Dataset explanation, implementation details, not there. Definition of Over and undercut a bit odd.

Demultiplexing Activities of Daily Living in IoT enabled Smarthomes[#]

Palanivel Kodeswaran*, Ravindranath Kokku*, Madhumita Mallick[$†], Sayandeep Sen*
*IBM Research, Bangalore [†]Indian Institute of Technology Kharagpur

***Abstract*—Powered by the emergence of the Internet of Things, smart homes containing a variety of sensors and actuators are expected to monitor and react to the activities of the residents with the goal of improving convenience, comfort and safety. However, in typical home settings, each human Activity of Daily Living (ADL) generates events from multiple sensors, and each sensor is triggered by multiple ADLs. Consequently, achieving high detection accuracy in these complex environments requires large amounts of training data for every possible multiplexing scenario, making it a complex problem. In this paper, we propose a data driven three-step de-multiplexing approach that simplifies the ADL recognition problem by first segmenting the event stream into periods of interest, before feeding to a classifier. We mine datasets to identify salient features which allow us to achieve a good segmentation. Extensive evaluation on ten public datasets shows that our approach achieves upto 77% segmentation accuracy, and a activity detection accuracy within 91% of the best possible.**

I. INTRODUCTION

Emerging *Internet of Things*(IoT) based smart home technologies [3], [4], with their purported ability to "'infer"' human requirements, are expected to comprehensively assist in addressing the problem of elderly care by significantly automating the important task of improved monitoring of the elderly [1]. Specifically, smart homes are expected to infer the wellness of the resident with the goal of alerting care providers/family in case of medical emergency, as well as providing early warnings of the onset of mental and physical degeneration [9].

Prior work [19], has reported that the completion of *Activities of Daily Life (ADL)* such as cooking, eating, daily hygiene are good indicators of the wellness of a person. For instance, caregivers of patients with Alzheimer's disease rank tracking and identifying activities of daily living of the patient at top of their list of needs for health assistance [17]. Similarly, medical researchers have also highlighted the importance of analyzing the *characteristics* in which certain activities are undertaken to determine the general wellness of a person. E.g., [8] has shown that, for the people with mild cognitive impairment(MCI), walking speed varies in the different stages of their disease, thereby making walking speed a good tracker of disease progression.

Fig. 1, depicts a typical smart home with multiple simple networked sensors to monitor human *ADL* . For instance, the kitchen has sensors to monitor the opening and closing of microwave, fridge and multiple cabinets. However, *ADL* s often generate complex sensor event patterns involving multiple sensors; i.e. an activity is characterized by a combination of heterogeneous sensors triggering over the activity duration.

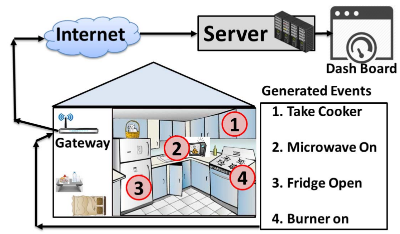


Fig. 1. Typical Smart home with sensors present to detect microwave, fridge and cabinet usage in kitchen. An gateway to aggregate sensor events and send to a server to analyze the data and finally depicts it on a dashboard to visualize the condition of the resident.

For instance, consider a remote family member (or care provider) who wants to ensure that an (elderly or post-operative) resident residing in the smart home depicted in Fig. 1 is cooking and eating regularly. Now, cooking by the resident may involve various sequences of events such as opening a gas burner, taking vegetables from a refrigerator, using the microwave for a sub-task, etc. This explains why activity detection approaches must learn combinations of sensor patterns to identify activity [13], [14]. Further, the same activity by a person on different days can have significant variations in time and triggered sensor patterns (might not use microwave always). Additionally, sensor events might actually be arising from by temporally overlapping activities either by a single resident or multiple residents.

Evidently, the above variabilities make detection of activities a computationally expensive and data intensive problem as a significant number of instances various patterns will need to be seen to learn and identify all the possible variations of activities (from a single or multiple resident) that can temporally overlap. The problem is exacerbated by the fact that myriad choice of available *IoT* devices from various vendors (with potential differences in features), would make smart home deployments different from one another.

In this work, we propose a segmentation technique that tries to address the above, by grouping events originating from a single activity into a single transaction, which can then be classified by a detection technique. Our core idea is similar to segmenting a multi-object image (and audio, video) into individual segments to simplify and improve image recognition [20]. The key contribution of this work lies in

Authors are listed in alphabetical order.
$ Work done as part of summer internship in IBM Research India.
[1]The argument is equally applicable in the case of automating post-operative care

identifying the right combination of discriminative features of human activities, which are manifested (and hence can be exploited) in the heterogeneous sensor event streams. The features are, *a) human activities last for a fixed duration*, b) *not all sensors that trigger together can be used to do a meaningful activity*, and, c) *the frequency characteristics of sensor events (from a single activity) change when received in conjunction sensor events from other activities*. We then design an efficient technique for event segmentation (and by extension activity recognition) by exploiting the insights.

Our benchmark results on ten smart home data sets The designed techniques lead to detection performance within 91% of the optimal. Apart from improving *ADL* detection accuracy, the segmentation technique also helps in enhancing the quality of analysis of activity characteristics for clinical tests, by removing spurious sensor events corresponding to other temporally overlapping activities. Experiments on data sets, show that our approach retains the frequency characteristics of the detected activities, with 80% of activities having mean frequency being less than two standard deviation.

To the best of our knowledge, this is the first effort to address the de-multiplexing problem in complex smart home environments with multiple overlapping activities of daily living. We make the following contributions: inhabitant's lifestyle by providing home

1. We highlight an interesting problem of de-multiplexing overlapping activities in IoT-enabled smart homes based on sensor triggering. The problem will be increasingly of interest, due to the rapid proliferation of a variety of sensing devices used in homes for elderly and patient care, wellness monitoring, safety, etc. **2.** We extensively mine datasets, to determine three highly discriminative features of sensor events generated from human activity, which lead to good segmentation of events. We then propose a data driven iterative approach for segmenting sensor event streams, using different criteria of separation of the events at each step, and demonstrate that the problem becomes more tractable by dividing and conquering different properties of human behavior (and hence the sensor patterns).

Finally, we note that while we focus remote health monitoring in this paper, our enhanced event segmentation coupled with activity recognition, are expected to improve functionality and performance of other emerging *IoT* enabled smart home applications such as home security, safety, user context modeling, energy efficiency, etc. as well.

