OCCUPANCY ESTIMATION USING NON INTRUSIVE SENSORS IN ENERGY EFFICIENT BUILDINGS

Abhay Arora, Manar Amayri, Venkataramana Badarla, Stéphane Ploix, Sanghamitra Bandyopadhyay

Proceedings of BS2015:
14th Conference of International Building Performance Simulation Association, Hyderabad, India, Dec. 7-9, 2015.

OCCUPANCY ESTIMATION USING NON INTRUSIVE SENSORS IN ENERGY EFFICIENT BUILDINGS

Abhay Arora[1], Manar Amayri[2],Venkataramana Badarla[1]
Stéphane Ploix[2],Sanghamitra Bandyopadhyay[3]
[1]Indian Institute of Technology Jodhpur, India
[2]G-SCOP laboratory / Grenoble Institute of Technology, Grenoble, France
[3]Indian Statistical Institute, Kolkata, India

ABSTRACT:

A general approach is proposed to estimate the number of occupants in a zone using different kinds of measurements such as motion detection, power consumption or CO_2 concentration. The proposed approach is inspired from machine learning. It starts by determining among different measurements those that are the most useful by calculating the information gains. Then, an estimation algorithm is proposed. It relies on a C4.5 learning algorithm that yields human readable decision trees using measurements to estimate the number of occupants. It has been applied to an office setting.

1 INTRODUCTION

Recently, research about building turns to focus on occupant behavior. Most of these works deal with the design stage: the target is to represent the diversity of occupant behavior in order to guarantee minimal measured performance. Most of the approaches use statistics about human behavior (Roulet et al., 1991; Page et al., 2007; Haldi and Robinson, 2009). (Kashif et al., 2013) emphasized that inhabitants' detailed reactive and deliberative behavior must also be taken into account and proposed a co-simulation methodology to find the impact of certain actions on energy consumption.

Nevertheless, human behavior is not only interesting during the design step, but also during operation. It is indeed useful for diagnostic analyses to discriminate human misbehavior from building system performance, and also for energy management where strategies depend on human activities and, in particular, on the number of occupants in a zone. Unfortunately, the number of occupants is not easy to measure. This paper tackles this issue. It proposes an occupancy estimator combining different measurements such as CO_2 concentration, motion detection, power consumption etc., because only one measurement proved to be not reliable enough to estimate the number of occupants. For instance, CO_2 concentration may be useful but in some configurations, when a window is opened for instance, estimations become unreliable. Motion detection and power consumptions depend on occupant activities. However, altogether, these measurements can be combined to get a more reliable estimator. The organization of the rest of the paper as follows. Section 2 presents a state of the art about occupancy estimation. Section 3 discusses the proposed process that yields to an occupancy estimator suitable for a specific context. Section 4 points out what are the most relevant measurements to consider for an estimator. Section 5 compares the occupancy estimations with actual ones in an office context. Finally, Section 6 presents conclusions and further directions.

2 STATE OF THE ART

Similar work for finding occupancy has been already tackled and various methods have been investigated. The methods vary from basic single feature classifiers that distinguish among two classes (Presence and Absence) to multi-sensor, multi feature models. A primary approach, which is prevalent in many commercial buildings is to use passive infrared (PIR) sensors for occupancy. However, motion detectors fail to detect presence when occupants remain relatively still, which is quite common during activities like working on a computer, or regular desk work. Furthermore, drifts of warm or cold air on objects can be interpreted as motion leading to false positive detections. This makes the use of only PIR sensors for occupancy counting purpose less attractive. Conjunction of PIR sensors with other sensors can be useful as discussed in (Agarwal et al., 2010) which makes use of motion sensors and magnetic reed switches for occupancy detection to increase efficiency in the HVAC systems of smart buildings, which is quite simple and non-intrusive. Apart from motion, acoustic sensors (Padmanabh et al., 2009) may be utilized. However, audio from the environment can easily fool such sensors, and with no support from other sensors it can report many false positives. In the same way, other sensors like video cameras (Erickson et al., 2011; Milenkovic and Amft, 2013b) which exploit the huge advances in the field of computer vision and the ever increasing computational capabilities, RFID tags (Philipose et al., 2004) installed on id cards, sonar sensors (Milenkovic and Amft, 2013a) plugged on monitors to identify presence of a person on the computer, have been used and have proved to be much better at solving the problem of occupancy count, yet can not be employed in most office buildings for reasons like privacy and cost concerns. The use of pressure and PIR sensors to determine presence/absence in single desk offices has been discussed in (Nguyen and Aiello, 2012); it further tags activities based on this knowledge.
However, for various applications like activity recognition or context analysis within a larger office space, information regarding the presence or absence of peo-

ple is not sufficient, and an estimation of the number of people occupying the space is essential. (Lam et al., 2009) investigates this problem in open offices, estimating occupancy and human activities using a multitude of ambient information, and compare the performance of HMMs, SVMs and Artificial Neural Networks. However, none of these methods generate human-understandable rules which may be very helpful to building managers.

