SenseTribute: Smart Home Occupant Identification via Fusion Across On-Object Sensing Devices

Jun Han
Carnegie Mellon University
Moffett Field, CA
junhan@cmu.edu

Shijia Pan
Carnegie Mellon University
Moffett Field, CA
shijiapan@cmu.edu

Manal Kumar Sinha
Carnegie Mellon University
Moffett Field, CA
manalkus@cmu.edu

Hae Young Noh
Carnegie Mellon University
Pittsburgh, PA
noh@cmu.edu

Pei Zhang
Carnegie Mellon University
Moffett Field, CA
peizhang@cmu.edu

Patrick Tague
Carnegie Mellon University
Moffett Field, CA
tague@cmu.edu

ABSTRACT

Occupant identification proves crucial in many smart home applications such as automated home control and activity recognition. Previous solutions are limited in terms of deployment costs, identification accuracy, or usability. We propose *SenseTribute*, a novel occupant identification solution that makes use of existing and prevalent on-object sensors that are originally designed to monitor the status of objects they are attached to. *SenseTribute* extracts richer information content from such on-object sensors and analyzes the data to accurately identify the person interacting with the objects. This approach is based on the physical phenomenon that different occupants interact with objects in different ways. Moreover, *SenseTribute* may not rely on users' true identities, so the approach works even without labeled training data. However, resolution of information from a single on-object sensor may not be sufficient to differentiate occupants, which may lead to errors in identification. To overcome this problem, *SenseTribute* operates over a sequence of events within a user activity, leveraging recent work on activity segmentation. We evaluate *SenseTribute* using real-world experiments by deploying sensors on five distinct objects in a kitchen and inviting participants to interact with the objects. We demonstrate that *SenseTribute* can correctly identify occupants in 96% of trials without labeled training data, while per-sensor identification yields only 74% accuracy even with training data.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**;

KEYWORDS

Occupant Identification; On-object Sensing; Sensor Fusion

BuildSys, November 8-9, 2017, Delft, Netherlands

ACM ISBN 978-1-4503-5544-5/17/11...$15.00
https://doi.org/10.1145/3137133.3137152

ACM Reference Format:
Jun Han, Shijia Pan, Manal Kumar Sinha, Hae Young Noh, Pei Zhang, and Patrick Tague. 2017. *SenseTribute*: Smart Home Occupant Identification via Fusion Across On-Object Sensing Devices. In *Proceedings of The 4th International Conference on Systems for Energy-Efficient Build Environments (BuildSys).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3137133.3137152

1 INTRODUCTION

Occupant identification is fundamental in providing many value-added services for smart homes. Personalized home control such as comfort adjustments for lighting and HVAC proves to be important for user convenience as well as energy and cost savings [5, 15, 39]. Furthermore, occupant identification supports activity recognition and/or occupant behavior analysis [43].

Prior works investigate the use of body-worn sensors for occupant identification [20, 21, 33]. Such solutions, however, are intrusive and are less practical because users are required to always carry or wear the sensors. To solve this problem, infrastructure-based solutions have also been explored. However, they make use of sensors that may invade privacy, such as cameras and microphones [30, 41]. To overcome such problems, researchers also introduce solutions leveraging special purpose sensors such as infrared or vibration sensors [24, 27, 36, 37]. Because these solutions deploy the sensors specifically for occupant identification purposes, the solutions come at high hardware and installation costs. Researchers also explore existing infrastructure, such as WiFi, to help identify occupants [43, 44]. However, they make strong assumptions – requiring a user to walk in a straight line, or to stay within a line-of-sight between transceivers – limiting their practicality.

Hence, to overcome the aforementioned limitations of prior work and provide a more practical and yet cost effective solution, we ask the following question – instead of building and deploying specific sensors to provide a practical occupant identification solution, can we leverage sensing capabilities of existing IoT devices within a smart home? To answer this question, we observe an emerging trend in commercial *on-object sensing devices* [1–3, 18, 31], which are detachable wireless sensor nodes that retrofit home objects such as doors, windows, drawers, and/or refrigerators, to monitor and report the object status over the home network. These devices are already prevalent, and are projected to be more ubiquitous throughout smart homes [16, 35].

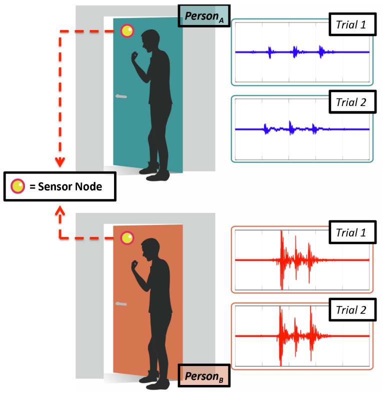


Figure 1: *SenseTribute* utilizes the physical phenomenon that different people interact with objects at home (e.g., knocking or opening a door) differently such that resulting signals captured by on-object sensing devices are (1) sufficiently differentiable across different home occupants; and (2) similar within different trials of the same occupant.

On-object sensing devices are typically equipped with accelerometers and/or gyroscopes to monitor object status (e.g., door opened or closed). However, we explore the possibility of re-purposing these devices to provide more expressive data rather than just object status. Specifically, we find that the way a person interacts with an object is rather unique and can differentiate among people. For example, different family members tend to open a door or refrigerator in different manners, possibly due to different physical build, strength, and habit. We present *SenseTribute*, a novel occupant identification mechanism for smart home settings, which takes advantage of this physical phenomenon using representative features from accelerometer and gyroscope measurements to distinguish home occupants. *SenseTribute* enables ***attribution*** of ***sensory*** measurements to the originating user, hence the name. Figure 1 depicts an example of repeated accelerometer measurements for two different users, highlighting the important capabilities to distinguish between users and match subsequent user readings.

SenseTribute utilizes supervised learning techniques to first train the model using collected bootstrapping data as training data, along with the corresponding training labels. Subsequently, upon collecting testing data, *SenseTribute* performs classifications using the trained model. While some application scenarios may ask the occupants to initially provide the training labels (e.g., names of persons associated with the training data) during a bootstrapping phase, such approach may be impractical in other scenarios due to usability problems. Hence, we design *SenseTribute* to be robust against this challenge, specifically, even in scenarios where the users do not provide the training labels. In such cases, *SenseTribute* is still able to identify the occupants, but with *pseudo-identifiers* instead of explicit identifiers such as names (e.g., *Persons A* and *B* rather than *Alice* and *Bob*). Pseudo-identifiers still support most of smart home applications such as aforementioned personalized home control and identifying occupants of recognized activities, and may even be suitable for privacy-preserving applications. This is made possible because *SenseTribute infers* the labels by utilizing unsupervised learning techniques to cluster the bootstrapping data into cluster identifiers. Subsequently, *SenseTribute* trains the model using the training data and the corresponding cluster identifiers as *quasi*-training labels. The *quasi*-training labels are labels that do not have information to map the cluster identifiers to occupant identities such as names.