The rest of the paper is organized as follows. In § II we explain the difficulties of event segmentation and motivate the challenges that we have to overcome in our approach for clearly identifying boundaries. In § III, we describe our approach to address the problem. In § IV, we present results based on benchmarking. We explain prior work in § V and then conclude in § VI.

II. Motivation

Our goal is to identify human activities of daily living (ADLs) in a smart home using a variety of sensors. In this section, we first explore the characteristics of such environments that make *ADL* detection challenging by studying various smart home datasets.

Label	**Name**	**# User**	**Dur. (days)**	**# ADL (# Activity)**	**# Sensor (# Event)**
kA	KasterenA [1]	1	25	16 (283)	14 (2006)
kB	KasterenB [1]	1	15	25 (172)	27 (22595)
kC	KasterenC [1]	1	19	27 (254)	23 (39861)
Cr	Cairo [5]	1	56	10 (600)	27 (159344)
Ar	Aruba [5]	1	219	11 (6477)	39 (805268)
Ml	Milan [5]	1	82	15 (2310)	33 (288498)
T1	Twor9-10 [5]	2	249	25 (3745)	100 (711421)
T3	Twor2009 [5]	2	57	14 (499)	100 (136504)
T2	TworSmr [5]	2	63	8 (1016)	100 (366075)
Mul	AdlNorm [5]	24	84	5 (120)	39 (6425)

TABLE I
Dataset table for Activity Recognition

A key characteristic of IoT enabled environments is that multiple applications can be run using sensors acquired from diverse vendors, these sensors can work in a distributed manner, collecting and transferring data to a gateway without knowledge of other sensors, and these sensors can be deployed organically by users without being tied to a specific application. In such a setting, consider a typical *IoT* enabled smart home, as shown in Fig. 1, instrumented with a variety of sensors. When a resident undertakes an *activity*, which is one of many ADLs such as cooking eating, bathing, etc, the activity manifests in a set of sensors generating *events*. These events are collected centrally by an aggregation entity (either within the home or in the cloud) for further analysis and actions. Several challenges arise when analyzing and interpreting the events collected, and detecting the actual activities undertaken from the sensor events. To study these challenges, we analyze a number of smart home datasets as shown in Tab. I. The table provides relevant details of the 10 datasets analyzed in this work, some of which are collected from single-resident homes, while others are from multiple-resident homes. The homes are equipped with a combination of motion/light sensors on the ceilings, temperature sensors as well as contact sensors on cabinets and doors [2]. The datasets are labeled and contain manually annotated entries specifying the IDs of sensors that triggered along with the corresponding time stamp and ADL label.

For de-multiplexing sets of events into transactions, the foremost difficulty arises from the fact that there exists no unique and distinguishing signatures of sensor usage for any *ADL* . We note that for a majority of the ADLs, multiple sensors are triggered by their occurrences. For instance, Fig. 2(a) shows the CDF of the number of distinct sensors that are triggered when a specific *ADL* is undertaken in all datasets. The figure shows that more than 96% of *ADL* across all datasets lead to

[2] We refer the reader to the cited literature for further details of the instrumentation and data collection setup.

Is this similar to NER, where IOB trained data is fed and HMM or CRF is used events from more than one sensor[3].

Secondly, a majority of sensors will generate events for more than one *ADL* . E.g., a refrigerator door may be opened before cooking, or every time a person would like to take a drink, or when picking a snack or a medicine. We study this phenomenon by plotting the CDF of number of *ADL* that each sensor participates in across all datasets in Fig. 2(b). As can be seen from the plot, more than 98% of the sensors are triggered for more than one *ADL* . The implication of the above observation is that no one sensor can deterministically imply the occurrence of a specific activity. Furthermore, there is a lot of variability in sensor event generation between different instances of the same *ADL* . We show this in Fig. 2(c), which plots the CDF of sensor event occurrence to activity occurrence ratio for every activity-sensor pair across all datasets. The graph shows that only less than 10% of the sensors *always* trigger for all instances of an *ADL* across all the datasets; i.e. in 90% of the activity instances, some sensors trigger only for a subset of the times an *ADL* actually happened.

Finally, events often get multiplexed from two or more activities due to (a) one user carrying out multiple activities simultaneously, or (b) multiple users undertaking activities concurrently. While interleaving of events is obvious, multiplexing has the secondary effects of significantly altering the activity signature compared to when the activity is performed in isolation. To highlight the effects, we look at the metric of distortion in mean frequency for an activity. Tab. II, reports the mean frequency of a single resident's activities across multiple datasets, when performed in isolation as well as when multiplexed with some other activity. E.g., the frequency of Relax activity in *Aruba* changes from 0.089Hz to 1.04 Hz a difference of 12×. Similarly, in Fig. 4 we depict the change in frequency for one activity when another resident is concurrently performing another activity. For e.g., the frequency of R1's bath activities in isolation in T1 dataset is 0.07 Hz (T1-R1). However when R2 is conducting any activity concurrently at the same time, the frequency of R1's bath (in T1 dataset) changes to 0.54 Hz (T1-R1+any), a jump of 8×. Also, the standard error of event frequency changes significantly when multiple activities are multiplexed.

III. Approach

We describe our approach to leverage three key insights: a) human activities last for a fixed duration, b) not all sensors that trigger together can be used to do a meaningful activity and, c) the frequency characteristics of sensor events (from a single activity) change when received in conjunction sensor events from other activities, for demultiplexing sensor events.

A. Definitions

Let the set of events $E^i = \{e_1, e_2, \ldots\}, E^i \subset E$ be generated by activity A_i. When collected together in a smart home,

[3]This phenomenon of heterogeneity in sensors distinguishes our problem from similar segmentation problems in the domain of image, audio and video processing, and necessitates the development of new techniques for segmentation in smart home settings.

