

RESEARCH ARTICLE

Use of Clustering Algorithms for Sensor Placement and Activity Recognition in Smart Homes

SIMON SIMONSSON^{ID}, FLÁVIA DIAS CASAGRANDE,
AND EVI ZOUGANELI^{ID}, (Member, IEEE)

Department of Mechanical, Electronics, and Chemical Engineering, OsloMet-Oslo Metropolitan University, 0166 Oslo, Norway

Corresponding author: Simon Simonsson (simonsson.simon@gmail.com)

This work was supported by the Norwegian Research Council.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Norwegian Authority for Research Data.

ABSTRACT This work presents a novel method for motion sensor placement within smart homes. Using recordings from 3D depth cameras within six real homes, clusters are created with the resident's tracked location. The resulting clusters identify the possible position of a sensor and its field of view. By using a sequence of clusters as input to a Recurrent Neural Network, we evaluate our method on the task of activity recognition and prediction. These results are compared to using sensor events as input sequence, from motion sensors that were installed empirically in the same homes. Different clustering methods are investigated and all outperform the installed motion sensors, achieving a significant increase of prediction accuracy and F1-score.

INDEX TERMS Assisted living, clustering, computer vision, sensor prediction, sensor data.

I. INTRODUCTION

Ambient assisted living technologies (AALT) can enable people to remain longer in their homes, and age well e.g. by assisting individuals in daily activities, monitoring health and safety at home, and by improving the cost-effectiveness and quality of health and social services [1]. AALT usually comprises information and communication technologies (ICT), stand-alone assistive devices, and smart-home systems. A smart-home can be defined as a dwelling in which sensors and controllers are installed to enhance one or more aspects of the resident's everyday life [2]. This can for example include comfort, energy efficiency, security, and safety features. It can also include more targeted assistive functions, however, such systems are to a very limited extent available today. Assessments of using smart home technology to support different health concerns, such as ADL monitoring, chronic obstructive pulmonary disease (COPD), cognitive

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan^{ID}.

decline and mental health struggles, fall prevention, and monitoring heart conditions; concluded that the technology readiness for smart home and health monitoring is still low [3].

Mild cognitive impairment and dementia (MCI/D) are cognitive declines that can affect attention, concentration, memory, comprehension, reasoning, and problem solving [4]. These interfere greatly with a person's ability to perform daily activities and therefore in the case of older adults it leads to disability and dependency of others. A fair amount of research on smart home functions has aimed at assisting older adults, with and without MCI/D, in their everyday life [1], [5]. These systems rely heavily on recognizing and predicting activities in the home in order to assist the resident. Recognition and prediction algorithms are created using data collected from ambient sensors (e.g. cameras and binary sensors), robots, and/or wearable sensors. Having *as few sensors* as possible in the home is beneficial in terms of cost, privacy-conserving, as well as home aesthetics. Choosing the number of sensors in the home and their *optimal placement*

are important tasks to comply with these. Ideally, sensor placement algorithm should preserve activity recognition performance while minimizing the number of sensors that are required.

In this paper, a method for finding the *optimal number of motion sensors and their placement* is presented. Our method is compared to a baseline which consists of placing the motion sensors empirically and predicting next sensor event from previous event sequences. The methods are evaluated and compared on the task of activity recognition.

Data was collected from six one-bedroom apartments at an elderly care unit in Oslo. A minimal number of binary sensors were installed in the apartments – magnetic, power and motion sensors – for periods between 75 and 385 days. These sensors were installed empirically in locations that were assumed to be optimal for detecting activities of interest for this study. In addition to the binary sensors, depth video cameras were installed to collect data during one to seven weeks. The recordings collected from the cameras allowed to create a labeled dataset of activities, from which a relation between a sequence of motion sensors to different activities could be created. Using recurrent neural networks, a model that recognizes activities based on a sequence of motion sensor events can be achieved.

In addition, using the recordings from the depth cameras and a person detection algorithm, allowed creating a dataset of the position of the resident within their home over time. The detected positions are in turn clustered, revealing cluster centroids. Our method uses these centroids as “optimal” locations for motion sensors. We then use a recurrent neural network that recognizes activity based on a sequence of centroids (possible motion sensors) and compare to the activity recognition model accuracies we achieved when the sensors were placed heuristically.

More specifically on our method, Fig. 1 presents a sequence of the process to generate centroids that would be sensor locations. Person detection is performed by applying the YOLO algorithm. Furthermore, three different methods of location clustering are analysed and compared: K-means, DBSCAN and BIRCH.

Our contributions include:

- A novel method for sensor placement within smart homes. One that adapts to a unique home and the movement patterns of it's occupant.
- Comparison of clustering algorithms for identifying key points within a home.
- Comparison with empirical method of sensor placement.

The work has been carried out in an interdisciplinary project, the Assisted Living Project (ALP), that involves experts in health, technology, and ethics [6]. The aim of the project was to develop assisted living technology (ALT) to support older adults with MCI/D live a safe and independent life at home.

The organization of this paper is as follows. Section II presents the related work on optimal sensors placement in the literature. Section III introduces our field trial and the

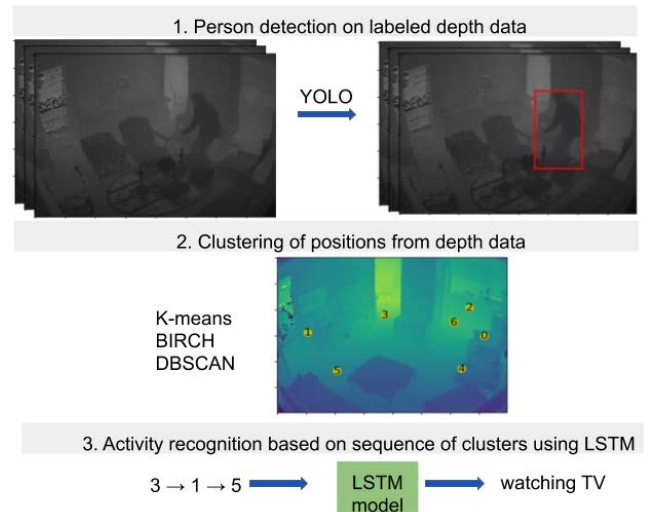


FIGURE 1. Workflow.

collected data. Section IV-A shows the data pre-processing steps for depth cameras data. Section IV-B presents the clustering algorithms applied to the depth video data. Section IV-C1 presents our neural network model for activity recognition using motion sensors. Section V presents the results and then discusses the findings of this work. Finally, Section VI concludes this paper with final remarks and suggestions for future work.

II. RELATED WORK

Finding the optimal number of sensors in an environment and their optimal location is a task many studies have addressed, for different applications. Empirical placement and several tests is one way of finding optimal sensor placement, as performed for categorizing type of physical activity and corresponding energy expenditure in older adults using wearable accelerometers [7]. The implementation of Markov chains and solving the boolean optimization problem derived from it, is used as method for optimally placing markers and minimizing their number in large public environments for robot localization [8]. These robots would assist older adults in moving around in such spaces.

Sensor placement has been presented formally as a binary variable problem, as standard optimization algorithms can be used in that case, for fault diagnosis systems for the industry [9]. Genetic algorithm has been used for sensor placement in structures to identify location and severity of damages [10], [11], and others have combined genetic algorithms with multi-objective algorithms for the same goals [12], [13]. Several analytical solutions and mathematical formulations, including path planning and trajectory tracking techniques, were implemented to identify the optimal geometrical configuration of sensors in autonomous underwater vehicles [14]. Optimal sensor placement has also been formulated as a cost minimization problem for outage detection in power distribution systems [15].

More specifically to the focus of this paper, optimal sensors placement has also been studied in the field of smart homes for ambient assisted living. For this application, sensors can be placed in the environment or on the body. Several feature selection techniques were investigated for sensors placement on the body for activity recognition in the home [16]. The work shows that the larger the number of accelerometers, the higher the classification accuracy becomes. Work has been done on identifying which inertial sensor contributes to best classifying activities on the *PAMAP2* dataset using LSTM models [17], [18]. The method of using inertial sensors is not preferable though, as wearable sensors are not ideal for older adults with mild cognitive impairment or dementia, as they can forget why they are using such sensors or that they need to put them on [19]. This many sensors on the body are, moreover, uncomfortable to wear all the time. Ambient sensors are therefore most preferable, and hence also the most present in the literature. A toolkit of ambient sensors was proposed as to track daily activities performed by subjects with Parkinson's disease and/or dementia, focusing on minimizing invasiveness and obstruction of the person as well as ease of installation [20].

A decision-making tool to help experts to do the best choice for sensors deployment is under development to minimize costs and the overlapping of motion sensor detections in a home environment [21], [22]. Their method represents the motion sensors in a space as binary grids and then integer linear programming techniques are applied to optimize their placement, as applied in [9]. The authors implemented their method in an open space in the home including the entrance hallway, the kitchen, the dining room and the living room and presented satisfactory results.

Optimal sensors and their placement for target tracking within Ambient Assisted Living have been investigated using an Integer Linear Programming model [23]. In this work, factors like layout of the Region of Interest, sensor's field of view and their orientation are taken into account while delivering satisfactory results. A user interface was also developed using this approach, enabling easy monitor and real-time location, energy consumption and comfort of an occupant within their home.

Motion sensors have also been used in a Japanese residential house, where 39 sensors were installed in the living room, the tatami room, two small rooms, kitchen, hallway [24]. They investigated the number of sensors required for sufficient localization accuracy in the home. Four algorithms for sensor placement optimization were examined: forward greedy algorithm, backward greedy algorithm, l_1 regression, and group lasso. The empirical results show that the backward greedy sensor selection algorithm achieves the most stable performance and that a few selected sensors (five to eight) presented competitive performance compared to the initial setting with 39 sensors.