Even with this classification approach, each on-object sensing device provides limited information content, yielding low identification accuracy. Performance degrades even further if training labels are not provided. In order to solve this challenge, we introduce *SenseTribute*'s *Ensemble Module* to amplify the information content across multiple on-object sensing devices, thereby boosting the accuracy of the overall occupant identification. The *Ensemble Module* relies on related research on *activity segmentation* [23, 28, 40], which segments out a sequence of events belonging to a single activity segment performed by the same person, out of entire sensor data streams of multiple persons' events. For example, a *cooking breakfast* activity may consist of multiple sensor events performed by a same user such as *opening the refrigerator*, followed by *taking out a frying pan*, followed by *turning on the stove*.

We design and implement *SenseTribute* and evaluate its feasibility by conducting real-world empirical experiments with five distinct sensor pairs (accelerometer and gyroscope), each attached to five different objects – door, refrigerator, drawer, towel dispenser, and window. We invite five participants to perform a sequence of events that interact with these objects. We choose five participants, as this number is greater than an average of 3.14 persons per home in the United States [10]. From our empirical analysis, *SenseTribute* is able to correctly identify occupants with 96% accuracy even when the training labels are not provided, while the average accuracy from per-object identification yields 74%, even with training labels. Overall, we make the following contributions in this paper.

- We design *SenseTribute* to extract expressive data from on-object sensors and identify occupants in a smart home.
- We demonstrate how *SenseTribute* achieves high identification accuracy by combining observations across several sensors on different objects, even without labeled training data.
- We evaluate *SenseTribute* by conducting real-world experiments with participants interacting with different objects in a kitchen, each interfaced with a sensor node.

The remainder of this paper is organized as following. We present background information and related work in Section 2. We then present the details of *SenseTribute*'s design and implementation in Section 3, and its evaluation results in 4. Subsequently, we present discussion and conclusion in Sections 5 and 6, respectively.

2 BACKGROUND AND RELATED WORK

We first present *on-object sensing devices* and their prevalence. We then introduce *activity segmentation* often studied in the field of activity recognition, and how *SenseTribute* utilizes it. Furthermore, we describe related work on*occupant identification*.

2.1 On-Object Sensing Devices

On-object sensing devices are popular smart home gadgets that enable home owners to monitor the status of various objects – such as doors and drawers – by simply attaching the device to each object. An on-object device is commonly equipped with inertial sensors (e.g., accelerometer and/or gyroscope), which sense the movement of object it is attached to. The sensor signals are then processed to output object status – such as door or drawer open/close – and reports the events to home owner's smartphone over the cloud. Companies such as Notion [2] and Samsung SmartThings [3] are industry leaders, while there are many other commercial solutions from various vendors [1, 18, 31]). These devices are projected to be more prevalent in smart homes in the near future [16, 35]. We design *SenseTribute* to extract more expressive data than mere status of objects, namely to infer the identities of occupants in a home. Hence, *SenseTribute* inherently eliminates the costly need to build and deploy specific sensing devices for occupant identification.

2.2 Activity Segmentation

Activity segmentation – an actively studied topic in activity recognition field – segments out a sequence of events that are performed by a single occupant. However, this is a difficult problem because different events are performed by different persons that may be temporally overlapping within a single stream of sensor data. Hence researchers make use of combinations of sensor patterns and temporal information to identify a sequence of *events* that constitute a single *activity segment* [23, 28, 40]. For example, consider $Person_A$ *cooking breakfast*, while $Person_B$ *watching TV* in the living room. The *cooking breakfast* activity segment may consist of a sequence of *events* such as: *{kitchen door opening*, *fridge door opening*, and *pasta drawer opening}*. On the other hand, *watching TV* activity segment may consist of a sequence of *events* such as: *{sitting down on sofa*, *taking out remote control*, *TV turning on}*. Each of the sequence of *events* belonging to the same *activity segment* are grouped together, even though there may be temporal overlaps between individual *events*. *Activity segmentation* is one of the important foundations when designing *SenseTribute*. Specifically, *Ensemble Module* exploits the above property that a sequence of events within an activity segment is performed by the same user, enabling *SenseTribute* to combine the confidence of a sequence of events (see Section 3.5).

2.3 Occupant Identification

Smart home occupant identification is an important problem. Personalized home control is gaining much attention such as user-specific comfort adjustments for lighting and HVAC for convenience as well as energy efficiency [5, 15, 39]. Due to potentially significant cost-savings, this is a real-world problem that are heavily studied by appliance manufacturers as well. Furthermore, occupant identification supports many activity recognition applications. This is because understanding *who* is performing the recognized activity is a building block to associating activities to individual occupants, rather than just knowing that someone at home has performed the activity [43] (e.g., splitting costs between roommates based on individual energy consumption or even simply providing feedback to which family member consumes most energy).

Due to the importance of occupant identification problem, prior works explore solutions by deploying infrastructure-based sensors. Researchers utilize ultrasonic-based doorway sensors to capture the movements and the physical characteristics such as height [24] and/or weight [27] of persons. Researchers also utilize structural vibration-based sensors to detect occupant's gait patterns [36, 37]. Occupants strike the floor with different gait patterns, inducing unique structural vibration waveform. Similarly, researchers also exploit changes in body electric potential due to walking [22]. While these solutions are promising first steps, all of them utilize hardware that are specifically built and deployed to solve the occupant identification problem. This inevitably incurs high cost both in terms of hardware as well as deployment costs.

Prior work also explore solutions that use existing infrastructure such as Wi-Fi to utilize channel state information (CSI) induced by occupant's walking pattern [43, 44]. While these solutions do not incur additional hardware or deployment costs, they face challenges in limited deployment practicality. This is because these solutions require the occupants to either (1) only walk in a straight line [43]; or (2) stay within the line of sight between WiFi transceivers [44].

As opposed to the related work, *SenseTribute* inherently reduces the hardware and deployment cost because it *utilizes existing and prevalent on-object sensing devices* deployed by users, and simultaneously provides a more practical occupant identification by performing simple software modifications to extract information necessary to identify the occupants.

