IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 18, 2018, accepted July 24, 2018, date of publication July 31, 2018, date of current version August 20, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2861004*

# A Scalable Hybrid Activity Recognition Approach for Intelligent Environments

**GORKA AZKUNE AND AITOR ALMEIDA**
DeustoTech–Deusto Institute of Technology, University of Deusto, 48007 Bilbao, Spain
Corresponding author: Gorka Azkune (gorka.azkune@deusto.es)

This work was supported by the H2020 Project City4Age, under Grant 689731.

**ABSTRACT** Human activity recognition is a key technology for ICT-based (infomation and communication technologies) assistive applications. The most successful activity recognition systems for intelligent environments in terms of performance rely on supervised learning techniques. However, those techniques demand large labelled data sets for specific sensor deployments and monitored person. Such requirements make supervised learning techniques not to scale well to real world deployments, where different sensor infrastructures may be used to monitor different users. In this paper, we present a novel activity recognition system, based on a combination of unsupervised learning techniques and knowledge-based activity models. First, we use a domain-specific data mining algorithm previously developed by Cook *et al.* to extract the most frequent action sequences executed by a person. Second, we insert knowledge-based activity models in a novel matching algorithm with the aim of inferring what activities are being performed in a given action sequence. The approach results on a scalable activity recognition system, which has been tested on three real data sets. The obtained performance is comparable to supervised learning techniques.

**INDEX TERMS** Intelligent environments, activity recognition, knowledge-based models, unlabelled datasets, unsupervised learning.

## I. INTRODUCTION

One of the objectives of public administrations is to promote active and healthy ageing, due to the positive consequences derived for the society and socio-sanitary services. In that sense, prevention has been identified as a key concept under the healthy ageing paradigm. The early detection of risks relating to a specific health condition improves the chances of planning suitable interventions that can stop or at least delay the condition itself, with beneficial effects on both patients' quality of life and costs of treatment [1].

Recent important technology developments may offer new ways to achieve a systematic health monitoring and early health risk detection approach. A good example is the *City4Age* project,[1] where MCI and frailty risks of elderly citizens want to be mitigated using unobtrusive technologies. The core idea of the project is to use urban infrastructures to monitor people's activity and behaviour and correlate their evolution with MCI and frailty, in order to plan suitable ICT-enabled interventions to minimise the consequences of those risks. Hence, automatic human activity recognition (AR) becomes a key enabler of the project.

[1]http://www.city4ageproject.eu/

Nowadays the most successful AR systems are based on supervised learning techniques (see Related Work in Section II). Those learning techniques rely on large annotated datasets of sensor information. However, as Chen *et al.* [2] noted, such approaches are not scalable. Notice that in the context of this paper, scalability is defined as the amount of work needed to deploy an algorithmic solution in a new environment with a new user. In the case of supervised learning, whenever the sensor infrastructure and/or user changes, a period of data collection and annotation has to be faced, resulting in poor scalability, according to the adopted definition.

In this paper we present a scalable and hybrid AR system called HARS (*Hybrid Activity Recognition System*), which has been developed inside the City4Age project. We call the AR system *scalable* because it avoids data collection and annotation phases, requiring only some initial activity modelling effort. We call the AR system *hybrid* because it combines data- and knowledge-driven approaches, which are reviewed in Section II. In a previous work we have already used this combination of both paradigms to solve other problems, for example, the creation of the models that describe activities [3]. In this paper, we focus on activity recognition, presenting novel insights, algorithms and experiments. The key idea is to use data-mining techniques to find the most frequent action patterns in the unlabelled dataset produced after monitoring a person's activity in an intelligent environment. Those patterns reflect specific executions of activities.

In order to know what activities are being performed in a given action pattern, we use Expert Activity Models (EAM). EAMs are simple knowledge-based computational models where the previous knowledge about target activities is represented. The spirit of EAMs is not to have a detailed activity model for a given person, but rather to represent a generic activity with minimum knowledge (Section III-B). One of the main advantages of EAMs is their simplicity, which allows any domain expert to model them easily. Furthermore, simplicity also makes EAMs flexible enough to be applied to varying activity executions. In the context of our research, the activities to be monitored are known beforehand by the domain experts. For example, Gold shows in [4] the relationship of Lawnton Instrumental Activities of Daily Living (IADL) with MCI assessment and its value in predicting cognitive decline. Some examples of those activities are using the telephone, preparing food or housekeeping. We assume that a domain expert will provide the activities to be detected as well as the EAMs that roughly describe how those activities are performed by monitored users.

With the purpose of discovering the activities for a given action pattern and a set of EAMs, we have developed a pattern-model matching algorithm. This algorithm is posed as a maximization problem, where the objective is to find the set of EAMs that better explains the given action pattern. We use actions, locations, duration and starting time to address the maximisation problem (Section III).

We tested the proposed approach on three real activity datasets collected in different intelligent environments obtaining comparable results to supervised learning techniques, such as Hidden Markov Models and Naive Bayes (Section IV and V).

As a result of the research pursued, there are two main scientific contributions in this paper:

1) A novel way to combine unsupervised learning techniques with knowledge-based models for AR systems, which solves the scalability problems of supervised learning techniques with comparable performance.
2) An algorithm to match action patterns with expert provided knowledge-based activity models in order to infer the activities being performed in such patterns.

## II. RELATED WORK

There are two main monitoring approaches for human AR; namely, vision-based and sensor-based monitoring. For a review of vision-based approaches, Weinland *et al.* [5] can be consulted. When human AR is targeted in intelligent environments, sensor-based AR is the most used solution [2], since vision-based approaches tend to generate privacy concerns among the users [6]. Sensor-based approaches are based on the use of emerging sensor network technologies for activity monitoring. The generated sensor data from sensor-based monitoring are mainly time series of state changes and/or various parameter values that are usually processed through data fusion, probabilistic or statistical analysis methods and formal knowledge technologies for AR. There are two main approaches for sensor-based AR in the literature: data-driven and knowledge-driven approaches. In the last years, a third approach has emerged, namely the hybrid approaches. Our work belongs to that third category.

