Fusing Sensors for Occupancy Sensing in Smart Buildings*

Nabeel Nasir, Kartik Palani, Amandeep Chugh, Vivek Chil Prakash, Uddhav Arote, Anand P. Krishnan, and Krithi Ramamritham**

Department of Computer Science
Indian Institute of Technology Bombay,
Mumbai, India
{nabeel12,kartik,amandeepchugh12,
vivekcp,uddhava,anandkp,krithi}@cse.iitb.ac.in
http://www.cse.iitb.ac.in

Abstract. Understanding occupant-building interactions helps in personalized energy and comfort management. However, occupant identification using affordable infrastructure, remains unresolved. Our analysis of existing solutions revealed that for a building to have real-time view of occupancy state and use it intelligently, there needs to be a smart fusion of affordable, not-necessarily-smart, yet accurate enough sensors. Such a sensor fusion should aim for minimalistic user intervention while providing accurate building occupancy data. We describe an occupant detection system that accurately monitors the occupants' count and identities in a shared office space, which can be scaled up for a building. Incorporating aspects from data analytics and sensor fusion with intuition, we have built a *Smart-Door* using inexpensive sensors to tackle this problem. It is a scalable, plug-and-play software architecture for flexibly realizing smart-doors using different sensors to monitor buildings with varied occupancy profiles. Further, we show various smart-energy applications of this occupancy information: detecting anomalous device behaviour and load forecasting of plug-level loads.

Keywords: Smart Door, Smart Building, Energy Saving, User Comfort, Electrical Energy.

1 Introduction

Designing new "green" buildings and retrofitting existing buildings with green technologies pose numerous research challenges but essential for society. Two of the main motivations for this transition towards a smarter and greener electricity grid have been capping total usage or flattening the peak and reducing the carbon footprint and costs. This has sparked new interest in developing smarter

* The authors would like to thank DeitY, Govt. of India and TCS for their generous support of this work.

** Corresponding author.

R. Natarajan et al. (Eds.): ICDCIT 2015, LNCS 8956, pp. 73–92, 2015.

Table 1. Approaches to tracking occupancy using various sensors and their fusion

Sensor	**Advantages**	**Disadvantages**	**Occupancy Information**
Passive Infra Red	Cheap; Scalable; RT Response; No User Intervention	When users become stationary (eg., working on PC) room occupancy detected as NIL	Presence of occupants in room
CO_2 [3]	Cheap; User intervention is not required; Scalable	Response is not real-time; Accuracy reduces when there is proper air circulation	Presence of occupants in room
Radio Frequency Identification	Accurate when proper measures taken;Real Time response; No user intervention req.	User should carry RFID tag; Tags must not be kept near metallic objects; Accuracy depends on speed of walking[5]	Count and identity
Face Recognition [9]	No user Intervention required	Computationally challenging; Expensive; Accuracy is less for moving objects; Not easily scalable; Requires 2 cameras to detect Entry and Exit	Count and identity
Sound Detection [10]	Cheap; Real-Time Response; Scalable	Not suitable for environments like labs and libraries	Presence of occupant in room
PIR+Reed[1]	Cheap; Scalable	The sensor fails to detect occupancy in a multi user environment	Presence of occupant in room
WiFi+Lan+ IM+Calender +Access Badge[6]	Cheap;Scalable	Accuracy reduces when users don't comply to the rules of the system, WiFi can't distinguish a person who is right outside the room, will be detected as inside the room	Count and identity

buildings, which can sense instances of undesired energy usage and intelligently take decisions towards curbing such occurrences. For instance, smart buildings may use sensors to track occupants and opportunistically disconnect loads in empty rooms; we use the term "load" to refer to any appliance or device that draws electricity.

Smart buildings inherently possess knowledge about their energy consumption at any given instant. Considering that smart meters that record aggregate power at fine granularity with high accuracies are ubiquitous in modern residential and commercial environments, it can be assumed that most new buildings will possess this level of smartness. What can accentuate the smartness, is the ability to calculate how much energy should optimally be consumed, given the various parameters (like temperature, relative humidity, etc.) that influence energy usage. One such parameter is the occupancy state of the building: the electricity demand of a building is driven by its occupants. Having real-time knowledge about the occupants adds to the building's intelligence significantly. This information can be put to use not only for energy savings but also for other important applications ranging from knowing the health of appliances to priority evacuation of children and the elderly in times of emergencies.

A review of existing occupancy monitoring systems shows that even the most accurate of them have certain bottle-necks. For example, with biometric identification systems, which score well on accuracy, people have to stop at a place to record their entry. This might be acceptable for a one-time check-in into a building. However, for room-level occupancy monitoring, the system becomes inconvenient due to the the fact that occupants need to register their identity every time they enter/exit the room. Another familiar occupant monitoring system is the Active RFID based system. RFID systems are generally used for access control in buildings, but it also logs the occupant identity which can be used to monitor occupancy. Active RFID systems, unlike passive RFID systems, do not require a stop and swipe mechanism. They are also known to have high accuracies. But in order to obtain high accuracy users need to carry the tag at all times, they also have to be careful not to keep the tag next to metallic objects and be wary of the speed with which they walk across the RFID reader. Moreover, RFID readers are expensive and can't be deployed in each and every doorway of the building. In addition, these systems are deactivated during evacuation of buildings so that the access control of the building doesn't hinder the free flow of occupants. If these systems are disabled during evacuation then finding who and how many are still inside the building becomes troublesome. In general, sensors need to be examined against characteristics such as, accuracy, cost, reliability, interruption/inconvenience caused, computational complexity and the occupancy data they help infer (how many, who and where). Table 1 summarizes the pros and cons of existing occupancy tracking systems along with the occupancy questions they answer. Given this, it is clear that for a building to have a real-time view of occupancy state, and use it intelligently, there needs to be a smart fusion of cost-effective, not-necessarily-smart, yet accurate-enough sensors. Such a sensor-fusion system should aim to respect the users' natural behavior by allowing for minimalistic user intervention while providing accurate occupancy data about the building. [7] talks about a probabilistic approach of identifying occupants in a home environment, focuses on using height sensors with the main goal as tracking the occupants. Our work, although has used few similar sensors, focuses more on the energy saving application of the occupant data and has been implemented using Machine Learning techniques in a lab environment where the number of users are much larger than a home environment. These considerations prompted the work reported in this paper which lead to the following contributions:

Firstly, we propose and report on the experiences with a set of novel solutions to the occupancy tracking problem:

- Incorporating aspects from data analytics and sensor fusion, combined with intuition, we have developed a *smart door* to tackle the occupancy detection problem.
- The experience with the building of multiple versions of the smart door lead to the design and creation of a plug-and-play architecture to flexibly address the door's controller's design and construction.
- We have installed the whole system in our lab's premises and have gathered extensive experiential data. We report on the occupancy prediction accuracy results, offer a comparative analysis of the operation and usefulness of different combinations of sensors and draw inferences that will be useful for researchers and practitioners alike.

It is important to point out that the smart door design along with embellishments such as a personalized appliance control system can help with matters related to occupancy, such as "how many are in a given space" and "who is in a given space".

Secondly, with occupancy-related data in hand, in conjunction with smart meter data, we show how some interesting energy-related practical questions can be answered:

- How can smart meter data be used to detect the occurrence of "unusual", "abnormal" or "unexpected" energy usage profiles?
- How can knowing occupant identities help forecast plug-level load better?
- Can knowing "who" help give personalized actionable energy savings advice?

These experiences clearly demonstrate that occupancy matters!

2 The Smart Door

In this section, we describe the *Smart Door*; a system capable of providing occupant identities and count, in a room. The smart door achieves occupancy identification and counting without any user interaction, supports easy sensor integration and at the same time is cost effective. The design philosophy of smart door is to enable its user to add sensors easily based on the accuracy required for the occupant detection/identification; thanks to the plug and play architecture described in Section 4. The base version of smart door has two LDR-laser pairs for occupancy counting and detecting direction of movement. The sequence of lasers being cut determine if an occupant is entering or exitting a room. Hence the base version of the Smart Door was capable of only keeping the real-time occupant count and not the user identification. Further versions have both paraphernalia and capability to infer identity of the occupants too. The plug and play architecture helped to experiment with multiple occupant identification sensors used to measure height, weight and skeletal parameters and are detailed in Section 2.1.1, 2.1.2 and 2.1.3 respectively. Their accuracies with various learning algorithms are also detailed in the section. We believe that the smart door can be implemented under $100 when manufactured in large quantities, making it a cheap system for building wide implementation.

2.1 Occupant Identification

We believed that sensing signatures from the human body, when people walked through the smart door, could be used to uniquely identify people. To achieve cost effectiveness, only signatures which could be sensed during both entry and exit, using a single sensor, were considered.

When a person passes through the smart door, his/her signature is obtained by the controller board, using the sensor. A Raspberry Pi board running Linux is used as the controller board. A tablet is deployed at the entrance using which people can manually tag their identity as they enter. The measured signature along with the tagged identity information is used as training data for a supervised learning algorithm. The algorithm, after sufficient training, would then be able to predict occupants' identities based on their body signature. The entire system, when used with a height sensor is designed as shown in Figure 1. Other versions maintain the same design except for the sensor, which is switched on depending on what body signature is being sensed.

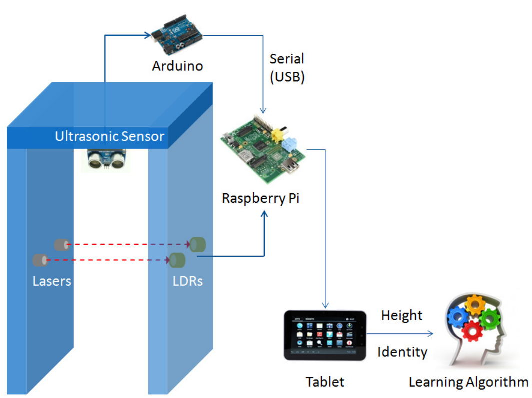

Fig. 1. Occupant Identification: Height

The learning algorithm used for our implementation was Support Vector Machines (SVM). The prediction results for each individual signature (height, skeletal parameters and weight) and a comparison of SVM with Naive Bayes classifier is presented in Section 2.3.

2.1.1 Occupant Identification: Height. Height was chosen as the first signature that was sensed, since it could be easily sensed and does not vary significantly over time, in adults.

An ultrasonic sensor is mounted on the smart door, which gives the distance to any object placed under it. The distance values are recorded between two subsequent laser obstructions. We take the minimum of all the recorded values, since the minimum distance from the ultrasonic sensor is when the beam hits the topmost point of the head of a person and reflects back. From this obtained minimum distance and the height of the smart door frame, the height of the person is estimated.

2.1.2 Occupant Identification: Weight. A weight mat can be used to measure weight even when a person is walking, and moreover only a single sensor is required for measurement during entry and exit. Although it can be argued that the weight measured for a moving object will be less when compared to a static measurement, it is important to note that the goal is to obtain a unique signature. This goal is still achieved considering that the difference in weight propagates through the data.

