Alaa Alhamoud, Pei Xu, Frank Englert, Philipp Scholl, The An Binh Nguyen, Doreen Böhnstedt, Ralf Steinmetz:
Evaluation of User Feedback in Smart Home for Situational Context Identification. In: Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), p. 20-25, March 2015.

Evaluation of User Feedback in Smart Home for Situational Context Identification

Alaa Alhamoud*, Pei Xu*, Frank Englert*, Philipp Scholl‡ , The An Binh Nguyen*,
Doreen Böhnstedt* and Ralf Steinmetz*
*Multimedia Communications Lab
Technische Universität Darmstadt, Darmstadt, Germany
Email: firstname.lastname@kom.tu-darmstadt.de
‡SAP AG
Walldorf, Germany
Email:pscholl@gmail.com

***Abstract*—In the recent years, smart home projects started to gain great attention from academic as well as industrial communities. However, an essential challenge that all smart home ideas face is the provision of the ground truth i.e. the labeled training data required to train the machine learning algorithms which achieve the smartness of the smart home. Another challenging task is to evaluate the correctness of the collected ground truth so that we can be sure that we train the system with correct data which represents the reality. In order to build a smart home which is interactive and adaptable to the behavior and preferences of its inhabitants, we need to have comprehensive information about the everyday behavior and preferences of the inhabitants of the smart home. This comprehensive information which needs to be collected represents the ground truth in the context of our smart home research. Many technologies have been utilized in order to collect this information. In this paper, we present our approach for collecting the ground truth in smart homes in a nonintrusive way. More importantly, we present our methodology for evaluating the correctness of the collected ground truth.**

I. Introduction

Knowing and obtaining the ground truth represents an essential challenge for all smart home projects. As adaptability represents one of the most important features the smart home must have, comprehensive information about the user's behavior and preferences should be available during the system development as well as deployment. Different ground truth information is required based on the functionality provided by each smart home project. In this paper, we present our system for energy conservation in smart home as an example for a smart home system in which ground truth information represents an essential part for the system functionality. We mainly focus on the collection of the ground truth as well as the evaluation of its correctness.

Our system for energy conservation in smart home depends on the fact that our everyday activities at home are strongly related to a set of electrical appliances which are necessary to perform these activities. Therefore, it is possible to provide the users with energy saving recommendations by informing them about appliances which are turned on but not related to the current activity.
The remainder of this paper is structured as follows. Section II introduces the hardware as well as the software components of the system. In Section III, we present the collected dataset. Section IV surveys several research projects with the main focus of collecting ground truth in the context of human activity recognition. In Section V, we introduce our methodology for ground truth evaluation. We conclude the paper in Section VI.

II. System Architecture

In this section we present the different system components as well as the communication technologies used to provide the communications between them. The main task of our system is to increase the energy awareness of users in smart homes by providing them with energy saving recommendations which inform the user nonintrusively about the energy saving potential. The system uses the user's energy consumption profile as well as other environmental parameters, namely motion, brightness, and temperature in order to infer the user's context in the smart home, thus to provide energy saving recommendations. Figure 1(a) shows the different components of the systems which are:

Sensor nodes: In our deployment we have used two types of sensor nodes, namely Plugwise sensors[1] which measure the power consumption of individual appliances and Pikkerton[2] sensors which monitor the temperature, brightness, and motion in the environment.