Dataset	**Activities**	**Proper** **Mean** **(Std. Err.)**	**Interleaved** **Mean** **(Std. Err.)**
KasterenA	Receive Guest	0.054(0.052)	0.342(0.337)
KasterenB	Go To Bed	0.0324(0.009)	0.0193(0.003)
KasterenC	Prepare Dinner	0.026(0.003)	0.009(0)
	Take Shower	0.341(0.063)	0.131(0)
Cairo	Work	0.049(0.005)	0.262(0.045)
	Breakfast	0.413(0.019)	0.399(0)
Aruba	Relax	0.089(0.001)	1.04(0.25)
	Leave Home	0.83(0.015)	276.39(69.44)
Milan	Bed To Toilet	0.127(0.008)	0.011(0.005)
	Sleep	0.024(0.003)	0.009(0.0007)

TABLE II

Effect of multi-tasking on event frequency pattern of activities for *KasterenB*

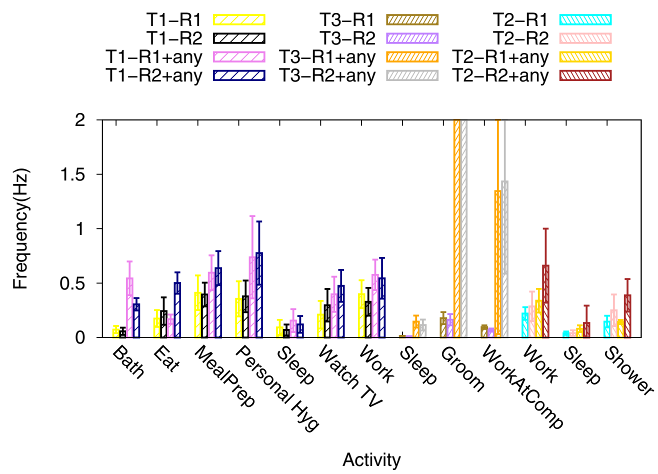
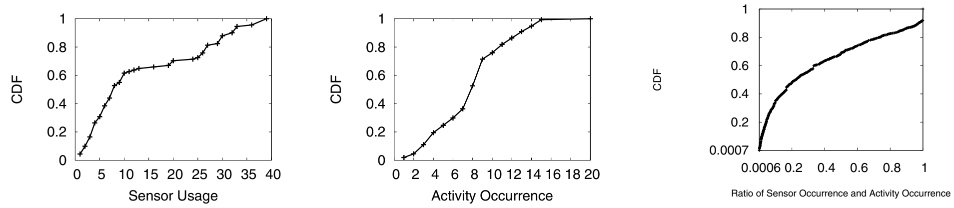


Fig. 4. Average event frequency in activity of a single user (R1 or R2) and interleaved with another user's activity (R1-any, R2-any) in T1, T2, T3 testbeds Tab. I

as demonstrated earlier, the sensor events for various activities can get interleaved due to many reasons. For explaining our iterative de-multiplexing approach, we define the notion of a *transaction*, which is a set of events clustered together using a specific criteria at each step of the iterative process.

At each step, we segment the event stream E into a set of transactions $Tr = \{tr_i\}$, where $tr_i = \{e_1, e_2, \ldots\}, e_i \in E$ based on that step's particular criterion as shown in Figure 5. We note that since segmentation is not perfect at each step, the resulting transactions at each intermediate step do not necessarily match any individual ADL, and can be of different types: We call a transaction tr_i a *ProperCut* , if the event set of the tr_i exactly matches that of the corresponding activity A_i. Similarly, if the event set of a tr does not contain all the events of the corresponding activity, we call it an *OverCut* transaction. In other words, the activity is split into multiple *OverCut* transactions. Finally, if the event set of a tr contains events that are from multiple different activities, we call it an *UnderCut* . Figure 5**a** pictorially shows the various types of cuts induced by segmentation.

The goal of each step that we describe next is to take the input event streams, and generate transactions so as to minimize the number of *OverCut* and *UnderCut* , and maximize the number of *ProperCut* . Specifically, the steps include an inter-arrival



(a) CDF of sensors count which generate events for an ADL.
(b) CDF of ADL count for which a sensor generates events.
(c) CDF of sensor occurrence to activity occurrence ratio.

Fig. 2. Characteristics of events generated in smart homes which present challenge for event segmentation technique. The plots are generated with events from all datasets listed in I.

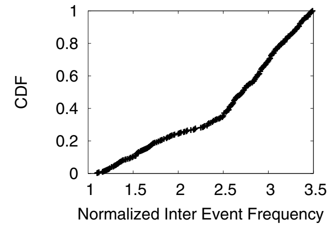


Fig. 3. Event pair arrival frequency from two different activities normalize to event pair caused by single activity, across all event pairs over all activities in all data sets. Average is $\sim 2.5\times$

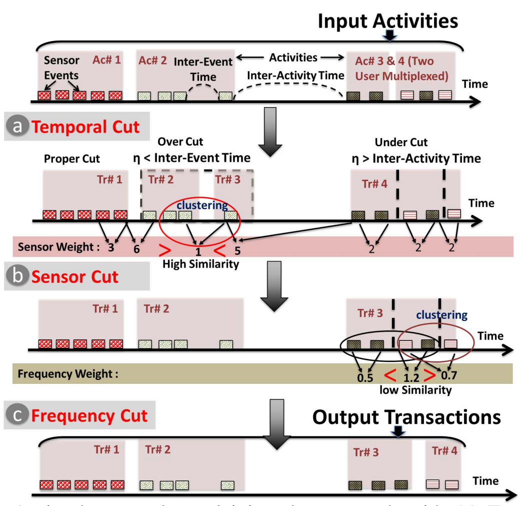


Fig. 5. A simple example explaining the approach with (a) Temporal (b) Sensor and (c) frequency cut

time based cutting of event streams, sensory-distance based , and finally a frequency-distance based cutting as shown in Figure 5. We next describe the various criteria used and how they are learnt from the datasets.

B. Inter-arrival based cutting of events

The goal of this step is to separate temporally distinct activities, based on sensor event inter-arrival times. For instance, events from morning activities will be mostly separated by time from afternoon activities and evening activities.

Temporal-cut criterion. Intuitively, for each *ADL* there is a maximum time interval within which an activity can begin and end. E.g., in *KasterenB* dataset 'Get A Drink' activity lasts 1 minute where as 'Prepare Brunch' lasts between 15 to 17 minutes. Therefore, to segment co-related events into a transaction, our technique needs to only focus on a *reasonable* time window. Again, our optimization goal is to maximize *ProperCut* , while minimizing both *OverCut* and *UnderCut* . Note that multiplexed events cannot be properly cut by temporal-cut criterion alone, as events of multiplexed activities are intermingled in temporal domain, and need further processing.