A weight mat was designed by attaching strain gauges underneath a wooden board - four gauges fixed near the corners of the board. The strain on a strain gauge produces a change in resistance and a Wheatstone bridge circuit can be used to measure this. Proper calibration of all four gauges can thus give the weight of a person standing or walking on the board. This board is placed between the legs of the walk-through frame of the Smart Door to obtain weights of people as they enter/exit the room.

2.1.3 Occupant Identification: Skeletal Parameters. Adhering to the design philosophy of choosing signatures which could be sensed in both directions using a single sensor, we decided to get data about occupants' skeletal structures. The Microsoft Kinect sensor, which uses its depth sensing technique to obtain these parameters, was used.

As shown in Figure 2 [8], the Kinect can deduce the skeletal structure for a person by obtaining the positions of about twenty joints of the body, using its infrared and depth sensors. In order to obtain signatures useful to uniquely identify humans, skeletal points that don't change much, across multiple sensing runs, were to be selected. A subset of points in the torso were identified as being the most consistent: shoulder width, torso length and hip width were picked as the signatures.

The Kinect was kept at a height, few feet away from the entrance into the room, to capture the signatures when people entered and exited the room. As with the height measurement, the Kinect values were record between subsequent laser cuts of the smart door.

2.2 Results

The data acquired comprised of around 5000 records - signatures along with the tagged identity information collected during entry and exit. The data set for all

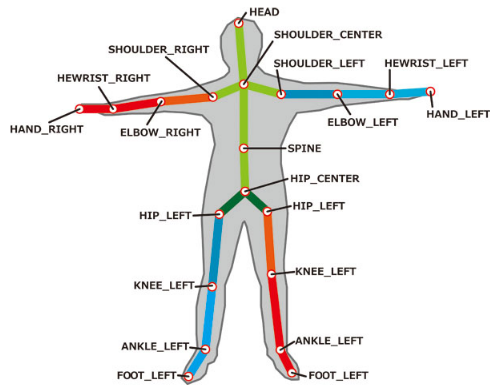

Fig. 2. Skeletal structure obtained from Kinect [8]

the five signatures taken individually - height, weight, shoulder width, hip width and torso length were put through an 10-fold cross validation using an SVM classifier as well as a Naive Bayes classifier and the accuracy was measured. The results are shown in Figure 3.

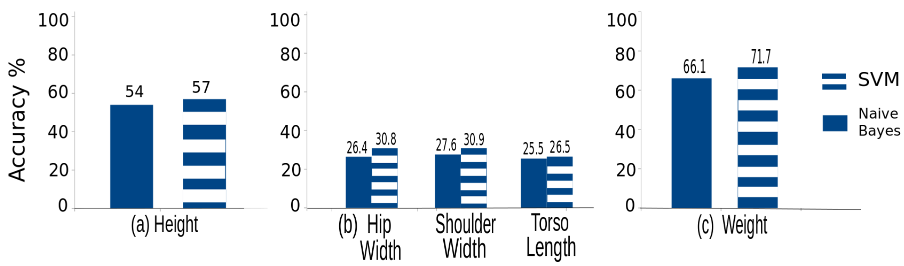

Fig. 3. Prediction Accuracy: (a) Height (b) Skeletal Parameters (c) Weight

Kinect parameters taken individually fared poorly (30%). Height individually achieved double the accuracy of the Kinect (60%) and weight fared even better than height with an accuracy around 70%. One worthwhile observation is that the low-cost height and weight sensors fared better in comparison to the costlier Kinect sensor (it can, although, be argued that the Kinect sensor can be put to a variety of other uses like face recognition, voice recognition, etc. but these do not adhere to our design philosophy of a single sensor sensing for both entry and exit). It was also observed that for the data set we had, SVM had a clear edge over Naive Bayes in terms of prediction accuracy.

3 Sensor Fusion

The results obtained in Section 2 indicate that individual signatures taken from users were not necessarily unique to the individuals. However, we hypothesized that multiple such signatures taken from individuals and fused together had the potential to increase identification accuracy. In this section we show how this accuracy significantly went up when intelligent sensor fusion was performed.

3.1 Results

The fusion of height data acquired from the ultrasonic sensor and the skeletal data obtained from the Microsoft Kinect provided much higher accuracy than they provided individually (Figure 4).

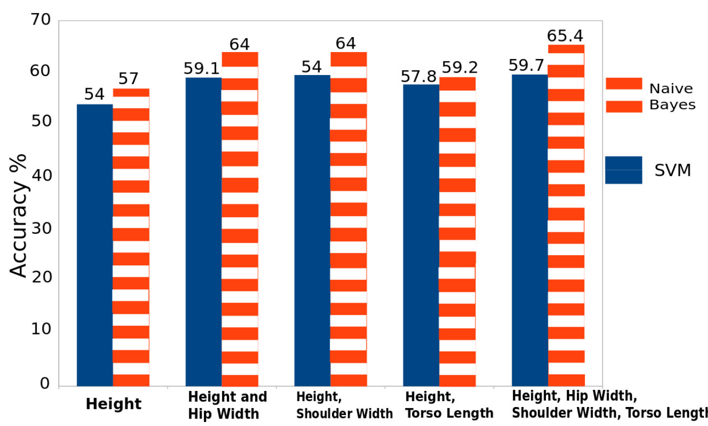

Fig. 4. Prediction Accuracy: Height and Kinect features

When multiple combinations of height and skeletal features are examined, an interesting trend was noticed: any combination of intuitively correlated features leads only to a small increase in accuracy. In order to mathematically examine this hypothesis, Pearson correlation was applied to combinations of two features. Pearson correlation is used to show how strong the association is between two variables. It ranges from +1 (indicating direct proportionality between variables) to -1 (indicating inverse proportionality). A correlation of 0 indicates that the two variables are independent of each other.The results are shown in Table 2.