### A. DATA-DRIVEN APPROACHES

The idea behind data-driven approaches is to use data mining and machine learning techniques to learn activity models. It is usually presented as a supervised learning approach, where different techniques have been used to learn activities from collected sensor data. Data-driven approaches need big datasets of labelled activities to train different kinds of classifiers. The learning techniques used in the literature are broad, going from simple Naive Bayes classifiers [7]–[12] to Hidden Markov Models [13]–[15], Dynamic Bayesian Networks [16], [17], Support Vector Machines [18], online (or incremental) classifiers [19] and dictionaries of patterns [20].

Although supervised learning reports excellent performance, the need of large-scale labelled datasets produces scalability problems for practical deployments. It seems unfeasible to obtain enough labelled data for real world scenarios, since the involved users and activities may be too numerous.

However, there are some efforts in the community directed to solve this problem through unsupervised learning techniques. For example, for smartphone-based AR, examples like [21] can be found. For intelligent environments, where the work of this paper is situated, a few unsupervised methods have been proposed to tackle the data annotation problem, such as the frequent sensor mining method [22], simultaneous frequent-periodic pattern mining method [23], episode discovery [24], activity modelling based on low dimensional Eigenspaces [25], probabilistic models [26], [27] and retrieving activity definitions using Web mining [28]. Although these methods address the data annotation problem, they consider a simplified version of the problem by ignoring the real world nature of data such as its sequential form, possible disruptions (a phone call while cooking), or variation of the same pattern.

Rashidi and Cook [29] tried to overcome the enumerated problems. They use an unlabelled dataset, where they extract activity clusters using unsupervised learning techniques. Those clusters are used to train a boosted Hidden Markov Model, which is shown to be able to recognise several activities. However, there are three fundamental downsides in this approach: (i) the modelled and recognised activities do not have any semantic meaning which makes harder for humans to understand what a user is actually doing, (ii) activity granularity, since the clusters found may refer to chaining activities such as washing dishes after having lunch, as only one activity, and (iii) the performance of current systems are still far from supervised learning approaches.

### B. KNOWLEDGE-DRIVEN APPROACHES

In order to maintain the scalability of unsupervised learning approaches, but overcome the downsides mentioned, we take ideas from knowledge-driven AR approaches. Knowledge-driven AR is based on real world observations that the list of objects and functionalities to perform an activity are always very similar. For example, to prepare coffee, a liquid container is needed alongside with some coffee and sugar. Although different people may use different coffee brands, some may add milk and some may prefer white sugar to brown sugar, there are some essential concepts that are always present for every activity. The idea is to use this prior knowledge to create rough activity models. The implicit relationships between activities, related temporal and spatial context and the entities involved (objects and people) provide a diversity of hints and heuristics for inferring activities.

The first step for knowledge-driven systems is to acquire the needed contextual knowledge. This is usually achieved using standard knowledge engineering approaches. Depending on the nature of the acquired knowledge, different approaches can be distinguished. Some researchers use logic-based approaches for AR, as Chen and Nugent [30]. Others adopt ontology-based approaches which allow a commonly agreed explicit representation of activity definitions independent of algorithmic choices, thus facilitating portability, interoperability and reusability. Good examples can be found in [31] and [32]. In a very recent work, Noor *et al.* [33] use Dempster-Shafer theory to combine uncertainty reasoning and ontologic reasoning. However, their system has only been evaluated in controlled laboratory experiments.

### C. HYBRID APPROACHES

With the aim of taking advantage of the positive features of data- and knowledge-driven approaches, hybrid approaches for AR have emerged recently. Representative methods have been presented by Riboni *et al.* [34] and Ye *et al.* [35]. Both systems combine data- and knowledge-driven approaches in a different way. For instance, Riboni *et al.* [34] produce very detailed ontologic models based on Web Ontologic Language (OWL2) to later map the knowledge to Markov Logic Networks (MLN), which allows them to use probabilistic reasoning. Their approach requires very detailed activity models which compromise the generality of the system. Furthermore, their evaluation is performed on datasets with low number of activities (8 at most) and they do not address the *idle* activity, i.e., time segments where no activities occur even though sensor events appear.

On the other hand, Ye *et al.* [35] present an AR system called USMART. They segment sensor events using the semantic similarity between the fired sensors. Based on previously modelled ontologic activity models they define the sufficient conditions for a sensor sequence to be mapped to an activity. Those activity models are very similar to the EAMs we use. However, contrary to HARS, their approach also requires knowledge-based models for the target environments and sensors in order to work properly. Once sensor segments are extracted, they use semantic reasoning to recognise activities. This imposes some burdens in their recognition capacities. As authors admit, they need at least a sensor event that uniquely describes an activity, in order to distinguish it from others. The evaluation they present does not consider the *idle* activity, whereas our approach does.

The AR system we present in this paper is another example of a hybrid approach and can address the enumerated problems for the two similar approaches explained before [34] and [35]. First of all, in contrast with Ye *et al.* [35], we do not segment sensors depending on their semantic similarity, but depending on the most frequent patterns executed by the monitored person, following the work of Cook *et al.* [36]. This has three advantages: (i) we avoid the detailed sensor modelling process, (ii) the approach can generalise better to other scenarios, and (iii) we can handle *idle* activities and sensor noise better. Second, we do not use semantic or probabilistic reasoning, but a novel matching algorithm which can handle interleaved activities naturally, can detect *idle* activities and offers a flexible framework where two activities composed by the same sensor events can be distinguished. Third, our approach demands very simple activity models, improving the generality and scalability of the system.

## III. DESIGN AND DEVELOPMENT OF THE HYBRID ACTIVITY RECOGNITION SYSTEM (HARS)

### A. SCOPE OF THE APPROACH

The AR system presented in this paper, called HARS, is designed to work in dense sensing-based monitoring scenarios [2], where activities are inferred by monitoring human-object interactions through the usage of multiple multi-modal miniaturised sensors.

Let us formally define the most important concepts we use through the paper:

*Definition 1 (Sensor Activation):* A sensor activation occurs when a sensor changes its state from the no-interaction state to interaction state and vice-versa. A sensor activation ($sa$) is represented by the sensor name or ID, the timestamp and the state: $sa = \{timestamp, sensor, state\}$.

*Definition 2 (Action):* Actions are short-timed conscious muscular movements and constitute the primitives of activities. Actions are detected by sensors. Hence, sensor activations can be directly mapped to actions $a = f_{map}(sa)$. An action ($a$) is represented by its name and timestamp: $a = \{timestamp, action\_name\}$.

