This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

Building Data-Aware and Energy-Efficient Smart Spaces

Neha Belapurkar, Jacob Harbour, Sagar Shelke, Baris Aksanli *Member, IEEE*

***Abstract*—With the increase in the dependency of our life on technology and data, smart spaces have become integral in providing an environment for data collection, analysis, and machine responses. This paper discusses the current research in this field and the challenges that arise in the execution of these smart spaces. We address the major challenges of hardware design, data analysis and energy efficiency in a new data aware smart environment that collects time-stamped data for position, movement, temperature and vibration sensors. Data collected from these sensors is used to achieve energy efficiency, for real time localization in conjunction with machine learning mechanisms to analyze human activities. We evaluate six different machine learning algorithms for human activity detection task, on a data set collected in our lab. Results show high classification performance for all methods giving up-to 99.95% classification accuracy. We also implemented energy-efficiency measures, leading to up to 30% energy efficiency improvement on top of our initial design. This ambient environment, along with data analytics and improved energy efficiency, provides information regarding the occupancy and behavior of people within its range. Spaces such as conference rooms, common areas such as libraries, classrooms, and even public spaces such as public transport can benefit from our design. Our system avoids privacy issues by using no audio/visual devices. This system thus provides an insight into smart spaces, their current trends, and what future direction research such as ours would lead them to.**

***Index Terms*—smart space, wireless sensor network, human behavior, data analysis, machine learning.**

I. Introduction

Smart spaces are environments equipped with sensors and intelligent devices. They also facilitate data collection, computing, and analysis based on the interaction of humans with the environment without requiring the occupants of this space to use or wear any special equipment. Smart spaces have become popular in the fields of health-care [1] [2], wellness [3] [4], education [5], safety [6], and commerce [7]. Figure 1 shows an example of a typical smart space deployment, demonstrating various types of sensors interacting with a user. Characterization of smart spaces comes from their real-time data-collection, response capabilities, as well as ease of deployment. This necessitates the use of wireless sensor networks in conjunction with the recent Internet of Things (IoT) solutions. The lack of wiring facilitates easy and fast installation and helps avoid any electrical rewiring or excessive remodeling to the existing infrastructure. The real-time data collection and analysis is vital for applications in the health-care and wellness industry; it can enable a quick response from concerned authorities in case of emergencies. Based on interactions of the inhabiting humans with the environment, a person losing consciousness, a person's pulse changing, or a similar situation can trigger an alarm. The constant monitoring and analysis of human-environment interactions can give a significant insight into behavior patterns in libraries, stores, museums, and wellness centers. These can, in turn, help predict customer trends in marketplaces, patient behavior in emergency rooms, and other future human behavior. They also help with management of resource such as inventory, staff, and supplies.

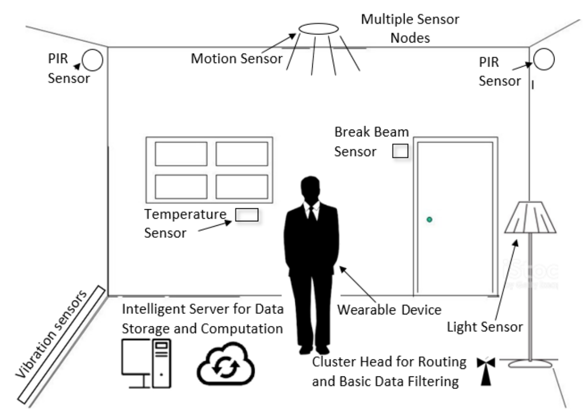


Fig. 1 : Typical smart space

Previous research in this area adopted audio-visual sensor data to analyze human behavior [8]. However, the computational overhead due to image and video processing, coupled with concerns about privacy and security, make this approach unsuitable for most environments. Power management for buildings using smart spaces and IoT is another well researched avenue [9]. The information gathered from the sensors finds use in effectively managing lights, HVAC systems, and other such high-power systems. It also can aid in the reduction of the carbon footprint. Ease of deployment encourages the use of battery operated sensors, independent of the existing infrastructure. As a result, energy management becomes essential to ensure a long lifetime for the system. Traditionally, energy efficiency is important only in body area networks or mobile networks, but it is possible to extend the concepts to smart ambient environments, as these environments tend to have various sensors and microcontrollers.

In this paper, we discuss various methods by which we

Neha Belapurkar, Jacob Harbour, and Sagar Shelke are with the Electrical and Computer Engineering Department, San Diego State University, CA 92182 USA. They have equal contribution to the work.

Baris Aksanli is with the Electrical and Computer Engineering Department, San Diego State University, CA 92182 USA. E-mail: baksanli@sdsu.edu.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

can address the main design considerations for smart spaces. These considerations include those described above as well as the challenges to such systems in terms of response time, computational overhead and energy efficiency. We demonstrate the methodologies we have adopted to address these challenges through our smart space implementation. This implementation employs a two-dimensional positional grid for localization and detection of human activity, in conjunction with sensors to detect motion and vibration for behavior modeling. Due to privacy concerns, our system avoids audio-visual sensors. Energy conservation is pursued by the implementation of algorithms that turn sensors on only when required and in such a way that integrity of data collection is not hampered. We observe that our proposed energy efficiency solution gives 30% better energy consumption than a system without energy considerations, and approximately 10% better performance than traditional algorithms. We localize user and classify user activities using low overhead machine learning methods. We show that support vector machine (SVM) algorithm gives the best result with 99.95% accuracy for activity detection.

II. Data Analytics in Smart Spaces

As sensor technology improves, the cost of a single sensor module is decreasing significantly, making it easier for smart spaces to deploy numerous sensors and collect continuous data from them. This section shows data collection and analysis techniques widely used in smart spaces.

A. Data Collection

This section focuses on data collection methods used in smart spaces. Data collection seeks to collect discrete representations of continuous time and space domains, making it impossible to have a perfect representation of the surrounding environment using data [10]. Examples data collection problems include noisy and/or redundant readings [11], and high power consumption due to frequent readings and/or a lot of sensors [12] [13].

In smart spaces, many data collection methods rely on wearable sensors. Chawla *et al.* [14] collected data from 8 different users for 6 activities using a single wrist-mounted module. However, wrist-mounted sensors restrict or bias human movements. Casale *et al.* [15] used a single chest mounted accelerometer sensor system to detect human behavior. This method requires providing each occupant with a wearable accelerometer. Similarly, Mannini *et al.* [16] used a single accelerometer sensor to detect activities. While the system performance was good, use of accelerometer on waist or ankle is not practical in areas where the occupants may resist their use and the occupants continually change.

With the previous drawbacks in mind, some data gathering projects seek to use data from devices people already carry with them, such as smartphones and watches. Vaizman *et al.* [17] collected labeled data from 60 subjects. However, to use this system every user needs to install a smart-phone application and companion application for Pebble Smartwatch. This still carries the burden of dependency on system users, ultimately having an intrusive system framework.

Several developments try to avoid the issues of wearable sensor modules. Luo *et al.* [18] created a one-dimensional array of ultrasonic HC-SR04 sensors to detect the number of occupants entering a stadium based upon the width of a human body. Using a two-dimensional grid of the same sensor module, Ghosh *et al.* [19] collected data and used it combined with machine learning to detect various human activities. This activity detection had limited application, as positional data was the only sensory input.

The privacy of anonymous bystanders in data collection remains a large concern [20] [21]. Lu *et al.* [8] developed an application in which phone's camera can more reliably track a person's steps compared to an accelerometer alone. However, this requires costly image processing and creates security concerns, such as unwanted tracking. [22].

B. Data Analysis

This section focuses on common data analysis methods, including supervised/unsupervised machine learning algorithms and deep learning techniques used in the literature to classify human activities in a smart space.