An alternate approach aims to understand the relationships between carbon dioxide concentration, IAQ (Indoor Air Quality) and the number of occupants. Such a physical CO_2 model built on sensor networks has been extensively used (Aglan, 2003) in smart office projects to improve occupant comfort and minimize building energy use. In this paper, the model has been studied to find out the value of using it in occupancy estimation.

In general, an occupancy count algorithm that fully exploits information available from low cost, non-intrusive, environmental sensors and provides meaningful information is an important yet little explored problem in office buildings.

3 PROCESS USED FOR ESTIMATION

Experiment setup

The test bed (Figure 1) is an office in Grenoble Institute of Technology, which accommodates a professor and 3 PhD students. The office has frequent visitors with a lot of meetings and presentations all through the week. The set-up for the sensor network includes:

- 2 video cameras for recording real occupancy numbers and activities.
- An ambience sensing network, which measures luminance, temperature, relative humidity (RH), motion at a sampling rate of 30 seconds.
- A centralized database with a web-application for retrieving data from different sources continuously.

Generating features

To perform the task of finding the number of occupants, a relation has to be discovered between the office environment and the number of people in it. The office environment can be represented as a set of parameters, $P_t = [p_1, p_2, \ \dots \ , p_n]_t$. This set of parameters P at any instance of time t must be indicative of occupancy. Such a parameter, can be termed as a feature, and therefore the set of features as feature vector. Similarly, the n-dimensional space that contains all possible values of such a feature vector is the feature space. The underlying approach for the experiments is to formulate the classification problem as a map from a feature vector into some feature space that comprises several classes of occupancy. Therefore, the success of such an approach heavily depends on how good (those which provide maximum separability among classes) the selected features are. In this case, features are attributes from multiple sensors accumulated over a time interval. The choice of interval duration is highly context dependent, and has to be done according to the granularity required. However, some features do not allow this duration to be arbitrarily small. As an example, it has been observed that CO_2 levels do not rise immediately, and one of the factors affecting this time is the ventilation of the space being observed. Regarding the results presented in this paper, an interval of T_s = 30 minutes (which has been referred to here as 1 quantum) has been considered.

Before any features are calculated for the training data, some basic preprocessing of data had to be done: *basic interpolation* for non-existent data and application of *an outlier removal algorithm*. The interpolation part is necessary for filling in missing values from the sensor data. This is frequent in devices which are event-triggered i.e. no data points are reported if there is no change in the feature being reported. Thus, the previous data point had to be copied into the voids.

Removing outliers

Despite having reliable sensors, some single data point spikes have been observed in the recordings, which are attributed to random faults from the sensor. The faults can be easily identified visually in a continuous time-series, but to identify and remove them statistically, it is necessary to understand what makes a data point an outlier. The removal afterwards is almost trivial. Contextual outliers are defined as data points, which, in the contact of previous and future data points, seem highly improbable. These points have been detected as the ones that follow the equations simultaneously:

$$\begin{aligned} \mathrm{pdiff}_k &= x_k - x_{k-1} \\ \mathrm{fdiff}_k &= x_{k+1} - x_k \\ |\mathrm{pdiff}_k| &> m_{\Delta x} + \lambda \sigma_{\Delta x} \\ |\mathrm{fdiff}_k| &> m_{\Delta x} + \lambda \sigma_{\Delta x} \\ \mathrm{pdiff}_k \,.\, \mathrm{fdiff}_k &> 0 \end{aligned}$$

where:

x_k	:	value of the feature at time quantum k
pdiff_k	:	difference between the current and previous data point
fdiff_k	:	difference between the current and next data point
$m_{\Delta x}$	:	average difference between two consecutive data points for the whole dataset
$\sigma_{\Delta x}$	:	standard deviation of the difference between two consecutive data points for the whole dataset
λ	:	configurable parameter, here $\lambda = 5$

All the data points that satisfy the above equation are removed.