Five algorithms for defining motion sensor placement in smart environments have been also compared: Human

Intuition-Based (HIB), Monte Carlo-Based (MC), Two-Dimensional Uniform Placement (Grid), Hill Climbing (HC), and Genetic Algorithm (GA) [25]. The authors use the CASAS dataset, collected in a 3-bedroom living-lab apartment and with a semi-grid of 48 sensors – infrared motion detectors, light sensors, door sensors, temperature sensors, light switch monitors, and object shake sensors [26]. They have compared the methods such that the placement of sensors (and possibly also a reduced number of sensors) derived from them should provide as good activity recognition accuracy as the original dataset. HIB, MC and Grid were baseline algorithms and generated layouts that covered more of the space, including areas that did not have any target activity happening. However, the GA and HC approaches grouped sensors around the areas that physically divide activities. The GA was more effective for placing the sensors and provided better results for the activity recognition. In that case, 26 motion sensors were in the home, as there were in the original dataset.

The method of combining algorithms has also been used for motion sensor placement within smart homes. One work used a hybrid of the Particle Swarm Optimization- and Whale Optimization algorithm with the goal of maximizing coverage and as well minimizing costs, evaluating their method on the CASAS dataset [27]. By combining the algorithms, the strengths of one would cover for the weaknesses of the other.

Other work has been carried out with the same CASAS dataset [28]. In this case, the authors investigate whether activity recognition with a smaller number of sensors could perform as well as with all sensors. The mutual information measurement (MI) was used to quantify the dependence of two variables – sensor and activity. Afterwards, sensors that had low MI would be removed. A second approach consists in selecting the number of sensors by applying hierarchical clustering of the sensors. This cluster would select the set of clusters with highest MI values and merge sensors that are close to each other in the space. The work shows that in their setup, an average of 21% of the sensors can be removed from the apartment without loss of accuracy when excluding sensors with low MI. When the hierarchical clustering is performed, an average of 58% of the sensors can be removed. The work also established that not only does an increase in sensors add to the cost of smart environment as well as maintenance and resource costs, but it can also actually degrade activity recognition performance. Notably, more sensors mean also more complex patterns to be learned by algorithms and possibly more collected data.

The works in [22] and [24] have applied their sensor placement optimization methods to localization accuracy. They indicate as future work their application to activity recognition, which should give more value for functions in smart homes. The work in [28] applies the method to activity recognition, however the method does not provide the best placement of sensors. All the works present the disadvantage

TABLE 1. Sample of binary sensor data.

Timestamp	Sensor ID	Sensor message
01.09.2017 07:58:05	2	1
01.09.2017 08:00:14	12	1
01.09.2017 08:01:01	4	1
01.09.2017 08:02:56	5	1
01.09.2017 08:03:05	12	0

of having to install the sensors in the home as a first step, which is costly operation both in terms of time and capital. In our approach, instead of several sensors, we would install a camera in each room prior to installing the final sensors. In addition, our method can be used to shed light to the best placement of sensors and use this as a guideline for future installations. Also, the knowledge can be transferred to other sensors than motion, as in [22], [24], and [25].

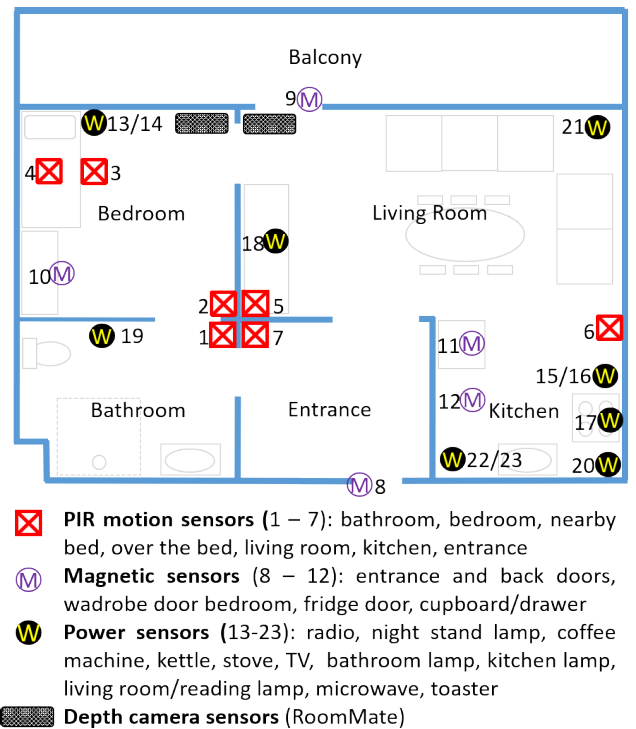
III. FIELD TRIAL

Six residents over 70 years old participated in our field trial. All apartments are part of a community care facility and have similar layouts – comprising a bedroom, a living room, an open kitchen area, a bathroom, and an entrance hall (Fig. 2).

Binary sensors were installed in each of the apartments, and were kept to a minimum number in order to minimize surveillance of the residents. The set of binary sensors was chosen so that it can enable the realization of useful functions for older adults with MCI/D as these were indicated at dialogue cafes with the users [6]. It contains motion (passive infrared sensor – PIR), magnetic, and power sensors. Motion sensors (Pyroelectric/Passive Infrared – PIR) detect motion through detecting a change in infrared radiation in the sensor’s field of view. The sensor generates an event with message “1” each time motion is detected. It otherwise sends no event. Magnetic sensors consist of two components, a reed switch and a magnet. They are fitted opposite to each other on doors, windows, and drawers to indicate whether they are open or closed. An electric current is created when the two pieces are close to each other, the circuit otherwise being broken. Events with message “1” are generated for open and “0” for closed. Power sensors measure the electricity usage of an appliance. They can therefore indicate whether the appliance is turned on or off, events with message “1” being for on and “0” for off.

These sensors enable inference of occupancy patterns (movement around the apartment) and some daily activities – kitchen related activities, dressing, being in bed –, and leisure activities — reading, watching TV, listening to radio. The data from these sensors include timestamp (date and time, precision in seconds), sensor ID, and sensor message (binary). Table 1 shows an example of data collected from the sensor network.

All the participants had the same initial proposal of set of sensors, as shown in Fig. 2. However, not all apartments could have the exact same set of sensors due to physical limitations (e.g. fridge door with a too big gap to enable the use of

**FIGURE 2.** Proposed sensors system for field trial apartments.

magnetic sensor) and/or different equipment (e.g. residents either have a coffee machine or a kettle). As it has been reported that ADL scores can be predicted from the long-term location and movement records obtained from solitary elderly people [29], in addition to these being common to all apartments, we chose to use only the *motion sensors* data.

In addition to the binary sensors, two RoomMate depth video cameras were installed in the apartments (Fig. 3). One of them monitors the living room and kitchen area, while the other monitors the bedroom area, as shown in Fig. 2. The RoomMate is an infra-red (IR)-based depth sensor and measures the distance of surfaces to the camera by time-of-flight (TOF) technology with pulses at 15MHz. The resolution is 160×120 pixels, with a rate of 25 frames per second. This is rather low resolution – a fact that is advantageous with respect to privacy, but makes data processing quite challenging. Adding to the challenge, the depth images produced by the RoomMate camera have a fair amount of noise. Fig. 4 shows an example raw image of RoomMate depth video camera.

IV. METHODOLOGY

As illustrated in Fig. 1, the work presented in this paper was performed through three steps:

- 1) Depth data pre-processing (Fig. 1 step 1), which is elaborated in section IV-A. Using the gathered depth data, this step consists of creating a dataset with the occupants location through time.



FIGURE 3. RoomMate depth sensor deployed in our field trial [30].



FIGURE 4. Raw RoomMate depth image example.

- 2) Clustering (Fig. 1 step 2) the results from the previous step using different algorithms, detailed in section IV-B.
- 3) Evaluation of our method (Fig. 1 step 3), which was done by training a classifier to perform activity recognition based on the input from the previous steps, and compared to using the empirically placed sensors instead. This part is further explained in section IV-C.

A. DEPTH DATA PRE-PROCESSING

Using the depth images we retrieved from the RoomMate cameras installed in the homes, we create a time series with the location of the residents within their homes. Instead of doing this manually, which would require a very large amount of time and effort, we performed person detection throughout the recorded material. *You Only Look Once* (YOLO, or YOLOv3) is a state-of-the-art object detection algorithm within RGB images [31], which works by dividing the image into a grid of cells and for each of them the algorithm tries to detect the bounding box of the object(s) it has been trained to find. The model only goes through each cell once to which it owes its name and also speed. The model



FIGURE 5. YOLO applied to raw image in Fig. 4.

can subsequently also be trained to detect people as in our case.

YOLO works best for RGB data which available pre-trained weights for the network are also optimized for, however depth images are mono-channeled so we therefore need to normalize the images and then duplicate the data into three channels, creating gray scale RGB images. We trained a detector using 4000 labeled images which we later used to create a dataset for the location of the residents within all the recordings. Empirical tests showed the YOLO model was very accurate despite the noise and lack of color in the images (see Fig. 5 for an example). Even though fast compared to alternative methods, YOLO can be a time-consuming algorithm without the proper hardware, to create our dataset faster we only tried to detect on each frame using a 5-second time step, making the assumption that the time gap is negligible.

When no person is detected in the frames, we assume the person is either outside of the apartment, or in the bathroom which is out of the field of view of the cameras. There can also be multiple people in the home at the same time (visitors, care-takers, etc.), in these occasions, we marked the possible cluster with a unique identifier that two people are present, and treated it as a unique input.

B. CLUSTERING ALGORITHMS

Clustering is an unsupervised technique that groups a set of samples based on the similarities between their attributes and/or proximity in the vector space. There are several types of clustering techniques – partitioning, hierarchical, and density-based.