Property	***SVM***	***kNN***	***Naive Bayes***	***Decision Trees***
Average accuracy	Highest	High	Low	Higher
Learning speed	Low	Highest	Highest	Highest
Classification speed	Highest	Low	Highest	Highest
Noise handling	High	Low	Higher	High
Transparency	Low	High	Highest	Highest
Handle overfitting	High	Higher	Higher	High

TABLE I: Supervised learning algorithms

Supervised learning solves problems in which each data point has a label. Table I gives an overview of supervised classification techniques. In smart space settings, Chawla *et al.* [14] used support vector machine (SVM), k-neighrest neighbors (kNN), decision trees and artificial neural network (ANN) to classify human activities in nine groups. With a training set of 522 instances, kNN performed best. Davis *et al.* [23] used machine learning techniques for activity recognition in ambient assisted living (AAL) consisting of a hybrid model of SVM and hidden Markov model (HMM) which outperformed both SVM and ANN. Cook [24] used naive Bayes classifier (NBC), HMM and conditional random field (CRF) model to study human activities. The motivation behind considering these three approaches is the capability to handle noise, sequential data, and being able to generate probability distributions over the class labels. The work created an ensemble of NBC, HMM, and CRF which outperformed individual classifiers by more than 8%. Fluery *et al.* [1] used SVM in smart homes to classify human activities. The performance of polynomial and Gaussian kernels for this task was compared where classification accuracy for Gaussian kernel was 10.7% better.

Unsupervised learning solves problems with unlabeled data. Clustering [25] and anomaly detection [26] are two common examples. Janakiram *et al.* [27] used Bayesian belief network (BBN) to detect anomalies and missing data points in sensor stream data. Zhang *et al.* [28] used a tree-based approach to find global outliers in a wireless sensor network

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

using kNN. Ide *et al.* [29] used a variant of kNN called stochastic nearest neighbors to detect anomalies in their sensor network. Changseok Bae *et al.* [30] compared performance of three unsupervised algorithms k-means, Gaussian mixture and hierarchical agglomerative clustering. GMM showed perfect recognition for all activities giving 100% accuracy.

Deep learning has become a very important field with applications in both research and industry. Neural networks are the core of the deep learning algorithms [31] where a neural network with many hidden layers is called a deep neural network. To use deep neural networks in smart space settings where energy and computational resources are limited, Yao *et al.* [32] developed the DeepIoT framework. In DeepIoT, commonly used neural network architectures such as fully-connected convolutional neural networks (CNN) and recurrent neural networks (RNN) are compressed for use in sensing applications. Cho *et al.* [33] used CNN for human activity recognition on data collected from acceleromters and gyroscopes. CNN outperformed traditional machine learning algorithms, achieving 94.79% accuracy to classify five activities with raw sensor data. In another work, Sharma *et al.* [34] used feed forward neural network to classify activities.

III. Energy Efficiency in Smart Spaces

Smart spaces rely on easily-deployable and low maintenance wireless sensor networks (WSN) for sensing and data-collection. For low maintenance, the batteries used in the sensor nodes need to last several years. To achieve this, smart spaces need to employ energy efficiency methods. This section discusses smart space applications and their energy efficiency requirements; and then shows common energy management methods used in smart spaces.

A. Smart Space Energy Requirements

Smart space have different energy requirements, based on their use-cases. These requirements are determined by various factors such as real-time application deadlines, data transmission rates, mobility in case of wearable devices, computational complexity of algorithms, controlled area size, and ability to expand [35]. We identify these parameters as: security, mobility, scalability, latency, coverage area, robustness, and fidelity of data. Table II shows a summary of energy efficiency methods suitable for various smart space applications (such as healthcare, public safety, environmental monitoring, and commercial applications) based on their requirements.

Smart Space Application	Requirements	Suitable Energy Efficiency Methods
Healthcare	low latency, mobility, high fidelity	Transmission control, sampling rate control
Public safety	Robustness, high coverage area	Energy harvesting, topology control, packet aggregation, duty cycling
Environmental monitoring	Robustness, high coverage area, mobility and scalability	Energy harvesting, duty cycling, topology control
Commercial applications	No life threatening implications, but low latency and fidelity desired	Duty cycling, sampling control, topology control

TABLE II: Energy efficiency methods for smart spaces

B. Energy Management Methods

This section explains the energy management methods most commonly used in smart spaces and connects them to the requirements listed in Table II.

Energy Harvesting and Energy Transfer methods are used where battery capacity is small, but changing batteries is not feasible because there are many sensor nodes or sensors are physically inaccessible. It adds to the hardware and makes sensor nodes bulky. If heavy duty sensors such as microwave sensors or high-power transmitters are used, the extra hardware can be an acceptable trade-off [36]. Energy harvesting methods that can be deployed in smart spaces include solar [37], thermo-coupled [38], piezo-electric [39], and RF-powered [36] options. The biggest disadvantage of these methods is the dependence on the environment. The inconsistent availability of the physical quantity in any smart space, such as vibration, makes it an unreliable resource. Hence, this method cannot guarantee a constant amount of energy harvested. A similar method of battery repletion is energy transfer. Dost *et al.* proposed a method to design mobile hosts [40] for wirelessly charging the depleted batteries of sensor nodes, using electromagnetic waves to charge capacitors.

Transmission Control is used to control transmission power consumption on sensor nodes. This method essentially changes the range to which the sensor node can transmit and thus adjusts connectivity. Transmission power reduction occurs dynamically and adaptively [35]. Whenever the remaining battery capacity of a node is low, the node reduces its transmission power. In response, the nodes in its vicinity with enough battery capacity, increase their transmission power. This evens out the battery life of the whole network, reducing maintenance overhead and increasing the system reliability.

Data Manipulation helps sensor networks reduce the cost of transmitting data by adjusting the amount of data sent. This is achieved by manipulating the data at the sensor node and transmitting only what is necessary. These methods include sampling rate control and data reduction. The most common way to manipulate energy usage is to control the data sampling rate itself. In adaptive sampling, the sampling rate increases or decreases adaptively, depending on the how frequently data changes in proportion to data sampling rate. In this method, the rate depends on a moving window of samples taken previously [41]. An alternative is to change the sampling rate based on a probabilistic model, pre-loaded into the sensor node. This is known as model based active sampling. This type of energy management adds computational overhead, but it is mostly negligible compared to the energy saved. In data reduction methods, instead of taking a smaller number of samples, transmissions include a reduced set of data.This reduces the net amount of data through the network, thereby reducing the power consumption [42]. Data aggregation helps avoid overwhelming the user with data notifications and helps reduce congestion in networks. However, the aggregation latency should be analyzed to make sure that the benefits of this method is not neutralized.

Topology Control is similar to duty cycling protocols. Most sensor networks, such as the ones for environmental

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

Study	Market	Energy Efficiency	Data Analytics	Wearable Sensor Dependency	Non Intrusive
Ya-Li Zheng *et al.* [2]	Healthcare	Included	Overview only	Yes	Yes
Vince Stanford [3]	Healthcare	Not included	Not included	Yes	Yes
Kiryong Ha *et al.* [45]	Elder care	Included	Included	Yes	No
Majd Alwan *et al.* [4]	Elder care	Not included	Minimal	No	Yes
Stephen S Yau *et al.* [5]	Education	Not included	Included	Yes	Yes
PV Vinu *et al.* [46]	Education	Not included	Not included	Some	No
A. Coronato *et al.* [47]	Office use	Not included	Included	Some	Yes
Serge Offerman *et al.* [48]	Office use	Not included	Not included	No	Yes
Jayashri Bangali *et al.* [49]	Security	Included	Not included	No	Yes
Julia Moehrmann *et al.* [50]	Group meetings	Not included	Included	No	No
Hammadi Nait Charif *et al.* [51]	Group meetings	Not included	Included	No	No
Zobl *et al.* [52]	Group meetings	Not included	Included	No	No
Helal *et al.* [53]	Healthcare	Not included	Included	No	Yes
Dan Yang *et al.* [54]	Home	Not included	Included	No	Yes
Hsu *et al.* [55]	Home	Included	Included	Yes	Yes
Alwan *et al.* [4]	Elder care	Not included	Included	Yes	Yes
Xiaomu Luo *et al.* [56]	Home	Not included	Included	Yes	Yes

TABLE III: Comparison of existing smart spaces in the literature

monitoring or smart stores, are redundantly deployed, which is exploited by topology control. Even with some of the sensors turned off, the smart environment can give adequate data. Sensors are adaptively turned on and off, as the requirements of the network change [43]. Topology control can be of two types: location driven, or connection driven. For the location-driven method, sensor nodes carry out spacial discovery. A grid forms depending on the location of sensor nodes and the redundant nodes cycle on and off. Connection driven control follows a similar strategy, where the redundant nodes connected to a cluster head cycle on and off.