In this section, both generic and specific features extracted from various sensors are discussed. Let $\mathcal{T}_k =$

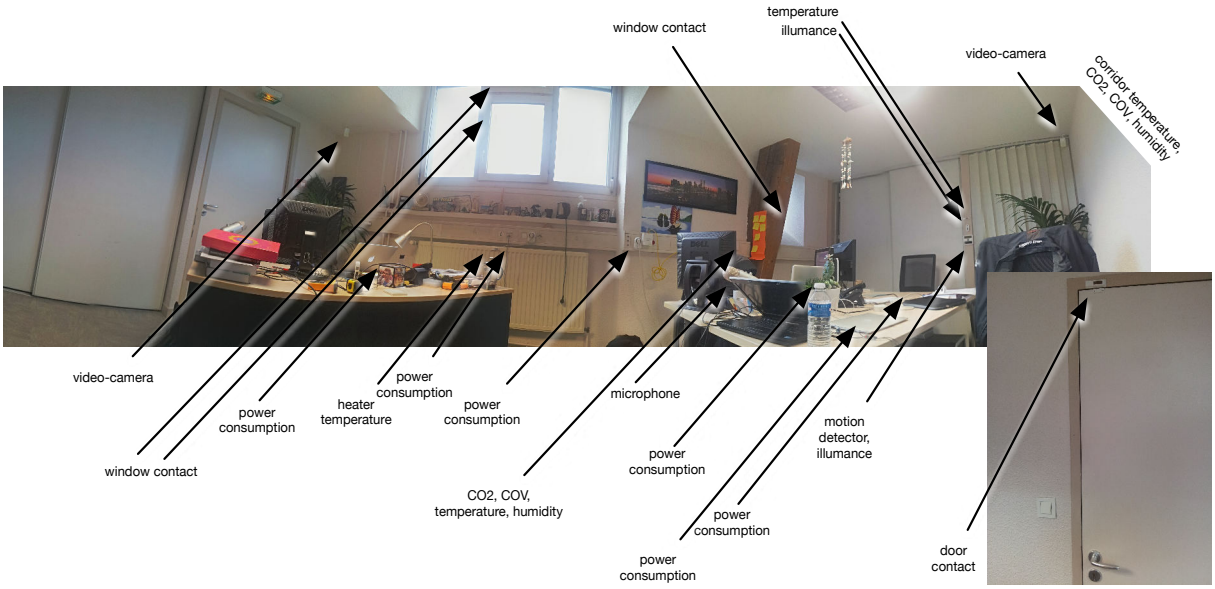

Figure 1: Sensor test bed at Grenoble INP

$\{t_i : t_i \in [kT_s, (k+1)T_s]\}$ be the time samples related to time quantum k:

- 30 minutes average: $\frac{1}{|\mathcal{T}_k|} \sum_{t_i \in \mathcal{T}_k} data(t_i)$, where $0 \leq k \leq 47$ for one day, since the number of 'half-hours'in a day are limited to 48. This feature is calculated for carbon dioxide and temperature sensors.
- Contact state: This feature is extracted for the door and window contact sensors. Possible values for this feature can be 0: door/window open, 1: door/ window closed, a real number $f_{state} \in [0, 1]$, which denotes the time ratio of door opening during a time quantum.
- Fluctuation count: the PIR sensor in use is a binary sensor that reports a value of 1 whenever it senses some motions. The number of times a motion is detected within the specified duration of 1 quantum has been computed. $\sum_{t_i \in \mathcal{T}_k} data(t_i)$
- Difference between outdoor and indoor temperatures.
- Time slot generated from calendar: NIGHT, PRELUNCH, LUNCH, POSTLUNCH. It corresponds respectively to time intervals [20-8),[8-12),[12-14),[14-20).
- Type of day generated from calendar: one among Weekday,Weekend

As occupancy in an office is a continuous event, there exist temporal dependencies within some of the features. To utilize those, some features which purely depend on sensor data in the past, have been extracted.

- Previous classification.
- First order difference: $data(t_k) - data(t_{k-1})$. This feature is calculated for CO_2, *temperature*, and the difference between *outdoor* and *indoor* temperatures.
- first order derivative: it gives the trend of data. The data points are interpolated to a first-order linear equation, and then the derivative of the resultant line is recorded. This feature is useful to quantify the rate of increase/decrease of occupancy relative to the previous time interval.