3 DESIGN AND IMPLEMENTATION

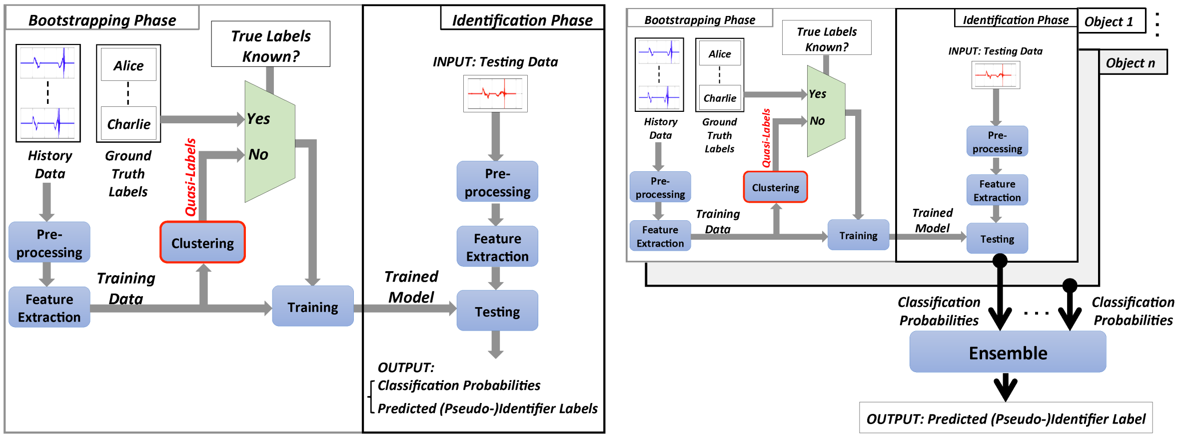
We now present *SenseTribute*'s design and implementation. We describe the details *SenseTribute*'s algorithm when the training labels are known and unknown. We also explain how *SenseTribute* *ensembles* different objects to amplify the identification accuracy.

3.1 *SenseTribute* Overview

SenseTribute's goal is to identify the occupants by leveraging signals of on-object sensors utilizing supervised learning techniques. *SenseTribute* is divided into two phases – a *Bootstrapping* and *Identification Phases*. First, during the *Bootstrapping Phase*, *SenseTribute* trains a classification model from the collected sensor data (i.e., history data). Subsequently, in its *Identification Phase*, *SenseTribute* tests newly collected sensor data, to finally identify the occupant.

In order to train the model for classification, the system requires training labels (i.e., ground truth occupant identity corresponding to the collected history data). However, it may be more practical for certain applications to not collect user provided training labels (e.g., to increase usability). We account for this problem, and design *SenseTribute* to automatically adapt its training scheme based on the availability of user-provided labels.

We present the flow chart diagrams to depict the overall *SenseTribute* design as shown in Figure 2(a). Specifically, when the training labels are provided to the system by the users (i.e., *known labels* scenario), *SenseTribute* utilizes the traditional supervised learning techniques, by taking as input for the *Training Module*, the (1) training label and (2) data. For the training label, *SenseTribute* utilizes the user-provided ground truth labels. For the training data, *SenseTribute* first processes the collected *history data* in *Pre-processing Module*), and then extracts necessary features in *Feature Extraction Module*. Finally, at the end of the *Identification Phase*, the *Testing Module* outputs the *Predicted Occupant Label*, along with the classification probabilities of all the potential classes.



(a) *SenseTribute*'s Flow Chart Diagram per Object

(b) Ensemble of n Objects

Figure 2: (a) Flow-chart diagram of *SenseTribute* for an individual object. During *Bootstrapping Phase*, collected *history data* and ground truth labels are used to train a model. If ground truth label is unavailable, *SenseTribute* clusters history data to *infer* training labels. Subsequently, during *Identification Phase*, *SenseTribute* uses the trained model to predict occupant identity. (b) Subsequent to the identification phase in (a), *SenseTribute* further *ensembles* classification probabilities from n different objects and predicts occupant identity with higher accuracy.

However, *SenseTribute* is also capable of operating even when users do not provide the ground truth labels (i.e., *unknown labels* scenario), by utilizing a hybrid approach of unsupervised and supervised learning techniques. Similar to *known labels* scenario, the *history data* are used to process and extract features. However, the features are now input to *Clustering Module*, which computes and outputs the clustered indices. We use these indices as *quasi-labels* that substitute the ground truth labels. *Quasi-labels* represent different clusters, or groups, corresponding to the history data. However, as opposed to the ground truth labels, *quasi-labels* (1) do not carry information to be directly mapped to specific occupant's explicit identities such as names; and (2) are prone to some amount of error due to clustering. Finally, the *Testing Module* outputs the predicted *pseudo-identifier labels*, along with the classification probabilities of all the potential classes. Similar to *quasi-labels*, *pseudo-identifiers* do not carry information to be directly mapped to the specific occupant identities such as names, but are still valuable because they can be used to sufficiently distinguish different occupants (e.g., $Person_A$ vs. $Person_B$). While clustering algorithms such as K-Means provide linear decision boundaries, we design *SenseTribute* using classification as the backbone framework for the simplicity of integrating both *known* and *unknown labels* scenarios.

Since the information content from a single object may not be sufficient to accurately identify the occupants, we introduce *SenseTribute*'s *Ensemble Module* subsequent to the *Identification Module* of each object, to "ensemble" the classification probabilities to arrive at a higher occupant identification accuracy. Figure 2(b) depicts the corresponding flowchart diagram.

3.2 Pre-processing and Feature Extraction

3.2.1 Pre-processing. Prior to extracting the features from the raw sensor data, we first perform noise reduction to increase the

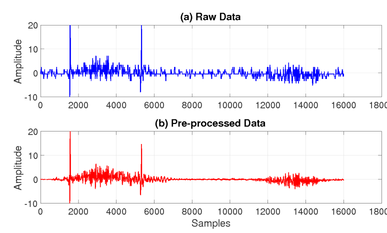


Figure 3: We use spectral subtraction to increase the Signal-to-Noise Ratio (SNR). (a) depicts raw time-series gyroscope signal; and (b) depicts the resulting spectral subtraction.

Signal-to-Noise Ratio (SNR) and the subsequent classification and clustering performance. We make use of *spectral subtraction* [8], used in speech recognition to remove background noise, because the ambient noise is similar to inherent sensor noise. Spectral subtraction performs the operation $S(\omega) = Y(\omega) - N(\omega)$, where $Y(\omega)$, $S(\omega)$, and $N(\omega)$, are the frequency-domain spectra of the noisy sensor reading, desired signal, and noise, respectively. We estimate the noise spectrum $N(\omega)$ by sampling the ambient noise, which can be performed by sensor nodes, in practice, prior to the *Pre-processing Module*. Figure 3 depicts an example of single-axis gyroscope signal corresponding to opening and closing a drawer.

3.2.2 Feature Extraction. *SenseTribute* then performs feature extraction on the pre-processed signal. We extract features from both time and frequency domains as characteristics of the induced signal. We list the features used in this work in Table 1. Vectors x_p and y_q are time and frequency domain representations of the data, respectively, and N and M are the number of elements in x and y, respectively. The Root Mean Square (RMS) (in time or frequency domains as RMS and FFT_{RMS}, respectively) reflects the variation

Feature	Domain	Expression
RMS	time	$\sqrt{1/N\sum_{p=1}^{N}x_p^2}$
FFT_{RMS}	frequency	$\sqrt{1/M\sum_{q=1}^{M}y_q^2}$
$Peak2RMS$	time	$\max(\lvert x\rvert)/\sqrt{1/N\sum_{p=1}^{N}\lvert x_p\rvert^2}$
$Energy$	time	$\sum_{p=1}^{N}log(x_p^2)$
SMA	time	$1/N\sum_{p=1}^{N}\lvert x_p\rvert$
FFT_{max}	frequency	$\max(y_q)$
$Mean$	time	$\frac{1}{N}\sum_{p=1}^{N}x_p$
$Median$	time	$median(x_p)$

Table 1: Features used in *SenseTribute*, where vectors, x_p and y_q are time and frequency domain representations of the pre-processed data, respectively. N and M are the number of elements in x and y, respectively.