In this work, several clustering algorithms were applied to cluster samples of position coordinates (x and y coordinates within each frame) that indicate the center of the person moving around in the apartment, information acquired using the method as explained in Section IV-A. Our hypothesis is that these clusters could then indicate how many sensors we should have in the apartment and more importantly where

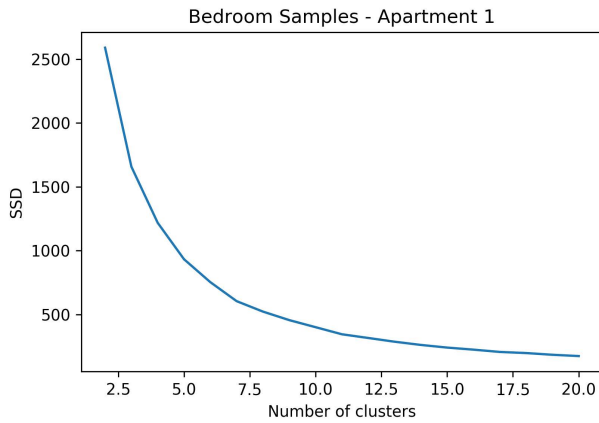


FIGURE 6. Graph for elbow method when K-means clustering the samples in the bedroom.

they should be placed using the center of the achieved clusters. The placement and number of sensors is then tested for activity recognition in the homes. In this section, we present the clustering results of three different techniques, one of each type we mentioned.

The clustering models were implemented using the Scikit-Learn library [32]. We model each room (bedroom and living room/kitchen) of each apartment individually, with part of the data – one week of depth data. Then, we label the rest of the data with the trained model.

1) K-MEANS

The K-means is an algorithm of the type partitioning. In the K-means algorithm, the position samples of each person are classified into K clusters such that the sum of square distances (SSD) within each cluster is minimized [33]. Each cluster contains a centroid, given by the mean value of each feature of the algorithm. We perform K-means for a number of clusters K between 1 and 20 for the room and choose the best K manually according to the elbow method [34]. This method consists of plotting an SSD vs. K graph and choosing the K that resembles an “elbow” (the point of inflection on the curve), which is the best fit for that problem. As a last step we tried to identify clusters that we deemed close enough to each other that they could be grouped into one.

Fig. 6 shows the graph of SSD vs. number of clusters, as dictated by the elbow method, for the bedroom area in apartment 1. A number of clusters equal to seven gives optimal results. For the living room, the graph looks similar and also with seven clusters as optimal. Hence, we create a separate model with seven clusters for each room – see Fig. 8 and 7. Notice that even though the number of clusters in the bedroom is 7, we group clusters 0, 4 and 6 as one, as they are very close to each other. Hence, a manual step after executing the algorithms is required.

2) BIRCH

The *Balanced Iterative Reducing and Clustering using Hierarchies* (BIRCH) is a hierarchical clustering technique [35].

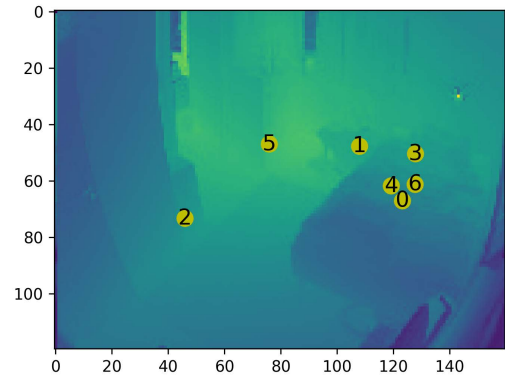


FIGURE 7. Clusters found with K-means in the bedroom.

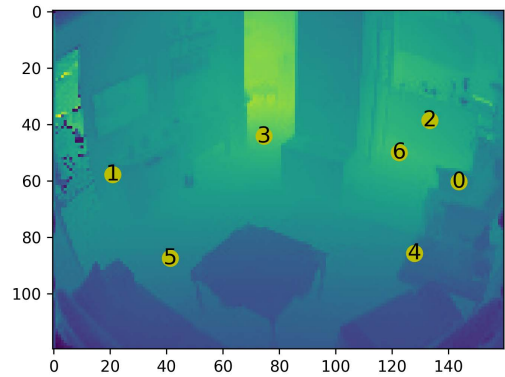


FIGURE 8. Clusters found with K-means in the living room.

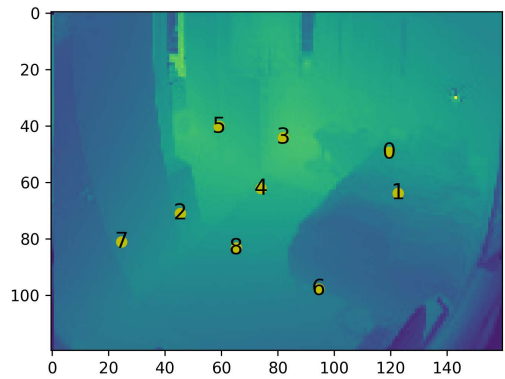


FIGURE 9. Clusters found with BIRCH in the bedroom.

The method has as a first step to build a Clustering Feature tree (CF tree) while scanning a given dataset. Each node in the CF tree is a cluster, that can contain sub-clusters.

Fig. 9 and 10 present the results for this technique when applied to the same room and data as in Fig. 8 and 7, using the following parameters:

- Branching factor of 50.
- Threshold of 0.10.
- Number of clusters set to 11.

As opposed to K-means, where we could use the elbow method to decide the number of clusters, here we had to choose the number of clusters empirically through testing, but up to the maximum number of sensors that were already installed within the room.

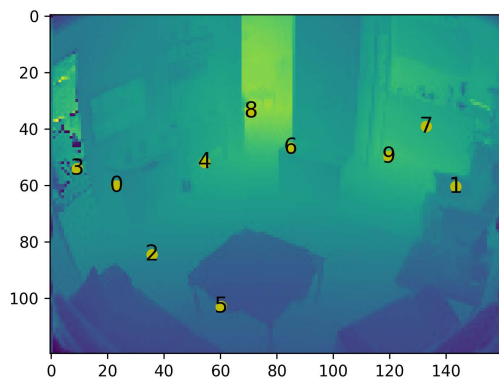


FIGURE 10. Clusters found with BIRCH in the living room.

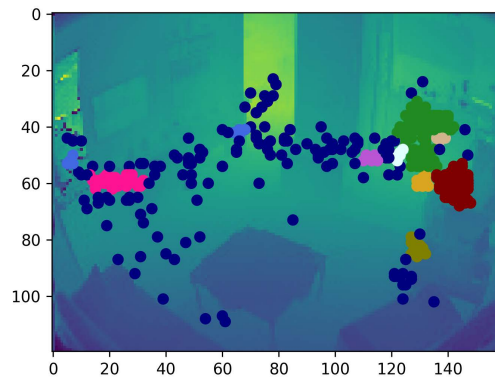


FIGURE 12. Clusters found with DBSCAN in the living room. Each color represents a cluster.

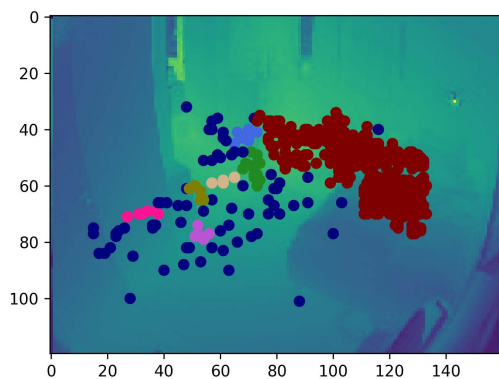


FIGURE 11. Clusters found with DBSCAN in the bedroom. Each color represents a cluster.

3) DBSCAN

Density-based Spatial Clustering of Applications with Noise (DBSCAN) is the last type of data clustering algorithm investigated. The main idea is to group data points that are close to each other in space [36].

Fig. 11 and 12 show the results for when DBSCAN is applied to the same room and data as in Fig. 8 and 7, using the following parameters:

- Maximum distance of 0.02 between samples to be considered of in the neighborhood of the other.
- Minimum of 5 samples within a neighborhood to be a core point.
- Euclidean distance as metric for measuring distance.

C. EVALUATION—ACTIVITY RECOGNITION

As mentioned in the introduction of this paper, in order to evaluate the success of our method we train a classifier for activity recognition using as input the events from the sensors installed within the homes, compared to using the clusters. In this section we describe our model for recognizing the ongoing activity based on the different stated inputs. We are using Recurrent Neural Networks (RNN) that take as input a sequence of *unique events*, and predict one of six labeled activities – getting (un)dressed, eating, walking, reading, watching-TV and sleeping. With the term *unique events* we mean that we pre-process and group all sequential similar

events and treat them as one event, so for example in the case where we have sequence $A \rightarrow B \rightarrow B \rightarrow C$ where the event B is triggered twice after each other, we group these two events into one and this becomes $A \rightarrow B \rightarrow C$.

For our purpose the activities were labeled manually by inspecting the recordings of the residents. For each labeled activity we retrieve N (after mentioned grouping) previous events in time to the activity, and train the network thereafter. In the Results section we evaluate and discuss optimal values for N that we refer to as memory length.

Not all activities occur equally often, creating an imbalance between the number of samples representing each class. We employ certain methods discussed in this section to help balance the training set, and use the F1-score as evaluation metric that is well suited for imbalanced datasets.

1) RECURRENT NEURAL NETWORK

Recurrent Neural Network (RNN) maintains an internal *memory* and has therefore been broadly applied to sequence prediction. It achieves good performance for inputs that are sequential in time and has been applied, for example, to text generation [37], speech recognition [38], and pattern recognition in music [39]. Long Short-Term Memory (LSTM) [40] is an RNN architecture that is designed to be better at storing and accessing information than the standard RNN [41].