Selecting features

From the large set of features discussed above, some of them may not be worthwhile to consider, to achieve our target of occupancy classification. These features are the ones which, when added to the classification algorithm make no difference to the overall output. In other words they are not useful enough for our purposes. For an example, using the absolute temperature readings would be useless, as it is not representative of occupancy at all. One quantitative measurement of the usefulness of a feature is *information gain*. Before detailing what is an *information gain*, it is imperative to discuss the concept of *entropy*. Entropy is an attribute of a random variable that categorizes its disorder. Higher the entropy, higher is the disorder associated with the variable i.e. the less it can be predicted. Mathematically, entropy is defined by:

$$H(y) = \sum_{i=0}^{n-1} -p(y_i) \log_2 p(y_i)$$

where:

y	:	a random variable whose value domain is $dom(y) = \{y_0, \ldots, y_{n-1}\}$
$H(y)$	:	entropy of a random variable y
$p(y_i)$	:	probability for y to be equal to the value y_i

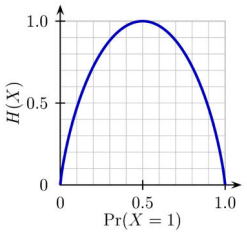

Figure 2: Entropy vs. probability [If probability of an event is 0.5, it means the highest disorder: entropy=1]

Now, Information gain can now be defined between two random variables, x and y as:

$$IG(x, y) = H(y) - H(y|x) \quad (1)$$

where:

y	:	target random variable
$H(y)$	:	entropy of y
$H(y\|x)$	:	conditional entropy of y given x

The higher the reduction of disorder by fixing feature x is, more is the information gained for determining y thus making x a good feature to use for classifying y.

Classification algorithm

A supervised learning approach has been used. Hence, for preparing training data, occupancy count was manually annotated using the video feed from two cameras strategically positioned in the office. The weighted average of the number of people visiting the office was recorded for each 30 minutes of the day. To complement this data, a keyboard connected to a *Raspberry PI* module had been set up to be used by the visitors for updating person-count in the office as they left/entered the office. Data recorded in this way is much easier to feed to a script, hence completely automating the training part, but only after sufficient care by the visitors and inmates the data can be termed reliable.

The *decision tree* classification technique has been selected as our prediction model because it provides human-readable results which can be analyzed and adapted by building managers easily. The decision tree algorithm selects a class by descending a tree of decision nodes where each internal node represents a comparison of a single feature value with a learned threshold. The leaf nodes represent the selected class for the given features. The target of the decision tree algorithm is to select features that are more useful for classification. As information gain (discussed earlier) approaches to zero, the difference between initial disorder (entropy) of the target variable, and the entropy of the variable after adding the observation from the test feature x, is negligible. Hence, the particular feature is not probably going to help very much during the decision making process.

A decision tree algorithm provides quite a few advantages. As per (Quinlan, 1986), the features with higher information gain are much higher up the tree, therefore making the process of feature selection intrinsic to the classifier. Since the path to the leaf may consist of many internal nodes, each of which may check different feature values, such paths exploit the correlation among the various features. The decision tree approach offers the advantage of generating rules that the path towards the leaf node is quite informative and it clearly points out direct causes for the selection of a particular class. Unlike methods that use decision boundaries (SVMs, regression techniques), decision tree analyses are independent of the scale of the input data, so no or little conditioning of the data is necessary.

Learning process

Using this raw training data, the features previously mentioned were extracted. A vector of features and target $\langle f_1, f_2, f_3, \ldots, f_N; y\rangle$ had been generated for each time quantum, where f_i stands for the i^{th} feature and y, for the level of occupancy. Here, level corresponds to an interval in the partition $\mathcal{L} = \langle I_0, \ldots, I_{l-1}\rangle$ where l in the number of occupancy levels.

Two decision tree models, D_1 and D_2, were trained for each level l. D_1 is trained with raw sensor features only and D_2 with raw as well as the temporal features. The reason these two models had to be created is because of the fact that using previous estimations as a feature during the test phase may (and in most cases does) introduce a classification error, which propagates and accumulates throughout the procedure. To prevent this from happening: (1) temporal dependencies altogether should be ignored by not using those features, or (2) some randomness to the process should be introduced, which does not let the error to propagate. Some investigations into method (2) led us to develop the following algorithm. Once the tree D_2 has given its classification, membership probability is estimated, say p_2 of the test instance for the prediction. Similarly, membership probability p_1 is also calculated for the tree (D_1) without the temporal features.

if $p_1 \geq p_2$ **then**
  Use classification from tree D_1
else
  if getRandom() ≥ 0.5 **then**
    Use classification from tree D_2
  else
    Use classification from tree D_1
  end if
end if

4 SELECTION OF BEST FEATURES

This section introduces the most relevant features that have been considered.