Table 2 indicates that the highest correlation is between the hip-width and the shoulder-width (0.733). Table 3 shows how their combination performs in terms of occupant identification accuracy.

The result indicates that as a virtue of the high correlation, no additional information is added to the model for it to improve. In order to test if less correlation meant higher accuracy, we calculated the correlation between height

Table 2. Correlation between features

Feature 1	Feature 2	Pearson Correlation
height	weight	0.599
height	hip-width	-0.008
height	shoulder-width	0.034
height	torso-length	0.173
weight	hip-width	0.066
weight	shoulder-width	0.088
weight	torso-length	0.385
hip-width	shoulder-width	0.733
hip-width	torso-length	-0.207
shoulder-width	torso-length	-0.152

Table 3. Prediction Accuracy: hip-width, shoulder-width

Features	Accuracy%
hip-width	30.8
shoulder-width	30.9
hip-width, shoulder-width	31.0

and hip-width which are almost uncorrelated (-0.008). As can be seen from Figure 4 the combination accounts for an accuracy of 64%.

In order to validate our original claim that adding sensors to the occupancy detection system makes it more intelligent, we tested how weight performs when combined with height. Figure 5 shows that a fusion of these two human parameters increases prediction accuracy to 87.1%. What makes this result even more exciting is the fact that a simple combination of two low-cost and readily-available sensors produces such high accuracy.

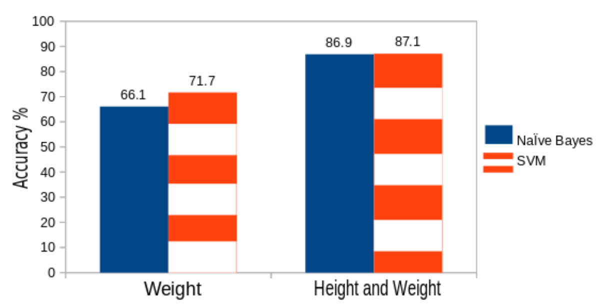

Fig. 5. Prediction Accuracy: Height and Weight

This performed extremely well considering that the two parameters have a relatively high correlation (0.59). Analyzing this led to multiple plots like the one shown in Figure 6. The first plot shows the probabilities with which an individual is identified among a certain subset of people with similar heights by the smart door described in Section 2. It becomes evident that it is hard to distinguish the individual uniquely. However, as shown in the second plot, for the same subset of people, by using weight as a metric for identification the system can identify them uniquely. Thus, along with correlation, it becomes important for the sensors fused to be able to understand the distribution of features among the occupants. This led to the formulation of a software architecture that seamlessly incorporates these learnings in order to monitor occupants for a room of any occupancy profile.

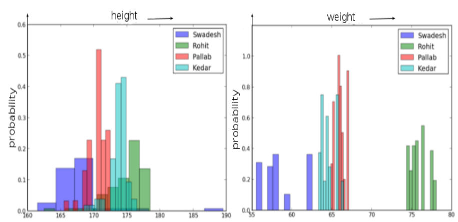

Fig. 6. Distribution of heights and weights for a subset of people

4 The Plug and Play Architecture

While installing various sensors to the smart door, we learnt a few key lessons. First, the accuracy of the smart door's prediction improved when an additional sensor was added, which meant that there needed to be support for adding multiple sensors to the setup. Second, the task of adding a new sensor can be very tedious and user-unfriendly. In order to create a system which, from the users' perspective, was a plug and play model where they could just plug in a sensor to the controller and hope to achieve improvement in prediction accuracy, we developed a scalable software architecture for the smart-door.

The foundation for such a model was based on two key ideas; first, the user should not have to make any changes in the code on the controller and second, any added sensor should seamlessly fit into the system and start improving the learning model and hence increase prediction accuracy.

4.1 Architecture

The Smart-Door has a master node to which all sensors(slave nodes) are wired directly or connected wirelessly. The master node defines the actions that the associated sensors must perform by exchanging messages with them. The master is responsible for detecting entry/exit events, framing meaningful messages for the slave nodes and reporting failures in the nodes to the administrator. The master also collects data from all the local nodes and sending it to a common database.

A slave node consists of a sensor that is attached to a micro controller board capable of storing sensed readings and performs local node aggregation. There are two kinds of slave nodes associated with the master: local nodes, which receive commands and sends data to master through wired connections and remote nodes, which performs message and data exchange with the master over a Wi-Fi network. Figure 7(a) gives a schematic detailing of these connections.

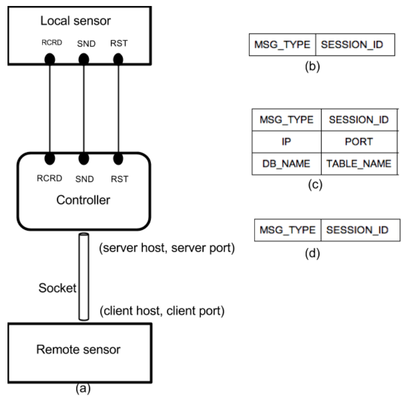

Fig. 7. (a) shows the schematic for sensor connections to controller. (b), (c), (d) illustrate formats of RCRD, RST and SND messages respectively, exchanged between network sensor and the controller.