For applying the temporal cut criterion, we need to learn a temporal cut threshold η, that is then used in the online segmentation phase to create transactions when the inter-event arrival time is larger than η. For each activity, we first compute the inter-event time intervals within an activity as well as inter-activity time intervals as shown in Figure 5. We plot the cdf of the measurements for various datasets in Figure 6. Observe that there is significant difference between the two distributions, with inter-activity time intervals being significantly higher than inter-event time intervals for most activities. E.g., in the base case for the *Cairo* dataset, only 1% of events have an inter-event arrival of 60 seconds and higher, whereas 93% of activities have inter-activity time intervals greater than 60 seconds. Similarly, in the average case *KasterenA* dataset, 88% of events have an inter-event arrival time less than 120 seconds, while only 22% of the activities have an inter-activity time interval less than 120 seconds. We leverage the above *key insight* in computing the best η that maximises *ProperCut* for the training data.

Our learning algorithm uses historical data to learn the best η (in terms of seconds of Day) that maximizes the goal while satisfying the constraint $0 < \eta < 86400$ (maximum activity duration, assuming that all activities necessarily repeat at day-boundaries). We note that for a given non-negative η, activity A_i is properly cut only if the following two constraints are satisfied

$$\eta \geq \tau^i \tag{1}$$

where, $\tau^i = max(\chi)$, $\chi = max(\tau^i_{k-1,k}, \tau^i_{k,k+1})$,$\forall k \in E_i$
where E_i is the set of all events of activity A_i , $\tau^i_{k,j}$ is the inter-event arrival time between k^{th} and j^{th} event of activity, A_i. And,

$$\eta \ \leq \ T^i \tag{2}$$

where, $$T^i = min(T^{i-1,i}, T^{i,i+1})$$

where $T^{i,k}$ is the inter-activity time interval between A_i^{th} and A_k^{th} activity.

The first constraint ensures that the choice of η does not over cut A_i and the second criteria ensures that A_i is not undercut with A_k. Of course, the above two criteria cannot be satisfied for a) activities A_i for which, $\tau^i > T^i$, in other words, the activities that have inter-event time intervals greater than the inter-activity intervals from neighboring activities, and b) interleaved activities as T^i in Eq. 2 will be negative since the start time of A_{i+1} will be lesser than the end time of A_i.
In our algorithm, we first calculate T^i and τ^i $\forall i \in A$, where A is the set of all training set activities. We then eliminate all activities and their corresponding events for which Eq.

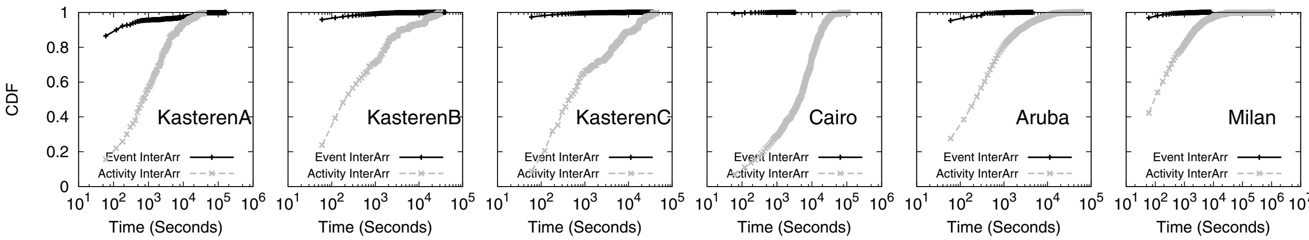


Fig. 6. The CDF of inter event and inter activity arrival times for six data sets. The figure shows a significant difference between the two inter-arrival distributions.

1 and 2 above do not hold, as these activities can never be properly cut. We then take the set of all remaining activities and search for the best value of η that satisfies our goal. Additionally, the search space of the optimization is reduced from real valued domain $\mathbb{R}$ to the discrete set of values $t \in \{T^i, \tau^i\}, i \in A$, by leveraging the fact that only these values affect the optimization metric of *ProperCut* , *OverCut* and *UnderCut* .

C. Sensory distance based cutting of events

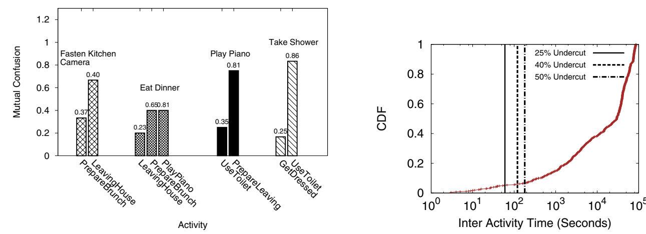
Sensor-cut criterion The goal of the sensor-cut criterion is to determine if sensors reporting events are triggered together by the same underlying activity. (In the rest of the section, we use the term *mistaking* to imply that events from different activities are wrongly assigned to the same transaction). Note that the probability of mistaking events coming from a neighboring activity are exponentially higher compared to mistaking events that come from activities that are far apart in time. Consequently, the strategy of delineating transactions will be successful, if a significant portion of neighboring activities *do not share* the same set of sensors i.e. if E_i be the set of sensors in A_i and E_j be the set of sensors in A_j, then the probability of mistaking an event involved in a different activity will be lesser as $(E_i \cap E_j)$ becomes close to ϕ. To ascertain the correctness of the above assertion, we calculate the mutual confusion between neighboring *KasterenB* activities [4]. We characterize the similarity in sensor sets between a pair of ADLs by calculating their *Cosine similarity*. For this, we represent an ADL_i as a vector $C_i = < s_1, s_2 \ldots, s_n >$ where $s_i = 1$ if sensor s_i frequently participates in ADL_i. The *Cosine similarity* between a pair of ADLs, ADL_i and ADL_j is defined as $\frac{C_i \cdot C_j}{\|C_i\|\|C_j\|}$. The mutual confusion between ADL_i and ADL_j is defined as the ratio of ADL_i activities that get classified as ADL_j. In figure 7(a) we see that as cosine similarity between a pair of ADLs increases, the mutual confusion also increases. We also plot the CDF of inter-arrival time between activities whose cosine similarity is high in Figure 7(b). We choose a threshold cosine similarity of 0.5 to indicate high similarity. From the plot, we observe that less than 6% of the activities that have high cosine similarity will be merged together, even for the most naive temporal cutter.