Occupancy from power consumption

Power consumption sensors are easy to deploy in most households and offices. In order to investigate the possibility of using power consumption data in occupancy recognition, 4 sensors have been connected to inhabitant laptops in the studied office. Through analyz-

ing the power consumption data, a threshold is determined to discriminate cases of computer standby from cases where someone is working on a laptop, which increases the power consumption and leads to wrong occupancy estimation (Kleiminger et al., 2013).

$$\pi_i = \begin{cases} 0 & \text{if } power_i < Threshold \\ 1 & \text{otherwise} \end{cases}$$

where $power_i$ stands for the actual laptop average power consumption during time quantum i and threshold $= 15W$. The number of occupants is then estimated by $\hat{l}_{consumption} = \sum_i \pi_i$. It has to be noticed that although this estimator is one of the most relevant feature, it is not reliable in case of visitors.

Occupancy from CO_2 physical model

An alternative approach for occupancy is to use a physical CO_2 model. According to (ASHRAE, 1985), the model given by (2) represents the relation between carbon dioxide generation, the volumetric flow rate of fresh air entering the office, the volumetric air flow rate outgoing from the office and occupancy (Aglan, 2003). The proposed approach relies on the data coming from CO_2 concentration sensor, door contact, window contact, occupancy labels extracted from video cameras for tuning air flows, and constant parameters associated to the office.

$$V.\frac{dC_{in}}{dt} = -Q.C_{in} + Q.C_{out} + n.S \qquad (2)$$

Equation (2) has been discretized, considering the time interval as 1 quantum:

$$n_k = \frac{Q}{S}\left(\frac{C_{in,k+1} - e^{\frac{-Q.T_s}{V}}}{1 - e^{\frac{-Q.T_s}{V}}} - C_{out}\right) \qquad (3)$$

where:

V	:	Volume of the space being modelled
C_{out}	:	Outdoor CO_2 concentration
$C_{in,k}$	:	Indoor CO_2 concentration at time k
Q	:	Rate of renewal air-flow
S	:	CO_2 production for 1 average person (7 ppm)
n_k	:	Number of people at time k

For this model, carbon dioxide readings averaged on 1 time quantum (hence T_s=30 minutes or 1800 seconds) were used, and the supervised occupancy count(n) extracted from the video cameras. The approximate volume of the office was $45m^3$. We take C_{out} to be 400 ppm, which is the standard outdoor carbon dioxide concentration.

Next, to find the best value for Q(office air flow rate), we need to consider all the configurations for the door and window, as Q would be different in each of the case. However, since the study was done in the month of January(winter season), the window were always closed, so the flow rates at open window configurations could not be found out. To minimize the objective function, the SLSQP optimization subroutine was utilized which provided the following air-flow rates after converging.

door's position	Qair(best)
close	Q_1=0.0237
open	Q_2=0.0333

Table 1: Qair corresponding to the positions of window and door

The next step is to use these best average airflow values for calculating the number of occupants over a time quantum lasting 30 minutes. Occupancy estimation is obtained from equation (2). Finally, the last step is to use this estimation of occupants as one feature in the classification model.

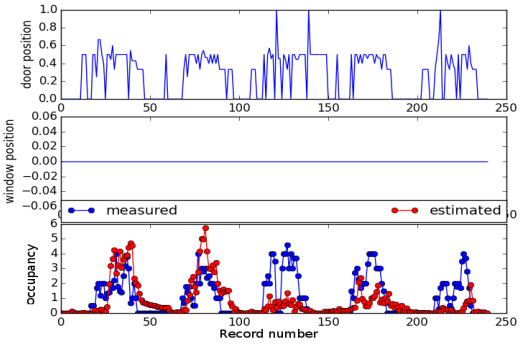

Figure 3: Occupancy estimation from CO_2

Figure 3 shows the results from the physical model for the period 26-Jan-2015 until 30-Jan-2015, with an accuracy of 41%. Obviously, the fault in estimation from the model happens when the door position changes, this problem can be solved by considering the actual corridor CO_2 concentration data instead of a constant value i.e. $400ppm$ to improve the results.