2) NETWORK ARCHITECTURAL SEARCH

The architecture of a neural network and the parameters that are used can heavily influence the end-performance of the model, it can as well prove difficult to find these parameters. For the task of finding the optimal architecture and training parameters we took use of genetic algorithms [42]. Genetic algorithms have shown to outperform other methods for hyperparameter-tuning by converging much faster. The basic idea behind genetic algorithms is to avoid trying all possible permutations of parameters and keeping track of the combinations that perform well. The algorithm is run in *generations* in each of which there is a given *population*. For the first generation all the members are assigned random parameters from a fixed *gene pool*. The gene pool are the

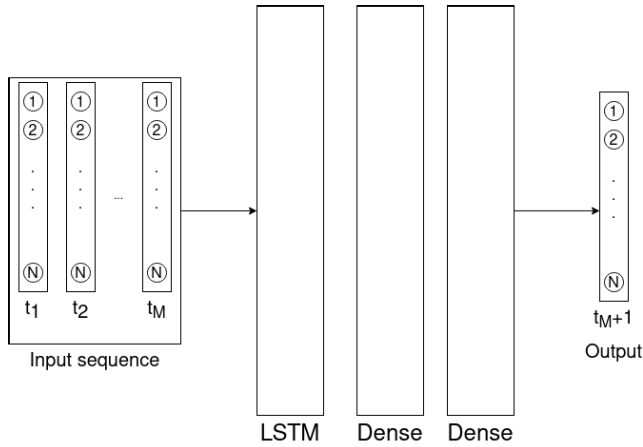


FIGURE 13. Final network architecture where M denotes the memory length (i.e. sequence length of the input) and N is the number of classes for the method (i.e. number of sensors or clusters).

different parameters that you want to test combinations of. Here they are the number of hidden layers, number of inputs for each layer, dropout rate and learning rate. At the beginning of each generation, each member of the population has a score computed for them which they are ranked after. The lowest scoring members are discarded, while the highest ones *breed* by random sampling their parameters to create offspring that take the place of the discarded members. To avoid the possibility of falling into a local minimum there is a random *mutation* of the population and we can also keep a number of the lower ranking members between generations. Using this method we therefore avoid computing the score for all members of the population as we already know it for those that are kept, we only need to do this for the offspring and any that mutated.

In our case, we implemented the genetic algorithm using a population size of 20, keeping the best 8 members and having a mutation chance 0.25%, which we ran for 10 generations. Each member was evaluated on the same training and validation datasets, and were ranked after the validation accuracy. Early stopping was used to speed up the process a little and avoid overfitting. The dataset was a merger of all the users’ data where the memory length was set to 10. This was done for each kind of input: sequence of sensor events, and sequence of clusters. After each finished we chose the common best performing parameter settings so we had one network architecture only to evaluate our methods on.

Our search gave the following parameters:

- 1 hidden LSTM layer.
- 128 neurons for each layer.
- 0.2 dropout to avoid overfitting.
- Learning rate of 0.001 for the ADAM optimizer.

The final architecture is shown in Fig. 13.

3) SMOTE

In order to tackle the class imbalance in our datasets, as some activities happen more often than others, the Synthetic

TABLE 2. Duration and Number of Samples for each Apartment.

Apt. ID	days bin	days depth	# samples
1	358	49	66711
2	311	49	28053
3	163	21	17697
4	385	35	34041
5	260	49	39748
6	75	28	23813

Minority Over-sampling Technique (SMOTE) is used [43]. SMOTE is an over-sampling technique that creates synthetic samples for the minority classes. The new samples are created by interpolating the values of the existing samples and are only used to train the model. The Imbalanced-Learn library was used to implement this technique [44].

V. RESULTS AND DISCUSSION

Table 2 shows the duration of data collection (for both motion sensors and depth data) and the number of samples each apartment had, which is dependent of having both motion sensors data and recordings with the depth camera working at the same time.

A. CLUSTERING ALGORITHMS

Section IV-B shows the results when applying the clustering algorithms K-means, BIRCH and DBSCAN to group the residents’ locations in the living room/kitchen and bedroom areas.

From the three algorithms, we can conclude that both K-means and BIRCH are more suitable than DBSCAN for placing sensors in a home as they provide only one point for each cluster, which can be seen as the location of the sensor. DBSCAN provides a grouping of the sensors by density, and do not provide centers of clusters indicating where sensors could be positioned. It can be however very suitable for applications where depth cameras are used in the final system. For example, in Fig. 11, the blue cluster possibly indicates movement whilst the others can indicate other activities – e.g. in bed or going to bed (red cluster), in front of the wardrobe/changing clothes (pink cluster). Hence, we chose K-means and BIRCH to analyze.

In addition, the methods for finding the optimal number of clusters are not yet perfect for clustering algorithms. After clustering the positions, there is a manual step where we remove some clusters that are too close to each other, even if the algorithm provided that as the optimal. For example, in Fig. 7, clusters 0, 4 and 6 are grouped as one cluster. Hence, a manual step after executing the algorithms is required. This is performed for both K-means and BIRCH. Note also that the optimal number of clusters is also affected by the type and number of activities that have been defined as further discussed in section V-E.

Table 3 shows the final number of K-means and BIRCH clusters for each apartment. We can notice that BIRCH provides either the same number or more clusters than K-means.

TABLE 3. Number of clusters found by each algorithm for each apartment.

Apt. ID	K-means	BIRCH
1	12	22
2	6	8
3	12	12
4	12	12
5	10	12
6	10	12

B. MEMORY LENGTH

We refer to memory length as the number of previous events used to predict the next one. Fig. 14a, 14b, 14c, 14d, 14e, 14f show the evolution of the mean F1-score by memory length, after running the LSTM network model 5 times with different training and validation sets for each apartment. The validation set would always have 3000 different samples in order to compare the F1-scores fairly. These results are also summarized in tables 4 and 5. The different apartments show similar evolution of increasing F1-score with increasing length of the input sequence to the model. We observe F1-scores ranging between 0.28 and 0.65 for the shortest memory length of 2, depending on the input and apartment, up to F1-score between 0.47 and 0.95 for the longest memory length of 100 subsequent unique events. In some cases we can observe a top F1-score already after a memory length of 15, but for most apartments, the peak F1-score is attained at memory length of 50 with F1-scores ranging between 0.48 and 0.95. Some apartments still improve with higher memory length, for example apartment 6 that improves with 0.03 for a memory length of 100 using k-means. Considering the small size of each apartment, the movement pattern of the resident therefore becomes quite similar whichever activity they are going to perform, which explains the need for a longer sequence for the model to be able to properly match the patterns to activities.

The trend that we observe for the F1-score is shared between all three kinds of inputs, where the sensor events are not able to acquire the same peak F1-score as the clusters, with the exception of apartment 3. The difference between the peak performance attained by the different methods varies significantly across apartments, where we observe a 0.44 F1-score increase between using the sensor data and using BIRCH clusters for apartment 5, and only a 0.02 increase for apartment 2. The only exception to this was apartment 3 where the sensor data attained 0.11 higher F1-score than K-means, which was the second-best method for this apartment.

Overall, BIRCH and K-means perform equally well, both achieving on average a peak F1-score of 0.90, while the baseline with motion sensors is the worst performing with an average peak F1-score of 0.77 across apartments.

C. SIZE OF TRAINING DATASET

We have two aims for analyzing the obtained prediction F1-score versus the size of the training set: firstly, to check whether the collected data is enough to train the model, and

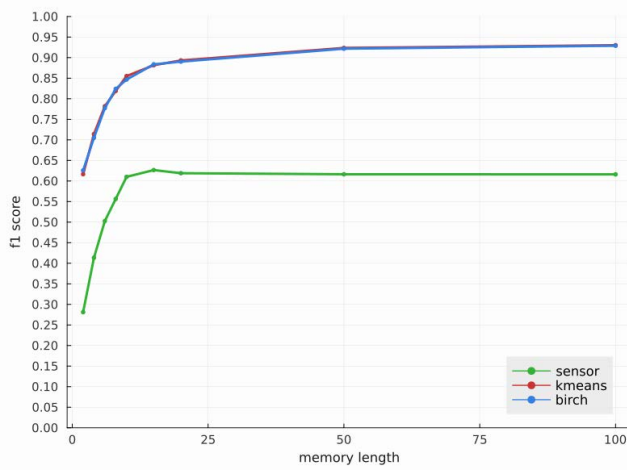
TABLE 4. Mean F1-score by memory length and input type for apartments 1, 2 and 3.

Apt. ID	Mem. len.	Sensors	K-means	BIRCH
1	2	0.28	0.62	0.63
	4	0.41	0.71	0.71
	6	0.50	0.78	0.78
	8	0.56	0.82	0.82
	10	0.61	0.86	0.85
	15	0.63	0.88	0.88
	20	0.62	0.89	0.89
	50	0.62	0.92	0.92
2	2	0.65	0.71	0.74
	4	0.74	0.77	0.81
	6	0.83	0.86	0.87
	8	0.90	0.88	0.90
	10	0.93	0.93	0.94
	15	0.95	0.94	0.95
	20	0.95	0.95	0.96
	50	0.95	0.96	0.97
3	2	0.36	0.63	0.60
	4	0.49	0.66	0.73
	6	0.62	0.68	0.69
	8	0.71	0.68	0.74
	10	0.75	0.75	0.72
	15	0.85	0.72	0.75
	20	0.87	0.75	0.69
	50	0.87	0.69	0.69
100	2	0.86	0.71	0.66

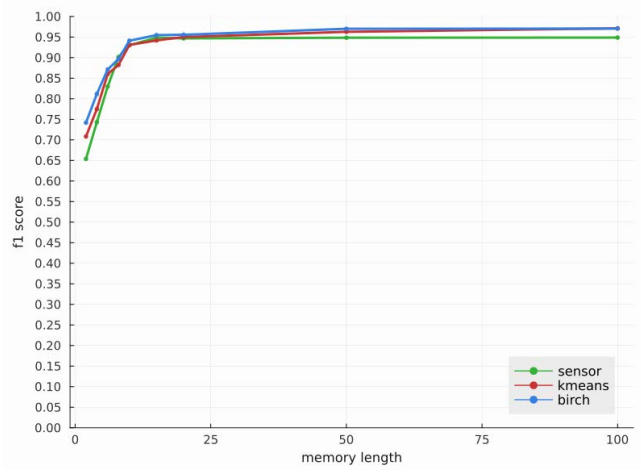
TABLE 5. Mean F1-score by memory length and input type for apartments 4, 5 and 6.