We next define our sensor based distance metric that we use for segmenting the event stream. For all event pairs $< e_i, e_j > \in A_i, \forall A_i \in A$, where A_i is a labeled activity and A is the set of all labeled activities, we calculate **Sensory co-occurrence distance** (W^{ij}_{sensor}) as the complement of the probability of co-occurrence of the sensor pair as part of any activity in the training data, multiplied by a empirically selected weight α.

$$W^{ij}_{sensor} = \alpha * 1 - [\frac{|< e_i, e_j >|}{|< e_i, e_k > + < e_j, e_k >|}], \quad (3)$$

where $i \neq j, k$ and $|< e_i, e_j >|$ is the number of occurrences of$< e_i, e_j >$ in an activity. We observe that, W^{ij}_{sensor} is inversely proportional to the frequency with which the event pairs $< e_i, e_j >$ co-occur within activities.

[4]The observations hold for other datasets and are omitted for the sake of brevity



(a) Cosine Similarity and Mutual Confusion between pairs of ADLs

(b) Inter arrival time between activities with high cosine similarity

Fig. 7. 7(a) Mutual confusion between ADL pairs in *KasterenB* testbed. This figure shows mutual confusion between pairs of ADLs in *KasterenB* dataset. In the x-axis, we show ADLs that confuse with the ADL labeled at the top. The figure shows that for high cosine similarity values(displayed on top of the bars) the mutual confusion between ADLs is also high(varies from 20% to 60%) 7(b) The CDF of inter-activity arrivals between activities whose cosine similarity > 0.5 in *KasterenB* testbed. The vertical line shows increasing η values.

D. Frequency distance based cutting of events

Frequency-cut criterion We next leverage the observation that when activities that use a common set of sensors get multiplexed together within the same transaction, the frequency between sensor triggerings will be different for different activities and this distinguishing feature can be used to demultiplex them into their component activities. In other words, assume that events e_i and e_j occur for both activities A_k and A_l. And, $f^k_{ij} = \frac{1}{t_i - t_j}, < e_i, e_j > \in A_k$[5] is the frequency of occurrence between e_i and e_j when occurring within activity A_k. Now, we assert that $f^k_{ij} \neq f^l_{ij} \neq f^{k,l}_{ij}$,where, $f^{k,l}_{ij}$ is the frequency of occurrence between $< e_i, e_j >$, when $e_i \in A_k$ and $e_j \in A_l$ (or vice versa).

To validate the above, we plot the CDF of multiplexed frequency, $f^{k,l}_{ij}$ normalized by f^k_{ij} across all data-sets. As seen in Figure II, for 50% of such multiplexed pairings, the

[5]Not that, $f^k_{ij} = f^k_{ji}$

frequency is higher by $2.5\times$ compared to the normal. This is expected as sensors are most of the time triggered at a particular phase while undertaking an activity. E.g., when eating inside the kitchen, a person will typically open and close the kitchen door only at the beginning and end of the activity. Whereas, if another person enters and drinks water, the same kitchen door sensor will trigger, but the time interval between the door open event due to *eating* activity will not have any correlation with the door close after the shorter *drinking water* has finished.

We now describe our frequency based distance metric for disambiguating activities. For this purpose, we calculate **Frequency Distance** ($W_{frequency}^{ij}$) as

$$W_{frequency}^{ij} = \beta * [1 - \frac{Count(f_{ij} - \frac{\sigma_{ij}}{2}, f_{ij} + \frac{\sigma_{ij}}{2})}{|F_{ij}|}] \quad (4)$$

where f_{ij} is defined above and $Count(f_a, f_b)$ is the count of f_l $\forall f_l \in F_{ij}$ such that $f_a < f_l < f_b$ and F_{ij} is the set of all f_{ij}'s. We define σ_{ij} as the standard deviation of the set of F_{ij} and β is an empirically determined constant. We empirically set α to 10 and β to 100 in our system.

E. Online Segmentation Phase

Figure 5 shows the overall flow of our approach and how the various criteria learnt in the previous section are used during online segmentation. Given an input event stream E, we first apply temporal cut to segment events into the transaction set T, when the inter-event arrival time exceeds the temporal cut threshold η, as shown in Figure 5**a**. We then apply sensor-cut on each transaction $T_i \in T$ as shown in Figure 5**b** by estimating the sensor distance W_{sensor}^{ij} for each sensor pair $< e_i, e_j > \in T_i$. We then perform hierarchical agglomerative clustering [2] on the events in T_i using W_{sensor}^{ij} as the distance metric to get a new set of transactions T'. Finally, we apply frequeny-cut as shown in Figure 5**c** by clustering the events in each transaction in T' using $W_{frequency}^{ij}$ as the distance metric. We then output the final set of transactions T'' which represents the segmentation of the input stream E into different transactions.

IV. Result

In this section, we present the experimental methodology and the evaluation of various components of our approach.

A. Experimental Methodology

For our experiments, we split each dataset, listed in Tab. I into 80% learning (training) and 20% test data. Unless otherwise stated each reported result is an average of 5 runs.

We now define several algorithms for comparison and a set of classification algorithms that take the output of our work for classifying ADLs.

Temporal Cut Criteria: We benchmark our *MinMax* temporal cutting strategy, against the following other possible temporal strategies.

MeanCutter : In this strategy, we set Temporal-cut threshold (η) in sec. III-B as $\eta = \mu + \sigma$, where μ and σ are the mean and standard deviation of inter-activity arrival time across all the activities in the training data; the strategy assumes that an inter-event gap greater than $\mu + \sigma$ implies end of activity.

MedianCutter : Another strategy where η = median + σ.

PercentageCutter : In this strategy, to calculate η, we determine the cumulative distribution of inter-activity as well as inter-event intervals. As shown in Fig. 6, this allows us to pick a value of η and obtain an estimate of number of transactions that will be either *UnderCut* or *OverCut* . We then select a value of η that minimizes the percentage of *OverCut* for a fixed percentage of *UnderCut* (empirically determined to be 20%) in our experiments [6].

Oracle Cutter: The best possible cutting strategy that segments events into transactions by reading the activity labels. Note that the cutter is a complete segmentation technique on its own and does not need any subsequent steps. Thus, the *Oracle cutter* allows benchmarking to the best possible segmentation.

Disambiguation Criteria: We use the following demultiplexing criteria for disambiguating undercut transactions.

Combined-cut criteria: We utilize both frequency and sensor-cut defined in Section III. **Sensor-cut only**: Only sensor-cut criteria is used. **Frequency-cut only**: Only frequency cut criteria is used. For both the methods use hierarchical clustering method from R [2].