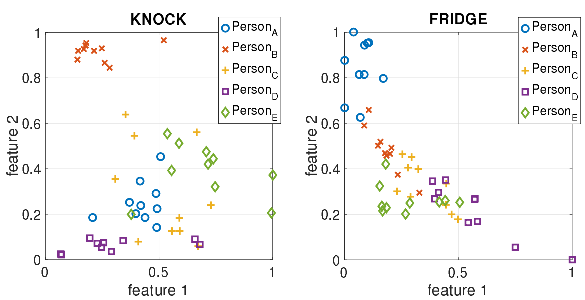


Figure 4: We plot feature pairs for *knock* and *fridge door open/close event types* for comparison. *Knock* plot depicts sufficient separation of features across *Persons A, B, D,* and *E,* while *Person C* has a large overlapping area. *Fridge* plot depicts sufficient separation for *Persons A, B,* and *E,* while *Persons C* and *D* have large overlapping areas.

within a signal segment, a relatively widely used feature that effectively describes the signal. The peak-to-RMS ratio of time domain signal, *Peak2RMS*, measures more detailed signal distribution in addition to *RMS*. For example, a person with thicker finger bones knocking on the door may trigger an impulse signal with a sharp waveform, which may lead to a higher *Peak2RMS* value. We also compute the log energy entropy [14, 34], *Energy*, which measures the signal distribution. Signal magnitude area [7, 9], *SMA*, measures the average of the signal amplitude. The maximum value of frequency domain signal, FFT_{max}, provides the peak amplitude of y_q. Finally, we use the common statistical *mean* and *median* of x_p as measurements of central tendency.

Furthermore, we compare the feature distributions of different occupants by plotting feature pairs. Figure 4 depicts two examples of feature pairs (*Peak2RMS* vs. *SMA*) from two distinct sensors on a door (capturing knocking events) and refrigerator (capturing refrigerator opening and closing events). Each marker represents a feature comparison per occupant (i.e., $Person_A$ to $Person_E$). We make the following two observations. First, we observe that within each sensor, the feature pair provides information to distinguish different occupants at a fairly sufficient manner. For example, for knocking event, $Person_A$, $Person_B$, $Person_D$, and $Person_E$ exhibits sufficient separation, while $Person_C$ exhibits large overlapping area with other occupants. Second, we also observe that across the two events from different objects, different feature pairs contribute to separating the occupants. For example, the feature pairs performed well in distinguishing $Person_D$ for knocking on a door but poorly for opening and closing a refrigerator.

3.3 *Known Labels* Scenario

In the application scenario where the user provides the ground truth labels for the training label, we leverage supervised learning techniques to perform occupant identification. We implement *SenseTribute*'s classification modules (i.e., *Training* and *Testing Modules*) with Support Vector Machines (SVM) [6] using Radial Basis Function (RBF) kernel. We choose SVM because it requires relatively small amount of training data to achieve high classification accuracy, compared to other classification methods such as neural networks. We implement the modules using publicly available LIBSVM [11]. We use multi-class SVM classification to classify n occupants in smart home settings, where $n \geq 2$. First, the *Training Module* takes as input aforementioned feature vector of the training data and the training label to compute the trained model. This module concludes the end of *Bootstrapping Phase*.

Second, the *Testing Module* in the *Identification Phase* takes as input the trained model and the feature vector of the testing data. This module performs the SVM classification to output the following: (1) classification probabilities, $Pr[O = o_i]$, of all possible classes (i.e., occupants), $o_1, ..., o_n$; and (2) final predicted label which is the occupant, o_i that yields highest $Pr[O = o_i]$.

3.4 *Unknown Labels* Scenario

When he user does not provide any training labels, we leverage a hybrid approach of supervised and unsupervised learning techniques to perform occupant identification. The **unknown** and **known labels** scenarios are equivalent in computing the feature vector. However, it differs in that the system no longer has the given training labels to be input to the classification modules. Hence, we ***infer*** the training labels using the unsupervised learning techniques.

Specifically, we implement the *Clustering Module* with K-Means clustering [25, 32], which takes a feature vector from the history data and the number of cluster groups K. We assume that K, i.e., number of occupants in a home is known (see Section 5.2). K-Means clustering algorithm groups each of the input observations into K clusters with the smallest distance to the corresponding computed centroid. This module outputs the clustered indices, which will be subsequently used as the training label, in the *Training Module*. We note that the clustered indices are *quasi-labels*, which does not map directly to occupants' explicit identifiers (e.g., occupants' names such as *Amy* vs. *Bob*). However, *quasi-labels* provide adequate information to identify occupants to their *pseudo-identifiers* (e.g., $Person_A$ vs. $Person_B$) at the end of the *Identification Phase*.

3.5 Ensemble Module

Each object's identification accuracy (output from Figure 2(a)) are limited because each object has either low resolution of information, or same occupant may occasionally interact with the object in slightly different manner. Furthermore, for the case of the *unknown labels*, accumulated errors from clustering contributes to lower per-object identification accuracy. Hence, we design *Ensemble Module* to amplify the occupant identification accuracy. *SenseTribute ensembles* identification probabilities of individual objects, as depicted in the flow chart diagram in Figure 2(b). Specifically, this module takes as input the resulting classification probabilities, $Pr[O = o_i]$, where $i = 1, ..., n$ (indicating n potential classes, i.e., n occupants), from each of the *Testing Modules* belonging to m different sensors each interfaced with different objects, defined as $\mathcal{S}_j$, where $j = 1, ..., m$. Subsequently, this module outputs the final predicted occupant identity, o^*, which has an amplified identification accuracy, which we evaluate in Section 4.

To implement the *ensemble algorithm*, we formulate this problem as the conditional probability depicted in Equation 1:

$$o^* = \arg\max_{o_i} Pr[O = o_i \mid \mathcal{S}_1, ..., \mathcal{S}_m], \quad (1)$$

This finds the most likely occupant o^* given sensor data $\mathcal{S}_1, \ldots, \mathcal{S}_m$. We assume independence across each sensor, $\mathcal{S}_j$, and use Bayes' theorem to rewrite this formulation as shown in Equation 2:

$$o^* = \arg\max_{o_i} \prod_{j=1}^{m} Pr[O = o_i \mid \mathcal{S}_j], \quad (2)$$

where each of the probabilities, $Pr[O = o_i \mid \mathcal{S}_j]$, is equivalent to the output probabilities, $Pr[O = o_i]$, of each sensor.