Analysis of the most relevant features

To mathematically calculate the information gain attribute discussed earlier, it is necessary to discretize the features which have values that are continuous in nature. A typical discretization function splits a large continuous range into several sub-ranges. However, such a function relies upon: a) sorting the values of the feature to be discretized b) determining a cut-point for splitting, according to some criterion (maximum and minimum value for each feature). The first condition requires the elements in the ranges to be comparable. The considered features (CO_2 concentration, motion fluctuations, number of people) are continuous ranges of real numbers, hence can be sorted. For the second, we calculated information gain by distributing our continuous features into 5, and 8 (chosen at random) ranges respectively to get a better insight into the relation between the number of ranges and usefulness of the feature. Using these information gain values, the most relevant features to estimate occupancy can be determined.Table 2 presents the information gains for the considered discretizations.

1. CO_2 (in ppm):

Disc.1	$\{[390, 450), [450, 690), [690, 900), [900, 1300), [> 1300]\}$
Disc.2	$\{[390, 420), [420, 500), [500, 600), [600, 700), [700, 800), [800, 900), [900, 1300), [> 1300]\}$

2. Occupancy number from physical model discretization (in average number of occupants during a time quantum):

Disc.1	$\{[0, 0.5), [0.5, 1.5), [1.5, 3), [3, 4.5), [> 4.5]\}$
Disc.2	$\{[0, 0.5), [0.5, 1), [1, 1.5), [1.5, 2), [2, 3), [3, 4), [4, 5), [> 5]\}$

Thus, it is imperative that methods for choosing the optimal discretization be applied, this is done implicitly by the C4.5 classification algorithm which is presented in the next section.

Feature	IG_1	IG_2
motion fluctuations	0.79	0.93
power consumption	0.59	0.91
CO_2 mean	0.54	0.68
time slot	0.63	0.63
CO_2 derivative	0.52	0.62
door's position	0.43	0.43
occupancy from physical model	0.33	0.41
mean ($temp_{outside}$-$temp_{inside}$)	0.19	0.22
day type	0	0
window position	0	0

Table 2: Information Gain values

The information gain coming from CO_2 physical model is not informative enough because of the missing data from corridor CO_2 sensor for the considered period. Finally, after removing less important features, the main informative features are found to be:

1. Motion detector counting
2. Occupancy estimation from power consumption
3. CO_2 average value
4. Time slot
5. CO_2 derivative
6. Door position

5 RESULTING OCCUPANCY ESTIMATORS

Generating decision trees

The C4.5 decision tree algorithm (Quinlan, 2014) had been used to perform recognition by using aggregated features, and the labels extracted from video cameras. 5 occupancy levels have been defined to generate decision trees because of the maximum number of occupants in the office. Training data covers for 11 days from 12-Jan-2015 to 22-Jan-2015, while testing data is collected over a span of 5 days from 26-Jan-2015 to 30-Jan-2015. During the training period, 110000 data points have been collected as indicated in Table 3.

Type of sensor	Datapoints
Power consumption (4 laptops)	18977
Motion detector	3301
CO_2 concentration	39349
Door contact	2100
Window contact	615
Indoor temperature	39373

Table 3: Raw sensor data collected over 11 days

Figure 4 shows the result obtained from the learned decision tree considering all the features as input to the detection model, where we plot both actual occupancy profile and the estimated profile as a graph of number of occupants with respect to time (quantum time is 30 minutes). The accuracy achieved is 46% (number of correctly estimated points divided by the total number of points), and average error 0.7 (average distance between actual points and estimated points). In addition, for each level some performance measures, and their support (the number of times an instance of a given class appears in the data) have been shown in Table 4. The following measures have been considered:

- Precision: Fraction of elements correctly classified as positive out of all the elements the algorithm classified as positive. $\frac{t_p}{(t_p+f_p)}$
- Recall: Fraction of elements correctly classified as positive out of all the positive elements $\frac{t_p}{(t_p+f_n)}$
- F1-score: The harmonic mean of precision and recall $2 * \frac{(precision*recall)}{(precision+recall)}$. The best score possible for a class is 1 and worst is 0.

where:

t_p: true positive i.e. element predicted to be in class i is really in class i

f_p: false positive i.e. element predicted to be in class i is really not in class i

f_n: false negatives i.e. element predicted not to be in class i, but is really in class i