Duty Cycling is switching certain sensor nodes on and off based on an algorithm. In this method, a sensor node should wake-up only when another node needs to communicate with it. Such a system utilizes a node with two radios. One radio transmits and receives data while the other is ultra-low power and used exclusively for wake-up purposes. However, the added hardware complexity and redundancy of having an extra radio on the board is undesirable. A modification to this is the scheduled wake-up protocol, where nodes wake-up at predetermined times and query one another for data. The trade-off between power saved and power required for wake-up and sleep is a critical parameter that decides the maximum frequency for wake-up and sleep protocols [44]. This is critical for smart spaces as the effectiveness of duty cycling severely affects the quality of the smart space application.

In the next section, we focus on full-scale smart space deployments and compare them in terms of their application domain, data analytics, energy efficiency, and user interaction.

IV. Existing Smart Space Deployments

Smart spaces are primarily deployed in the fields of health-care, public safety, education, agriculture, and workplaces, as shown in Table III. The capability to decide by learning from the surroundings differentiates normal spaces from smart spaces. Sensors used to learn from the ambient environment plays an important role in deciding the use-case of a smart system. Sensors are either embedded in the environment or the environment interacts with a mobile phone or other data collecting devices that the user carry. Cameras and microphones, placed either on wearable devices or within the ambient environment, can provide accurate information about the users but they also lead to user privacy concerns.

Haet *et al.* [45] proposed a Google Glass based smart system for elder care which includes energy efficiency and data analysis. However, it requires wearable technology, a problem that is particularly pronounced with the elderly. Vinu *et al.* [46] proposed a smart classroom model, based on smart sensors embedded in the room and devices in use. But their system uses audio/video recording, raising privacy issues. The work also does not consider energy efficiency or data analytics. Moehrmann *et al.* [50] introduced a vision based system for monitoring in group meetings. They used hidden Markov model to classify idle, sitting and writing activities of the users. In a similar work, Zobl *et al.* [52] detected and recognized the actions of a single person in a meeting room. Charif *et al.* [51] represented a smart meeting room able to analyze activities of its occupants.They used multiple cameras to track multiple users using particle filter and method of maximum likelihood. Despite the good accuracy, video recording created user privacy concerns, reducing system usability in wide.

Alwayn *et al.* [4] created a vibration sensing system for elderly to detect falls. The work does not consider energy efficiency and provides minimal data analytics. Coronato *et al.* [47] created sensor networks for smart offices and smart homes. The work relies heavily on data analytics, but does not analyze energy efficiency for frequently used wearable sensors. Hsu *et al.* [55] used multi-sensor data fusion techniques with wearable motion sensing devices. They used machine learning to classify user gestures with sensor unit in hand. For

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

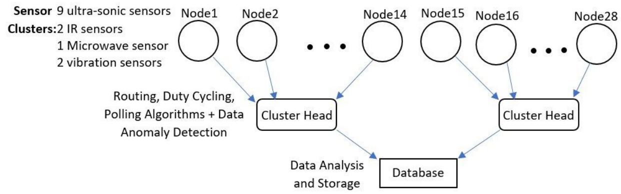


Fig. 2 : Connectivity diagram of our smart space deployment

localization, they placed sensor units on the user's feet. But instrumenting the user is not practical for real-life applications. Yang *et al.* [54] used multiple passive infrared sensors for indoor positioning in smart homes whereas Luo*et al.* [56] used pyroelectric infrared sensors for both localization and activity detection in smart homes. In the healthcare domain, Helal *et al.* [53] used temperature, motion, and light sensors to build a smart health platform to analyze and alter behavior of diabetes patients. They were able to classify patient activities using a Markov model, with up-to 98% recognition accuracy. Zheng *et al.* [2] embedded sensors in daily objects such as mirror, sleeping bed etc. to collect health information but did not the collected analyze data.

A mostly unexplored area in previous smart space environments is the energy efficiency of the system. Non-healthcare related smart space applications largely neglect energy efficiency. But, even in non-life-critical applications, increasing the lifetime of a smart space by decreasing energy consumption of the sensor nodes, is a desirable attribute, as it reduces maintenance costs and increases the reliability of the system. Table III lists and compares the smart space deployments analyzed in this section. The table shows whether each smart space deployment includes energy efficiency, data analytics, wearable sensors, and non-intrusive methods.

In contrast to the existing smart spaces, our smart system provides an energy-efficient environment, without affecting the quality of our smart space application, i.e. user activity detection. We do not require wearable sensors, thus provide a non-intrusive system. Also, we localize the user and classify user activities using machine learning algorithms, achieving up to 99% accuracy in detecting user activities.

V. Our Smart Space Deployment

In this section, we first present our smart space deployment and the design decisions we make, and then demonstrate the capabilities of our smart space in terms of localization, activity detection, and energy efficiency.

A. System Setup

We adopt a hierarchical hardware setup as shown in Figure 2 . Plain sensor nodes are the lowest-level in the hierarchy and they are connected to the cluster heads for data collection and initial data processing. And finally, cluster heads connect to the main database, representing the cloud in our system, where all data is gathered for further analysis and storage.

We select Raspberry Pi 3B as the cluster node for hardware data collection due to its multitude of I/O pins, computing

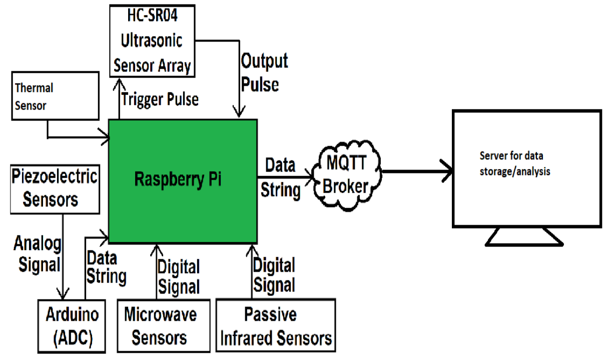


Fig. 3 : Block diagram of the data collection hardware

3V3 (+)	1	2	5V (+)
GPIO2 (*)	3	4	5V (+)
GPIO3 (*)	5	6	GND (0)
GPIO4 (z)	7	8	GPIO14 (z)
GND (0)	9	10	GPIO15 (z)
GPIO17 (x)	11	12	GPIO18 (y)
GPIO27 (x)	13	14	GND (0)
GPIO22 (x)	15	16	GPIO23 (y)
3V3 (+)	17	18	GPIO24 (y)
GPIO10 (x)	19	20	GND (0)
GPIO9 (x)	21	22	GPIO25 (y)
GPIO11 (x)	23	24	GPIO8 (y)
GND (0)	25	26	GPIO7 (y)
DNC	27	28	DNC
GPIO5 (x)	29	30	GND (0)
GPIO6 (y)	31	32	GPIO12 (y)
GPIO13 (z)	33	34	GND (0)
GPIO19 (i)	35	36	GPIO16 (y)
GPIO26 (i)	37	38	GPIO20 (x)
GND (0)	39	40	GPIO21 (x)