4 EVALUATION

In this section, we first present the experiment setup and evaluate *SenseTribute*'s performance.

4.1 Experiment Setup

4.1.1 Apparatus. We conduct our experiment by facilitating five objects in a kitchen each with a sensor node. The objects include – *door*, *fridge door*, *drawer*, *towel dispenser*, and *window*. Each sensor node comprises of an Arduino Uno [4] interfaced with ADXL335 tri-axis accelerometer [17] and LPY503AL dual-axis gyroscope [42], sampling each axis at 5KHz. These sensor are attached to the objects so that the accelerometer's *Z-Axis* is perpendicular to the object's surface, and the gyroscope's *X-Axis* revolves around an imaginary line perpendicular to the floor, as depicted in Figure 5. The rest of this evaluation only considers using the two axes, and we discuss practical considerations later in Section 5.3.

4.1.2 Data Collection. We invited five participants, which is higher than an average people per family of 3.14 persons [10]. We ask each participant to perform a predefined activity of operating the aforementioned objects – i.e., opening closing *door*, *fridge*, *cabinet drawer* and window, and pulling towel from *towel* dispenser. We now refer to *event type* as the *event type – object* pair (e.g., *door* represents opening/closing door). We performed the study after obtaining approval of Institutional Review Board (IRB) and conducted the experiment in compliance to the IRB's recommendations.

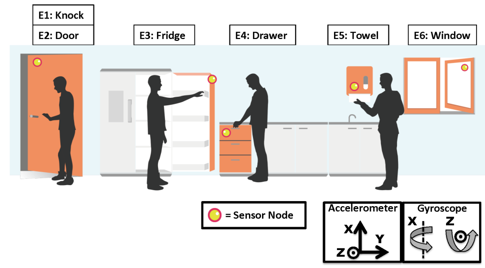


Figure 5: We depict experiment setup conducted in a kitchen. Participants are asked to perform prescribed events.

	Gender	Height	Weight	Age
Person A	Female	1.59m	51kg	28
Person B	Male	1.75m	76kg	33
Person C	Male	1.65m	45kg	27
Person D	Female	1.65m	50kg	30
Person E	Female	1.85m	95kg	26

Table 2: Table presents demographics of five participants.

4.2 Known Labels Scenario

We evaluate *SenseTribute* when the system is given the training labels, performing an SVM classification as described in Section 3.3. We report the classification accuracy by varying number of occupants from $i = 2, ..., 5$, where each variation is an average of all possible combinations, $\binom{5}{i}$ (e.g., *3 occupants* case is an average of $\binom{5}{3} = 10$ instances). Each instance of combination is an average result of a 10-fold cross-validation (i.e., Leave-One-Out) as we have ten trials per occupant. Figure 6 depicts the result for all six *event types*. We observe that as the number of occupants increases, the classification accuracy decreases, for each of the *event types*. This is intuitive as introducing more *classes* (i.e., occupants) to the classifier introduces more room for error. The average of all six *event types* with five number of occupants yields 74%, as reported in Section 1.

We further note that different *event types* result in different accuracy, due to certain objects being more distinctive. We observe that objects that provide relatively consistent interaction yielded better classification accuracy. For example, *knocking on door* and *opening and closing a drawer* leads to more information to sufficiently distinguish occupants, while dispensing towel did not produce sufficient information on its own. We notice that the towel often got ripped during dispensing in multiple trials, and consequently yielded different interactions even within same subject.

We also report the classification accuracy per occupant ($Person_A$ through $Person_E$), per *event type* (Figure 7).Certain *event type* yields high classification accuracy for one person, but low for another person while a different *event type* yields flipped results for the same pair of persons. For example, *Knock* and *Door event types* yield relatively high and low classification accuracy for $Person_A$, respectively. However, the two *event types* conversely yield relatively low and high accuracy for $Person_B$, respectively. *SenseTribute* takes advantage of such phenomenon to amplify its identification accuracy in its *Ensemble Module* (Section 3.5).

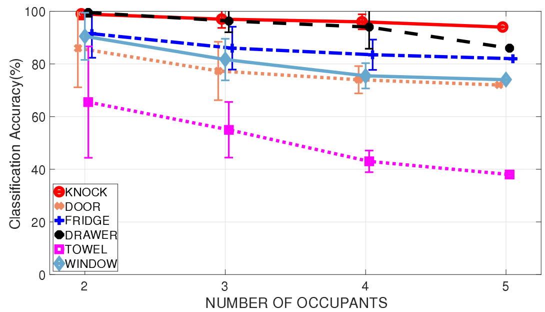


Figure 6: Figure depicts classification accuracy by varying number of occupants of each *event type* (for *known labels* scenario). Each data point is an average accuracy of all combinations within each number of occupants. As the number of occupants increases, classification accuracy decreases. The average accuracy of different *event types* for five occupants case yields 74%.

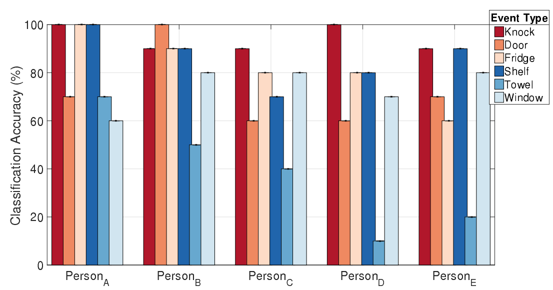


Figure 7: Figure depicts classification accuracy due to different participants. Certain pair of *event types* yield contradicting accuracy across different participants. *SenseTribute* takes advantage of such phenomenon to amplify the final accuracy in its *Ensemble Module*.

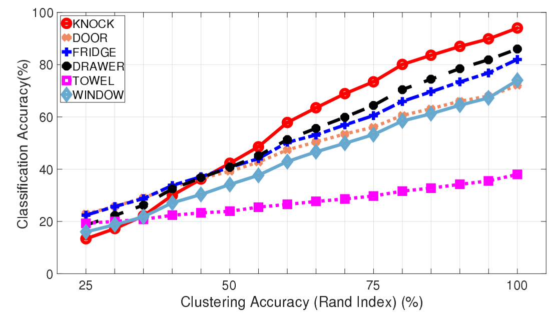


Figure 8: Figure depicts how varying clustering accuracy affects classification accuracy by artificially setting clustering accuracy from 25% to 100% for each *event type* (for five occupants case). Classification accuracy increases as clustering accuracy increases for all *event types*.

