a)

Temperature

$$\begin{cases} F_{A} = (3,) \\ F_{B} = (4,\ CR_{AB} >= 0.088\ ^{\circ}\text{C/min}) \\ F_{BC} = (5,\ \text{Min}_{BC} <= 30.21\ ^{\circ}\text{C},\ AD_{BC} <= 0.5\ ^{\circ}\text{C}) \\ F_{C} = (4,) \\ F_{D} = (3,\ D_{CD} >= 0.5\ ^{\circ}\text{C}) \end{cases}$$

Relative humidity

$$\begin{cases} F_{A'} = (\ ,\ ) \\ F_{B'} = (4,\ CR_{A'B'} >= 0.96\%) \\ F_{B'C'} = (5,\ ACR_{B'C'} >= 0.72\%/\text{min}) \\ F_{C'} = (\ ,\ ) \\ F_{D'} = (\ ,\ D_{C'D'} >= 0.96\%) \end{cases}$$

(b)

Fig. 7. Curve templates for one "AC usage" instance generated by CD: (a) typical curve shapes for temperature and relative humidity, and (b) feature vector combinations

Based on the templates in Fig. 7, the recognition rules can be expressed as the following steps:

1) Transform the new collected air temperature and relative humidity data into morpheme chains;

2) Search along the air temperature morpheme chain section by section according to the time sequence. If the characteristics of the section from A to D meet the conditions of Feature vectors F_A through F_D, as shown in Fig. 7(b), then Section A to C is a candidate value for the dataset of "AC usage" instances;

3) Search along the relative humidity morpheme chain within the time period of A to D. If there is a sub-section A' to D' that meets the conditions of Feature vectors $F_{A'}$ through to $F_{D'}$ in Fig. 7(b), then the section from A' to C' is confirmed as a value for the dataset of "AC usage" instances. Otherwise, repeat step 2;

4) If a sample point is within the range of values in the dataset of "AC usage" instances, label it "AC on"; if not, label it "AC off."

3.2 Testing results of the two algorithms

Table 5 displays the recognition results for C4.5 and CD from Dataset A (also the training set) and the other two testing sets.

Table 5. Testing results of the two algorithms

	C4.5			CD		
	Correctly classified percentage (*CCP*)	Correctly recognized "AC on" percentage (*CRN*)	Correctly recognized "AC off" percentage (*CRF*)	Correctly classified percentage (*CCP*)	Correctly recognized "AC on" percentage (*CRN*)	Correctly recognized "AC off" percentage (*CRF*)
Dataset A	93.1%	92.2%	94.4%	94.6%	89.5%	98.3%
Dataset B	87.3%	86.3%	98.3%	99.2%	98.1%	99.3%
Dataset C	89.1%	88.8%	96.7%	96.9%	34.7%	99.9%

3.3 Performance evaluation and comparison of the two algorithms

Based on the testing results shown in Table 5, we evaluated and compared the performance of C4.5 and CD in recognition AC operations from air temperature and relative humidity.

The *CCP* values for the three datasets were all higher than 90% when CD was used, and they were better than that for C4.5, especially for Datasets B and C. Since both algorithms were trained with Dataset A, it can be concluded that CD has better performance when generalized to Datasets B and C, than C4.5 does.

The *CCP* values from the two algorithms for Dataset A were almost the same, however, the classification capability of CD was much better than that of C4.5. Fig. 8 shows the ground truth data with the values recognized by the two algorithms from Dataset A for a two-day period. CD recognized the two "AC usage"

instances precisely with clear starting and ending time points, as is illustrated by the curve of “Recognized by C4.5” in Fig. 8. By contrast, C4.5 divided the two periods into parts, as if the AC was being turned on and off frequently by the occupants, as is illustrated by the fluctuations in the curve of “Recognized by CD” in Fig. 8. It was difficult to conduct further analysis on the basis of fragmented instances. A similar phenomenon occurred in the recognition results for Datasets B and C. Differences in matching mode between the two algorithms may account for this phenomenon.