Description	Key
+ voltage	(+)
Ground	(0)
Thermal sensor	(*)
Microwave sensor	(i)
Ultrasound trigger	(x)
Ultrasound echo	(y)
PIR sensor	(z)

Fig. 4 : Raspberry Pi 3B pin connection diagram

power, ease of use, and an on-board Wi-Fi module. Figure 3 depicts how Raspberry Pi (as a cluster head) is connected to a variety of sensors to collect data. We also show how we utilize Raspberry Pi pins to communicate with the sensors in a more detailed way in Figure 4 . The Raspberry Pi creates a JSON string by collecting data from all the sensors and sends it to the server (cloud in our hierarchical setup) using MQTT (Message Queuing Telemetry Transport) client-server protocol. A python program on the server runs two threads. One fetches data being published on an MQTT broker by the Raspberry Pi. The other parses the JSON string to get time-stamped sensor data and then write to the MySQL database. Data analytics algorithms then use this data for localization, activity detection and to achieve energy efficiency. Figure 5 shows our implementation and construction of the data collection environment, demonstrated in abstract in Figure 3 . The sensors in Figure 5 correspond to the sensor nodes in Figure 2 , and the two Raspberry Pis act as the cluster-heads. The database server is located at a different remote location.

We use *HC-SR04 ultrasound range sensors* to discern the position of many different objects or occupants in a given space and their exact location relative to objects such as furniture, open floor spaces, seating, etc. We observe that the largest indicator of leaving or entering a room, as opposed to simply continuing to occupy it, would be the packing or unpacking of belongings. As a result, vibration-related data would greatly aid us in predicting when occupants would soon

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

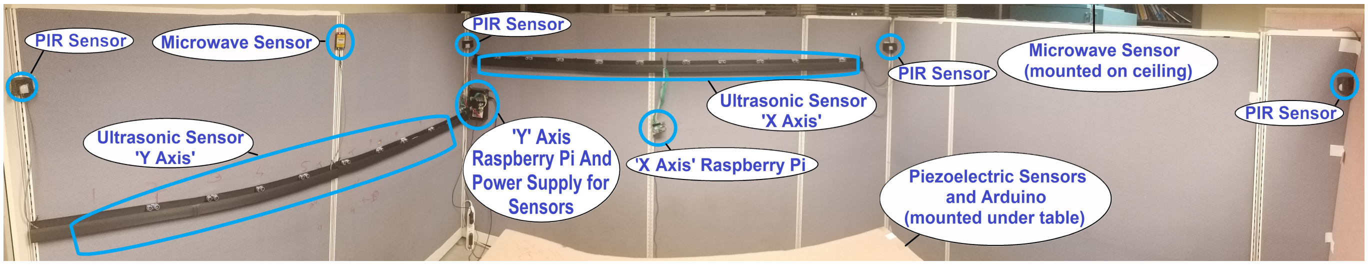
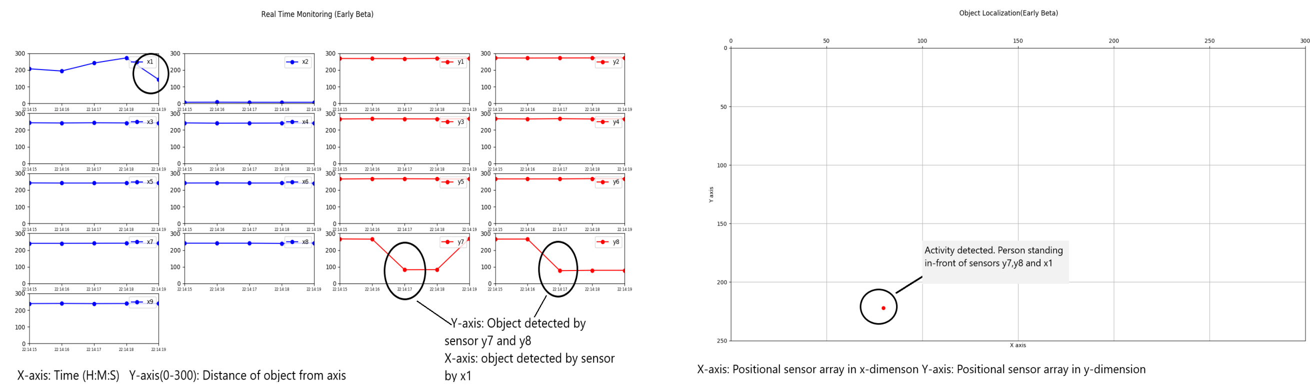


Fig. 5 : A panoramic view of experimental setup



(a) Real-time sensor data monitoring

(b) Real-time localization

Fig. 6 : Correlating real-time sensor output and localization

leave. To collect this vibration data, we employ *piezoelectric sensors*. Data collection related to motion sensing is useful for supplementing the positional grid data. Not only does it aid in correcting positional grid data errors, but it also allows differentiation between a more robust range of human behaviors that do not necessarily translate to a change in position. For example, reading and typing on a keyboard are the same action when using only positional data, but motion-sensitive algorithms can still differentiate between them. We use two types of motion sensors. *passive infrared (PIR) sensors* are inexpensive but not very sensitive to subtle motions such as fingers moving. By contrast, *digital microwave sensors* (model: SEN0191) are more expensive but allow for detection of more subtle movements. Finally, *thermal sensors* (model: MLX90621) is used to capture changes in the temperature across the given environment.

The main goal of our system is to localize the person inside, detect their activity and achieve energy efficiency (without affecting activity detection performance) with intelligent power switching. We use ultrasonic, microwave, PIR and piezoelectric sensors for localization and to cycle power only through active components depending upon occupancy status, whereas we use thermal sensors only for activity detection in conjunction with machine learning algorithms.

B. Localization Capability

Localization is one of the most important aspects of ambient assisted living (AAL), especially regarding elderly care and health-care applications where it is essential not to instrument the person to be tracked (for non-intrusiveness purposes). To assist data collection and monitoring, we design a Python application to plot all sensor values in real time with time on the x-axis. When the sensors detect an object, the output is shifted down from its steady value. The delay between actual event occurrence and it's plotting is the same as that of a packet traveling over a network and a server program decoding the corresponding value averaging up-to *0.37 seconds*. A sample demonstration of this process is shown in Figure 6 a. We design another application to locate objects within the smart space. Figure 6 bshows object a localization map where the chart borders represent our smart environment. To differentiate between standing and sitting objects at the same (x,y) position, we use blue and red colors, *e.g.* if a person is standing at position (80, 230), a red dot will represent that, whereas a blue dot represents a sitting person. We mount X-axis sensor array at different elevations than their Y-axis counterparts, as shown in Figure 5 , to allow for differentiation between standing and sitting. We can analyze the functionality and accuracy of the system by looking at Figure 6 band Figure 6 atogether. Deflection of sensor y7 and y8 on the y-axis is occurring due to an object in-front of them (due to little spacing between sensors, one object might deflect two sensors). Thus, the y position of the dot is 230. X position is the output of these two sensors i.e. 80. Now the dot is *Red* due to deflection of sensor x1, which is at an elevated height. This indicates that the person is standing.