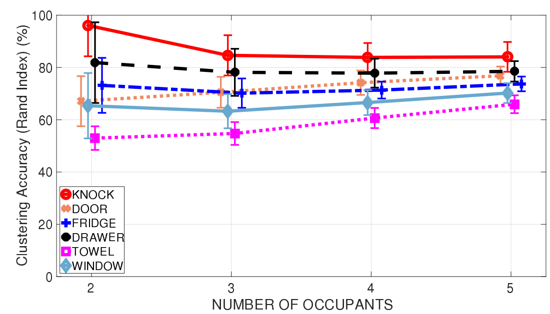


Figure 9: Figure depicts clustering accuracy when varying number of occupants for each *event type* (for *unknown labels* scenario). *Knock* and *Drawer* yield decreasing accuracy while other *event types* yield increasing accuracy, as the number of occupants increase.

4.3 Unknown Labels Scenario

We now evaluate *SenseTribute* when the training labels are not provided by the user. As presented in Section 3.4, *SenseTribute* utilizes a hybrid approach of unsupervised and supervised learning – i.e., using clustering result as *quasi-labels*, to replace the unknown training labels. To provide a comprehensive view of how clustering accuracy affects the classification accuracy, we set clustering accuracy artificially from 25% to 100%, for each *event type*, as depicted in Figure 8. For example, a clustering accuracy of 50% indicates that half of the training labels selected at random are made incorrect on purpose. We repeat this process a thousand times and report the average for each data point in this figure. We show the result of five occupants case as an example. This figure illustrates that as the clustering accuracy increases, the corresponding classification accuracy also increases (with 100% corresponding to known labels).

We now evaluate the performance of the *Clustering Module*. The clustering accuracy is computed as *Rand Index* [38], which is defined as Equation 3:

$$\textit{Clustering Accuracy (Rand Index)} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3)$$

where TP, TN, FP, and FN, are True Positive and Negative, and False Positive and Negative, respectively. Figure 9 depicts the clustering accuracy (i.e., Rand Index), when we vary the number of occupants $i = 2, \ldots, 5$. Each of the data points is an average of all possible combinations of i occupants, $\binom{5}{i}$. Furthermore, we report the average of a thousand iterations for all instances. We note that the clustering accuracy decreases as the number of occupants increase for *Knock* and *Drawer event types*. However, the rest of the *event types* yield results that have increasing clustering accuracy as the number of occupants increase. This is because *Knock* and *Drawer*, which yield high classification accuracy for *Known Labels* scenario, have features that are sufficiently differentiable, while the rest of the *event types* do not follow this trend. Hence, during clustering of two occupants case, the two centroids may be very close to each other, yielding low clustering accuracy. However, when the number of occupants increase, more centroids are introduced, yielding higher clustering accuracy.

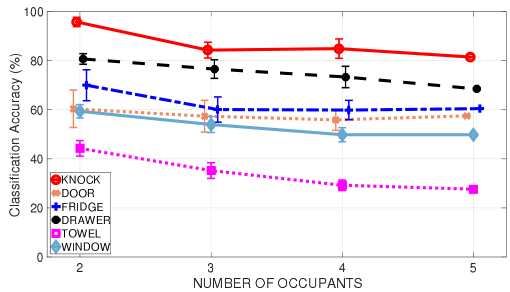


Figure 10: Figure depicts the classification accuracy when varying number of occupants for each *event type*, for the *unknown labels* scenario. As number of occupants increases, corresponding accuracy also decreases.

Finally, we evaluate the classification accuracy of *SenseTribute*'s *Unknown Labels* scenario (i.e., output of *Testing Module*). We compute the classification accuracy in a similar manner to the aforementioned Figure 8, namely purposely degrading the correctness of the training label. Only this time, we take the actual empirical results of clustering accuracy from Figure 9 instead of the artificial numbers. We apply this strategy rather than directly applying the output of the clustering indices as the training label. This is because *Clustering Module* outputs clustered indices, which is at times difficult to map to corresponding ground truth labels. However, this is necessary when computing the final classification accuracy for evaluation purposes. While improving clustering algorithm would certainly help to solve this issue, we concentrate on evaluating the effects of clustering accuracy on classification accuracy. Figure 10 depicts the effect of the classification accuracy as we vary the number of occupants, where each data point, again depicts an average of all possible $\binom{5}{i}$ combinations, and each combination is an average of 10-fold cross validation (i.e., Leave-One-Out). We make two interesting observations. First, similar to Figure 6 of the *Known Labels* scenario, this figure depicts an intuitive trend of decreasing classification accuracy as the number of occupants increase. This trend exists even for the *event types* that have increasing clustering accuracy with number of occupants from Figure 9. This is because the effect of increasing the number of SVM classes outweighs the effect of correct labels. Second, we also observe that classification accuracy are relatively lowered compared to Figure 6 of *Known Labels* scenario due to the incorrect labels.

4.4 *Ensemble* Classification Accuracy

We now evaluate *SenseTribute*'s *Ensemble Module* for both the *known* and *unknown labels* scenarios. To provide a comprehensive view of how the number of *ensemble event types*, and availability of training labels affect the classification accuracy, we present Figure 11. We report the classification accuracy when varying number of events to ensemble from $j = 2, \ldots, 6$, where each variation is an average of all $\binom{6}{j}$ combinations. Again, each combination is an average of 10-fold cross valuation (i.e., Leave-One-Out). We artificially assign equal clustering accuracy per *event type*, again by artificially degrading the correctness of training label accordingly. We degrade different training label at random, and repeat this process for a

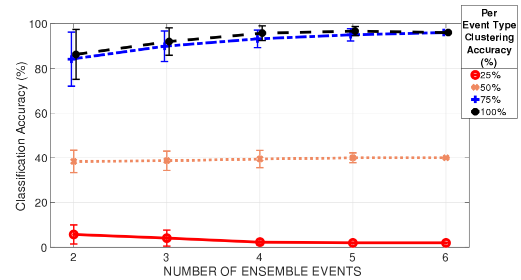


Figure 11: Figure depicts how (1) number of *ensemble event types*; and (2) availability of training labels affect classification accuracy. We artificially assign equal clustering accuracy per *event type*. As number of *ensemble event types* increases, accuracy increases, except for the 25% case. Also, lower per *event type* clustering accuracy yields lower classification accuracy due to more mislabeled training data.

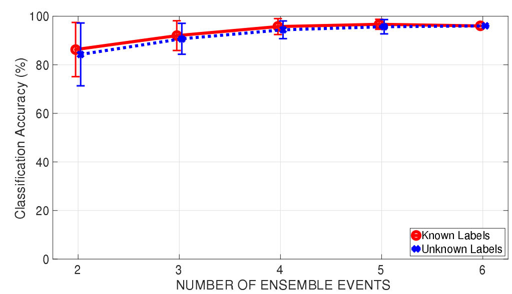


Figure 12: This figure depicts increasing classification accuracy as we ensemble more number of *event types*, for both *Known* and *Unknown Labels* scenarios. We observe high classification accuracy even if the training labels are not known.

thousand times to report an average value. Each of the lines plots depict different clustering accuracy – 25%, 50%, 75%, and 100% – assigned per *event type*. The 100% clustering accuracy line graph represents the *known labels* scenario. We observe the trend of increasing classification accuracy as we ensemble more *event types*. This is intuitive as we have more information content to amplify the confidence of occupant identification. The 25% per *event type* curve does not follow this trend, however, due to the fact that most of the training labels are incorrect, which would actually hurt the performance as the number of *event types* increases.