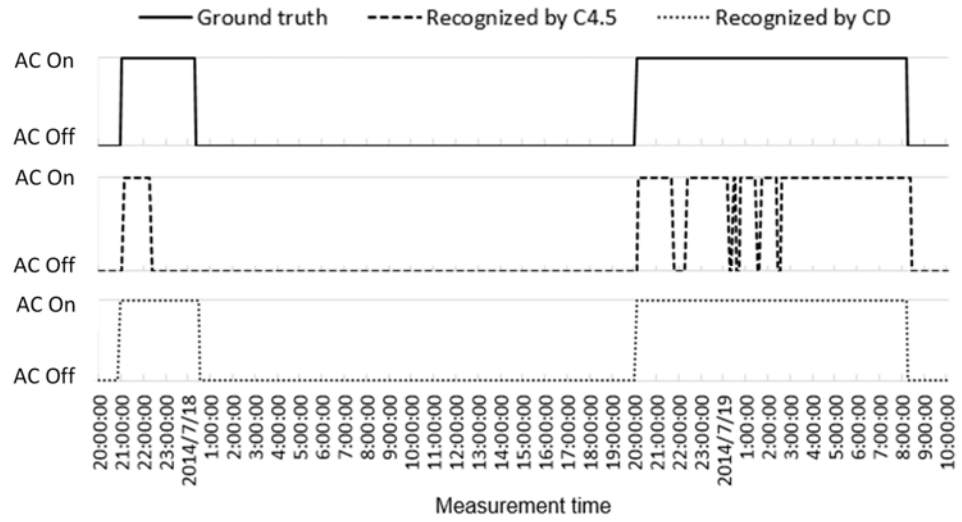


Fig. 8. Comparison of ground truth data and values recognized by the two algorithms, C4.5 and CD

During recognition, C4.5 investigated the temperature and relative humidity curves point by point, using the eight rules from the decision tree in Fig. 6. When the feature values at a given point did not match the right decision rule, the point was labeled with the opposite class. This occurred frequently at points with feature values close to the splitting threshold. For example, according to one of the rules, “If temperature <= 29.327 ^{o}C, the class is ‘AC on’.” If there was a short period during which the indoor air temperature fluctuated at around 29.3 ^{o}C, the AC state recognized by C4.5 would change from “AC on” to “AC off” and back to “AC on” repeatedly, leading to frequent fluctuations. Another type of error was the short-term matching of air temperature and relative humidity data with the rules. For example, during the short period after the windows in the test rooms were opened in the morning, there would be a simultaneous drop in both air temperature and relative humidity, which caused the samples to be recognized as “AC on.” Such error led to a lower *CRF* for C4.5 shown in Table 5.

Unlike C4.5, CD tested the curves section by section, using the templates in Fig. 7. Each “AC usage” instance was either recognized or unrecognized as a whole. That is, a section was recognized as an “AC usage” instance only when its characteristics completely matched the templates. This resulted in very few “AC off” samples being misrecognized as “AC on” samples. Therefore, CD achieved a much higher *CRF* than did C4.5. In fact, all misrecognition of samples by CD were caused by the location deviation at the starting and ending time points, and this deviation was found within three sampling intervals. However, by contrast, if any feature of the section did not match the templates, the AC state for this section was recognized wholly as “AC off” by CD, causing a relatively low *CRN*. For example, the occupants sometimes kept the AC running for only a few minutes and shut it off before the temperature could drop below 30.21 ^{o}C (a temperature feature shown in Fig. 7(b)). This led to misrecognition during this period.

The *CCP* values recognized by CD for Datasets B and C were both very high, indicating that the curve templates learned from Dataset A closely matched the “AC usage” characteristics in Datasets B and C. Fig. 9 shows the typical shapes of air temperature and relative humidity curves within one “AC usage” instance selected from Dataset B and C. No similar fluctuations in the temperature curves are found in both Fig .9 (a) and Fig. 9 (b) to that found in Fig. 7 (a). Difference of AC types and usage preferences between Room A, B and C can account for this. Different from Room A, Room B used an AC with an inverter, while the occupant in Room C preferred to set a very low temperature set-point and turn off the AC before the room temperature reached the set-point. Therefore, the phenomenon that AC automatically, periodically switched on and off was not observed in Room B and C. Although, the curve shapes in Fig .9 (a) and Fig. 9 (b) are similar to those in Fig. 7 (a). However, both curve shapes in Fig .9 (a) are more matching with Fig. 7(a) than those from Fig. 9 (b). Thus, Dataset B had a higher *CCP* than Dataset C did. However, detailed characteristics at each point on the curves of both Datasets B and C

differed greatly from those in Dataset A. This is demonstrated by the low *CCP* values for Datasets B and C with C4.5.