*Definition 3 (Activity):* An activity is composed of a sequence of actions which are executed with a common purpose in a specific location, at a given time and with a given duration. An activity ($A$) is represented by a sequence of actions ($[a]$), a starting and end time and a location: $A = \{start\_time, end\_time, [a], location\}$.

*Definition 4 (Expert Activity Model (EAM)):* An EAM is a simple and rough activity model which encapsulates the prior knowledge of how an activity is performed in terms of the executed actions ($\{a\}$), typical duration ($d$), possible starting times ($\{st\}$) and locations ($\{l\}$): $EAM = \{\{a\}, d, \{st\}, \{l\}\}$.

In dense sensing-based activity monitoring an action of a user interacting with an object is detected through the sensor attached to the object, as defined in Definition 2. As such, the activation of a sensor implies that an action has been taken and hence, sensor activations can be mapped to actions. In this sense, HARS is best suited for intelligent environments which use binary sensors. A state change of a binary sensor can be directly mapped to an action. However, analogous sensors could also be used with HARS, for example, using some thresholds to distinguish sensor activations, as suggested by van Kasteren *et al.* [37].

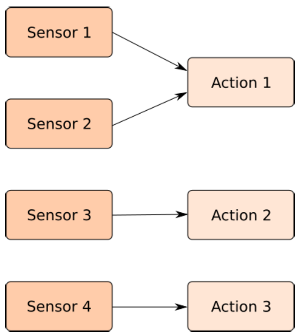


**FIGURE 1. Sensor-action mappings: several sensor activations can be mapped to the same action given that they share the same semantics.**

Mapping sensor activations to actions allow a higher level of abstraction and modelling capabilities, since several sensor types may map to the same action (see Figure 1). For example, we might add different sensors to saucepans and casseroles to monitor their usage. However, the described action for both sensor types might be 'use cooking utensil', since we are interested on knowing whether a person uses them for cooking purposes. That is why HARS works in the action space rather than in the sensor space.

On the other hand, we consider single user - concurrent activities scenarios, i.e. only one user is being monitored, but activities can be executed both sequentially and concurrently. For example, a user might start cooking and meanwhile, go to the toilet. Another feature to be taken into account is that HARS is an offline recognition system. We describe its deployment and usage in Section III-C.

### B. DETAILED DESCRIPTION OF THE APPROACH

As introduced in Section I, HARS is divided into two main steps:

1) Action Pattern Discovery (APD): the objective of this module is to extract frequent action sequences which represent specific executions of activities. Before mining for frequent action sequences, unlabelled sensor activations are mapped to actions, for which sensor-action mappings are used as input.
2) Pattern-Model Matching (PMM): using EAMs as input, this module aims at matching the best set of EAMs to every action pattern discovered by APD. It might be the case that an action pattern is composed by several activities, only one activity or 'None' (a label used for *idle* activities and unknown activities, i.e. those activities that do not have any EAM). The output of PMM is a set of detected activities for every action pattern.

Figure 2 depicts the system diagram, where the two main software components of HARS can be seen with the required input files.

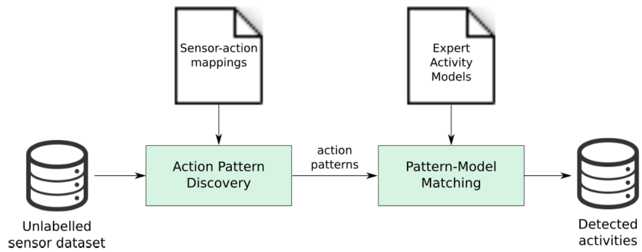


**FIGURE 2. The system diagram of HARS.**

*Action Pattern Discovery:*

- Input data: (i) unlabelled dataset of sensor activations for a given user; (ii) sensor-action mappings.
- Input parameters: the number of iterations for the pattern mining algorithm (HARS uses the option to repeat until no new patterns can be found to compress the data).
- Output: a labelled dataset of actions, where every action is tagged with the pattern number found by the pattern mining algorithm.

For the first software component, i.e. APD, we use the software developed and published by Cook *et al.* [36] in their web page.[2] This approach builds on previous research on pattern discovery, including methods for mining frequent sequences [38], [39], mining frequent patterns using regular expressions [26], constraint-based mining [40], mining frequent temporal relationships [41], and frequent-periodic pattern mining [24]. However, activity discovery for dense sensing-based monitoring approaches implies certain requirements which cannot be addressed by those former algorithms: (i) input data in an intelligent environment is composed by just a single stream of sensor events with no clear boundaries between different activities, and (ii) activity patterns can be discontinuous, varied order, and of arbitrary length. Thus, for activity discovery a specially designed pattern mining algorithm is better suited, as proposed by Cook *et al.* [36]. They developed an unsupervised learning algorithm to extract the most frequent sensor patterns from a dataset. Their idea is to search the space of sensor event sequences in order to find those sequences that maximise the compression of the original dataset. For that purpose, they developed a greedy search approach, with the special feature

[2] http://ailab.wsu.edu/casas/tools.html

of using the edit distance to compute the similarity of sensor sequences. This is motivated by the fact that human activities are rarely performed the same way. Hence, small variations in the sequence should be detected as the same pattern or activity. Their algorithm can be used to find as many patterns as possible, using as input an unlabelled dataset. Those sensor events that do not pertain to any pattern are directly tagged as 'None' activities. The logic behind this decision is that if a sensor sequence is not frequently executed, it may not be a representation of an activity.

HARS runs the pattern mining algorithm in the action space, rather than in the sensor space. The result is a new dataset where actions are annotated with discovered pattern numbers.

*Pattern-Model Matching:*

- Input data: (i) a labelled dataset of actions, where every action is tagged with a pattern number (output of APD); (ii) a set of EAMs for a given user.
- Input parameters: weights for the cost function ($w_i \in \mathbb{R}$): (i) action weight ($w_a$), (ii) duration weight ($w_d$), (iii) starting time weight ($w_s$), and (iv) location weight ($w_l$).
- Output: a labelled dataset of actions, where every action is tagged with the detected activities.