A configuration file exists on the master which contains a list of the local and remote nodes and their locations (IP address in case of remote nodes and pin numbers in case of local nodes). Whenever a new sensor is added to the system, the administrator adds a new entry to the configuration file stating the type of sensor and its location. Thus, no change is made to any code. Internally, the master reconfigures and automatically incorporates the new sensor into the mix.

4.2 Messaging Protocol

A unique messaging protocol is designed in order for the master to communicate with the slave nodes. The messaging protocol is different for the local nodes and remote nodes. The activity diagram for the master and the state diagram for the slaves are shown in Figures 8 and 9.

4.2.1 Local Node Messaging

- Upon detecting a possible entry/exit event (recorded by emitting sensor 1 (Section 2.1)), the master sets the RCRD pin HIGH on the local nodes. This causes an interrupt in the nodes, which then starts data sensing.
- A possible entry/exit event may either be successful (recorded by emitting sensor 2) in which case, the RCRD pin is set LOW and the SND pin is set HIGH, which is an instruction to the node to calculate an average of recorded values and send the data to the local node manager via the serial interface.

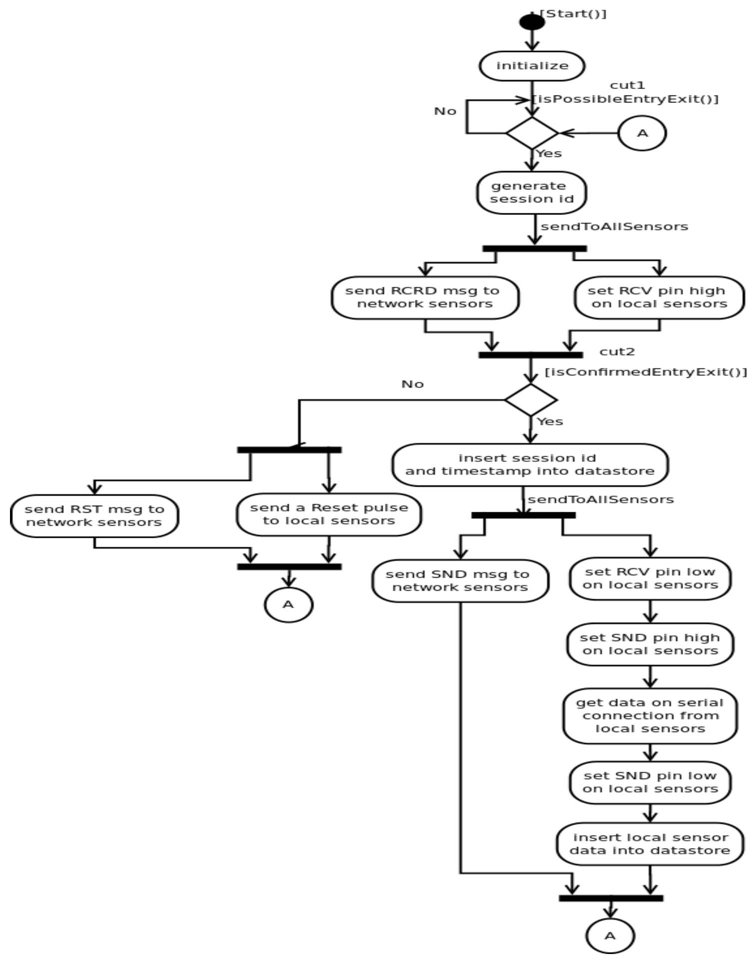

Fig. 8. Activity diagram for controller

- If the possible entry is not successful, i.e., it times out, the RCRD pin is set LOW and a RST pulse is sent which clears the data structures on the local nodes.
- In case of a successful entry/exit event, the host inserts an entry into the database table with the unique session_id and the timestamp. The data received by the local node manager is then updated in the the table.

4.2.2 Remote Node Messaging

- A possible entry/exit event triggers the master to frame a RCRD message (Figure 7 (b)) which is transmitted to the client ports on all the remote nodes. This prompts the remote sensors to begin storing sensed data.
- If the entry/exit event is successful, the master frames a SND message (Figure 7 (c)) which contains details about the common database where the

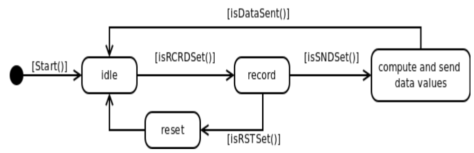

Fig. 9. State diagram for local and network sensors

sensor values are stored. The remote node then calculates the average and updates the entry in the database table.

– In case of a timeout event, a RST message is sent (Figure 7 (d)) which clears all the sensed data stored in the remote node.

All remote node messages are sent over Wi-Fi, thereby using existing infrastructure.

Since most low cost processing boards do not have a real time clock on them, a unique session_id is used for synchronization instead of the conventional timestamp synchronization. The time of the event is marked by the host which is Network Time Protocol synchronized. Thus, exact times of entry/exit events are recorded with high precision (to the second).

4.3 Database Design

Considering the goal of the smart door is to achieve maximum scalability while allowing minimal user intervention, another design choice that has been made is the use of a NoSQL database [12] instead of a traditional relational database model. With the addition of sensors, we wanted horizontal scalability and high write operation performance, both of which were achieved very well by the open source NoSQL database, MongoDB. This made it the database of choice for the software architecture.