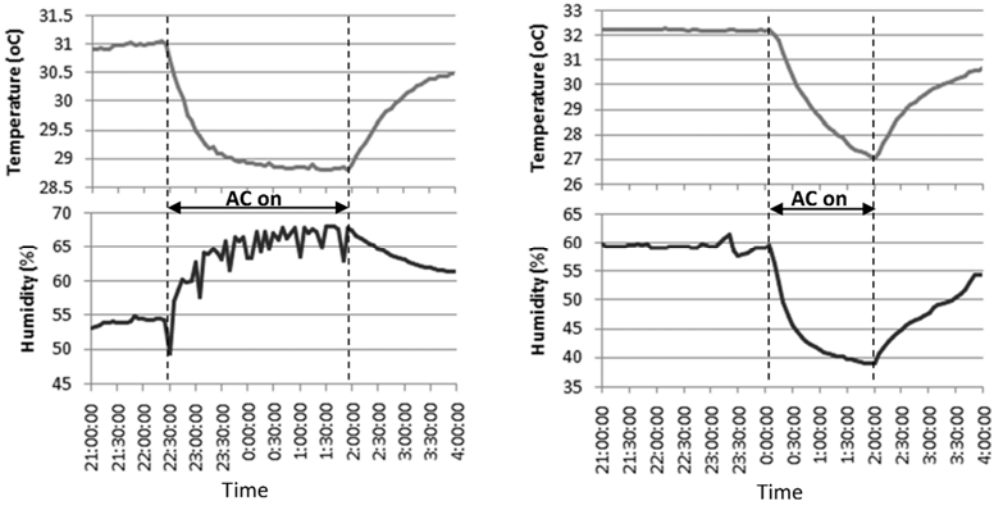


Fig. 9. Typical curve shapes for air temperature and relative humidity within one "AC usage" instance: (a) from Dataset B, and (b) from Dataset C

The *CRN* for Dataset C with the use of CD was much lower than that with C4.5, so the results for Dataset C were analyzed in detail. Fig. 10 compares the ground truth data with the values recognized by the two algorithms in Dataset C within a four-day period. Although C4.5 achieved a higher *CRN*, there were numerous fluctuations as can be seen in Fig.10, so the *CRF* was lower than that achieved by CD. As is shown in Fig.10, very few "AC off" samples were misclassified as "AC on" by CD (*CRF* = 99.9%). However, only 5 out of total 11 "AC usage" instances in Dataset C were correctly recognized, which led to a low *CRN*. To find out the reasons, we trained CD again with Dataset C, and found that optimized thresholds of features in the feature vector combination had deviated from those obtained from Dataset A. For example, the temperature change rate after AC was turned on (CR_{AB}) in Dataset C was calcuated to be larger than 0.078 ^{o}C, while recognition rules generated from Dataset A required that CR_{AB} should be larger than 0.088 ^{o}C (shown in Fig.7 (b)). Such deviations resulted in the misrecognition of "AC useage" instances in Dataset C.