Once the patterns have been discovered, the PMM algorithm is applied. PMM uses as inputs an action pattern and the set of defined EAMs. An EAM is defined as a computational model of activities which contains:

1) Actions: the minimum number of actions that are usually executed to perform a given activity.
2) Duration: a rough estimation of the typical duration of a given activity.
3) Starting Time: approximate time ranges when a given activity is usually started (multiple time ranges are supported).
4) Locations: semantic tags for the places where a given activity is usually performed, for example: bathroom or bedroom (multiple locations for an activity are supported).

```
"MakeCoffee": {
  "actions": ["hasContainer", "hasCoffee"],
  "duration": 300,
  "start": [[7:00-10:00], [13:00-15:00]],
  "locations": ["Kitchen"]
}
```

**FIGURE 3. Example of an EAM for hypothetical activity MakeCoffee.**

Figure 3 shows an example of an EAM for a hypothetical activity called MakeCoffee. As can be seen, EAMs are very simple activity models, which is very important to assure the scalability of HARS. So simplicity is a virtue from the scalability point of view, rather than a disadvantage. EAM simplicity allows domain experts to easily design descriptive models, without the need of using complex techniques which involve more detailed knowledge, greater effort and heavier processing requirements. We have to stress that it is very important to reduce the modelling effort for real world deployments since the number of users and the variability of activities may be too high to model manually. Notice also that EAMs allow modelling diverse combinations of starting time and locations for an activity, improving the flexibility of activity models.

We assume that EAMs are provided by domain experts, based on their knowledge about the users, activities and monitoring systems. For the experiments carried out in this paper, we define EAMs manually in a JSON file (JavaScript Object Notation), which the software of HARS can read at execution time. However, in the City4Age project, for example, we provide domain experts a REST API endpoint where they can upload the EAMs to a shared database. In that case, HARS retrieves the EAMs from a relational database. In any case, to define an EAM experts have to provide a JSON file following the format shown in Figure 3. Although convenient applications can be developed to make this process easier for experts, this is out of the scope of this paper.

To assess the activities that best describe a given action pattern, PMM poses a maximisation problem. The objective of the algorithm is to find the set of EAMs that maximises the cost function of Equation 1.

$$\theta = w_a A(P, EAM) + w_d D(P, EAM) + w_s S(P, EAM) + w_l L(P, EAM) \quad (1)$$

Equation 1 computes the suitability of EAMs using the four concepts that are represented in EAMs: actions, duration, starting time and locations (see Figure 3). We define a suitability function for each of the concepts, which outputs a value between $[-1, 1]$, being 1 maximum suitability. For instance, $A(P, EAM)$ computes the action suitability, where $P = \{a_0, a_1, \ldots a_n\}$ is the action pattern and $EAM = \{EAM_0, \ldots EAM_m\}$ is a set of EAMs. We use several weights $w_i$ to assess the weight of a suitability function in the cost function. In the following, we describe each of the suitability functions.

### 1) ACTION SUITABILITY

$A(P, EAM)$ is calculated as the number of actions shared between the set of EAMs and the pattern, relative to the total number of actions (Equation 2). Shared actions contribute positively to the score, while actions of EAMs that are not in the pattern, contribute negatively. $A(P, EAM) = -1$ means that there are no shared actions between the pattern and the set of EAMs, whereas a value of 1 indicates that the correspondence between pattern actions and EAMs is perfect.

$$A(P, EAM) = \frac{|A_{EAM} \cap A_P|}{|A_P|} - \frac{|A_{EAM} - (A_{EAM} \cap A_P)|}{|A_{EAM}|} \quad (2)$$

where $A_{EAM}$ refers to the actions in the set of EAMs and $A_P$ to the actions of $P$.

### 2) DURATION SUITABILITY

$D(P, EAM)$ is defined in Equation 3. The idea is that the result is 1 if the sum of the duration of each EAM of the set equals the actual duration of the pattern, decreasing linearly the score, as both duration values get further. This responds to the intuition that an action pattern composed by several activities will have a longer duration than any individual EAM. Hence, to detect the existence of various activities, the sum of the estimated duration of every EAM in the set has to be computed.

$$D(P, EAM) = max\{-1, 1 - k|Duration(P) - D_{EAM}|\} \tag{3}$$

where $k$ is an external parameter to define the slope of the line and $D_{EAM} = \sum_i Duration(EAM_i)$, i.e. the duration of a set of EAMs is the sum of the duration estimation of its EAMs.

#### 3) STARTING TIME SUITABILITY

$S(P, EAM)$: to compute the suitability of the starting time, we use Equation 4. The average of the suitability of every EAM of the set is calculated. As we can handle more than one activity per detected pattern, we use the average suitability of each of the EAMs in a given set. This means that incorporating an EAM whose starting time is far from the actual pattern's starting time, will decrease the suitability value.

$$S(P, EAM) = \frac{1}{N}\sum_{i}^{N} s_p(P, EAM_i) \tag{4}$$

where $s_p(P, EAM_i)$ is the function that computes the starting time suitability of an individual EAM relative to a pattern (see Equation 5).

$$s_p(P, EAM_i) = \begin{cases} 1 & \text{if } P_{start} \in [EAM_{start}, EAM_{end}] \\ max\{-1, \frac{m}{\Delta_s(EAM_i, P)} - b\} \\ \quad \text{else} \end{cases} \tag{5}$$

The function $\Delta_s$ $(EAM_i, P)$ calculates the time-distance between the start of the pattern and the starting range of the EAM. Thus, an individual EAM is suitable for a pattern, if the starting time of the pattern is inside of one of the starting ranges of the EAM. As the starting time of the pattern gets further from the EAM range, the score decreases linearly, for which parameters $m$ and $b$ have to be defined.

#### 4) LOCATION SUITABILITY

$L(P, EAM)$: Equation 6 shows the location suitability. It is defined as the action suitability function, but using locations instead of actions. In order to know where an action has been executed, we use the location of the sensor in the environment, which is well-known at execution time.

$$L(P, EAM) = \frac{|L_{EAM} \cap L_P|}{|L_P|} - \frac{|L_{EAM} - (L_{EAM} \cap L_P)|}{|L_{EAM}|} \tag{6}$$

Those suitability functions are designed to be generic, trying to capture the knowledge we have about how humans tend to perform activities. The concepts of actions, locations, duration and starting time are generic to any AR problem. Thus the cost function in Equation 1 can be applied to any scenario and in principle, although suitability functions can be changed, they are also thought to work in diverse scenarios. Indeed, Section IV shows the generality of these functions, which are used unchanged for three different datasets. However, suitability functions are used in Equation 1 with respective weight values ($w_a$, $w_d$, $w_s$ and $w_l$), which have to be set before running HARS. Notice that weight values represent the relative importance of the concepts described in EAMs for activity recognition. Having a labelled dataset would offer a chance to apply optimisation algorithms to tune those weight values, for example. But given that HARS is designed for unlabelled datasets there is no way to set those values automatically. Hence, they have to be provided by domain experts, depending on the monitoring approach of the intelligent environment and the nature of the target activities. Domain experts have to provide the prior knowledge to assess the relative importance of actions, duration, starting time and location for the activities they want to recognise.