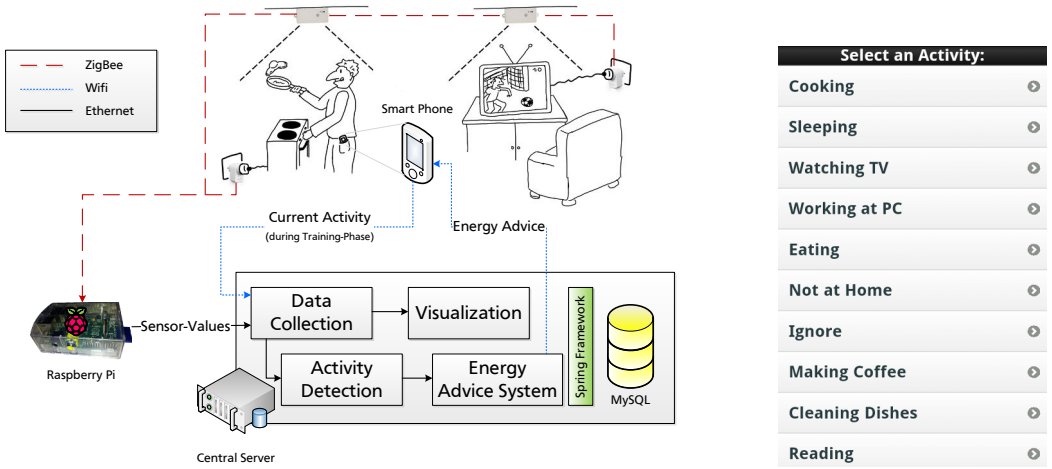
Gateway (Raspberry Pi): The Raspberry Pi[3] collects all the sensor readings and forwards them to the control server.

Smartphone: The smartphone plays two roles in our deployment. It is used by the user to provide us with the ground truth, namely her/his current activities. Furthermore, we use it to inform the user about any energy saving potential.

Control Server and Data Collection: The core functionality of the system is implemented by the control server where we build the machine learning model which uses the sensor data stored on the control server to generate the energy saving recommendations.

The co-author Philipp Scholl is employed by SAP AG; however, the opinions and results expressed in this paper are his own and do not denote the point of view of SAP AG.

[1]http://www.plugwise.com/
[2]http://www.pikkerton.com/
[3]http://www.raspberrypi.org/



(a) System architecture and network topology

(b) User Interface

Fig. 1. System Architecture and User Interface

As we can see in Figure 1(a), ZigBee, Wi-Fi, as well as Ethernet have been utilized in order to establish the communications between the different system components. Furthermore, Figure 1(a) shows an example in which the system informs the user about saving potential. In this example, we see that the power consumption of the TV as well as the oven are monitored with power sensors. Moreover, we monitor the temperature, brightness and motion in the kitchen and the living room. By processing the information provided by the sensors, the system recognizes that the current activity of the user is cooking and therefore considers the energy consumed by the TV in the living room as an energy wastage.

In order to build a machine learning model which is able to recognize the user's current activity, we need to collect feedback from the user and use it in combination with sensor readings as a training dataset for the machine learner. We have developed a user interface which can be used by the user on her/his smartphone to provide us with her/his current activity so that we build our machine learning model by relating this feedback to the collected sensor readings. Figure 1(b) shows a snapshot of the user interface.

III. Dataset Structure

We have deployed our system in two apartments where we collected our dataset. During both deployments we were able to collect around 42 million sensor readings combined with the user feedback. As we clarified, the sensor readings were collected by two types of sensors: the Plugwise sensors which monitor the power consumption as well as the Pikkerton sensors which monitor the temperature, brightness, and motion in the environment.
The first apartment (Deployment 1) in which we deployed the system was a shared apartment in which the experimenter was one of our researchers. In this apartment, a wireless sensor network monitoring his daily activities was deployed for 82 days. More than 22 million sensor events were generated by the corresponding sensors. The second apartment (Deployment 2) was a single apartment in which the experimenter had no experience with the system architecture and development. The duration of deployment 2 was 62 days during which about 20 million sensor readings were recorded.