4.4 Cost Considerations

From the results presented, the smart door is able to predict identities of people at fairly high accuracy and is scalable owing to the plug and play architecture, enabling fusion of additional sensors with ease. It is interesting to analyze the deployment cost of such a system including only the most essential sensors which were identified from the experiments. Such a minimalistic Smart Door comprises of a height sensor, a weight sensor, two Laser-LDR pairs, an Arduino board, and two low-cost Android Tablets. From Table 4 we can see that these components amount to around $150 and this cost can be further cut down upon mass production. At this cost, the system can be deployed at all doors in a building without incurring much. The applications that arise from deploying in such a scale are appealing and described in the following section.

5 Applications Enabled by Occupancy Information

The fusion of occupancy data and electricity consumption data can enable a rich set of applications necessary for smart buildings to become smarter and

Table 4. ComponentCost for Smart Door

Component	Cost ($)
Ultrasonic Sensor	3
Weight Sensor	10
2 Laser-LDR Pairs	10
Arduino Board	25
2 Android Tablets	100

greener. In this section we explore some applications which use this fused data, focusing on cases other than the conventional ones like load forecasting of HVAC loads [2] [4] and room automation – in order to provide insights into other important energy saving applications. The applications described here stem from our experiences with buildings at IIT Bombay.

5.1 Auxiliary Sensing and Actuation for Energy Applications

5.1.1 Smart Meter Setup. The smart meter's ability to provide high accuracy consumption data at fine frequencies makes it an important sensor. We use three EM6400 smart meters (named LSM-A, LSM-P, and LSM-F respectively) in order to understand the consumption profile of our lab. Table 5 shows what appliances' usage the respective smart meters monitor.

Table 5. SEIL smart meter connection and device profile

<table>
<tr><td rowspan="3">LSM-A</td><td>Phase 1</td><td>AC 1 & AC 4</td></tr>
<tr><td>Phase 2</td><td>AC 2</td></tr>
<tr><td>Phase 3</td><td>AC 3</td></tr>
<tr><td>LSM-P</td><td>Phase 1, 2, 3</td><td>Computers and Wall Sockets</td></tr>
<tr><td rowspan="2">LSM-F</td><td>Phase 1</td><td>Light Arrays and Fans</td></tr>
<tr><td>Phase 2, 3</td><td>Null</td></tr>
</table>

5.1.2 Relay Control. The fans, lights and air conditioners in the lab are controlled using a relay system. Occupants turn ON/OFF their devices after logging into a web portal, which sends actuation messages to the relays. The resulting knowledge of who uses what devices allows for the appliance preferences of the occupants to be learnt.

5.2 Anomaly Detection

The plot in figure 10 compares the electricity consumption profile of our academic building on a day in which one of the 185 air conditioners in the building was malfunctioning, to the day on which the anomalous device was rectified. The exact details the anomaly are discussed later in this section. When we examine the peak power and total energy consumption, shown in Table 6, on these days, we notice that on the day of the AC anomaly, the peak was higher by 31 kW and the energy consumed was higher by almost 222 kWh. Considering that our electricity usage is charged at $0.10/unit, we could have saved $22.2/ day had the anomaly been identified earlier. This is admittedly very small compared to the average electricity bill for the academic building, which is around $13300/month.

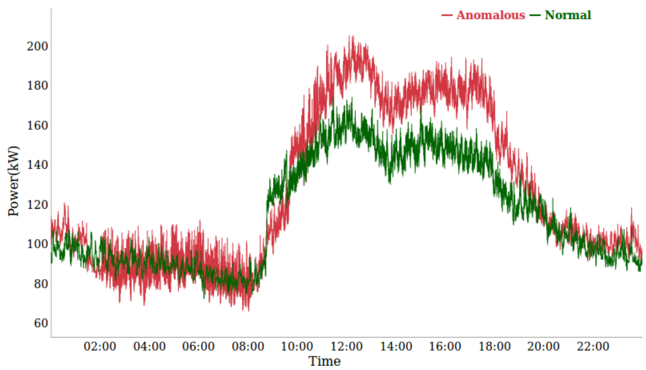

Fig. 10. Comparison of an anomalous day to a normal day

Considering the relatively negligible saving, identifying the anomaly might not seem like an issue worth addressing. However, the seriousness can be realized when we understand that in a building with a large number of such AC units (185 in our case), with each room fitted with about 3-4 of them, these anomalies go easily undetected. The primary cause for this is the fact that the non-anomalous ACs in the room compensate for the lack of cooling by the malfunctioning one. Now, when we look back at the problem and realize that an excess usage like this might go unnoticed for months, as it has been in our case before we installed smart meters, we realize that a single malfunctioning AC accounts for almost 5% ($666) of the monthly electricity bill for the building.

Table 6. Load profile – on an anomalous day (Jul 10) compared to the day it was rectified (Jul 11)

Date	Peak Power (kW)	Energy Consumed (kWh)
July 10	204.98	2975.87
July 11	173.93	2753.02

Smart-meters have been put to use, beyond their conventional usage [11] of monitoring electricity consumption, to detect such anomalous behavioral patterns. If the plot for the anomalous device is examined, we notice periodic spikes in the power drawn. It has been found that these spikes are due to a commonly occurring fault in ACs: the compressor overload trip, which is caused by compressor malfunction or non-function. In this section we provide an algorithm that successfully detects such an occurrence, using the smart meter data for the building, and isolates the fault to a small set of devices.