Once we have described the suitability functions, let us explain how we approach the maximisation of Equation 1. When solving the maximisation problem, actions play a capital role. Assume the number of defined EAMs for a given dataset is $N$. In the beginning of the PMM algorithm, we calculate all the possible combinations of those $N$ EAMs for $k \in [0, N]$. So the total number of EAM sets is given by Equation 7:

$$|EAM| = \sum_{k=0}^{k=N} \frac{N!}{k!(N-k)!} \tag{7}$$

The number of EAM combinations grows exponentially, thus for big numbers of $N$, checking all the possible combinations for maximisation is not feasible. For that reason, when calculating the value of the cost function we do not consider the EAM sets that do not share any action with a given pattern. This heuristic is derived from the fact that even though all the other parameters of an EAM may fit well in a given action pattern, if no action is shared, the EAM cannot explain properly the action pattern. This pruning strategy reduces the number of combinations considerably, requiring no more optimisation strategies.

Hence, the maximisation problem computes the score for all valid sets of EAMs and chooses the set which obtains the maximum score as the set which better explains a given pattern. We further refine the obtained results with a filtering heuristic. This heuristic labels as 'None' those action sequences which produce more activities than actions. For some cases, PMM finds that a sequence of three actions can be explained with four activities, for instance. Those cases are clearly the result of trying to explain something that cannot be explained with the current EAMs. Thus, we apply the filtering heuristic and PMM automatically labels those sequences as 'None'.

### C. HARS DEPLOYMENT AND USAGE EXAMPLE

HARS is designed to be executed periodically offline. The idea is to recognise activities that are already executed by

**TABLE 1.** An extract of the dataset used to describe how HARS works.

| Timestamp | Sensor Activation | Action | APD | PMM | Ground truth |
|---|---|---|---|---|---|
| 00:20:14 | HallBedroomDoor | HallBedroomDoor | Pat2 | GoToBed | None |
| 09:33:41 | HallBedroomDoor | HallBedroomDoor | Pat2 | GoToBed | GoToBed |
| 09:33:47 | HallBedroomDoor | HallBedroomDoor | Pat2 | GoToBed | GoToBed |
| 09:36:43 | HallToiletDoor | HallToiletDoor | None | None | None |
| 09:37:20 | HallBathroomDoor | HallBathroomDoor | Pat6 | UseToilet | UseToilet |
| 09:37:51 | ToiletFlush | ToiletFlush | Pat6 | UseToilet | UseToilet |
| 09:37:55 | ToiletFlush | ToiletFlush | Pat6 | UseToilet | UseToilet |
| 09:37:58 | HallBathroomDoor | HallBathroomDoor | Pat6 | UseToilet | UseToilet |
| 09:49:27 | PlatesCupboard | PlatesCupboard | Pat32 | PrepareBreakfast | PrepareBreakfast |
| 09:49:31 | PlatesCupboard | PlatesCupboard | Pat32 | PrepareBreakfast | PrepareBreakfast |
| 09:49:39 | Fridge | Fridge | Pat32 | PrepareBreakfast | PrepareBreakfast |
| 09:49:53 | Fridge | Fridge | Pat32 | PrepareBreakfast | PrepareBreakfast |
| 09:49:58 | Microwave | Microwave | Pat94 | PrepareBreakfast | PrepareBreakfast |
| 09:50:39 | GroceriesCupboard | GroceriesCupboard | Pat94 | PrepareBreakfast | PrepareBreakfast |
| 09:52:11 | Microwave | Microwave | Pat94 | PrepareBreakfast | PrepareBreakfast |
| 09:53:22 | GroceriesCupboard | GroceriesCupboard | Pat94 | PrepareBreakfast | PrepareBreakfast |

**TABLE 2.** A summary of the EAMs used to illustrate how HARS works.

| Activity | Actions | Locations | Starting time | Duration |
|---|---|---|---|---|
| GoToBed | HallBedroomDoor | Bedroom | 22:00 - 10:30 | 28800 |
| UseToilet | ToiletFlush, HallBathroomDoor, HallToiletDoor | Bathroom | 00:00 - 23:59 | 60 |
| PrepareBreakfast | PlatesCupboard, Fridge, GroceriesCupboard, Freezer | Kitchen | 8:00 - 10:30 | 300 |
| PrepareDinner | PlatesCupboard, Freezer, Fridge, PansCupboard, GroceriesCupboard | Kitchen | 18:00 - 23:30 | 300 |

a given user, once all data has been collected. For example, in the City4Age project, we are executing HARS once per week for every user. Thus, frequent action patterns are searched for a week-time and afterwards recognised using the PMM algorithm. This allows us to adapt our recognition algorithm to changing behaviour, i.e. if a user starts executing a given activity in a different way, APD will be able to catch the new action pattern, since past patterns do not influence current executions. To analyse the needed steps to make HARS an online algorithm is out of the scope of this paper.

As such, the deployment of HARS for a given scenario comprises the following steps:

1) Design and implement sensor activation - action mappings. If binary sensors are being used to monitor user activities, this step becomes straightforward.
2) Identify target activities and design EAMs. We assume a domain expert will provide the knowledge for those EAMs.
3) Set the weights for the PMM algorithm (Equation 1). We assume a domain expert will provide the knowledge to set those weights appropriately.
4) Set the execution frequency of HARS. We provide some experiments that show how the quantity of collected data influences the performance of HARS (Section IV). Those experiments can be used to set the frequency in other use cases.