During both deployments, the user has provided us with his current activity using his smartphone. Nine different daily activities listed in Table I were chosen to be monitored in each apartment. This list of activities has been chosen based on our discussion with the user as well as the available electrical appliances at home.
As shown in Table I and in Figure 1(b), besides the normal activities at the first nine rows, each deployment has an additional "Ignore" activity. By giving the users the choice of providing "Ignore" as a feedback, we protect their privacy by only monitoring the activities they agreed to be monitored. Furthermore, we maintain the correctness of our dataset. Since the duration of both deployments is two to three months, besides the listed activities, the experimenter could perform other activities not listed in our activity set which can cause unrelated sensor readings being generated. An example of such an activity could be the cleaning of the house without using a vacuum cleaner. This activity can't be monitored with the set of sensors we decided to deploy. Hence, the "Ignore" should be returned by the user to remind the researcher to ignore these sensor readings in later analysis.

TABLE I. Activities performed in both deployments.

Deployment 1	Deployment 2
Cooking	Sleeping
Sleeping	Watching TV
Watching TV	Eating
Working at PC	Listening Radio
Eating	Making Tea
Not at Home	Ironing
Making Coffee	Slicing Bread
Cleaning Dishes	Not at Home
Reading	Reading
Ignore	Ignore

The focus of this paper is the analysis and the evaluation of the user feedback which represents the ground truth of our project. We will use clustering techniques in order to find out whether the feedback collected during both deployments

is correct and matches the nine activities which we have monitored.

IV. Related Work

Obtaining ground truth has always been a challenging process in smart home projects, especially in projects which deal with the problem of activity detection in indoor environments. One possible approach to tackle this issue is the use of cameras to monitor all the user's activities and then manually annotating the data [1]. However, this approach results in serious privacy concerns for the house inhabitants. Another problem with the camera is that it might affect the user's normal daily behavior in the house which leads to a system which is not correctly trained . Moreover, the use of the camera comes with a lengthy and difficult process of video annotation. Other researchers have used speech recognition techniques to automatically annotating the activities by the user. In [2], authors have used a Bluetooth headset combined with a speech recognition software to let the user annotate her/his activities. This approach has achieved accurate speech recognition results with regard to the annotation process. However, it requires the users to always have their headsets mounted which could become inconvenient especially when the user has to provide the feedback over a long period of time.

Another idea which helps reducing the overhead of ground truth collection is to collect the ground truth in one house and use it for training purposes in other houses. Researchers in [3] have tried this approach. However, as human activities and the way they are being done vary significantly from one user to the other, this approach is difficult to be used in real world scenarios.

Different from the aforementioned projects, we present an approach which fulfills the following criteria:

- Avoiding to interfere with the user's privacy or at least keeping this at the minimum level.
- Reducing the annotation overhead for the researcher as well as the user as much as possible.
- Being able to automatically evaluate the collected user feedback which represents the ground truth in our case and guarantee its correctness.

In order to achieve this goal, we employ clustering techniques which proved to be a powerful tool for the analysis of huge datasets. Clustering aims at grouping the data based on some kind of similarity measure (e.g. distance measure). For instance, in computer vision, clustering is a typical approach for image segmentation [4] which normally contributes for object recognition and is the basis for high level image processing. Pixels within the same cluster share common properties (e.g. color, texture).

V. Evaluation of User Feedback

The ground truth provided by the user feedback represents an essential part of most smart home deployments which depend on it to provide smart services such as energy conservation and user comfort. Therefore, we should have means which are able to evaluate this feedback and to make sure that it has been correctly provided by the user with the minimum number of errors. As we mentioned before, our dataset contains feedback which indicates the beginning and type of each activity the user is doing. According to the user feedback, there are nine different activities performed by the user in each deployment. Since each activity is accompanied by a set of sensor readings, the collected dataset should contain sensor readings related to these nine activities after the exclusion of the activities marked as "Ignore" and all the sensor readings related to them. This means the dataset should contain nine different clusters with each cluster representing one activity. In order to validate this fact, we apply clustering algorithms on the dataset in order to find out how many clusters can be generated from the dataset. In the following section, we explain the clustering process and present the results of the feedback evaluation.