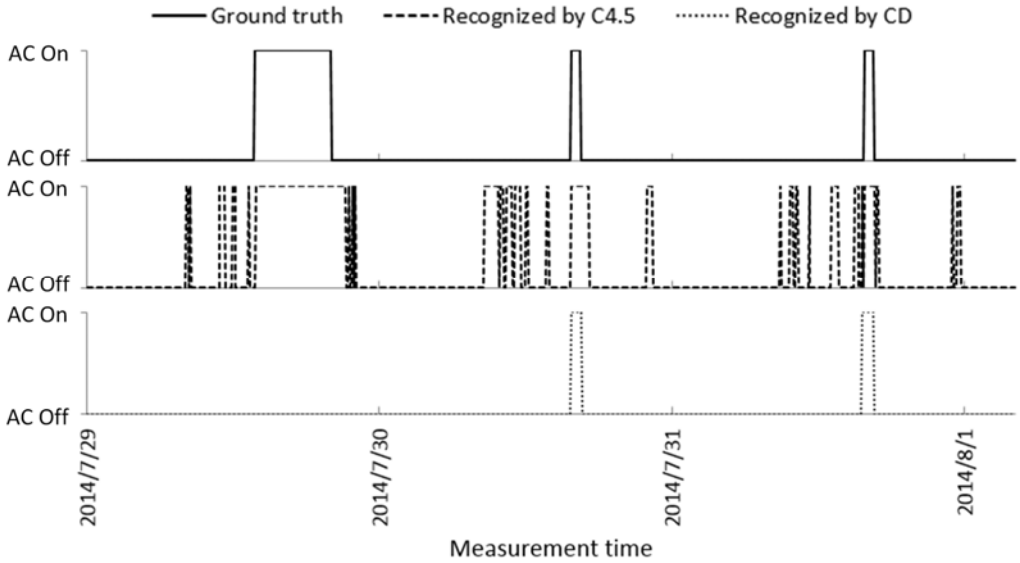


Fig. 10. Comparison of ground truth data and values recognized by C4.5 and CD in Dataset C

4. Discussion

In our experiment, we collect the ground truth AC operations through participants' self-recording on an app

installed in their mobile phones. This method is not as accuracy as directly collection with an electrical meter on the AC feed in cases that the participants forget to record. However, it is much more convenient when we want to collect some behavioral related information, such as recording the air temperature set-point when AC is turned on, and even the vote from participants to the current thermal environment. Besides, it is inexpensive, especially when several different ground truth behavioral data are collected together. In fact, we did collect the ground truth information of other behaviors with the app for our next work, including operations related to lights, windows, curtains and room occupancy.

Data mining approach has been used to detect occupant activities [34], occupancy [35], and behaviors [36]. Most of these cases used time-independent parameters and features as the input of classification algorithms, so they often achieved good performance of C4.5. Zhao et al. [21] used the power data of appliances to recognize the occupant presentation in the office, and obtained an average accuracy of 90.29%. However, most of environmental parameters are highly time dependent, and only the integration of a set of continuous points on the time-series curve can present accurately the impacts of behaviors on the parameters. Therefore, as a statistical classifier, C4.5 is proved not suitable to process environmental data in this study. It was found that even when the decision tree grew completely without any disturbances, *CCP* increased by only 2%, from 93.1% (listed in Table 5) to 95.1%. Meanwhile, the size of the tree expanded to 14 times as large as that shown in Fig. 8. We also tried to incorporate all the backward differences in Table 4 into the tree input (a total of eight features). *CCP* turned into 95.5% when the tree grew freely. Thus, the added information yielded hardly any improvement.

By contrast, CD is an approach of classification based on contextual information in curves rather than a statistical algorithm. It can cope with various curve shape deformations, so it is often used in image and signal identification, such as handwriting recognition [37] and vehicle classification [38]. It was proved that CD also has good performance in describing the time-series curves from environmental parameters in this work. No other studies have used CD to mine occupant behaviors from indoor environment data yet.

However, though CD has good performance in processing time-series curves of environmental parameters, it also suffers from over-fitting. CD achieved high accuracies in Dataset A, but obtained a low *CRN* in Dataset C. Over-fitting is a common issue for a supervised learning algorithm, and may be caused by different reasons. In this study, the amount of training data can account for over-fitting. As the amount of training data is not infinite, samples in the training set cannot cover all the features in different classes and different testing sets. Therefore, if the algorithm attempts to perform as well as possible in the training set, it will probably fail to maintain good generalization to a new testing set. Such failure may be significant when deviation of feature thresholds between the training set and testing sets is large, as is showed by Dataset A and C. Therefore, further work will be done to mitigate this problem from the following measures: 1) to increase the amount of training data by collecting data from more rooms; 2) to adjust the number of features used in the feature vector combination; and 3) to integrate a regularization method [39] with CD.