With the objective of understanding how HARS works, let us describe a real example. This example is taken from the experiments carried out in Section IV. We start with an unlabelled dataset, where only timestamps and sensor activations can be found (columns 1 and 2 of Table 1). In the APD step, first of all, we map sensor activations to actions. In this case, we decided to apply a 1:1 mapping (column 'Action' in Table 1). Afterwards, the pattern finding algorithm is applied. Take into account that for this example, we only provide a small part of the whole dataset. In the 'APD' column of Table 1 we can observe how APD finds and tags some patterns. At this step, APD can already distinguish actions that are not part of any activity (look at the 9:36:43 row in Table 1). Hence, using a fully unsupervised algorithm, frequently executed action sequences are extracted. However, notice how APD could not capture the whole activity 'PrepareBreakfast' in a unique pattern. Instead, APD finds two different patterns, namely Pat32 and Pat94. This is due to the variability of execution of the activity.

In the second step of HARS, i.e. PMM, the action sequences found by APD have to be identified with the target activities. EAMs are used for that purpose. In this case, we provide a summary of the EAMs used in Table 2. As can be seen, those EAMs define the common sense knowledge about the target activities. PMM maximises the cost function of Equation 1 for all EAM sets that share actions with the detected pattern, using the suitability functions defined in Section III-B. For example, in the case of Pat2, the actions pertaining to the pattern are only used in the EAM of the 'GoToBed' activity. Moreover, any additional EAM added to 'GoToBed' decreases the value of the cost function. Thus, the inference problem becomes very simple. However, for Pat32, two activities can be considered: 'PrepareBreakfast' and 'PrepareDinner'. PMM finds out that the best way to explain Pat32 is to label it as 'PrepareBreakfast'. Even though both activities share many actions and are performed in the same location, the starting time suitability makes 'PrepareBreakfast' the best candidate. We show the activity inferred by HARS in the PMM column of Table 1. Additionally, we also show the ground truth activity in the last column of the table.

**TABLE 3. Details of the datasets used for evaluation extracted from [37] and [10].**

| | House A | House C | Tapia dataset |
|---|---|---|---|
| Age | 26 | 57 | 30 |
| Gender | Male | Male | Female |
| Setting | Apartment | House | Apartment |
| Rooms | 3 | 6 | 7 |
| Duration | 25 days | 19 days | 16 days |
| Sensors | 14 | 21 | 76 |
| Activities | 8 | 18 | 21 |
| Annotation | Bluetooth headset | Bluetooth headset | PDA App |

## IV. EXPERIMENTS AND RESULTS

### A. MATERIALS AND METHODS

In order to validate the performance of HARS, we use the activity datasets[3] published by van Kasteren *et al.* [37]. More concretely, we use the datasets from House A and C. We discarded House B, due to annotation problems, probably derived from the annotation technique used by authors (a personal diary). Table 3 contains a summary of the datasets. Notice that the activity numbers of Table 3 take into account the 'None' activity. Another important detail regarding House C is that we divide the 'Use Toilet' activity into 'Use Toilet Upstairs' and 'Use Toilet Downstairs', since there are two different toilets in the house.

House A is the result of monitoring a 26-year-old man in a three-room apartment where 14 binary sensors were installed. Those sensors were installed in locations such as doors, cupboards, refrigerator, freezer or toilet. Sensor data for 25 days was collected for a total of 2120 sensor events and 245 activity instances. The annotated activities were: 'Leave House', 'Use Toilet', 'Take Shower', 'Go To Bed', 'Prepare Breakfast', 'Prepare Dinner' and 'Get Drink'.

House C presents a different scenario. A 57-year-old man is monitored in a house of two floors. 21 binary sensors were installed in the house to monitor 18 activities. As the total number and detail level of activities is higher than in House A, having activities such as 'Brush Teeth' or 'Shave', House C is a more challenging dataset. For further details about the datasets refer to [37].

We believe that the combination of both datasets is a good reference of the potential of HARS, since we combine two persons with significantly different ages, different environments and most notably, scenarios with low-grain and fine-grain activities. However, to show that HARS can be applied to diverse scenarios, we also show the obtained results in another dataset, completely unrelated to House A and C. More concretely, we use the dataset of Tapia *et al.* [10],[4] which we call *Tapia dataset*. There are two different apartments where binary sensors are deployed to monitor a single inhabitant. We use the data produced by Subject 1, because up to 76 sensors are used to monitor 21 activities during 16 days.

As summary, we run all the experiments using the three datasets described in Table 3: House A, House B and Tapia dataset. In order to have a deeper view of the three datasets in terms of activity instances, Table 4 provides some useful statistics. As it can be seen, all three datasets are very imbalanced in terms of activity instances.

[3]https://sites.google.com/site/tim0306/datasets

[4]http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/

**TABLE 4. Statistics of the three datasets: for each of them the maximum, minimum, average and standard deviation of instances per activity is depicted.**

| Instances/activity | House A | House C | Tapia dataset |
|---|---|---|---|
| Maximum | 114 | 52 | 84 |
| Minimum | 10 | 1 | 1 |
| Average | 35 | 20.23 | 14 |
| Standard deviation | 32.91 | 14.95 | 18.11 |

### B. PARAMETER SETTING

To run the experiments with HARS, for every target activity, we defined an EAM as described in Section III. We defined EAMs manually, using only the description of the sensors, activities and environment provided in the corresponding datasets. This process aims at showing that EAMs can be defined without a deep knowledge of the monitored person and the technical details of the intelligent environment. The simple nature of EAMs is one of the key features for the scalability of HARS.

Regarding sensor-action mappings, for the considered datasets, we decided to apply a 1:1 mapping function, since the semantics of each sensor activation can be considered unique with respect to the target activities.

Furthermore, HARS needs to configure some parameters, such as the weights of suitability functions (Equation 1) as well as some internal parameters of these suitability functions. For these experiments, we used the following values:

- $w_a = 1.3; w_d = 0.1; w_s = 1.5; w_l = 1$. Those values reflect the importance of actions and starting time respect to the other concepts.
- $k = 0.001$ to define the slope of the decreasing line for duration suitability function (Equation 3); $m = 1.0$ and $b = 0.1$ to define the line for starting time suitability function (Equation 5).