A. Feature Extraction and Clustering Process

As the objects which have to be clustered are the reported sensor readings during the activities, we use the maximum sensor readings in timeslots of two minutes as features for the clustering process.The instance dataset is composed of a set of instances represented in Equation 1 where $S_{n\_max}(slot_i)$ represents the maximum sensor value of sensor n in ith timeslot, and m is the total number of timeslots.

$$Instance_i =< S_{1\_max}(slot_i), ..., S_{n\_max}(slot_i) > i \in [1, m] \quad (1)$$

After obtaining the instance dataset, we implement the clustering process. Due to the size of the instance dataset, we need to choose an efficient yet a simple clustering algorithm. Therefore, we choose the typical clustering method K-means [5] as it achieves the trade-off between efficiency and simplicity when dealing with large datasets. K-means involves two major steps:

Data point assignment: This step consists of assigning each data point to its nearest centroid. A centroid of one cluster represents the mean of all data points within this cluster. The centroids are initialized before the algorithm.

Centroid update: This step consists of recalculating the centroids with respect to the new assigned data points.

These two steps are iteratively executed until the generated clusters are stable. In other words, no new centroid is generated. For computing the distance of two data points, we have chosen the euclidean distance measure. One disadvantage of the K-means algorithm is that the number of clusters should be assigned beforehand. Although we know that there are nine activities (clusters) performed in each deployment, this number cannot be used as it comes from the user feedback and therefore represents the value which needs to be verified. In order to make this algorithm suitable for dealing with this experiment, we tested different numbers of clusters and evaluated the clustering results to determine the optimal number of clusters.

B. Break point based approach

One approach for evaluating the clustering results is the breakpoint based approach. It focuses on finding the point that obviously breaks the tread of one cluster quality measure. In other words, a breakpoint is considered to be the actual number

of clusters, when the cluster quality measure decreases or increases rapidly before this breakpoint, and then respectively gradually rises or declines thereafter. The quality of a cluster is normally evaluated according to two basic types of measures [6]:

Cohesion: It represents the closeness of the data points within the same cluster or to their centroid. Measures for cohesion evaluation includes radius, diameter, and the most commonly used sum of squares error (SSE). As indicated in Equation 2, for each cluster, the distance between all data points in this cluster and the centroid are computed. Then, the SSE is obtained by summing up all these distances. k is the total number of clusters, x_i is a data point in cluster j.

$$SSE = \sum_{j=1}^{k} \sum_{x_i \in C_j} (x_i - C_j)^2 \tag{2}$$

Separation: It aims to examine how well the clusters are separated from each other. One of the typical measures of separation is the so called sum of squares between (SSB), whose computation is defined in Equation 3, where C_j represents the centroid, $\overline{X}$ is the mean of the whole dataset, and N_j is the number of instances contained in corresponding cluster.

$$SSB = \sum_{j=1}^{k} N_j (C_j - \overline{X})^2 \tag{3}$$

We evaluated the cluster quality based on the cohesion as well as the separation measures. The testing starts with an instance dataset of one week, then we increase the dataset by the data of another week each testing until all sensor values are covered. For deployment 1, 13 weeks of data are in total analyzed while for the other deployment 9 weeks of data are analyzed. For each dataset, the K-means clustering algorithm with a cluster number $k \in [1, 30]$ is applied. The evaluation is conducted on each individual instance dataset. The reason for not testing the whole dataset at one time is that, it is important to know whether it is really necessary to use such a large dataset for building the activity recognition model later on. In other words, we try to find out if it is necessary to collect the user feedback for such a long period of time(deployment 1: about 3 months, deployment 2: about 2 months).

However, after testing all instance datasets with regard to the number of weeks as shown in Figure 2, the results clearly indicate that the breakpoint based approach is incapable of telling the correct number of clusters, at least in the case of this work. One example result is shown in Figure 2 in which we conduct the test with a dataset of four weeks from the second deployment. In this figure, the trends of SSE and SSB with regard to the number of clusters are demonstrated. It can be seen from the figure that the breakpoints can be found in several positions (i.e. cluster number equals to 4, 9 and 13). An explanation for this is that breakpoint based approach requires the dataset to be clean enough which is not the case in this work. Our dataset contains to some extent noisy data due to errors in user feedback as well as sensor readings. Moreover, this approach is relatively subjective and relies on the perception of the ones who judge the results. Although there are some methods for dealing with this problem such as determining the number of clusters by calculating the difference between each pair of adjacent points, it still highly depends on the cleanness of the analyzed dataset.