Apt. ID	Mem. len.	Sensors	K-means	BIRCH
4	2	0.48	0.71	0.72
	4	0.67	0.77	0.79
	6	0.76	0.85	0.85
	8	0.83	0.90	0.89
	10	0.87	0.92	0.92
	15	0.90	0.95	0.94
	20	0.89	0.95	0.94
	50	0.89	0.95	0.95
5	2	0.32	0.62	0.63
	4	0.38	0.71	0.71
	6	0.45	0.79	0.78
	8	0.46	0.83	0.82
	10	0.47	0.85	0.84
	15	0.47	0.88	0.89
	20	0.48	0.90	0.89
	50	0.48	0.91	0.91
6	2	0.30	0.51	0.51
	4	0.54	0.60	0.62
	6	0.64	0.68	0.69
	8	0.69	0.70	0.72
	10	0.71	0.74	0.75
	15	0.76	0.80	0.80
	20	0.75	0.82	0.82
	50	0.77	0.83	0.84
100	0.77	0.86	0.86	

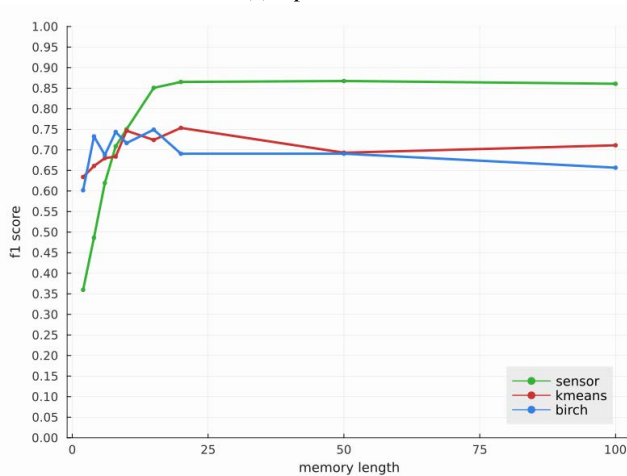
can achieve a stabilized F1-score; and secondly, to identify the amount of data that needs to be collected on each resident for the purpose of accurate activity recognition. A validation set of 3000 samples is allocated for each apartment, and is used for each defined training set size to make a fair comparison. Here we also chose to use the dataset with



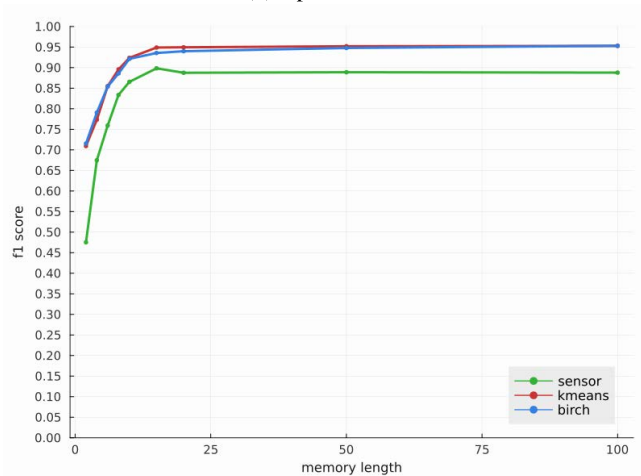
(a) Apartment 1.



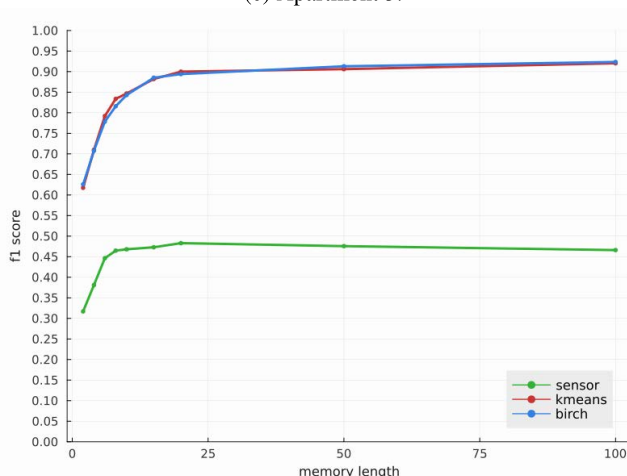
(b) Apartment 2.



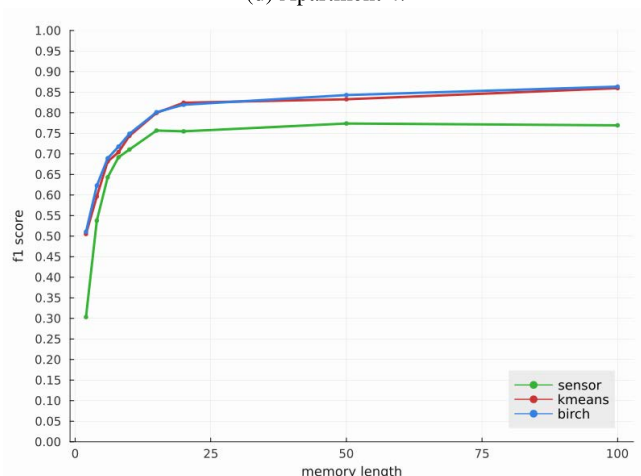
(c) Apartment 3.



(d) Apartment 4.



(e) Apartment 5.



(f) Apartment 6.

FIGURE 14. F1-score of prediction of the activity vs. memory length for different apartments.

a memory length of 50 as this was the shortest memory length for which many of the apartments achieved their peak performance, as discovered in section V-B. Training set sizes

of 200, 500, 1000, 2500, 5000, 7500, 10000, 15000 and 20000 samples were used. Apartment 3 and 6 are exceptions for the tests as they both did not have enough samples of

TABLE 6. Mean F1-score by training set size and input type for apartments 1, 2 and 3.

Apt. ID	Training samples	Sensors	K-means	BIRCH
1	200	0.50	0.75	0.76
	500	0.47	0.76	0.80
	1000	0.55	0.81	0.81
	2500	0.51	0.84	0.81
	5000	0.56	0.85	0.84
	7500	0.61	0.88	0.89
	10000	0.58	0.89	0.88
	15000	0.62	0.89	0.89
	20000	0.61	0.90	0.90
2	200	0.86	0.88	0.86
	500	0.89	0.86	0.87
	1000	0.89	0.89	0.90
	2500	0.89	0.91	0.90
	5000	0.91	0.93	0.93
	7500	0.93	0.95	0.94
	10000	0.93	0.95	0.95
	15000	0.93	0.96	0.96
	20000	0.94	0.96	0.96
3	200	0.79	0.68	0.64
	500	0.83	0.69	0.65
	1000	0.78	0.73	0.72
	2500	0.86	0.65	0.68
	5000	0.87	0.74	0.69
	7500	-	0.69	0.74
	10000	-	0.72	0.72
	15000	-	-	-
	20000	-	-	-

motion sensor events for the labeled data set due to technical errors, however, the results are clear also for these apartments.

The dependence of F1-score on training set size for each apartment is shown in Fig. 15a, 15b, 15c, 15d, 15e and 15f. At a first glance, we can see that the results are all in accordance with the findings in section V-B when it comes to comparative performance between clusters and binary sensor data. We can also observe that relatively little data is needed to create a well-performing model: most apartments achieve their peak F1-score with only 5000 samples. The trend is very similar for all apartments, namely that the performance reaches a plateau after an initial quick increase with training dataset size. The results for all apartments are summarized in 6 and 7.

Only slight improvement may be possible for larger datasets. The exception to this rule appears to be apartment 1 that may appear to still improve with larger dataset sizes. This can be due to the resident of this apartment having more unique movement patterns within their home, requiring more data samples for better activity recognition. Besides this, the high F1-score results for relatively modest dataset sizes can imply that the activity patterns are distinct enough, and that further performance improvement would require additional inputs to the model.

D. ACTIVITY RECOGNITION

In this section, we use confusion matrices to analyze the performance of the activity recognition in detail for two of the apartments – 1 and 3. In apartment 1, as in most apartments, K-means and BIRCH attain much higher F1-score than the

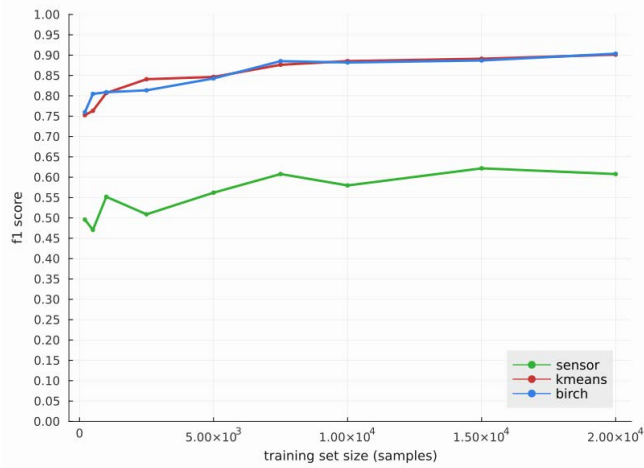
TABLE 7. Mean F1-score by training set size and input type for apartments 4, 5 and 6.