Noting the effects of number of *event types* and availability of training labels on classification accuracy, we now evaluate the performance of ensemble for both *known* and *unknown labels* scenarios, as depicted in Figure 12. From these two plots, we make the following two observations. (1) We observe that the classification accuracy increases as we ensemble more *event types* for both *known* and *unknown labels* scenarios. For example, we observe for the *unknown labels* scenario, an increase from 84% to 96%. This is intuitive, and in fact, one of the main contributions of *SenseTribute*, as increasing information content ultimately amplifies the accuracy of occupant identification. (2) We observe only a small difference in the resulting classification accuracy between the *known* and *unknown labels*

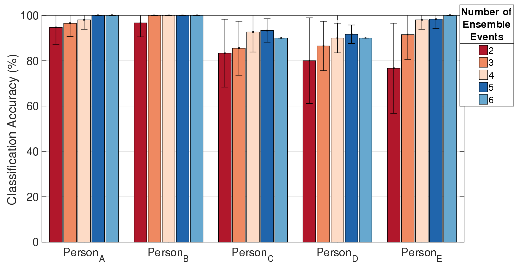


Figure 13: Figure depicts classification accuracy of ensemble of *event types* for different occupants when labels are *known*. As the number of ensemble *event types* increases, the accuracy also increases.

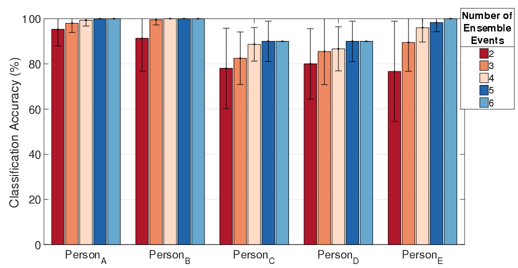


Figure 14: Figure depicts classification accuracy of ensemble of *event types* for different occupants when labels are *unknown*. As the number of ensemble *event types* increases, the accuracy also increases.

scenarios. We further observe that the difference reduces as we ensemble more *event types*. This important observation means that *SenseTribute* provides a practical solution that *does not require users to provide manual labels* with no significant impact on occupant identification.

We also present the classification accuracy per occupant for different number of *event types*. Again, we report the average over all combinations. Figures 13 and 14 depict the corresponding results for *known* and *unknown labels* scenarios, respectively. For both figures, we observe the similar trend as we ensemble more number of *event types*, we achieve higher classification accuracy.

5 DISCUSSION

In this section, we further discuss practical considerations and directions for further study with respect to activity segmentation, unsupervised learning techniques, and sensor calibration.

5.1 Additional Contextual Information

We highlight two additional contextual information that may potentially be helpful for *SenseTribute*, namely *order* and *time* of events. In this work, we design *SenseTribute* to perform occupant identification based on the results of activity segmentation, which provides a sequence of events that are performed by a single person. In Section 4, we evaluate scenarios where the *order* of events (in an activity segment) are same across different participants. However, in practice, there is a high probability that the *order* may vary. For example, when making a bowl of cereal, $Person_A$ may take out a bowl from cabinet, milk from fridge, and cereal from cupboard, while $Person_B$ may perform the same activity in an opposite order. In addition, different occupants may conduct the same activity at different *times* of the day. For example, $Person_A$ usually makes cereal around 8 a.m., while $Person_B$ does the same at 10 a.m. Taking the above two observations into account, we hint at the possibility of a hybrid approach of solving both the activity segmentation and occupant identification problem simultaneously. This hybrid approach would potentially increase the performance with the additional contextual information. Furthermore, the hybrid approach may even increase the identification accuracy despite inconsistencies in different interactions by the same user over time, or similar interactions by different users.

5.2 Unsupervised Learning

Recall that when the training labels are not provided by the user, *SenseTribute* utilizes clustering to *infer* the quasi-training labels. We evaluate our results by clustering the history data during bootstrapping phase. When *SenseTribute* is deployed in practice, we can utilize online learning techniques [12, 13, 29] to improve the results of clustering. This is because, over time, the clustering accuracy would increase as the system collects more data, ultimately leading to potentially higher identification accuracy.

Furthermore, in our evaluation, we assume the knowledge of "K" (i.e., number of occupants) in the K-means clustering algorithm. We make such assumptions because it is practical to have such prior knowledge of how many people live at home. Granted, we note that if guests are introduced to smart home, it may lead to less accurate results. In practice, however, there are clustering methods to estimate the optimal "K", such as Elbow method [26]. Also, there are other clustering algorithms that do not require the number of clusters [19]. However, we leave this study for future work.

5.3 Sensor Calibration

Recall from our evaluation that we deploy sensors on different objects with consistent orientation of accelerometers and gyroscopes as presented in Section 4.1. While we conducted the experiment as a proof-of-concept, in practice, we cannot assume such deployment. Hence, the system would need a simple but important calibration phase, to identify the axes that have relatively richer information content. *SenseTribute* may benefit from the calibration phase, as identifying a specific set of features and axes per object and/or event type would ultimately increase the identification performance.

6 CONCLUSION

We present *SenseTribute*, a smart home occupant identification system that leverages existing and prevalent *on-object sensing devices* equipped with inertial sensors, which are traditionally designed to monitor status of objects such as doors. *SenseTribute* re-purposes these devices, and exploits machine learning techniques to provide a low-cost, non-intrusive, and practical occupant identification system in a smart home with high accuracy, even when training labels are unavailable. Furthermore, *SenseTribute* combines information from multiple sensors on different objects to amplify the identification accuracy. We evaluate *SenseTribute* using real-world experiments with five on-object sensors deployed on distinct objects. The system achieves identification accuracy of 96% when the training labels are unknown, while only achieving per-object accuracy of 74% on average even when the labels are known.

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation (under grants CNS-1645759, CNS-1149611 and CMMI-1653550), Intel and Google. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of CMU, NSF, or the U.S. Government or any of its agencies.