C. Validity based approach

The validity based approach combines the intra-distance and inter-distance of the clusters, and produces a value for determining the number of clusters. Since no exact number of clusters can be found by using the breakpoint based approach, we evaluated the clustering results according to their validity. According to [7], the validity is defined as the average intra-distance of all clusters divided by the minimum inter-distance as shown in Equation 4. The intra-distance is the distance between each point and its centroid. The inter-distance is the distance between two cluster centroids. The idea behind the validity criterion is to minimize the intra-distance while maximizing the inter-distance. The number (k) of clusters which results in the smallest validity can be considered as the optimal number of clusters.

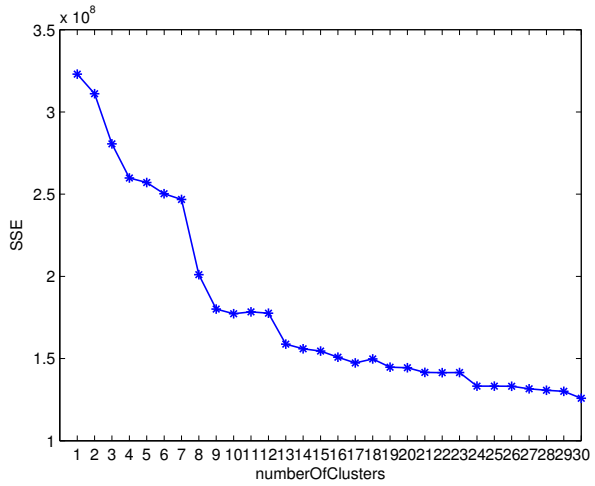
$$Validity_k = \frac{avg(Intra)}{min(Inter)} \tag{4}$$

However, we have to be careful when finding out the smallest validity as it might falsely occur when testing with a small number of clusters (e.g. $k = 2$). This happens due to the fact that a small number of clusters may lead to a large inter-distance resulting in the smallest validity. Hence, [7] defines that, the smallest validity can only be found after the first occurrence of a local maximum. A local maximum refers to a validity value which is larger than both its predecessor and successor.

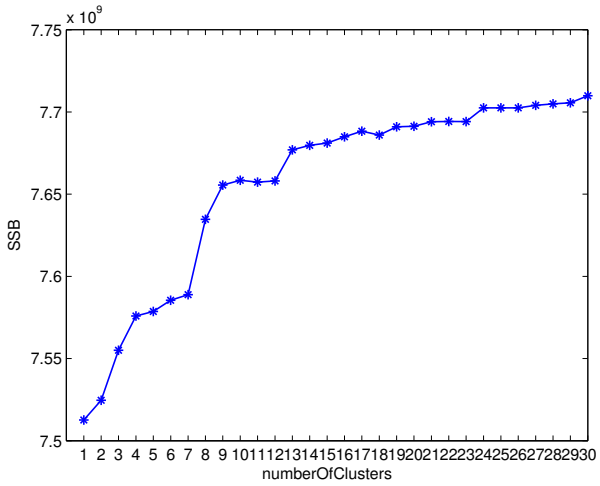
In order to find out the optimal number of clusters, we followed the same experimental setup we followed in the breakpoint based approach. We conducted the first test with a dataset of one week, and enlarged the size of the dataset by one week with each new test. For each test, we executed K-means with cluster numbers ranging from 2 clusters to 30 clusters. Thereafter, we evaluated the clustering results by calculating the associated validity. Figure 3 demonstrates one example for determining the optimal number of clusters from a dataset of four weeks. According to the aforementioned principle, the optimal cluster number produces the smallest validity which can be found after the occurrence of the first local maximum. As we can see from Figure 3(a), the optimal number must be between 2 and 11 clusters if the first local maximum can be found within this range. By further inspecting the validity within this range as shown in Figure 3(b), we can determine that the optimal number of clusters is 9. This is because the validity at 9 clusters is the minimum validity value which occurs after the first local maximum which can be found at 4 clusters.