ADL Classification Algorithms: Once segmentation is done using any of the approaches, ADL classification algorithms use the output of segmentation for accurate classification. To show effectiveness of our approach, we benchmark ADL detection accuracy by the following algorithms using a variety of segmentation approaches above.

Frequent Item set Mining (FIM) based Detector: Akin to the detector described in [16], we use association rule mining on the training data to 'find most common' sets of generated events for an *ADL* . We then classify the test event pattern as the *ADL* whose event set signature it matches best.

Naive Bayes (NBC) Detector: A NBC detector, in training phase, uses standard Bayes theorem for determining the probability of occurrence of an *ADL* when a specific set of events, along with time of occurrence etc. is seen. In test phase, it predicts *ADL* with highest match to have occurred. We use an open source implementation of NBC [16] in our experiments.

Hidden Markov Model (HMM) Detector: The HMM detector learns and runs a hidden markov model for classifying transactions into ADLs. In our setting, the ADLs represent the hidden states, while the sensor events represent the observables of the HMM model. We use an open source implementation of HMM [16] in our experiments.

B. Overall Performance

In this section, we characterize the performance of our end-to-end scheme in improving *ProperCut* , detection accuracy as well as retaining frequency characteristics of activities.

Improvement in Segmentation: We define *ProperCut* ratio as the fraction of total activities which were properly cut.

[6]Fixing the percentage of *OverCut* and minimizing the percentage of *UnderCut* , performs worse, and is omitted for sake of brevity.

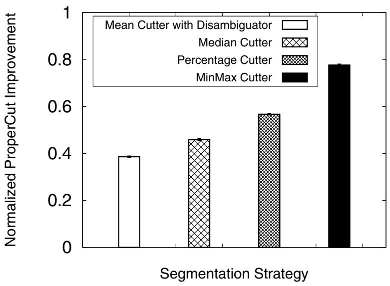


Fig. 8. The *Minmax* segmentation strategy outperforms other strategies in yielding proper cut transactions with an average normalized improvement of 77 % over mean cutter

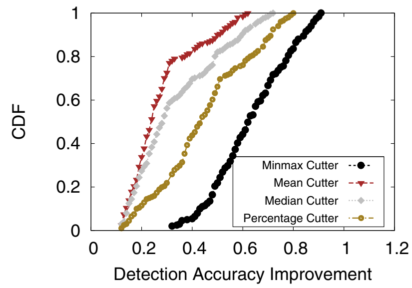


Fig. 9. The *Minmax* strategy significantly outperforms other strategies with respect to ADL detection accuracy and has only a 9% misclassification rate compared to oracle cutter's detection accuracy

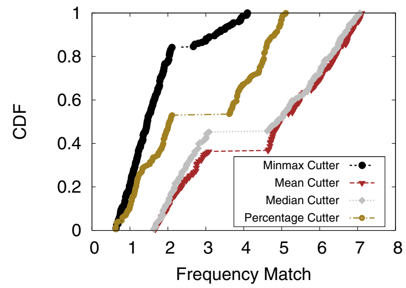


Fig. 10. *Minmax* strategy is able to accurately retain sensor frequency characteristics with frequency deviations being less than two standard deviations away for 80% of the times

Detector Detector	Mean Cutter	Median Cutter	Percentage Cutter	MinMax Cutter
NBC	32.1 (.12)	37.3 (.41)	57.1 (.8)	74.6 (.21)
HMM	34.2 (.11)	39.4 (.96)	62.4 (.7)	85.5 (.31)

TABLE III

Average (Std. Error) Detection accuracy of segmentation techniques

We compare the propercut ratio obtained by using different temporal cutting strategies followed by running the combined sensory-frequency disambiguator on undercut transactions. We normalize the results against the propercut ratio returned by mean-cutter without disambiguator and show the results in Fig. 8. The *Minmax* segmentation provides the best propercut ratio with an average improvement of 77% over the mean cutter. On the other hand, the percentage and median segmentation strategies provide average improvements of only 56% and 46% respectively. Our *Minmax* approach performs very well since its temporal criteria is able to jointly optimize on both the maximum inter-event and minimum inter-activity time intervals to optimally choose the right cutting interval. Furthermore, a high proportion of the undercut activities are able to be disambiguated correctly by the combined sensory-frequency disambiguator. In summary, the cumulative advantages of optimal temporal cut followed by a high accuracy disambiguator enables *Minmax* to outperform other strategies by up to 68%.

Improvement in Activity Detection Accuracy: Next, we evaluate the improvement in ADL detection accuracy obtained by our approach. We compare the performance of our scheme against the detection accuracy obtained by running a FIM based detector on transactions returned by the oracle cutter. Recall that the oracle cutter is a supervised cutter with zero segmentation error, and therefore the resulting detection accuracy is a function of only the classification algorithm. Fig. 9 shows the CDF of normalized detection accuracy of the different segmentation strategies over 30 iterations. Our *Minmax* approach performs significantly better compared to the other strategies yielding a normalized detection accuracy of 60% at the median, compared to 20-40% accuracy for other strategies. Significantly, the *Minmax* detection accuracy goes up to 91% of the oracle cutter's, which illustrates the very

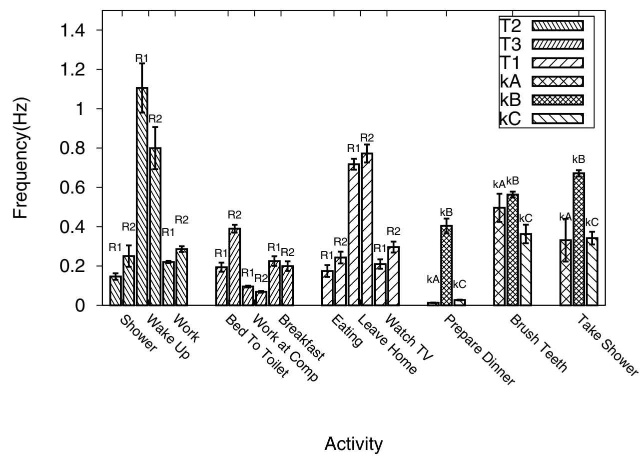
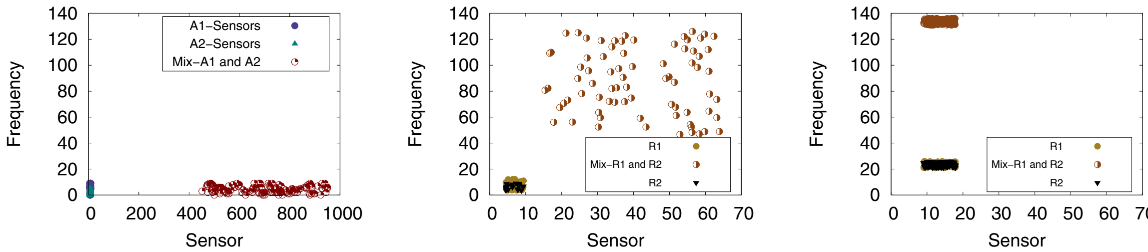


Fig. 11. Human subject have different frequencies for performing the same activity

high precision with which our *Minmax* approach is able to cut events into transactions for classification by the detector. Similar observations can be made for other classifiers as depicted in Table III.