Apt. ID	Training samples	Sensors	K-means	BIRCH
4	200	0.82	0.87	0.87
	500	0.79	0.88	0.87
	1000	0.85	0.90	0.86
	2500	0.87	0.90	0.91
	5000	0.87	0.92	0.93
	7500	0.88	0.94	0.93
	10000	0.88	0.94	0.94
	15000	0.89	0.94	0.94
	20000	0.88	0.95	0.94
5	200	0.39	0.78	0.80
	500	0.39	0.81	0.83
	1000	0.43	0.83	0.82
	2500	0.46	0.87	0.88
	5000	0.46	0.87	0.89
	7500	0.45	0.89	0.89
	10000	0.47	0.90	0.90
	15000	0.47	0.90	0.91
	20000	0.47	0.91	0.91
6	200	0.76	0.80	0.78
	500	0.76	0.79	0.80
	1000	0.78	0.80	0.79
	2500	0.77	0.82	0.82
	5000	0.77	0.82	0.83
	7500	-	0.83	0.83
	10000	-	0.83	0.83
	15000	-	0.83	0.84
	20000	-	0.84	0.84

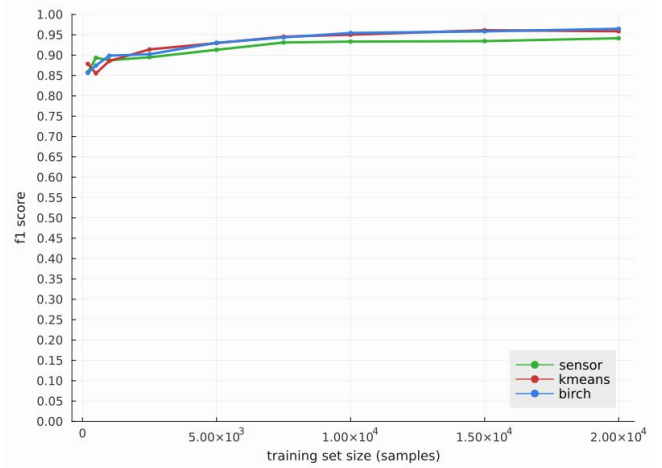
motion sensor dataset. In apartment 3 however, the highest F1-score is achieved by the motion sensor dataset.

Fig. 16a presents the confusion matrix of the motion sensor dataset of apartment 1. We can notice that only *walking* and *sleeping* can be well recognized with 78% and 93%. The other four activities are very much confused with *walking*. This is as expected since having only one sensor in each room, there is not enough information to distinguish between activities. K-means and BIRCH are however able to recognize these, as shown in Fig. 16b and 16c. Indeed, there is no significant confusion between activities in either of the algorithms. It is interesting to notice that BIRCH can recognize *watching-tv* 6% better than K-means, even though it has very few samples. In apartment 1, BIRCH derived 10 more clusters than K-means, and this has shown to be significant for recognizing this activity.

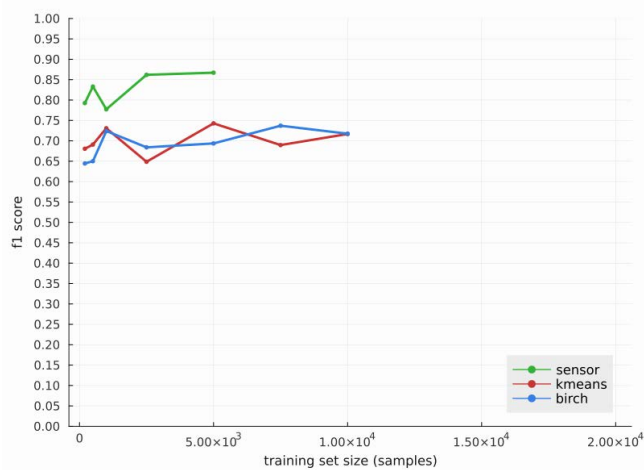
Fig. 17a, 17b and 17c show the confusion matrices for the activity recognition with the motion sensor, K-means and BIRCH datasets for apartment 3. As discussed earlier, this is the apartment where an exception occurred, the sensor dataset provided better results than K-means and BIRCH. We can notice from the confusion matrix with the motion sensor dataset there is no real confusion between classes. This resident does not move much around the apartment due to physical impairments. In addition, there is a lot of furniture, limiting the paths between rooms. As a result, the number of patterns in this apartment is limited, and the confusion matrices of the datasets indicate that a simpler sensor system, fewer sensors, would suffice in this case. Having more sensors or placing them in other positions than



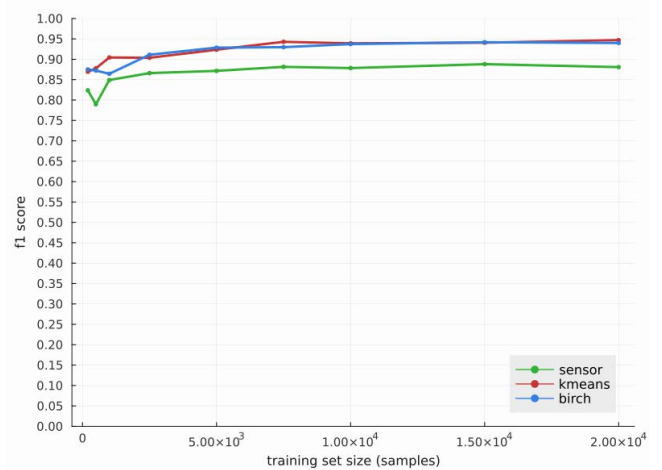
(a) Apartment 1.



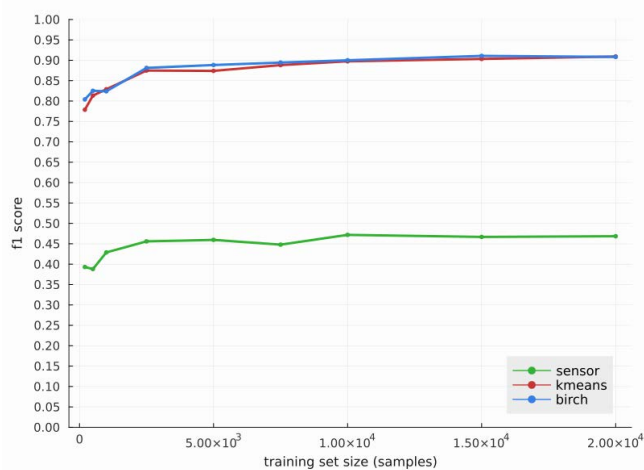
(b) Apartment 2.



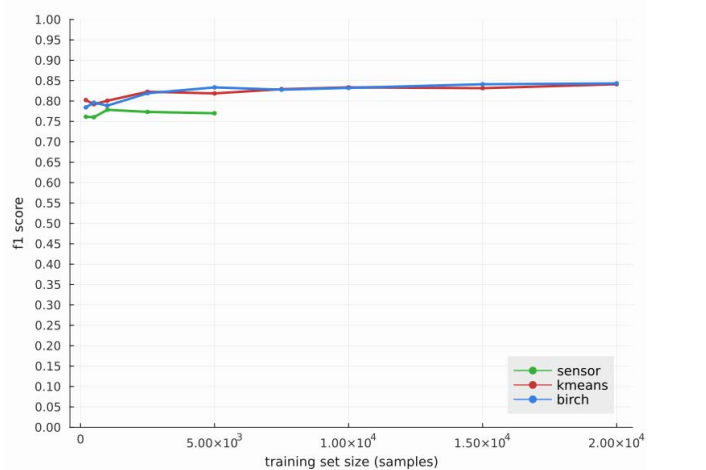
(c) Apartment 3.



(d) Apartment 4.



(e) Apartment 5.

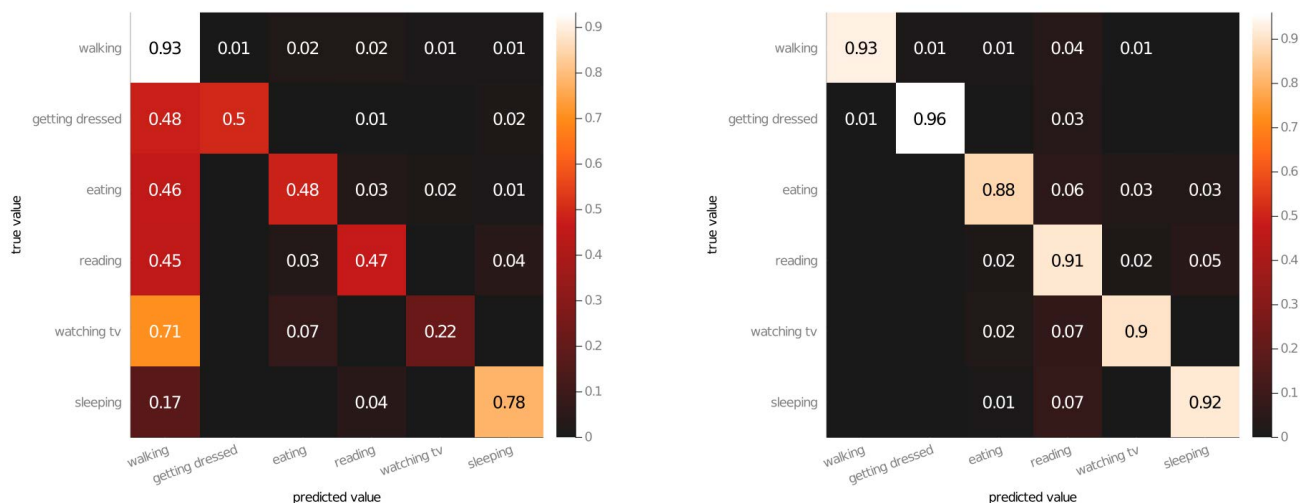


(f) Apartment 6.

FIGURE 15. F1-score of predictions of the activity vs. training set size for different apartments.

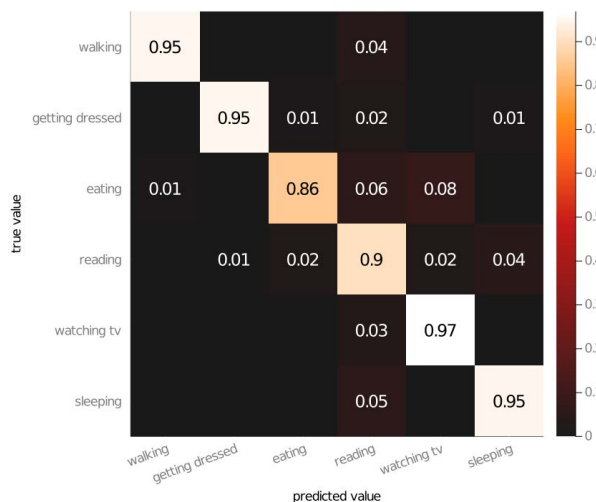
the empirical ones does not have a significant effect as the patterns seem to be simpler and fewer. We can also notice

that K-means and BIRCH are biased towards the class with most samples for this resident, *watching-tv* activity.