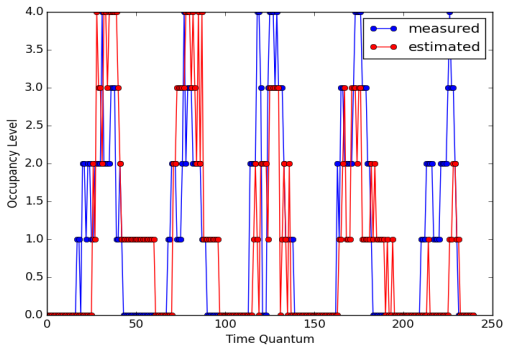

Figure 4: Occupancy estimation from DT using all features

	precision	recall	f1-score	support
level 0	0.73	0.712	0.72	130
level 1	0.11	0.19	0.14	31
level 2	0.40	0.18	0.25	45
level 3	0.17	0.19	0.18	21
level 4	0.00	0.00	0.00	13
avg/total	0.50	0.46	0.47	240

Table 4: Decision tree classification results with considering all features

Figure 5 shows the result obtained from the decision tree considering only the main features. It leads to improvement in occupancy estimation with an accuracy of 65% and an average error of 0.47 occupant in average. Additionally, the results indicate that motion detector, power consumption, CO_2 concentration and door contact have the largest correlation with the number of occupancy detection.

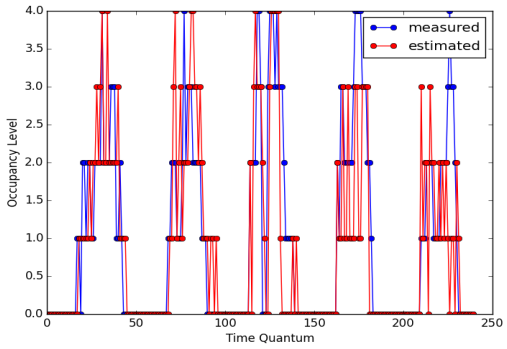

Figure 5: Occupancy estimation from DT using main features

	precision	recall	f1-score	support
level 0	0.89	0.92	0.90	130
level 1	0.30	0.42	0.35	31
level 2	0.50	0.31	0.38	45
level 3	0.25	0.29	0.27	21
level 4	0.36	0.31	0.33	13
avg/total	0.66	0.65	0.65	240

Table 5: Decision Tree classification results after selecting main features

Comparing the results obtained from the decision tree using all the features (Table 4) with the one using only the main features (Table 5), a significant improvement in occupancy estimation for all levels can be observed.

Decision tree structure

Decision tree structure represents the classification rules, data are split at each node in a tree according to a decision rules, which corresponds to nested if-then-else rules. In the if part of such a rule, the decision is made based on a feature of the data record. These rules are for both binary ($f_i \in \{0,1\}$) and continuous data. For example, Figure 2 shows part of the final tree structure with root node and some leaf nodes. Nodes in the tree are named as follows.

X0: CO_2 average value
X1: CO_2 derivative
X2: Motion detector fluctuations
X3: Occupancy estimation from power consumption
X4: Time slot
X5: Door position

In this example, the root is the feature test $X2$ i.e. CO_2 average value is the most meaningful feature in terms of information gain, and the subtrees occur due to the rules:

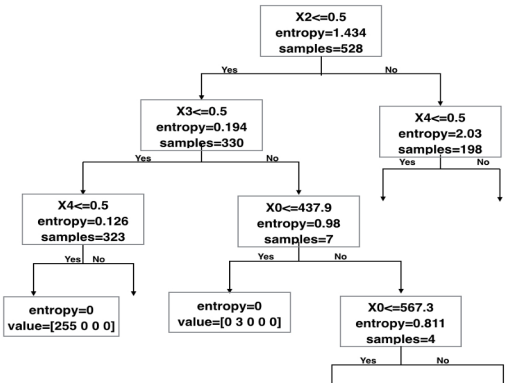

Figure 6: Part of final Decision Tree structure

if $X_i \leq$ *threshold* **then**
  left child node
else
  right child node
end if

These rules are repeated in order to build all sub-trees and determined the classification levels, depending on the most informative feature in each subtree i.e.:

- X2∧X3∧X4 → classification level1.
- X2∧X3∧X0 → classification level2.

For binary data the threshold is 0.5, while for continuous data is defined by sorting the data ascending, then calculate the mid point between each two points (f_i,f_{i+1}), and choose an accurate value for the threshold to be used in the decision tree (Mitchell, 2007).