REFERENCES

[1] *MetaSensor: Meet Sensor-1.* https://www.metasensor.com/.
[2] *Notion: Home awareness, simplified.* http://getnotion.com/.
[3] *Samsung SmartThings Multipurpose Sensor.* https://support.smartthings.com/hc/en-us/articles/205382174-Samsung-SmartThings-Multipurpose-Sensor.
[4] Arduino. 2017. *Arduino/Genuino UNO.* https://www.arduino.cc/en/Main/arduinoBoardUno.
[5] Bharathan Balaji, Jian Xu, Anthony Nwokafor, Rajesh Gupta, and Yuvraj Agarwal. Sentinel: Occupancy Based HVAC Actuation Using Existing WiFi Infrastructure Within Commercial Buildings *(ACM SenSys '13)*.
[6] Kristin P. Bennett and Colin Campbell. 2000. Support Vector Machines: Hype or Hallelujah? *SIGKDD Explor. Newsl.* (2000).
[7] Sebastian D. Bersch, Djamel Azzi, Rinat Khusainov, Ifeyinwa E. Achumba, and Jana Ries. 2014. Sensor Data Acquisition and Processing Parameters for Human Activity Classification. (2014). http://www.mdpi.com/1424-8220/14/3/4239
[8] S. Boll. 1979. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing* (1979).
[9] C. V. C. Bouten, K. T. M. Koekkoek, M. Verduin, R. Kodde, and J. D. Janssen. 1997. A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE Transactions on Biomedical Engineering* (1997).
[10] U.S. Census Bureau. 2017. *Average number of people per household in the United States from 1960 to 2016.* https://www.statista.com/statistics/183648/average-size-of-households-in-the-us/.
[11] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* (2011).
[12] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental Clustering and Dynamic Information Retrieval. In *ACM Symposium on Theory of Computing (STOC '97)*.
[13] Anna Choromanska and Claire Monteleoni. Online Clustering with Experts. In *Proceedings of the 2011 International Conference on On-line Trading of Exploration and Exploitation 2 - Volume 26 (OTEAE'11)*.
[14] R. R. Coifman and M. V. Wickerhauser. 1992. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory* 38, 2 (March 1992), 713–718. DOI:https://doi.org/10.1109/18.119732
[15] Giorgio Conte, Massimo De Marchi, Alessandro A. Nacci, Vincenzo Rana, and Donatella Sciuto. BlueSentinel: A First Approach Using iBeacon for an Energy Efficient Occupancy Detection System *(ACM BuildSys '14)*. ACM.
[16] Deloitte. 2015. *The Digital Predictions – The Deloitte Consumer Review.*
[17] Analog Devices. *ADXL335 Datasheet.* http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf.
[18] Ecolink. 2017. *Ecolink Z-Wave Garage Door Tilt Sensor.* http://discoverecolink.com/product/tiltzwave2-eco/.
[19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise *(KDD'96)*.
[20] Davrondzhon Gafurov and Einar Snekkenes. 2009. Gait Recognition Using Wearable Motion Recording Sensors. *EURASIP J. Adv. Signal Process* (2009).
[21] D. Gafurov, E. Snekkenes, and P. Bours. 2007. Gait Authentication and Identification Using Wearable Accelerometer Sensor. In *2007 IEEE Workshop on Automatic Identification Advanced Technologies*.
[22] Tobias Grosse-Puppendahl, Xavier Dellangnol, Christian Hatzfeld, Biying Fu, Mario Kupnik, Arjan Kuijper, Matthias R. Hastall, James Scott, and Marco Gruteser. Platypus: Indoor Localization and Identification Through Sensing of Electric Potential Changes in Human Bodies *(ACM MobiSys '16)*.
[23] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A Probabilistic Ontological Framework for the Recognition of Multilevel Human Activities *(ACM UbiComp '13)*.
[24] Timothy W. Hnat, Erin Griffiths, Ray Dawson, and Kamin Whitehouse. Doorjamb: Unobtrusive Room-level Tracking of People in Homes Using Doorway Sensors *(SenSys '12)*.
[25] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. 2002. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002).
[26] David J Ketchen and Christopher L Shook. 1996. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal* (1996).
[27] Nacer Khalil, Driss Benhaddou, Omprakash Gnawali, and Jaspal Subhlok. Non-intrusive Occupant Identification by Sensing Body Shape and Movement *(ACM BuildSys '16)*.
[28] P. Kodeswaran, R. Kokku, M. Mallick, and S. Sen. Demultiplexing activities of daily living in IoT enabled smarthomes. In *IEEE INFOCOM 2016*.
[29] Percy Liang and Dan Klein. Online EM for Unsupervised Models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '09)*.
[30] Qiguang Lin, Ea-Ee Jan, and James Flanagan. 1994. Microphone arrays and speaker identification. *IEEE Transactions on Speech and Audio Processing* (1994).
[31] Cao Gadgets LLC. 2017. *Wireless Sensor Tag System: Monitor Everything from the Internet.* http://www.wirelesstag.net.
[32] S. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* (1982).
[33] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. M. Makela, and H. A. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. In *ICASSP 2005*.
[34] Mathworks. 2017. *Entropy (Wavelet Package).* https://www.mathworks.com/help/wavelet/ref/wentropy.html.
[35] Samantha Murphy. 2015. *Samsung: By 2020, all of our products will be connected to the web.* http://mashable.com/2015/01/05/samsung-internet-of-things/#c3_K6GzxJ65N.
[36] Shijia Pan, Ningning Wang, Yuqiu Qian, Irem Velibeyoglu, Hae Young Noh, and Pei Zhang. Indoor Person Identification Through Footstep Induced Structural Vibration *(ACM HotMobile '15)*.
[37] Shijia Pan, Tong Yu, Mostafa Mirshekari, Jonathon Fagert, Amelie Bonde, Ole J. Mengshoel, Hae Young Noh, and Pei Zhang. 2017. FootprintID: Indoor Pedestrian Identification through Ambient Structural Vibration Sensing. *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies* (2017).
[38] William M. Rand. 1071. Objective Criteria for the Evaluation of Clustering Methods. *J. Amer. Statist. Assoc.* (1071). DOI:https://doi.org/10.2307/2284239
[39] Juhi Ranjan, Erin Griffiths, and Kamin Whitehouse. Discerning Electrical and Water Usage by Individuals in Homes *(ACM BuildSys '14)*.
[40] Daniele Riboni, Timo Sztyler, Gabriele Civitarese, and Heiner Stuckenschmidt. Unsupervised Recognition of Interleaved Activities of Daily Living Through Ontological and Probabilistic Reasoning *(ACM UbiComp '16)*.
[41] Scott T Stillman, Rawesak Tanawongsuwan, and Irfan A Essa. 1998. *A system for tracking and recognizing multiple people with multiple camera.* Technical Report. Georgia Institute of Technology.
[42] STMicroelectronics. *LPY503AL Datasheet.* http://www.st.com/web/en/resource/technical/document/datasheet/CD00237199.pdf.
[43] Yunze Zeng, Parth H. Pathak, and Prasant Mohapatra. WiWho: Wifi-based Person Identification in Smart Spaces *(ACM/IEEE IPSN '16)*.
[44] J. Zhang, B. Wei, W. Hu, and S. S. Kanhere. WiFi-ID: Human Identification Using WiFi Signal. In *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*.