(a) Confusion matrix of motion sensors dataset for apartment 1, 20000 training samples.

(b) Confusion matrix of K-means dataset for apartment 1, 20000 training samples.



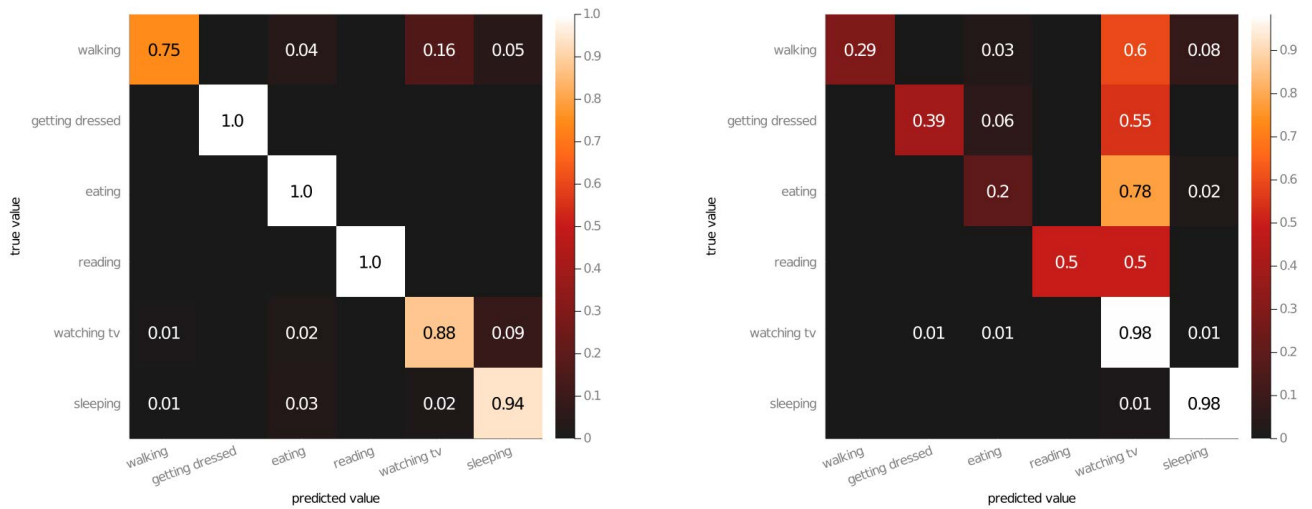
(c) Confusion matrix of BIRCH dataset for apartment 1, 20000 training samples.

FIGURE 16. Confusion matrices using different clustering algorithms for apartment 1 with 20000 training samples.

When looking at the different sequences generated for apartment 3 using the K-means and BIRCH methods for further explanations we found that a few sequences dominated the dataset. As previously mentioned, as a post-processing step of the clustering we grouped any cluster that we deemed similar enough. This was a manual step, and in the case of this particular apartment there seemed to be two clusters that were not grouped, but were still similar enough so that the resident just shifting position, created sequences that resulted in moving back and forth between the clusters. This resulted in losing vital information for activity recognition, and presumably explains why the sensor data, that did not have this problem, performed better. This illustrates the limitations of using a manual step after the clustering.

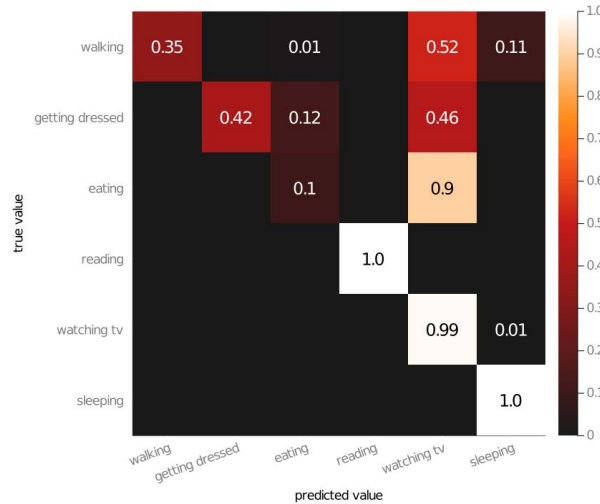
E. SUMMARY COMPARISON—CLUSTERS VS. SENSORS

Overall the results show that using the location clusters generates better results than the sensor event data, the F1-score improvement ranging from 0.02 (apartment 2) to 0.44 (apartment 5) for our data. Both memory length and dataset size tests indicate this discussed in detail in V-B and V-C. One exception sticks out, apartment 3, where the results are actually better for the installed sensors. As discussed, this is thought to be due to the physical limitations of this resident that lead to fewer and simpler motion patterns around the apartment. Having more sensors or in different locations than the empirical placement, does not improve, but rather decreases the F1-score as more inputs generate more complex patterns acting like noise when in reality the patterns are rather simple.



(a) Confusion matrix of motion sensors dataset for apartment 3, 5000 training samples.

(b) Confusion matrix of K-means dataset for apartment 3, 5000 training samples.



(c) Confusion matrix of BIRCH dataset for apartment 3, 5000 training samples.

FIGURE 17. Confusion matrices using different clustering algorithms for apartment 3 with 5000 training samples.

Among the different clustering methods the results are very similar in the end, where one has some small advantage over the other in some cases, but reversed in others. This is quite surprising as the clusters generated by the different methods could differ a great deal. Especially BIRCH resulted in more granularity. However, this also depends on the chosen activities of daily living and their granularity. For example, to the extent a certain location reflects a certain activity, keeping this location cluster, or equivalently placing a binary sensor at that position, will improve activity recognition and prediction. On the other hand, if this activity has not been defined in the list, then the cluster will not lead to better recognition or prediction. Hence the original selection of activities to monitor in the home may pre-determine the optimal granularity, and this in turn will affect which

clustering method will perform best. Increasing the number of sensors or clusters beyond what is reflected by the set of activities will not be manifested in the attained prediction F1-score but will only contribute to redundant data. Similarly, defining activities that are not possible to monitor by any sensor will create inaccuracies. By comparing the placement of K-means cluster centers in Fig. 8 and 7, with those of BIRCH in 10 and 9, it is evident that in locations for performing activities (bed for sleeping, kitchen for cooking, chair for watching TV, etc.) they share the exact same clusters. However, BIRCH adds more clusters for transition paths between these locations. Considering the small size of the apartments, and the defined set of activities, these added clusters do not contribute much to additional unique patterns, and thus do not increase the F1-score of

prediction of the next activity - in agreement with the results.

Our results show that longer memory lengths are favorable, however, the F1-score does not increase considerably beyond a memory length of 50, and in some cases we only need 15 previous clusters as input to reach the peak F1-score. This suggests that anything lower makes sequences that are not distinguishable enough to deduce the next activity.

VI. CONCLUSION

In this paper we describe a novel method for placing motion sensors in the home of older adults. Data is collected from six real apartments where seven motion sensors were placed empirically. We use depth video cameras, collect data in each apartment and apply clustering techniques – namely K-Means and BIRCH – to identify position clusters. The cluster centers indicate the positions where motion sensors should be placed in the apartment, based on real movement patterns. We evaluated the performance of this method by comparing the attained F1-score of activity recognition of the K-means and BIRCH datasets with that attained by the motion sensors dataset that then acted as a baseline. Our work sheds light to the optimum positioning of sensors.

K-means and BIRCH dataset presented significantly improved F1-score results for all apartments except one. Activity recognition was performed with F1-scores between 0.86 and 0.97 using K-Means and BIRCH, except in apartment 3 (0.75 for both clustering methods). In this apartment, the sensor dataset performed best (0.87). This case shows that our method might have some limitations in certain specific use cases.

Future work will include analyzing the optimal number of sensors, in addition to their optimal placement. Again, both the optimal placement and the optimal number of sensors will depend on the activities we aim at identifying. Finally improved results might be achieved by further focusing on the different steps of our method. For example, in addition to the traditional clustering algorithms employed in this study, other algorithms such as *spectral clustering*, could be explored in future work [45].

ACKNOWLEDGMENT

The authors wish to thank Oda Olsen Nedrejord and Wonho Lee, previously students at Oslo Metropolitan University, for designing and building the data collection system; the residents and the housekeepers at the seniors' care unit Skøyen Omsorg+; Dr. Torhild Holthe and Prof. Anne Lund (OsloMet) for recruiting participants for the field trial and communicating with the residents throughout; RoomMate AS for practical assistance; and the rest of the participants of the Assisted Living Project for a fruitful interdisciplinary collaboration.