We set weight values exploring House A and C datasets individually. As the duration of each activity execution is very irregular for both cases, we set the duration weight to a low value. On the other hand, starting time plays an important role to distinguish between activities such as 'Prepare Breakfast', 'Prepare Dinner' and 'Get Drink', since they share a lot of actions. Their difference comes mainly from when the activity is performed. Regarding suitability functions' internal parameters, we set them at design time, so their configuration has been kept aside of the dataset. We would like to highlight that the same configuration values gave the best results for both datasets, even though the environments, persons and activities were not the same.

For the experiment with Tapia dataset, which was designed to show the applicability of HARS, we used the same parameters too.

### C. ACTIVITY RECOGNITION EXPERIMENTS

#### a: PERFORMANCE TESTS

Using as inputs the unlabelled datasets for House A and House C and the defined EAMs, we launched HARS for both datasets. We compared the obtained results to the respective ground truths, i.e. the datasets annotated by the monitored persons themselves. In order to assess the performance of the system, we use *precision*, *recall* and *F-Measure* in their *macro* variant, because the problem we are facing is a multi-class classification problem. Table 5 shows the obtained results for House A, whereas Table 6 is for House C (look at the HARS row in both cases).

**TABLE 5.** Comparison of the performance of different approaches for House A. USMART is marked with an asterisk since it does not consider 'None' activities.

| Category | Approach | Precision | Recall | F-Measure |
|---|---|---|---|---|
| Hybrid | HARS | 75.97% | 78.57% | 77.01% |
| | USMART* | – | – | 74.0% |
| Supervised | HSMM | 70.5% | 75.0% | 72.4% |
| | HMM | 70.3% | 74.3% | 72.0% |
| | NB | 67.3% | 64.8% | 65.8% |
| | CRF | 73.5% | 68.0% | 70.4% |

**TABLE 6.** Comparison of the performance of different approaches for House C. USMART is marked with an asterisk since it does not consider 'None' activities.

| Category | Approach | Precision | Recall | F-Measure |
|---|---|---|---|---|
| Hybrid | HARS | 51.17% | 40.55% | 43.42% |
| | USMART* | – | – | 46.0% |
| Supervised | HSMM | 43.8% | 52.3% | 47.4% |
| | HMM | 40.7% | 53.7% | 45.9% |
| | NB | 40.5% | 46.4% | 42.3% |
| | CRF | 37.7% | 40.4% | 38.9% |

In order to have meaningful performance references for the same datasets, we also depict the results obtained by van Kasteren *et al.* [37] for the corresponding dataset. More concretely, authors test the usage of four supervised learning approaches, namely, Naive Bayes (NB), Hidden Markov Model (HMM), Hidden semi-Markov Model (HSMM) and Conditional Random Field (CRF). They combine those four approaches with three different sensor representations, but we only show the results of the so called *change of point* representation which achieves the best F-Measures consistently (Tables 5 and 6). We also depict the results published for USMART [35]. They do not provide average precision and recall, only the F-Measure. It is very important to highlight, though, that USMART does not consider the 'None' activity in their experiments. Thus, the provided results cannot be compared to ours or to Kasteren's [37] on equal basis. As USMART authors say [35]: *"We do not consider the null type of activities in our evaluation simply because they are time periods with no associated activity annotation in the ground truth, and thus it is impossible to define necessary conditions on them."*

Additionally, we also provide the confusion matrices obtained with HARS for House A (Table 7) and House C (Table 8). The numbers depicted in the confusion matrices are percentages. Those confusion matrices allow us to further analyse the behaviour of HARS, showing in detail the performance obtained activity by activity.

**TABLE 7.** The confusion matrix of HARS for House A. NO: None, GB: Go To Bed, UT: Use Toilet, PB: Prepare Breakfast, TS: Take Shower, LH: Leave House, GD: Get Drink, PD: Prepare Dinner.

| | NO | GB | UT | PB | TS | LH | GD | PD |
|---|---|---|---|---|---|---|---|---|
| **NO** | **42.86** | 8.16 | 9.18 | 9.18 | 3.06 | 2.04 | 5.10 | 20.41 |
| **GB** | 0.00 | **100.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **UT** | 13.33 | 0.00 | **86.00** | 0.00 | 0.67 | 0.00 | 0.00 | 0.00 |
| **PB** | 12.77 | 0.00 | 0.00 | **78.72** | 0.00 | 0.00 | 6.38 | 2.13 |
| **TS** | 4.17 | 0.00 | 8.33 | 0.00 | **87.50** | 0.00 | 0.00 | 0.00 |
| **LH** | 13.95 | 0.00 | 2.33 | 0.00 | 0.00 | **83.72** | 0.00 | 0.00 |
| **GD** | 12.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **80.00** | 8.00 |
| **PD** | 24.53 | 0.00 | 0.00 | 1.89 | 0.00 | 0.00 | 3.77 | **69.81** |

Let us now explain some meaningful examples to better understand how HARS works and add more context to the obtained results. We will illustrate the capacity of our system to handle action patterns which describe multiple activities. In House A, it is quite common to see the monitored person going to the toilet by night just before going to bed. In consequence, APD discovered that a typical action pattern is *{HallToiletDoor, HallToiletDoor, HallToiletDoor, HallBedroomDoor, HallBedroomDoor}*. This is the case where a person performs two chaining activities frequently. Using the defined EAMs, the matching algorithm of PMM can infer that there are two ongoing activities in the action pattern; namely, 'Use Toilet' and 'Go To Bed'. Consulting the ground truth, we observe the inference is correct.

To see how HARS can also handle *idle* activities, we have another example, extracted again from House A. There is an action pattern composed by *{HallBedroomDoor, HallToiletDoor, HallBedroomDoor}*. The ground truth indicates that the monitored person was not performing any activity at that moment. The pattern resembles an erratic behaviour, going from bedroom to toilet and back without any apparent purpose. APD discovered that the action pattern is not frequent, so it directly tagged it as 'None', inferring that the action sequence is not actually representing any activity.

Another exemplary case shows an unknown activity which has been correctly labelled as 'None'. APD discovered a pattern of *{Dishwasher, Dishwasher}*, which seems to describe a washing dishes activity. However, there are no annotations for such an activity in the ground truth, thus, we did not model any EAM for it. As the actions of the pattern have zero correspondence with the defined EAMs, HARS labels the pattern as 'None'. Notice that in this case, HARS actually discovered an unknown activity, which could be used to model a new EAM.