5. Conclusions

Based on the experiments in three bedrooms, the study developed two classification algorithms to recognize AC operations from indoor air temperature and relative humidity. Two types of recognition rules for AC operations were generated by training in one dataset, and were tested in all the three datasets. With the testing results, we evaluated and compared the performance of the two algorithms. The following conclusions are drawn:

1) As a statistical classifier, the C4.5 decision tree algorithm (C4.5) is not suitable for mining AC operations from environmental parameters. It obtained relatively low *CCP* and *CRN*, and led to frequent fluctuations on the time-series curve of recognition outcomes from all the tree datasets.
2) As a contextual curve classifier, the curve description algorithm (CD) has good performance in processing time-series curves of environmental parameters. It achieved relatively high *CCP* and *CRF* during the testing, and each recognized “AC usage” instance showed clear starting and ending time points.
3) AC operations can be recognized from indoor air temperature and relative humidity by CD. However, CD also suffers from over-fitting as a supervised classifier. More work should be done to further improve the performance of CD.

The study has proposed a promising approach for large-scale data collection of AC operations in buildings. It could be integrated with applications of behavior modelling and building energy simulation. Application of this approach for other occupant behaviors, such as operations related to light switches and windows, and occupancy, should be validated by further experiments.

Acknowledgements

The authors gratefully acknowledge the enthusiastic cooperation of the three families who participated in the experiment in this study. An undergraduate student from Tianjin University and several workers from Intelligent Green Solutions Ltd helped with the design of the wireless behavioral data collection system. Their contributions are greatly appreciated.

References

[1] J.J. Hirsch, eQuest Introductory Tutorial, Version 3.64, Camarillo, CA, (2010) 59.
[2] D. Yan, J. Xia, W. Tang, F. Song, X. Zhang, Y. Jiang, DeST--An integrated building simulation toolkit, Part I: Fundamentals, Building Simulation 1 (2008) 95–110.
[3] Z.J. Li, J. Yi, Q. Wei, Survey and analysis on influence of environment parameters and residents' behaviors on air conditioning energy consumption in a residential building, HV&AC 37(2007) 67–71. (In Chinese)
[4] P. De Wilde, The gap between predicted and measured energy performance of buildings: A framework for investigation, Automation in Construction 41 (2014) 40-49.
[5] C. Menezes, A. Cripps, D. Bouchlaghem, R. Buswell, Predicted vs. actual energy performance of non-domestic buildings: using post-occupancy evaluation data to reduce the performance gap, Applied Energy 97 (2012) 355–364.
[6] T. Hong, S. D'Oca, W.J.N. Turner, S.C. Taylor-Lange, An ontology to represent energy-related occupant behavior in buildings, Part I: Introduction to the DNAs framework, Building and Environment (2015), http://dx.doi.org/10.1016/j.buildenv.2015.02.019.
[7] International Energy Agency, EBC Annex 66 Text, http://www.annex66.org/?q=Subtasks (accessed 22.5.15).
[8] X. Ren, D. Yan, C. Wang, Air-conditioning usage conditional probability model for residential buildings, Building and Environment 81 (2014) 172-182.
[9] R. Yasue, H. Habara, A. Nakamichi, Y. Shimoda, Modeling the occupant behavior relating to window and air conditioner operation based on survey results, Proceedings of BS2013: 13th Conference of International Building Performance Simulation Association, Chambery, France, August 26-28, 2013.
[10] H.C. Shih, A robust occupancy detection and tracking algorithm for the automatic monitoring and commissioning of a building, Energy and Buildings 77 (2014) 270-280.
[11] B. Song, H. Choi, H.S. Lee, Surveillance tracking system using passive infrared motion sensors in wireless sensor network, International Conference on Information Networking, 2008, pp. 1-5.
[12] D.J. Hess, J.S. Coley, Wireless smart meters and public acceptance: The environment, limited choices, and precautionary politics, Public Understanding of Science 23 (2014) 688-702.
[13] V. Fabi, R. V. Andersen, S. Corgnati, B. W. Olesen, Occupants' window opening behaviour: A literature review of factors influencing occupant behaviour and models, Building and Environment 58 (2012) 188-198.
[14] M. Jin, N. Bekiaris-Liberis, K. Weekly, C. Spanos, A. Bayen, Sensing by Proxy: Occupancy Detection Based on Indoor CO2 Concentration. UBICOMM 2015, pp. 14.
[15] Lam K P, Höynck M, Zhang R, B. Andrews, Y. S. Chiou, B. Dong, D. Benitez, Information-theoretic environmental features selection for occupancy detection in open offices. Eleventh International IBPSA Conference, 2009, pp. 1460-1467.
[16] S. Wold, C. Albano, W. J. Dunn, K. Esbensen, S. Hellberg, E. Johansson, M. Sjöström, Pattern recognition: finding and using regularities in multivariate data, Food research and data analysis (1983) 147-188.
[17] S. Fang, Y. Liu, D. Yuan, C. Guo, Wireless monitoring and control system for indoor environment based on ZigBee technology, Future Information Engineering (2 Volume Set) (2014) 49-67.
[18] Fisk W J, How IEQ affects health, productivity, ASHRAE journal 44(2002) 56-56.
[19] J. G. Allen, P. MacNaughton, J. G. C. Laurent, S. S. Flanigan, E. S. Eitland, J. D. Spengler, Green Buildings and Health, Current environmental health reports 2(2015) 250-258.
[20] Bishop, Christopher, Pattern recognition and machine learning, Berlin: Springer. ISBN 0-387-31073-8, 2006.
[21] J. Zhao, B. Lasternas, K.P. Lam, R. Yun, V. Loftness, Occupant behavior and schedule modeling for building energy simulation through office appliance power consumption data mining, Energy and Buildings 82 (2014) 341-355.
[22] L. Atallah, M. ElHelw, J. Pansiot, D. Stoyanov, L. Wang, B. Lo, G.Z. Yang, Behaviour profiling with ambient and wearable sensing, 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), Springer Berlin Heidelberg, (2007) 133-138.