C. Activity Detection Capability

In our smart environment setup, we divide activities into *static activities* (sitting, standing, sitting on chair, sitting on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

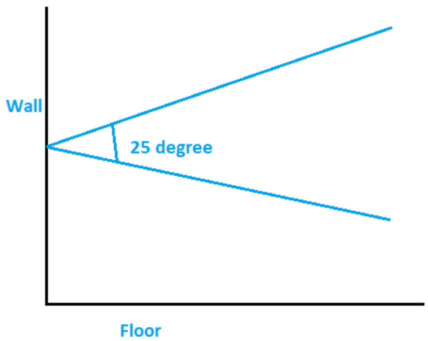


Fig. 7 : Initial sensor placement (side view)

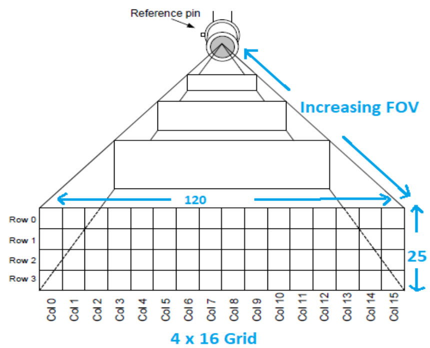


Fig. 8 : Thermal sensor (front view) [26]

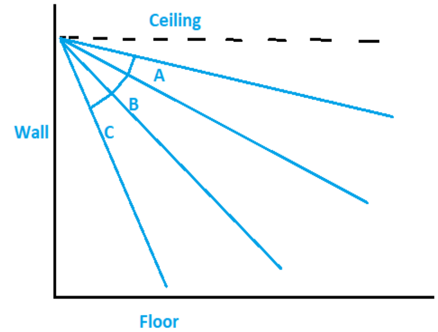
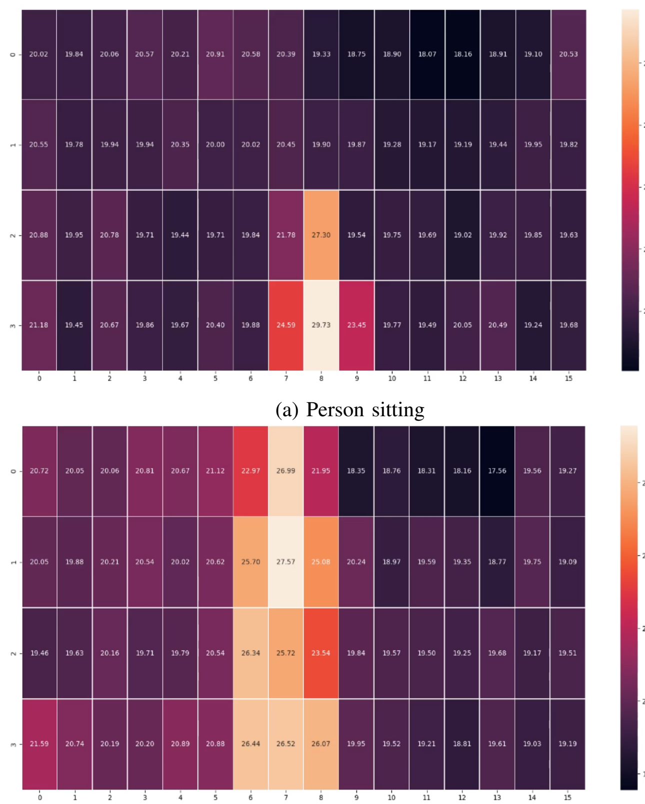


Fig. 9 : Future sensor placement

Activity	***Sensor Pattern***
Enter or leave	Ultrasonic sensor and piezoelectric sensor
Sitting, idle	Ultrasonic sensor from Y axis activate, high values in lower-middle part of thermal sensor array and microwave sensor
Sitting, working	ultrasonic sensor from Y axis activate, high values in lower-middle part of thermal sensor array and microwave sensor activate.
Standing	Ultrasonic sensors from both X and Y axis activate, high values in the center of thermal sensor array
Walk around	Ultrasonic sensors array activate sequentially, changing values in thermal sensor array and PIR sensors

TABLE IV: Example activities and sensor patterns



(a) Person sitting

(b) Person standing

Fig. 10 : Activity heat-maps

ground, laying on ground) and *dynamic activities* (move right, move left, move towards the sensor, move away from the sensor). This section shows how our grid-based smart system setup detects activities, and then presents accuracy results for activity detection. Table IV shows intuitive relationship between sensor output and corresponding activities. In the remainder of this section, we demonstrate our proof-of-concept analysis to detect two activities along with a detailed accuracy analysis using different machine learning methods.

a) Data Collection: We use non-contact thermal sensors to collect data for activity recognition. These sensors have 120° horizontal field of view (FOV) and 25° vertical FOV with output in 4x16 array as shown in Figure 8 . Since we cannot construct a high-quality image from such a low resolution thermal sensor output, user privacy is not exposed. Each value in 4x16 matrix gives temperature in that area which was directly fed to the machine learning algorithms after flattening, without any feature engineering. With a single sensor placed on the middle of vertical wall as shown in Figure 7 , we capture two activities which are standing (STAND) and sitting on the chair (SIT). Figure 10 shows the heat-map of data collected for a person standing and sitting on the chair. Each sensor gets 4 frames/second, and we collect 111,225 labeled examples in ten days with five users. Class STAND has 49384 labeled examples whereas class SIT has 61841 labeled examples. To capture more than two activities, in our future work, we will increase vertical FOV by installing multiple sensors, as in Figure 9 with all A,B,C equal to 25°.

b) Data Analysis: We implement and compare the performance of six machine learning algorithms using open source frameworks SciKit-Learn [57] and TensorFlow [58]:

1. Logistic Regression (LogisticR): This is a binary classification method estimating the probability of an instance belonging to a particular class. We use a linear solver with regularization strength parameter. C=1 and L2 regularization.

2. Support Vector Machine(SVM): SVM which is another strict binary classifier, extensively used for human activity classification [59], [60]. Given labeled data, SVM outputs an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

Classifier	*Accuracy*	*Precision*	*Recall*	*Specificity*	*F1*
Logistic	97.83	98.01	98.09	97.50	98.04
SVM	99.95	99.93	99.98	99.91	99.95
DecTree	99.34	99.42	99.39	99.28	99.40
RandFor	99.67	99.54	99.88	99.42	99.70
NaiveB	91.67	91.58	93.19	89.16	92.36
NN	99.44	99.45	99.55	99.31	99.49

TABLE V: Performance metrics for algorithms in (%)

		Predicted SVM	
		SIT	*STAND*
Actual	*SIT*	61829	12
	STAND	42	49342

		Predicted NaiveB	
		SIT	*STAND*
Actual	*SIT*	56398	5443
	STAND	3867	45517

TABLE VI: Confusion matrix. SVM (left) and NaiveB (right)

Algorithm	**NN**	**LogisticR**	**SVM**	**DecTree**	**RandFor**	**NaiveB**
Training time	89.93	10.23	46.45	10.58	6.23	0.156
Test time	0.021	0.002	1.627	0.005	0.018	0.019

TABLE VII: Algorithm train/test time evaluation (seconds)

optimal hyperplane which divides current training data and categorize new examples. We use *one-vs-all* and *one-vs-one* techniques to extend SVM for multiclass classification. We use SVM algorithm from Scikit-Learn with *"rbf"* kernel, *"hinge"* loss and regularization parameter, C=1.

3. Decision Tree(DecTree): Scikit-Learn uses *CART* algorithm to train decision trees and *Gini* impurity to check the quality of classification. *CART* is a *greedy algorithm* which looks for optimum split at top level and repeats the procedure for all levels. It selects feature k and threshold t_k such that it refines class prediction at each level.

4. Random Forest(RandFor): Ensemble methods combine predictions from several base estimators reducing bias and variance. Random forest is an ensemble method with decision tree as a base estimator and *Bagging* (Bootstrap Aggregating) to create different random subsets of the training set. We use 10 decision tress as base estimators.

5. Naive Bayes(NaiveB): Naive Bayes is a supervised learning algorithm, applying Bayes theorem and assuming all features are independent. Naive Bayes classifiers are very fast and differ by the assumption they make about $P(X|y)$ where X is feature and y is label. Our implementation is based on Gaussian distribution.