REFERENCES

- [1] S. Blackman, C. Matlo, C. Bobrovitskiy, A. Waldoch, M. L. Fang, P. Jackson, A. Mihailidis, L. Nygard, A. Astell, and A. Sixsmith, "Ambient assisted living technologies for aging well: A scoping review," *J. Intell. Syst.*, vol. 25, no. 1, pp. 55–69, Mar. 2015.
- [2] D. J. Cook, M. Youngblood, E. O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja, "MavHome: An agent-based smart home," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2003, pp. 521–524.
- [3] L. Liu, E. Stroulia, J. Nikolaidis, A. Miguel-Cruz, and A. R. Rincon, "Smart Homes and home health monitoring technologies for older adults: A systematic review," *Int. J. Med. Inform.*, vol. 91, pp. 44–59, Jul. 2016.
- [4] B. Winblad, K. Palmer, M. Kivipelto, V. Jelic, L. Fratiglioni, L. O. Wahlund, A. Nordberg, L. Backman, M. Albert, O. Almkvist, and H. Arai, "Mild cognitive impairment—beyond controversies, towards a consensus: Report of the international working group on mild cognitive impairment," *J. Internal Med.*, vol. 256, no. 3, pp. 240–246, 2004.
- [5] V. Sanchez, C. Pfeiffer, and N.-O. Skeie, "A review of smart house analysis methods for assisting older people living alone," *J. Sensor Actuator Netw.*, vol. 6, no. 3, p. 11, Jul. 2017.
- [6] E. Zouganeli, F. D. Casagrande, T. Holthe, A. Lund, L. Halvorsrud, D. Karterud, A. Flakke-Johannessen, H. Lovett, S. K. Mørk, J. Strøm-Gundersen, E. Thorstensen, R. Norvoll, R. T. Meulen, M.-R. Kennedy, R. J. Owen, M. Ladikas, and E.-M. Forsberg, "Responsible development of self-learning assisted living technology for older adults with mild cognitive impairment or dementia," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. Ageing Well e-Health*, 2017, pp. 204–209.
- [7] A. Davoudi, M. T. Mardini, D. Nelson, F. Albinali, S. Ranka, P. Rashidi, and T. M. Manini, "The effect of sensor placement and number on physical activity recognition and energy expenditure estimation in older adults: Validation study," *JMIR mHealth uHealth*, vol. 9, no. 5, May 2021, Art. no. e23681.
- [8] F. Zenatti, D. Fontanelli, L. Palopoli, D. Macii, and P. Nazemzadeh, "Optimal placement of passive sensors for robot localisation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4586–4593.
- [9] A. R. Oliva, R. S. Estruch, and F. N. Akhi-clarab, "Optimal sensor placement for FDI using binary integer linear programming," in *Proc. Int. Workshop Princ. Diagnosis*, 2009, pp. 235–242.
- [10] R. Hou, Y. Xia, Q. Xia, and X. Zhou, "Genetic algorithm based optimal sensor placement for L₁-regularized damage detection," *Struct. Control Health Monitor.*, vol. 26, no. 1, pp. 1–14, 2019.
- [11] Y. Tan and L. Zhang, "Computational methodologies for optimal sensor placement in structural health monitoring: A review," *Struct. Health Monitor.*, vol. 19, no. 4, pp. 1287–1308, Jul. 2020.
- [12] C. Yang, "An adaptive sensor placement algorithm for structural health monitoring based on multi-objective iterative optimization using weight factor updating," *Mech. Syst. Signal Process.*, vol. 151, Apr. 2021, Art. no. 107363.
- [13] M. Civera, M. L. Pecorelli, R. Ceravolo, C. Surace, and L. Z. Fragonara, "A multi-objective genetic algorithm strategy for robust optimal sensor placement," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 9, pp. 1185–1202, Sep. 2021.
- [14] X. Bo, A. A. Razzaqi, and G. Farid, "A review on optimal placement of sensors for cooperative localization of AUVs," *J. Sensors*, vol. 2019, pp. 1–12, Jul. 2019.
- [15] A. N. Samudrala, M. H. Amini, S. Kar, and R. S. Blum, "Optimal sensor placement for topology identification in smart power grids," in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2019, pp. 1–6.
- [16] L. Atallah, B. Lo, R. King, and G.-Z. Yang, "Sensor placement for activity detection using wearable accelerometers," in *Proc. Int. Conf. Body Sensor Netw.*, Jun. 2010, pp. 24–29.
- [17] S. Mekruksavanich, A. Jitpattanakul, and P. Thongkum, "Placement effect of motion sensors for human activity recognition using LSTM network," in *Proc. Joint Int. Conf. Digit. Arts, Media Technol. Northern Sect. Conf. Electr., Electron., Comput. Telecommun. Eng. (ECTI)*, Mar. 2021, pp. 273–276.
- [18] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.
- [19] G. Demiris, M. J. Rantz, M. A. Aud, K. D. Marek, H. W. Tyrer, M. Skubic, and A. A. Hussam, "Older adults attitudes towards and perceptions of 'smart home' technologies: A pilot study," *Med. Informat. Internet Med.*, vol. 29, no. 2, pp. 87–94, Jun. 2004.
- [20] J. Twell, D. O'Sullivan, N. Maiden, J. Lockerbie, and S. Stumpf, "Monitoring meaningful activities using small low-cost devices in a smart home," *Pers. Ubiquitous Comput.*, vol. 23, no. 2, pp. 339–357, Apr. 2019.

- [21] M. P. Fanti, M. Roccotelli, J.-J. Lesage, and G. Faraut, "Motion detector placement optimization in smart Homes for inhabitant location tracking," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–6.
- [22] M. P. Fanti, M. Roccotelli, G. Faraut, and J.-J. Lesage, "Smart placement of motion sensors in a home environment," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 894–899.
- [23] S. Gholizadeh-Tayyar, R. B. Bachouch, Y. Fousseret, Y. Parmantier, and N. Ramdani, "Optimal sensor deployment according to a new approach for target tracking in smart Homes," *IRBM*, vol. 41, no. 6, pp. 321–330, Dec. 2020.
- [24] M. Shimosaka, O. Saisho, T. Sunakawa, H. Koyasu, K. Maeda, and R. Kawajiri, "ZigBee based wireless indoor localization with sensor placement optimization towards practical home sensing," *Adv. Robot.*, vol. 30, no. 5, pp. 315–325, Mar. 2016.
- [25] B. L. Thomas, A. S. Crandall, and D. J. Cook, "A genetic algorithm approach to motion sensor placement in smart environments," *J. Reliable Intell. Environ.*, vol. 2, no. 1, pp. 3–16, Apr. 2016.
- [26] A. S. Crandall, "Behaviometrics for multiple residents in a smart environment," Dept. Elect. Eng. Comput. Sci., Washington State Univ., 2011.
- [27] S. Nasrollahzadeh, M. Maadani, and M. A. Pourmina, "Optimal motion sensor placement in smart Homes and intelligent environments using a hybrid WOA-PSO algorithm," *J. Reliable Intell. Environ.*, vol. 8, pp. 345–357, Sep. 2021.
- [28] D. J. Cook and L. B. Holder, "Sensor selection to support practical use of health-monitoring smart environments," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 4, pp. 339–351, 2011.
- [29] M. Shimosaka, S. Masuda, K. Takeichi, R. Fukui, and T. Sato, "Health score prediction using low-invasive sensors," in *Proc. ACM Conf. Ubiquitous Comput.* York, NY, USA: Association for Computing Machinery, 2012, p. 1044.
- [30] *Roommate As*, Sandvika Norway. [Online]. Available: <https://www.roommate.no>
- [31] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
- [33] K. M. Bataineh, M. Najia, and M. Saqera, "A comparison study between various fuzzy clustering algorithms," *Jordan J. Mech. Ind. Eng.*, vol. 5, no. 4, pp. 335–343, 2011.
- [34] K. D. Joshi and P. S. Nalwade, "Modified K-means for better initial cluster centres," *Int. J. Comput. Sci. Mobile Comput.*, vol. 2, pp. 219–223, Jul. 2013.
- [35] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM SIGMOD Int. Conf. Manag. Data.* York, NY, USA: Association for Computing Machinery, 1996, pp. 103–114.
- [36] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.
- [37] J. Martens, "Generating text with recurrent neural networks," *Neural Netw.*, vol. 131, no. 1, pp. 1017–1024, 2011.
- [38] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, May 2013, pp. 6645–6649.
- [39] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, no. 1, pp. 115–143, 2003.
- [40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [41] A. Graves, "Generating sequences with recurrent neural networks," 2014, *arXiv:1308.0850*.
- [42] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994.
- [43] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
- [44] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5, 2017.
- [45] H. Jia, S. Ding, X. Xu, and R. Nie, "The latest research progress on spectral clustering," *Neural Comput. Appl.*, vol. 24, nos. 7–8, pp. 1477–1486, 2014.



SIMON SIMONSSON was born in Stockholm, Sweden, in 1987. He received the M.Sc. degree in computer science from Kungliga Tekniska Högskolan (KTH), Stockholm, in 2019. Since then he has participated in research projects within assisted living using machine learning and computer vision at OsloMet–Oslo Metropolitan University, Norway. Since 2020, he has been working as a Data Scientist with the Norwegian Postal Service. His research interests include machine learning and computer vision, automated systems, and robotics.



FLÁVIA DIAS CASAGRANDE was born in Vitoria, Espírito Santo, Brazil, in 1990. She received the B.S. degree in electrical engineering from the Federal University of Espírito Santo, Brazil, in 2013, the joint M.S. degree in computer vision and robotics (Erasmus Mundus joint programme) from the University of Burgundy, France, Heriot-Watt University, Scotland, and the University of Girona, Spain, in 2016, and the Ph.D. degree in informatics from the University of Oslo, Norway, in 2019. She is an externally employed at OsloMet–Oslo Metropolitan University, Norway. Her research interests include machine learning and computer vision, with a focus on medicine, assisted living, and health applications.



EVI ZOUGANELI (Member, IEEE) received the B.Sc. degree in applied physics from the University of Patras, Greece, the M.Sc. degree in telecommunications engineering and the Ph.D. degree in semiconductor nanodevices from University College London, U.K. (on Greek State Scholarship, IKY), and the master's degree in management from the Norwegian School of Management (BI). Since 2008, her research has gravitated toward machine learning and artificial intelligence, first for the Internet of Things, then for medical informatics at the Norwegian Radium Hospital, and later for assistive technologies at Oslo Metropolitan University, where she heads the research group on automation, robotics, and intelligent systems, ARIS. Since 2020, she has been a Full Professor with Oslo Metropolitan University. She has also done consultant work on network technology strategy, roadmap, and new business opportunities. After a Postdoctoral Researcher at ETH Zürich, she joined Telenor Research and Development, Norway, where she was active in a series of European research projects, among others in broadband networks, optical switching, and the Internet of Things, and European Networks of Excellence. Her research interests include architectures and algorithms for intelligent systems, and AI-enabled robotics, for a range of applications. She is a Board Member of the Norwegian Artificial Intelligence Society (NAIS).