As a conclusion, we can say that this experiment can help evaluating the user feedback. However, for some tests we got different optimal number of clusters than 9. This might be due to the fact that the sensor readings can be noisy from time to time due to battery as well as network problems. Moreover, activities such as “Sleeping” and “Not at Home” might trigger similar feature vectors as both of them cause almost no power consumption.

Table II summarizes all the optimal numbers of clusters we found during all the experiments conducted for both deployments. As we can see from the table, an optimal number of

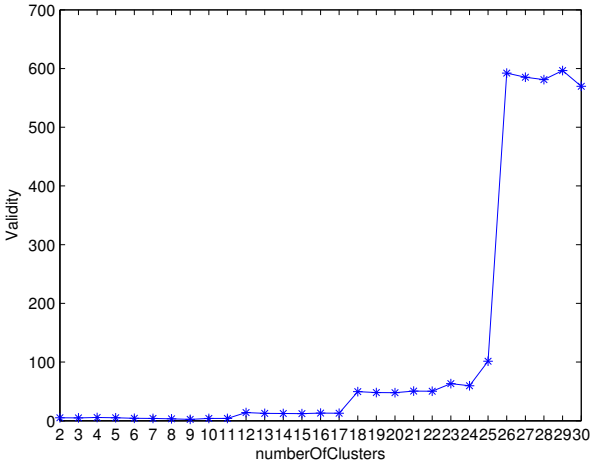


(a) SSE (Sum of Squares Error)

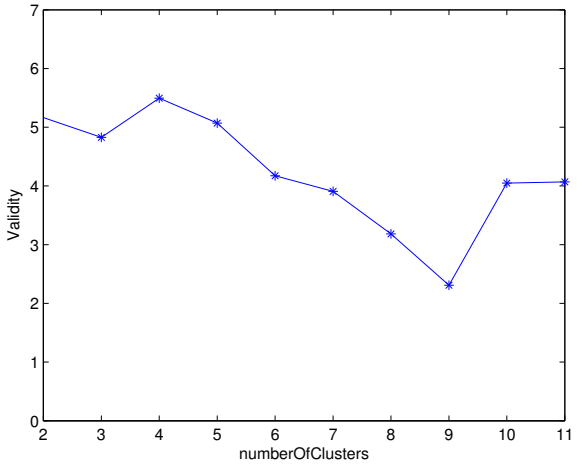


(b) SSB (Sum of Squares Between)

Fig. 2. SSE and SSB with regard to the number of clusters. Tested dataset: four weeks sensor data from deployment 2.



(a) Validity of 2 to 30 clusters



(b) Validity of 2 to 11 clusters

Fig. 3. Validity with regard to the number of clusters.

TABLE II. OPTIMAL NUMBER OF CLUSTERS FOR BOTH DEPLOYMENTS

	Optimal k	
n weeks data	Deployment 1	Deployment 2
1	4	6
2	7	6
3	9	4
4	5	9
5	4	4
6	6	5
7	18	11
8	7	6
9	6	12
10	13	-
11	4	-
12	6	-
13	6	-

clusters which is equal to the number of activities specified by the user, namely 9 can be found in the first deployment when testing with a dataset of three weeks. For the second deployment, we find this number with a dataset of four weeks.

When changing the size of the dataset, the optimal number of clusters varies. The small optimal numbers of clusters can be caused by similar features of the activities. For instance, both the activities “Sleeping” and “NotAthome” require no electronic devices. Moreover, the motion sensors during these two activities were either returning very low or no values. These two reasons might cause the instances of these two activities to be falsely grouped together, thus reducing the number of clusters.