6 CONCLUSIONS

A supervised learning approach has been proposed in this paper to estimate the number of occupants in an office setup. It results in a virtual sensor that relies on other sensors but with a superior performance. The proposed process makes it possible to determine valuable sensors using the concept of information gain. In the proposed work, motion fluctuation counters using PIR sensors, power consumption sensors, CO_2 mean and derivative, and the door position are found to be the most interesting sources of information. The estimation of the number of occupants using a physical CO_2 model is also very promising but an additional CO_2 sensor has to be installed in the corridor to improve results when the door is opened. Decision trees have been obtained using C4.5 classification algorithm. Occupancy estimation using these trees gave

a superior performance with an average estimation error of 0.47 occupants for a test period of one week.
Supervised learning has been done using 2 video cameras but this approach is limited because of privacy issues. Another option has been envisaged: using discrete feedback from occupants themselves such as with a keyboard or any other means. In addition, because decision trees are human readable, they can be adjusted using expert knowledge. For instance, thresholds can be adjusted and nodes for which information is not available can be removed depending on the considered living areas. The two extensions can be combined to avoid the use of video cameras. It will be investigated further in the future. Estimating the number of occupants is very interesting in many areas: simulating occupant behavior at design stage, predicting the number of occupants at energy management stage, discriminating physics from usage at diagnosis stage etc.. The proposed approach can be extended to the estimation of occupant activities, which is useful to develop interactive systems where relevant advice can be provided to occupants at relevant times.

REFERENCES

Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., and Weng, T. 2010. Occupancy-driven energy management for smart building automation. In *Proc. 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 1–6. ACM.

Aglan, H. A. 2003. Predictive model for co2 generation and decay in building envelopes. *Journal of applied physics*, 93(2):1287–1290.

ASHRAE, Atlanta, G. 1985. *Fundamentals American Society of Heating, Refrigerating and Air-Conditioning Engineers*. Fundamentals American Society of Heating, Refrigerating and Air-Conditioning Engineers.

Erickson, V. L., Carreira-Perpiñán, M. Á., and Cerpa, A. E. 2011. Observe: Occupancy-based system for efficient reduction of hvac energy. In *Information Processing in Sensor Networks (IPSN), 2011 10th Int. Conf. on*, pages 258–269. IEEE.

Haldi, F. and Robinson, D. 2009. Interactions with window openings by office occupants. *Building and Environment*, 44(12):2378–2395.

Kashif, A., Dugdale, J., and Ploix, S. 2013. Simulating occupants'behavior for energy waste reduction in dwellings: A multiagent methodology. *Advances in Complex Systems*, 16(04n05):1350022.

Kleiminger, W., Beckel, C., Staake, T., and Santini, S. 2013. Occupancy detection from electricity consumption data. In *Proc. 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8. ACM.

Lam, K. P., Höynck, M., Dong, B., Andrews, B., Chiou, Y.-S., Zhang, R., Benitez, D., Choi, J., et al. 2009. Occupancy detection through an extensive environmental sensor network in an open-plan office building. *IBPSA Building Simulation*, 145:1452–1459.

Milenkovic, M. and Amft, O. 2013a. An opportunistic activity-sensing approach to save energy in office buildings. In *Proc. Fourth int. conf. on Future energy systems*, pages 247–258. ACM.

Milenkovic, M. and Amft, O. 2013b. Recognizing energy-related activities using sensors commonly installed in office buildings. *Procedia Computer Science*, 19:669–677.

Mitchell, T. M. 2007. *Machine Learning*. Number 432. McGraw-Hill Science/Engineering/Math.

Nguyen, T. A. and Aiello, M. 2012. Beyond indoor presence monitoring with simple sensors. In *Proc. PECCS*, pages 5–14.

Padmanabh, K., Malikarjuna V, A., Sen, S., Katru, S. P., Kumar, A., Vuppala, S. K., Paul, S., et al. 2009. isense: a wireless sensor network based conference room management system. In *Proc. First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 37–42. ACM.

Page, J., Robinson, D., and Scartezzini, J.-L. 2007. Stochastic simulation of occupant presence and behaviour in buildings. In *Proc. Int. IBPSA Conf: Building Simulation*.

Philipose, M., Fishkin, K. P., Perkowitz, M., Patterson, D. J., Fox, D., Kautz, H., and Hähnel, D. 2004. Inferring activities from interactions with objects. *Pervasive Computing, IEEE*, 3(4):50–57.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning*, 1(1):81–106.

Quinlan, J. R. 2014. *C4. 5: Programs for machine learning*. Elsevier.

Roulet, C., Cretton, P., Fritsch, R., and Scartezzini, J. 1991. Stochastic model of inhabitant behavior in regard to ventilation. Technical report, Ecole Polytenique Federale de Lausanne.