6. Artificial Neural Networks(NN): NN structure is composed of several layers of nodes connected by weighted links. The behavior of the neural network is decided by it's depth, activation functions used, learning rules and architecture itself. We use a three layer feed forward neural network from TensorFlow. Number of inputs to the neural network are 64(4 x 16 output of thermal sensor). Each hidden unit has 10 neurons. We use batch gradient descent optimizer with batch size of 200. With no regularization, we set the learning rate to 0.01 and use activation function *leaky relu*.

c) Classification Results: To achieve the best classification performance, we apply stratified k-fold cross-validation (*with k = 10*) technique to each classifier. This reduces the variance of the resulting estimates because every data point was only once used in the test set. The performance metrics we use to compare the performance of different classifiers are shown in Table V. Since accuracy is not always a good performance measure, we also use *Precision*, *Recall*, *Specificity*, and *F1* score. *Precision* is the accuracy of positive predictions of the classifier. *Recall* is the portion of positive instance that are correctly detected by the classifier. *Specificity* is the proportion of negatives that are correctly identified as such. *F1* score is the harmonic mean of precision and recall, giving more weight to low values. Specifically:

- Accuracy = $\dfrac{T_P + T_N}{T_P + T_N + F_P + F_N}$
- Precision = $\dfrac{T_P}{T_P + F_P}$
- Recall/Sensitivity = $\dfrac{T_P}{T_P + F_N}$
- Specificity = $\dfrac{T_N}{T_N + F_P}$

where T_P is true positives, T_N is true negatives, F_P is false positives and F_N is false negatives.

In Table V, we notice a very high accuracy of 99.95% with SVM whereas naive Bayes classifier gives the lowest accuracy of 91.67%. For these two methods, we also include their confusion matrices in Table VI. Each row in the confusion matrix represents the actual classes and each column represents the predicted classes. For example, in the left table, the number "61829" corresponds to the data points that actually belong to the class SIT and that are also classified as SIT using the SVM method (true positive - T_P). Similarly, the number "12" in the left matrix shows the data points that are actually in the SIT class but classified as STAND using the SVM method (false negative - F_N). Overall, the numbers in Table VI help us calculate the performance metrics listed above. Accuracy of machine learning algorithms depends on bias and variance. Naive Bayes algorithm assumes that data distribution is Gaussian which introduces bias leading to low overall accuracy. Ensemble models in most of the cases outperform the base estimator. Single decision tree gives average accuracy of 99.34% on 10 folds whereas an ensemble of 10 decision trees slightly outperform giving overall accuracy of 99.67%. Deep feed forward neural network with three layers gives an average accuracy of 99.44% which is slightly worse than SVM. Even simplest training algorithm of logistic regression gives 97.83% accuracy.

For the practical deployment of a model, train and test time of machine learning algorithms are crucial. We evaluate train (on 90% of an entire data) and test time(on 10% of an entire data) for all models. Workstation used for bench-marking has *Intel(R) Xeon(R) CPU E3-1270v5@3.60GHz* processor and 8GB RAM. Table VII shows the comparison of train and test time in seconds for machine learning algorithms. Out of all machine learning algorithms SVM takes the highest time of 46.45 seconds. Computational complexity for SVM is $O(m^2 * n)$ where m is number of the training examples and n is number of features. Computational complexity for decision tree is $O(n * mlog(m))$, having log relationship with number of examples leads to faster training. Therefore, decision tree and random forest takes 10.58 and 6.23 seconds, respectively. To make predictions, each node in tree based algorithm requires checking only one feature making time complexity $O(log_2(m))$. Naive Bayes based model has train-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

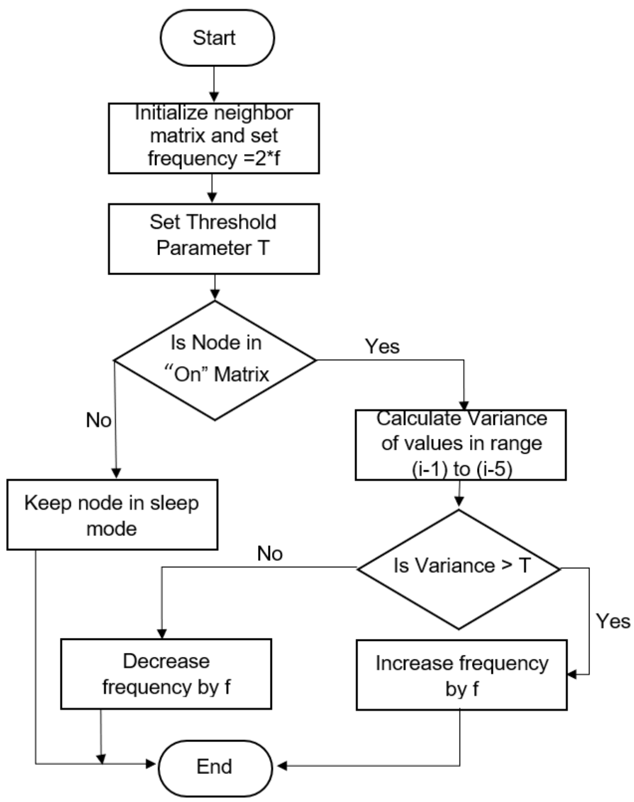


Fig. 11 : Energy efficiency algorithm flowchart

ing time complexity $O(n * m)$ making it fastest to train, but with the lowest average accuracy. Training time for a neural network with three hidden layers trained in 100 epochs with 10 neurons in layer is approximately 90 seconds.

D. Energy Efficiency

In our smart space, we propose a two-fold method to improve energy efficiency. The energy efficiency methods we use are based on the type of data we collect: 1) ambient environment data and 2) user/subject-dependant data. The former is for collecting data such as temperature, humidity, vibration, which provides context for data analysis. The latter involves data directly related to the subject's movement, location, etc. Both data are used for behavior analysis and prediction. We first focus on the ambient data (temperature, pressure, vibration, etc.). These are independent of people, prevalent throughout the smart space, and also not event triggered. For sensors providing this kind of data, adaptive duty cycling is an effective way to reduce energy consumption. Thus, in our cluster based topology, we implement adaptive duty-cycling. In this method, we calculate the percentage of time intervals in which there is no change in data (miss-M) and there is change in data (hit-H). If there are too many misses in a row, we change to a lower frequency. If a hit occurs, change to a higher frequency. To control this change, we define a threshold value, to control the number of redundant readings allowed, changing the duty-cycle of the system.

To further improve the energy efficiency, we also consider subject locality in a smart space. After detecting the initial location of a subject, we do not need to keep the entire set of sensors active. Instead we can keep a subset active and adaptively change this subset based on the movements of the subject. We combine this localized activation of sensors with adaptive duty cycling mentioned above. Figure 11 shows the flowchart for the proposed method. To achieve this, we need to have the physical location matrix of all the sensor nodes (sensor topology). There are two methods used for the creation of the location matrix: It can be hard-coded into the system, or a neighbor location discovery algorithm can be used [61]. The advantage of having a neighbor discovery algorithm is that it makes the system scalable and dynamic. If the system uses additional sensor nodes deployed in the environment, their incorporation becomes easy. Hence, we choose this method. The energy can be optimized even more, by adding a duty cycling algorithm to localized activation, in cases where the same subset of sensors stays active for a long duration. This can occur in activities, such as subject sitting idle/working, standing in same location, etc.