In order to first identify the anomaly in real time, a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is run over a 15 minute window of data. DBSCAN is a scalable and almost linear algorithm which identifies clusters in large spatial data sets using only one input parameter and gives out information about outliers in the same. As an input, the algorithm takes the magnitude of all the power values that are observed within the interval of interest. The output is a cluster of step-ups that are unusual for the profile. It is assumed that no two devices turn on at the same instant (1 second as per our smart-meter's resolution).

input : Power data (per second), Power-surge threshold (T), minimum neighbor distance (min_dis), minimum points for a cluster (min_pts)
output: Unusual clusters of *power-surges* based on magnitude
Calculate *power-surges* based on consecutive data points;
Filter them based on T, store in list_PowerSurge;
Calculate global_mean and global_std_deviation of list_PowerSurge;
Run DbScan (min_dis, min_pts) on list_PowerSurge;
for *cluster with mean > (global_mean+global_std_deviation)* **do**
  Mark the cluster as **Unusual**;
end

Algorithm 1. Clustering *Power-Surges* with unusual magnitudes

Once the spikes are identified, the anomaly is isolated to a small set of devices. The flowchart in Figure 11 succinctly describes the fault localization algorithm. If a spike is detected by the peak detection algorithm, the spike is evaluated to find its phase information. This information is passed to a process that uses the occupancy of the building in the relevant 15-minute interval and the set of devices that those occupants are known to have used in the past (learnt over a period of time using the system discussed in Section 4.1.2) to decipher which given appliances on that phase are active.

In our experience, the output list generally contains only a set of 3-4 devices. Thus, adding occupancy information leads to quick, almost real-time, identification of anomalous devices. This algorithm has been put to use in the academic building and has helped identify five major anomalies in the two months that it has run.

5.3 Load Forecasting: Plug-Level Loads

In most offices and academic buildings, most of the loads are at the plug-level. These loads, like desktop computers, laptops, printers and copiers are considered to consume significantly lesser energy than HVAC loads. But our experiences in the lab, which closely parallels an office space, taught us that this was not completely true. Figure 12 shows the plot of the energy consumed by various devices

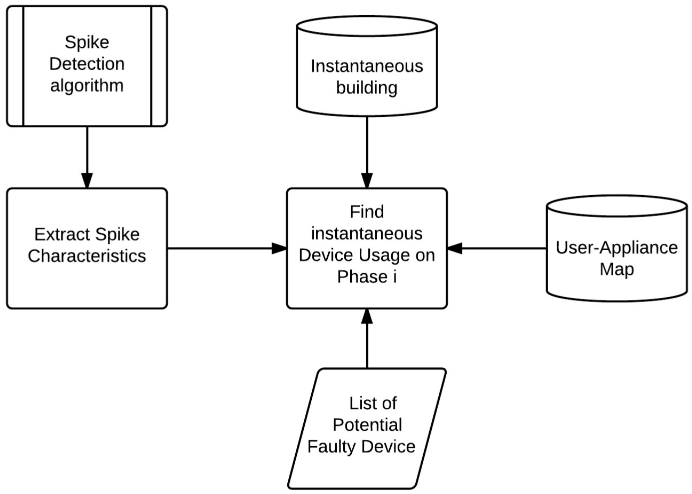

Fig. 11. Algorithm for detecting faulty devices

in this environment. Although the expected behavior of air conditioners consuming significantly higher power than the other loads is noticed in the summer months (April, May, June), during the winter months, the plug-level usage becomes comparable to the cooling load, with plug-level consumption being higher in the month of January. With this trend we expect that for at least half the year, plug-level energy consumption is of prime importance. It is also worth noting that in developing countries like ours, the majority of office spaces lack air conditioning in which case their primary usage comes from plug loads.

These findings motivated us to research how the accuracy of predicting plug-level load is affected by occupancy information.

5.3.1 Methodology. Since we had accurate occupancy and electrical consumption data only for 25 days at the time of experimentation, the time intervals for learning/prediction was chosen as 15 minutes so as to have more records for the experiment. The features considered for learning and their representation is as follows:

- Time of Day - A feature vector representing every 15^{th} minute of the day
- Week End/Week Day - Binary feature representing if its a weekend or weekday
- Who - A feature vector of size k is maintained for each time interval. k denotes the total number of unique persons present in the occupancy records. Each cell indicates the amount of time a person was present in the room, for that interval, normalized to unity. For example, for the 15 minute interval of 21:00 -21:15, if Person_1 was inside the lab for half the interval time, then his feature value would be 0.5. Similarly, if Person_2 was present for 75% of the time, his feature value would be 0.75.

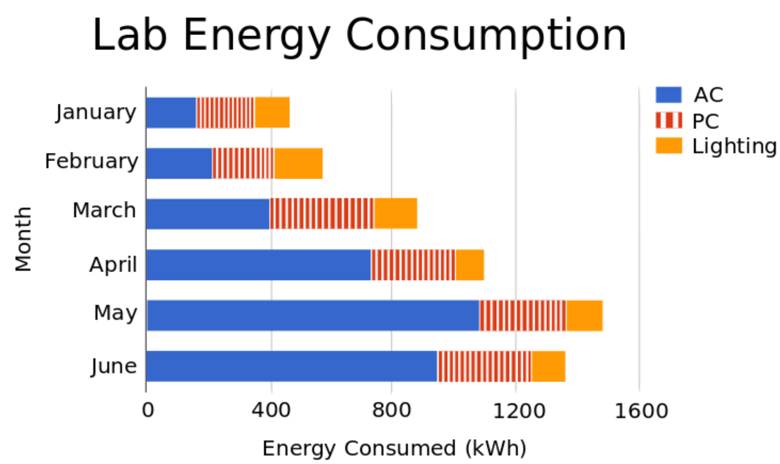

Fig. 12. Month-wise Energy Consumption of Our Lab

– Occupancy Count - Since, a room's occupancy count can change within a time interval, average occupancy count of the room is taken as a feature. This is computed using occupancy count numbers and corresponding time.