[23] B. Dong, Integrated Building Heating, Cooling and Ventilation Control, Carnegie Mellon University, 2010.
[24] B. Dong, B. Andrews, K.P. Lam, M. Höynck, R. Zhang, Y.S. Chiou, D. Benitez, An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network, Energy and Buildings 42 (2010) 1038-1046.
[25] C. Drummond, R.C. Holte, C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling, Workshop on Learning from Imbalanced Datasets II, 2003.
[26] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 2013.
[27] H. Nishida, Curve description based on directional features and quasi-convexity/concavity, Pattern Recognition 28 (1995) 1045-1051.
[28] J.R. Quinlan, C4.5 Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993.
[29] Shannon C E, A note on the concept of entropy, Bell System Technical Journal 27(1948) 379-423.
[30] R. Kohavi, Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid, KDD, 1996, PP. 202-207.
[31] T. Fushiki, Estimation of prediction error by using K-fold cross-validatio, Statistics and Computing 21 (2011) 137-146.
[32] University of Waikato, Weka 3, http://www.cs.waikato.ac.nz/~ml/weka/downloading.html (accessed 4.3.15).
[33] Python Software Foundation, Python 2.7.6, https://www.python.org/ (accessed 4.3.15).
[34] N. Ravi, N. Dandekar, P. Mysore, M.L. Littman, Activity recognition from accelerometer data, AAAI 5 (2005) 1541-1546.
[35] S.K. Ghai, L.V. Thanayankizil, D.P. Seetharam, D. Chakraborty, Occupancy detection in commercial buildings using opportunistic context sources, International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012, pp. 463-466.
[36] Z.J. Yu, F. Haghighat, B.C.M. Fung, E. Morofsky, H. Yoshino, A methodology for identifying and improving occupant behavior in residential buildings, Energy 36 (2011) 6596-6608.
[37] M. Cui, J. Femiani, J. Hu, P. Wonka, A. Razdan, Curve matching for open 2D curves, Pattern Recognition Letters 30 (2009) 1-10.
[38] P.Q. Lin, J.M. Xu, A description and recognition method of curve configuration and its application, Journal of South China University of Technology (Natural Science Edition) 2 (2009) 17 (in Chinese).
[39] Andrew G, Gao J, Scalable training of L1-regularized log-linear models, Proceedings of the 24th international conference on Machine learning, ACM, 2007, pp. 33-40.