In our system, the response time of an ultrasonic sensor is 40 ms. Both the PIR sensors and microwave motion sensors have a response time of 0.5 seconds, while it is negligible (2 ms) for the vibration sensors. The data analysis is to detect human behavior and thus locality within the positional grid is important. Thus, we turn six sensors ON in the vicinity of the subject at a time. These include four ultrasonic sensors, one PIR sensor, and one microwave sensor. The vibration sensors are collecting ambient vibration data and remain on at all times. This means that in case of a single occupant, only $one - third$ of the sensors are consuming power at a time. This enables the system to save 30% of its power. As the number of occupants increases, more sensors start to turn on. This reduces the effectiveness of the localized activation method, and the main contributor to reducing power consumption is the dynamic duty cycling. The duty cycling algorithm needs to take into consideration the response times of the sensors. The smallest allowable ON time for the duty cycling algorithm is 500 ms for PIR and Microwave and 40 ms for ultrasonic sensors. To preserve duty cycle uniformity for all sensors, we take the largest response time which, is 500 ms. This ensures high quality for the data used in analytics. The energy efficiency algorithm is deployed at the cluster head level (see Figure 2 as these nodes monitor the overall status of multiple nodes, hence can decide how duty cycling and adaptive localization be applied.

To analyze the performance of our energy efficiency method, we simulate an equivalent of our experimental setup on NS3 [62]. NS3 is an open source, discrete-event network simulator to analyze network systems. We create the network diagram of our smart space, as shown in Figure 2 , in simulation, with a total of 36 sensors and 2 cluster heads, where the sensors correspond to the lower level of the hierarchy and cluster heads, represented by Raspberry Pis in our system, are responsible for controlling the sensor nodes. We model all sensors as simple WiFi end nodes, with no computational intelligence using the standard end node configuration in NS3. Furthermore, for the sensor nodes, we add WiFi station application, provide a device id for each node, and define the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

Algorithm	Energy (J)	% improvement
None	37.44	0 (default)
Standard duty cycling	31.45	15.9
Standard topology control	28.08	25
Proposed combination	26.27	29.83

TABLE VIII: Comparison of energy efficiency methods

mobility as stationary (i.e. their location is fixed).

The cluster heads are modelled as access points, using the WiFi Access Point (AP) node configuration. The basic differences between the WiFi AP node and the standard sensor node are the additional computational and routing capabilities. The routing and sleep/wake-up decisions are made at these cluster heads. We establish the channel and connection between these nodes by using *Helper APIs* provided by NS3. *WifiMacHelper* and *YansWifiPhyHelper* are two sample APIs used in our simulation. Each cluster head is responsible for determining the transmission time of any sensor that is connected to it in the device hierarchy. Hence, there is no interference among sensors, even though they share the same physical channel.

Finally, we use the standard energy module available in NS3 for every transceiver node to calculate the overall energy consumption across all the sensors. In this module, the base power consumption is 0.1W per transmission considering default voltage as 3.3V. The energy module has a *Sleep* state, representing non-transmission intervals, during which the power consumption is 0.01W. This energy module is built considering the standard WiFi transceiver energy consumption, not considering the energy consumed by the sensing element. We use this module because reducing the energy consumption of the sensing unit is not in the scope of our paper. Finally, for each energy efficiency method, we run 25 simulations. Each simulation run represents an activity detection session, which is around 10 seconds.

The results of this experiment are presented in Table VIII. The first row of this table corresponds to the base case, where no additional energy efficiency measure is implemented. In this base case, all sensors stay on and transmit to the access point each second. The second and third rows show standard duty cycling and topology control methods, respectively, that we have described previously in this section. And the last row is the proposed energy efficiency method for our smart space. We observe that individual methods can achieve 15.9% and 25% energy efficiency but with the method we choose, the savings go up to 30% without affecting the performance of the activity detection application.

VI. Conclusion

In this paper, we present a comprehensive study of smart systems in different application domains with a specific focus on data analytics and energy efficiency. We consider the major factors affecting the type of deployment that would be beneficial for a particular application. Based on the results for the design discussion, we demonstrate the potential use of an energy-efficient, 2-D grid based smart space for human localization and activity detection. The proof-of-concept demonstration is capable of localizing human object and identifying activities under energy efficiency constraints. We implement six machine learning algorithms and compare their performances in terms of activity detection accuracy. Experimental results reveal that support vector machine classifier performs the best for human activity detection task with 99.95% average accuracy. However, the time complexity of SVM has cubic relation with number of training examples, making it slow for very large training sets. In contrast, random forest method results in 99.67% average accuracy, slightly less than the best, but much faster training time (around 8x faster than SVM). We also propose a duty cycling and adaptive topology based energy efficiency method for our system, achieving up to 30% system energy reduction with no activity detection performance degradation. Our future work will include more complex static and dynamic activity recognition, predicting human behavior based on recognized patterns, and having a fully dynamic energy efficient system. Our vision is that this research will expand the application of non-intrusive, non-wearable smart space environments.

References

[1] A. Fleury, M. Vacher, and N. Noury, "Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *IEEE transactions on information technology in biomedicine*, vol. 14, no. 2, pp. 274–283, 2010.

[2] Y.-L. Zheng, X.-R. Ding, C. C. Y. Poon, B. P. L. Lo, H. Zhang, X.-L. Zhou, G.-Z. Yang, N. Zhao, and Y.-T. Zhang, "Unobtrusive sensing and wearable devices for health informatics," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 5, pp. 1538–1554, 2014.

[3] V. Stanford, "Using pervasive computing to deliver elder care," *IEEE Pervasive computing*, vol. 1, no. 1, pp. 10–13, 2002.

[4] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," in *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 1. IEEE, 2006, pp. 1003–1007.

[5] S. S. Yau, S. K. Gupta, F. Karim, S. I. Ahamed, Y. Wang, and B. Wang, "Smart classroom: Enhancing collaborative learning using pervasive computing technology," in *ASEE 2003 Annual Conference and Exposition*. sn, 2003, pp. 13 633–13 642.

[6] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 254–263.

[7] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*. ACM, 2010, pp. 1–6.

[8] Y. Lu and S. Velipasalar, "Autonomous footstep counting and traveled distance calculation by mobile devices incorporating camera and accelerometer data," *IEEE Sensors Journal*, 2017.

[9] M. Pizzotti, L. Perilli, M. del Prete, D. Fabbri, R. Canegallo, M. Dini, D. Masotti, A. Costanzo, E. Franchi Scarselli, and A. Romani, "A long-distance rf-powered sensor node with adaptive power management for iot applications," *Sensors*, vol. 17, no. 8, p. 1732, 2017.

[10] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 588–599.

[11] C. C. Aggarwal, *Managing and mining sensor data*. Springer Science & Business Media, 2013.

[12] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 4. IEEE, 2001, pp. 2033–2036.

[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[14] J. Chawla and M. Wagner, "Using machine learning techniques for user specific activity recognition." in *INC*, 2016, pp. 25–29.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2834907, IEEE Internet of Things Journal

[15] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," *Pattern Recognition and Image Analysis*, pp. 289–296, 2011.

[16] A. Mannini, S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell, "Activity recognition using a single accelerometer placed at the wrist or ankle," *Medicine and science in sports and exercise*, vol. 45, no. 11, p. 2193, 2013.

[17] Y. Vaizman, K. Ellie, and G. Lanckriet, "Recognizing detailed human context in-the-wild from smartphones and smartwatches," *arXiv preprint arXiv:1609.06354*, 2016.

[18] Y. Luo, M. Huo, and Z. Zhu, "Design of the automatic detection system for the number of people based on the ultrasonic ranging array," in *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016 IEEE*. IEEE, 2016, pp. 1697–1700.

[19] A. Ghosh, A. Sanyal, A. Chakraborty, P. K. Sharma, M. Saha, S. Nandi, and S. Saha, "On automatizing recognition of multiple human activities using ultrasonic sensor grid," *Unknown*, unknown.

[20] I. Frederix, "Internet of things and radio frequency identification in care taking, facts and privacy challenges," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*. IEEE, 2009, pp. 319–323.