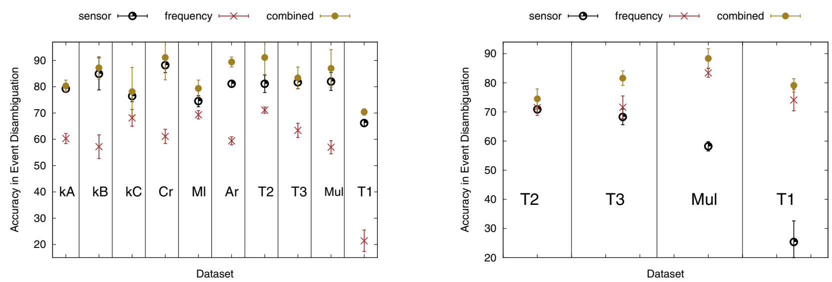
Retaining Frequency level Characteristics: A number of applications of ADLs utilize information on the frequency of occurrence of events within an activity. E.g., the frequency at which a resident who recently had an arm surgery cooks would be markedly different before and after the surgery. Furthermore, different residents have different characteristic frequencies for performing the same activity as shown in Fig. 11.

We compare the ability of various segmentation strategies to retain such low level sensor frequency information of the activities by estimating the event arrival frequency inside detected activities and comparing them against the corresponding ADL baseline frequency. In Fig. 10, we plot the CDF of event frequency deviation for each detected activity, normalized by the standard deviation of the corresponding ADL. From the plot, the *Minmax* segmentation strategy significantly outperforms other strategies. In the case of *Minmax* , the frequency deviation is within two standard deviations for 80% of the times, while the deviation is four standard deviations in the



(a) Same resident Interleaved (b) Two resident Interleaved 1 (c) Two resident Interleaved 2

Fig. 12. The Sensor and frequency distance metrics are able to disambiguate events that belong to single resident interleaved activities (subFig. 12(a)) as well as multi resident interleaved activities(subFig. 12(b)). SubFig. 12(c) shows the special case where multiple residents are in involved in concurrent activities that use the same set of sensors such as *watching tv* and *wandering in room* in the T2 dataset. We see that while the sensory distance is low, the temporal distance is very high, allowing the system to disambiguate between the two activities.



(a) Disambiguation Accuracy of Single resident interleaved activities

(b) Disambiguation Accuracy of Multi-resident interleaved activities

Fig. 13. For single resident interleaved activities, (a) The sensor-cut criteria has high accuracy. (b) For multi-resident concurrent activities, the frequency-cut criteria provides better disambiguation performance as the event arrival frequency induced by multi-residents is different from that induced by single resident. The combined performance is at least as good as sensor, frequency-cut criteria.

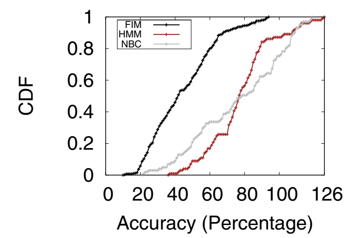


Fig. 14. Normalized detection accuracy improvement. The sensor and frequency-cut criteria improve detection accuracy across all classifiers up to a maximum of 126% for HMM.

worst case. In comparison, only 10% of the cases are less than two standard deviations for mean and median strategies. The ability of *Minmax* to accurately retain frequency characteristics of activities is due to its ability to generate significantly high propercuts and high detection accuracy.

C. Sensitivity Analysis

Performance of disambiguation approaches As an evaluation metric, we consider the number of events in an undercut transaction that are properly assigned to their corresponding activities.

We first evaluate the discriminating power of our sensory and frequency metrics for disambiguating undercut transactions. Fig. 12 shows the distance in the sensor and frequency axes for interleaved and non-interleaved activities in both single and multi resident homes. We can clearly see that the distribution of distances along both the axes are different for interleaved and non-interleaved activities. Particularly, single-resident interleaved activities have high sensor distances as there is not much deviation in the frequency pattern since there is only one resident involved in multiple activities, and frequency is constrained by the resident's characteristic speed. On the other hand, for multi-resident interleaved activities, the distances on both the sensor and frequency axes are high due to the interference caused by an additional set of sensors as well as mixing of frequencies between both residents. Similarly, even if the residents use similar sets of sensors concurrently, the combined frequency is very different from each resident's individual frequency resulting in a high frequency distance as shown in Fig. 12(c). This validates the utility of our sensor and frequency metrics in disambiguation of undercut transactions.

Next, we evaluate the performance of each of the disambiguator schemes viz. sensory, frequency and combined. For single-resident interleaved activities, the sensory based disambiguator provides high disambiguation accuracies of up to 90% for *Cairo* datasets as shown in subFig. 13(a) since there is not much deviation in the frequency axis given that there is only one resident. On the other hand, for multi-resident concurrent activities, the frequency based disambiguator provides better disambiguation performance yielding up to 88% accuracy in the case of *Mul* dataset, as shown in subFig. 13(b), as the event frequency induced by multiple residents is very different from a single resident's frequency. In all cases, the combined disambiguator performs at least as good as the sensor or frequency disambiguator, obtaining up to 90% disambiguation accuracy in both single and multi resident interleaved activities.