In order to forecast the load requirement, we experimented with two well known models: Support Vector Regression (SVR) and Decision Tree Regression (DTR). The performance of the model was judged using the CV-RMSE (Coefficient of Variation of Root Mean Square Error) which measures the difference in values predicted by the model to the observed values.

5.3.2 Results. Different combinations of the features described in the previous section were supplied to the forecasting models. The performance is detailed in Table 7.

Table 7. Plug-level load forecasting results

Interval	Weekday/Weekend Time of Day	Who	Occupancy count	DTR CV(RMSE) %	SVR CV(RMSE) %
15	✓	✓		21.94	18.40
15	✓		✓	25.08	28.1
15	✓			30.36	29.12

One behavior observable from the table is that greater the occupancy information (count, identity) the building has, the better it is able to forecast its energy requirements. The model performs best, with only a 18.4% error, when the identities of the occupants are available. This indicates that a building which knows its occupants' identities performs load forecasting of plug-level loads 36.8% more accurately than a building that just uses calendar information to do the same, thus making it smarter.

6 Conclusions

We have reported on a repertoire of techniques contributing to occupancy detection and using the detected/inferred information for better energy management. Our design and implementation choices were driven by the following considerations:

- *Cost effectiveness*: For example, using occupant signatures which could be sensed using the same sensor during both entry and exit.
- *Minimal User Intervention*: The occupants should be allowed to freely walk in and out of a room and the system should still be able to monitor the occupancy.
- *Resilient to Errors Thereby giving High Accuracy Prediction*: Given the nature of office environments, errors like electromagnetic interference were resolved and accuracies as high as 87.1% were achieved.
- *Extensible Architecture*: Allow seamless addition of sensors, in a plug-and-play fashion, into the mix of existing sensors to improve the functionality/accuracy of the system.

It is clear that a single solution will not fit all occupancy detection scenarios. For example, since many of the actions triggered following occupancy detection are themselves prone to further validation, (for examples see Section 5), depending on the applications at hand, some amount of inaccuracy can be tolerated in occupancy detection and that can be exploited in trading-off between design choices. It is in this context that the flexible architecture described in this paper has a significant role to play.

Incorporating the lessons learnt in sensor fusion from Section 3.1, we are currently building a *sensor-recommender-tool* that aids users in selecting the best set of sensors for their room profile and intended purpose of deployment. Sensor recommendations are made to the user based on how the features of the room's occupants are distributed, existing sensors on the smart door, the desired accuracy and the required cost.

In our future work we also plan to look at further applications, including some that are unrelated to energy management, but are related to smarter building management, such as those that are necessary for emergency management and disaster recovery.

References

1. Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., Weng, T.: Occupancy-driven energy management for smart building automation. In: Proceedings of the 2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, BuildSys 2010, pp. 1–6. ACM, New York (2010), http://doi.acm.org/10.1145/1878431.1878433

2. Balaji, B., Xu, J., Nwokafor, A., Gupta, R., Agarwal, Y.: Sentinel: Occupancy based hvac actuation using existing wifi infrastructure within commercial buildings. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, SenSys 2013, pp. 17:1–17:14. ACM, New York (2013), http://doi.acm.org/10.1145/2517351.2517370
3. Emmerich, S., Persily, A.: State-Of-The-Art Review of Co2 Demand Controlled Ventilation Technology and Application. Diane Publishing Company (2001), http://books.google.co.in/books?id=1hr0Nzju3IYC
4. Erickson, V.L., Carreira-Perpiñán, M.A., Cerpa, A.E.: Occupancy modeling and prediction for building energy management. ACM Trans. Sen. 10(3), 42:1–42:28 (2014), http://doi.acm.org/10.1145/2594771
5. Floerkemeier, C., Lampe, M.: Issues with rfid usage in ubiquitous computing applications. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 188–193. Springer, Heidelberg (2004), http://dx.doi.org/10.1007/978-3-540-24646-6_13
6. Ghai, S., Thanayankizil, L., Seetharam, D., Chakraborty, D.: Occupancy detection in commercial buildings using opportunistic context sources. In: 2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 463–466 (March 2012)
7. Hnat, T.W., Griffiths, E., Dawson, R., Whitehouse, K.: Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors. In: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, pp. 309–322. ACM (2012)
8. Labs, K.: Kinect sdk xna [Programming Guide] (2011), http://www.kosaka-lab.com/tips/2011/06/kinect-sdk-xna.php
9. Lee, K.C., Ho, J., Yang, M.H., Kriegman, D.: Video-based face recognition using probabilistic appearance manifolds. In: Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. I–313–I–320 (June 2003)
10. Padmanabh, K., Malikarjuna V, A., Sen, S., Katru, S.P., Kumar, A., Vuppala, S.K., Paul, S.: isense: A wireless sensor network based conference room management system. In: Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, BuildSys 2009, pp. 37–42. ACM, New York (2009), http://doi.acm.org/10.1145/1810279.1810288
11. Palani, K., Nasir, N., Prakash, V.C., Chugh, A., Gupta, R., Ramamritham, K.: Putting smart meters to work: Beyond the usual. In: Proceedings of the 5th International Conference on Future Energy Systems, e-Energy 2014, pp. 237–238. ACM, New York (2014), http://doi.acm.org/10.1145/2602044.2602084
12. van der Veen, J., van der Waaij, B., Meijer, R.: Sensor data storage performance: Sql or nosql, physical or virtual. In: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), pp. 431–438 (June 2012)