Moreover, motion sensors will only be triggered when there is motion in the environment. For some activities such as “Working at PC”, as the user keeps the same position for most of the time, small or no motion values will be generated. This might lead to the problem that the instances of a specific activity might be grouped into different clusters just because of the differences in their motion values. In order to examine the impact of the motion sensors, we evaluated the clustering results without the values of the motion sensor. Table III lists

TABLE III. OPTIMAL NUMBER OF CLUSTERS FOR BOTH DEPLOYMENTS WITHOUT MOTION VALUES

	Optimal k	
n weeks data	Deployment 1	Deployment 2
1	4	6
2	5	5
3	20	4
4	4	9
5	19	4
6	4	5
7	7	6
8	4	5
9	4	8
10	5	-
11	8	-
12	6	-
13	4	-

the optimal number of clusters found for both deployments after removing the values of the motion sensors. As we can see from the table, the results deteriorate after removing the motion sensors. An optimal number of clusters which is equal to the number of activities was not found during the dataset analysis of the first deployment. However, removing motion sensors does have little impact on the results obtained from the second deployment. An optimal number of clusters equal to 9 still happens with a dataset of 4 weeks. For the datasets of 7, 8, and 9 weeks, the optimal number has even decreased. This supports our conclusion that the values of the motion sensors might contribute to an increasing optimal number of clusters.

VI. CONCLUSION

In this paper we presented our approach for obtaining the ground truth in smart home projects designed to recognize human everyday activities. As a main contribution, we introduced our methodology for the evaluation of the collected ground truth i.e. the user feedback needed to train the model of activity recognition. We built our work based on the K-means clustering algorithm. Looking at the results of the analysis conducted in the course of this work, we can say that our methodology were able in some of the experiments to find out an optimal number of clusters which is equal to the number of activities specified by the user i.e. to prove the correctness of the user feedback. However, in the other experiments the algorithm found out an optimal number of clusters which is not equal to the number of activities performed. This could be explained by one of two reasons. One reason could be that our proposed methodology was very sensitive to the size of the used datasets i.e. the number of weeks used to conduct the experiment and therefore was not able to produce the same optimal number of clusters in all experiments. Another reason could be that the user feedback itself was not accurate and that the user performed in some cases different activities from the one specified in his feedback. This leads us to the fact that this work proves to be promising and can be built on. However, it needs to be improved using the means of machine learning and data mining.

VII. ACKNOWLEDGMENTS

This work has been financially supported by the German Research Foundation (DFG) in the framework of the Excellence Initiative, Darmstadt Graduate School of Excellence Energy Science and Engineering (GSC 1070). Furthermore, it has been co-funded by the Social Link Project within the Loewe Program of Excellence in Research, Hessen, Germany.

REFERENCES

[1] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille, "A long-term evaluation of sensing modalities for activity recognition," in *Proceedings of the 9th International Conference on Ubiquitous Computing*, UbiComp '07, (Berlin, Heidelberg), pp. 483–500, Springer-Verlag, 2007.

[2] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, (New York, NY, USA), pp. 1–9, ACM, 2008.

[3] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Proceedings of the 8th International Conference on Pervasive Computing*, Pervasive'10, (Berlin, Heidelberg), pp. 283–300, Springer-Verlag, 2010.

[4] E. J. Pauwels and G. Frederix, "Finding salient regions in images: non-parametric clustering for image segmentation and grouping," *Computer Vision and Image Understanding*, vol. 75, no. 1, pp. 73–85, 1999.

[5] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2012.

[6] V. K. PangNing Tan, Michael Steinbach, "Cluster analysis: Basic concepts and algorithms," in *Introduction to Data Mining*, pp. 487–568, February 2005. Online available: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf.

[7] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in colour image segmentation," in *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, pp. 137–143, 1999.