We also evaluate the effect of the combined disambiguation scheme on overall ADL detection accuracy. For this, we uniformly and randomly sample 400 undercut transactions from all the datasets, and pass them through three different classifiers viz. Naivebayes, HMM and our FIM based classifier. Fig. 14 shows the CDF of normalized detection accuracy improvement compared to just a temporal cutting strategy. We see that the disambiguator scheme improves detection accuracy for all the classifiers, with the improvements ranging between 40-

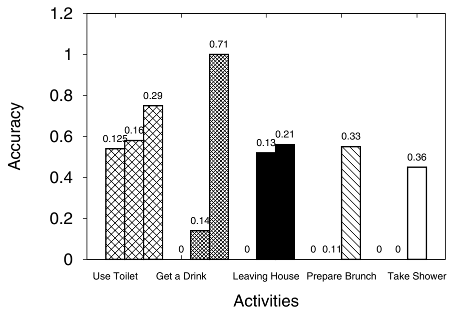


Fig. 15. Overall Detection for various *ADL* , (with *ProperCut* ratio values in legend). *ADL* detection accuracy improves with propercut ratio

80% at the median. Additionally, the disambiguation scheme is further able to provide accuracy improvements of up to 126% in the best case for the HMM classifier. This result signifies the crucial role played by the disambiguation scheme in improving overall detection accuracy of all classifiers.

Validating choice of *ProperCut* as the optimization metric We now characterize the effects of propercut ratio on other metrics, and establish the importance of propercut ratio as an important metric for optimizing our event segmentation strategies. Fig. 15 shows the detection accuracy as a function of proper cut ratio for different ADLs of *KasterenB* dataset. Observe that as the propercut ratio increases, the detection accuracy for the ADLs also increases, illustrating the significant role played by proper cuts in overall detection accuracy.

Effect of Seasonality Human subjects perform activities at different frequencies at various times of the day. We term this effect, *seasonality*. We found that there does exist seasonality in human activities. However, incorporating seasonality into segmentation strategies, reduced along all metrics due to the confusion caused by multiple ADLs co-occurring within the same time interval. We omit results for brevity.

V. Related Work

ADL Monitoring The work of Feuz et al [10] is closest to our work, in that they also detect transitions between activities. The authors propose using a supervised learning based approach as well as change point detection based unsupervised approach for detecting activity transitions. However their approaches do not consider inter-leaved activities and multi-resident settings.

There has been significant interest in the recent past in monitoring ADLs in the home. A number of machine learning based approaches viz. NaiveBayes, HMM, CRF [16], SVM [7], association rule mining [15] etc. have been employed for activity recognition. The authors in [6] provide a good survey of the significant body of work in activity recognition in homes. However, our work is complementary to the above approaches, and the output of our segmentation process can be fed to these approaches to improve their classification accuracy as shown in § IV.Furthermore, our approach uses only infrastructure sensors without the need for additional sensors like smartphones as in [18].

Event Segmentation in other Domains There has been significant related work on segmentation in other domains as well such as audio [20], image [11] and web browsing [12]. However, all these approaches are based on processing homogeneous sensory signals and do not work in smart home settings that involve multiple heterogeneous sensors.

VI. Conclusion

In this paper, we propose a novel iterative de-multiplexing approach leveraging key insights of human behavior, which manifest in sensor event stream. Our approach simplifies the identification of human activities of daily living in an IoT-enabled smart home. Our approach combines multiple techniques to significantly improve detection accuracy.

References

[1] Benchmark datasets; datasets for activity recognitions. https://sites.google.com/site/tim0306/datasets.

[2] Hierarchical clustering in r. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html.

[3] Smart home: How apple, google and samsung will take over your home. http://www.trustedreviews.com/opinions/home-automation-how-apple-google-and-samsung-will-take-over-your-home.

[4] Smart home systems and the code for sustainable homes. http://www.domo-energie.com/usr_file/Pdf/presse/ibexcellence_smart_home_systems.pdf.

[5] Wsu casas dataset. http://ailab.wsu.edu/casas/datasets/.

[6] G. e. a. Acampora. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12):2470–2494, 2013.

[7] C. Chen, B. Das, and D. Cook. A data mining framework for activity recognition in smart environments. In *Intelligent Environments (IE), 2010 Sixth International Conference on*, pages 80–83, July 2010.

[8] H. Dodge, N. Mattek, D. Austin, T. Hayes, and J. Kaye. In-home walking speeds and variability trajectories associated with mild cognitive impairment. *Neurology*, 2012.

[9] B. M. M. El-Basioni, S. M. A. El-Kader, and H. S. Eissa. Independent living for persons with disabilities and elderly people using smart home technology. *International Journal of Application or Innovation in Engineering & Management*, 2014.

[10] K. Feuz, D. Cook, C. Rosasco, K. Robertson, and M. Schmitter-Edgecombe. Automated detection of activity transitions for prompting. *Human-Machine Systems, IEEE Transactions*, 2014.

[11] J. Freixenet, X. Muoz, D. Raba, J. Mart, and X. Cuf. Yet another survey on image segmentation: Region and boundary information integration. In *In ECCV*, pages 408–422, 2002.

[12] D. Gayo-Avello. A survey on session detection methods in query logs and a proposal for future evaluation. *Inf. Sci.*, 179(12), 2009.

[13] T. V. Kasteren, A. Noulas, G. Englebienne, and B. Krose. Accurate activity recognition in a home setting. In *UbiComp*, 2008.

[14] L. Liao, D. Fox, and H. Kautz. Location-based activity recognition using relational markov networks. In *Proc. Int. Joint Conf. Artif. Intell.*, 2005.

[15] E. Nazerfard, P. Rashidi, and D. J. Cook. Using association rule mining to discover temporal relations of daily activities. In *Toward Useful Services for Elderly and People with Disabilities*. 2011.

[16] P. Rashidi, D. Cook, L. Holder, and M. Schmitter-Edgecombe. Discovering activities to recognize and track in a smart environment. *Knowledge and Data Engineering, IEEE*, 23:527–539, 2010.

[17] V. Rialle, C. Ollivet, C. Guigui, and C. Herv. What do family caregivers of alzheimer's disease patients desire in smart home technologies? *Methods of information in medicine*, 47:63–69, 2008.

[18] N. Roy, A. Misra, and D. Cook. Infrastructure-assisted smartphone-based adl recognition in multi-inhabitant smart environments. In *Pervasive Computing and Communications*, 2013.

[19] M. Schmitter-Edgecombe1 and C. P. et al. Development and psychometric properties of the instrumental activities of daily living: Compensation scale. *Journal of the National Academy of Neuropsychologists*.

[20] T. Theodoros and M. I. et al. An overview of automatic audio segmentation. *International Journal of Information Technology and Computer Science (IJITCS)*, 6(11):1, 2014.