[21] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the internet of things," *Cluster of European Research Projects on the Internet of Things, European Commision*, vol. 3, no. 3, pp. 34–36, 2010.

[22] T. Cho, J.-H. Kim, H.-J. Cho, S.-H. Seo, and S. Kim, "Vulnerabilities of android data sharing and malicious application to leaking private information," in *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*. IEEE, 2013, pp. 37–42.

[23] K. Davis, E. Owusu, V. Bastani, L. Marcenaro, J. Hu, C. Regazzoni, and L. Feijs, "Activity recognition based on inertial sensors for ambient assisted living," in *Information Fusion (FUSION), 2016 19th International Conference on*. IEEE, 2016, pp. 371–378.

[24] D. J. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE intelligent systems*, vol. 27, no. 1, pp. 32–38, 2012.

[25] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[26] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015.

[27] D. Janakiram, V. Reddy, and A. P. Kumar, "Outlier detection in wireless sensor networks using bayesian belief networks," in *Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on*. IEEE, 2006, pp. 1–6.

[28] K. Zhang, S. Shi, H. Gao, and J. Li, "Unsupervised outlier detection in sensor networks using aggregation tree," *Advanced data mining and applications*, pp. 158–169, 2007.

[29] T. Idé, S. Papadimitriou, and M. Vlachos, "Computing correlation anomaly scores using stochastic nearest neighbors," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 523–528.

[30] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6067–6074, 2014.

[31] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *arXiv preprint arXiv:1702.05747*, 2017.

[32] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher, "Deepiot: Compressing deep neural network structures for sensing systems with a compressor-critic framework," *unknown*, 2017.

[33] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016.

[34] A. Sharma, Y.-D. Lee, and W.-Y. Chung, "High accuracy human activity monitoring using neural network," in *Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on*, vol. 1. IEEE, 2008, pp. 430–435.

[35] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104–122, 2014.

[36] J. A. Khan, H. K. Qureshi, and A. Iqbal, "Energy management in wireless sensor networks: A survey," *Computers & Electrical Engineering*, vol. 41, pp. 159–176, 2015.

[37] W. K. Seah, Z. A. Eu, and H.-P. Tan, "Wireless sensor networks powered by ambient energy harvesting (wsn-heap)-survey and challenges," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*. Ieee, 2009, pp. 1–5.

[38] Y. K. Tan and S. K. Panda, "Energy harvesting from hybrid indoor ambient light and thermal energy sources for enhanced performance of wireless sensor nodes," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4424–4435, 2011.

[39] M. P. Buric, G. Kusic, W. Clark, and T. Johnson, "Piezo-electric energy harvesting for wireless sensor networks," in *Wireless and Microwave Technology Conference, 2006. WAMICON'06. IEEE Annual*. IEEE, 2006, pp. 1–5.

[40] R. Doost, K. R. Chowdhury, and M. Di Felice, "Routing and link layer protocol design for sensor networks with wireless energy transfer," in *Global telecommunications conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–5.

[41] A. Pughat and V. Sharma, "Performance analysis of an improved dynamic power management model in wireless sensor node," *Digital Communications and Networks*, vol. 3, no. 1, pp. 19–29, 2017.

[42] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*. IEEE, 2002, pp. 575–578.

[43] Y. Wang, "Topology control for wireless sensor networks," in *Wireless sensor networks and applications*. Springer, 2008, pp. 113–147.

[44] X. Yang and N. H. Vaidya, "A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay," in *Real-Time and Embedded Technology and Applications Symposium, 2004. Proceedings. RTAS 2004. 10th IEEE*. IEEE, 2004, pp. 19–26.

[45] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 2014, pp. 68–81.

[46] P. Vinu, P. Sherimon, and R. Krishnan, "Towards pervasive mobile learning–the vision of 21st century," *Procedia-Social and Behavioral Sciences*, vol. 15, pp. 3067–3073, 2011.

[47] A. Coronato, G. De Pietro, and M. Esposito, "A semantic context service for smart offices," in *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, vol. 2. IEEE, 2006, pp. 391–399.

[48] S. Offermans, A. K. Gopalakrishna, H. van Essen, and T. Özçelebi, "Breakout 404: a smart space implementation for lighting services in the office domain," in *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*. IEEE, 2012, pp. 1–4.

[49] J. Bangali and A. Shaligram, "Design and implementation of security systems for smart home based on gsm technology," *International Journal of Smart Home*, vol. 7, no. 6, pp. 201–208, 2013.

[50] J. Moehrmann, X. Wang, and G. Heidemann, "Motion based situation recognition in group meetings," in *Image Processing: Machine Vision Applications III*, vol. 7538. International Society for Optics and Photonics, 2010, p. 75380N.

[51] H. Nait-Charif and S. J. McKenna, "Tracking the activity of participants in a meeting," *Machine Vision and Applications*, vol. 17, pp. 83–93, 2006.

[52] M. Zobl, F. Wallhoff, and G. Rigoll, "Action recognition in meeting scenarios using global motion features," in *Proc. of Fourth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS), Graz, Austria*, 2003.

[53] A. Helal, D. J. Cook, and M. Schmalz, "Smart home-based health platform for behavioral monitoring and alteration of diabetes patients," *Journal of diabetes science and technology*, vol. 3, no. 1, pp. 141–148, 2009.

[54] D. Yang, B. Xu, K. Rao, and W. Sheng, "Passive infrared (pir)-based indoor position tracking for smart homes using accessibility maps and a-star algorithm," *Sensors*, vol. 18, no. 2, p. 332, 2018.

[55] Y.-L. Hsu, P.-H. Chou, H.-C. Chang, S.-L. Lin, S.-C. Yang, H.-Y. Su, C.-C. Chang, Y.-S. Cheng, and Y.-C. Kuo, "Design and implementation of a smart home system using multisensor data fusion technology," *Sensors*, vol. 17, no. 7, p. 1631, 2017.

[56] X. Luo, Q. Guan, H. Tan, L. Gao, Z. Wang, and X. Luo, "Simultaneous indoor tracking and activity recognition using pyroelectric infrared sensors," *Sensors*, vol. 17, no. 8, p. 1738, 2017.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[58] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[59] A. Mannini and A. M. Sabatini, "On-line classification of human activity and estimation of walk-run speed from acceleration data using support vector machines," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 3302–3305.

[60] S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, and P. S. Freedson, "Multisensor data fusion for physical activity assessment," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 3, pp. 687–696, 2012.

[61] G. Molina and E. Alba, "Location discovery in wireless sensor networks using metaheuristics," *Applied Soft Computing*, vol. 11, no. 1, pp. 1223–1240, 2011.

[62] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

Neha Belapurkar is an MSc student in the Electrical and Computer Engineering Department of San Diego State University. She received her Bachelor of Engineering in Electronics and Telecommunication from Savitribai Phule Pune University (formerly University of Pune), India. Her research interests include energy efficient sensor networks, human behavior modeling for the Internet of Things, and adaptive control in robotics and autonomous systems.

Jacob Harbour is a BSc student in Electrical and Computer Engineering Department of San Diego State University. His research interests include embedded systems and VLSI.

Sagar Shelke is an MSc student in the Electrical and Computer Engineering Department of San Diego State University. He received his Bachelor of Engineering in Electronics Engineering from University of Mumbai, India. His research interests include use of computer vision and machine learning techniques for activity recognition in smart spaces.

Baris Aksanli is an assistant professor in the Electrical and Computer Engineering Department of San Diego State University. He received his PhD and MS degrees in Computer Science from UC San Diego, and two BS degrees in Computer Engineering and Mathematics from Bogazici University, Turkey. His research interests include energy efficient cyber physical systems, human behavior modeling for the Internet of Things, big data for energy efficient large-scale systems. He won the Internet2 IDEA Award with his work in Lawrence Berkeley National Laboratory and Spontaneous Recognition Award from Intel.