We can also see cases where PMM fails at inferring the ongoing activity. For example, in House C, in the action pattern composed by *{Fridge, Freezer, Fridge, Cutlery, Fridge, Freezer}*, PMM infers two activities: 'Eating' and 'Prepare Breakfast'. Actually, the ground truth shows only an activity: 'Prepare Breakfast'. In this case, due to the actions modelled in the EAMs, PMM gives a higher score to the composition of the two activities, introducing an error.

**TABLE 8.** The confusion matrix of HARS for House C. NO: None, LH: Leave House, UTD: Use Toilet Downstairs, GD: Get Drink, RE: Relax, GS: Get Snack, BT: Brush Teeth, GB: Go To Bed, UTU: Use Toilet Upstairs, GDR: Get Dressed, PB: Prepare Breakfast, EA: Eating, PL: Prepare Lunch, PD: Prepare Dinner, TM: Take Medication, TS: Take Shower, SH: Shave, CM: Put Clothes in Washing Machine.

| | NO | LH | UTD | GD | RE | GS | BT | GB | UTU | GDR | PB | EA | PL | PD | TM | TS | SH | CM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **NO** | **11.40** | 4.78 | 7.35 | 3.68 | 22.43 | 5.51 | 8.82 | 0.00 | 10.66 | 7.72 | 1.84 | 1.84 | 1.47 | 2.94 | 4.41 | 4.78 | 0.37 | 0.00 |
| **LH** | 9.84 | **72.13** | 1.64 | 0.00 | 11.48 | 0.00 | 0.00 | 0.00 | 0.00 | 1.64 | 0.00 | 0.00 | 0.00 | 1.64 | 0.00 | 0.00 | 1.64 | 0.00 |
| **UTD** | 3.85 | 0.00 | **80.77** | 0.00 | 3.85 | 0.00 | 0.00 | 0.00 | 0.00 | 3.85 | 3.85 | 0.00 | 0.00 | 0.00 | 0.00 | 3.85 | 0.00 | 0.00 |
| **GD** | 36.36 | 0.00 | 0.00 | **36.36** | 0.00 | 0.00 | 0.00 | 0.00 | 18.18 | 0.00 | 0.00 | 9.09 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **RE** | 13.41 | 0.00 | 6.10 | 2.44 | **62.20** | 2.44 | 3.66 | 0.00 | 3.66 | 0.00 | 0.00 | 2.44 | 1.22 | 2.44 | 0.00 | 0.00 | 0.00 | 0.00 |
| **GS** | 18.18 | 0.00 | 0.00 | 0.00 | 9.09 | **63.64** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 9.09 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **BT** | 15.15 | 0.00 | 4.55 | 0.00 | 12.12 | 0.00 | **36.36** | 0.00 | 13.64 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 12.12 | 3.03 | 3.03 | 0.00 |
| **GB** | 18.82 | 0.00 | 0.00 | 0.00 | 2.35 | 0.00 | 5.88 | **54.12** | 4.71 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.59 | 3.53 | 0.00 | 0.00 |
| **UTU** | 42.42 | 0.00 | 0.00 | 0.00 | 6.06 | 0.00 | 6.06 | 0.00 | **27.27** | 3.03 | 0.00 | 0.00 | 0.00 | 0.00 | 3.03 | 9.09 | 3.03 | 0.00 |
| **GDR** | 30.00 | 0.00 | 0.00 | 0.00 | 2.00 | 0.00 | 10.00 | 0.00 | 4.00 | **42.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 12.00 | 0.00 | 0.00 |
| **PB** | 27.78 | 0.00 | 2.78 | 5.56 | 8.33 | 0.00 | 0.00 | 0.00 | 2.78 | 0.00 | **47.22** | 5.56 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **EA** | 43.14 | 0.00 | 1.96 | 3.92 | 17.65 | 1.96 | 0.00 | 0.00 | 0.00 | 0.00 | 5.88 | **17.65** | 0.00 | 7.84 | 0.00 | 0.00 | 0.00 | 0.00 |
| **PL** | 25.93 | 0.00 | 0.00 | 3.70 | 3.70 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 11.11 | 0.00 | **55.56** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **PD** | 21.51 | 0.00 | 1.08 | 1.08 | 6.45 | 0.00 | 0.00 | 0.00 | 0.00 | 5.38 | 0.00 | 0.00 | 2.15 | **62.37** | 0.00 | 0.00 | 0.00 | 0.00 |
| **TM** | 50.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 16.67 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **33.33** | 0.00 | 0.00 | 0.00 |
| **TS** | 27.27 | 0.00 | 0.00 | 0.00 | 2.27 | 0.00 | 15.91 | 2.27 | 25.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 2.27 | **25.00** | 0.00 | 0.00 |
| **SH** | 23.68 | 2.63 | 0.00 | 0.00 | 13.16 | 0.00 | 39.47 | 0.00 | 7.89 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 7.89 | 2.63 | **2.63** | 0.00 |
| **CM** | 33.33 | 0.00 | 0.00 | 33.33 | 0.00 | 0.00 | 0.00 | 0.00 | 33.33 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.00** |

**TABLE 9.** Comparison of the performance of different weight combinations for both houses, in terms of F-Measure. Line 1 shows the best performance. Each line shows in bold the weight that has been modified respect to line 1.

| | $w_a$ | $w_l$ | $w_d$ | $w_s$ | House A F-Measure | House C F-Measure |
|---|---|---|---|---|---|---|
| 1 | 1.3 | 1.0 | 0.1 | 1.5 | 77.02% | 43.42% |
| 2 | 1.3 | 1.0 | **1.0** | 1.5 | 76.33% | 42.39% |
| 3 | 1.3 | 1.0 | 0.1 | **2.0** | 75.45% | 42.54% |
| 4 | **2.0** | 1.0 | 0.1 | 1.5 | 74.48% | 42.64% |
| 5 | **0.5** | 1.0 | 0.1 | 1.5 | 75.45% | 42.40% |
| 6 | 1.3 | **3.0** | 0.1 | 1.5 | 76.54% | 42.66% |
| 7 | 1.3 | **0.1** | 0.1 | 1.5 | 76.67% | 41.46% |

Another typical error is when HARS detects activities where no activity is happening actually. We can see in House C an action pattern composed by *{Frontdoor, Frontdoor, Frontdoor, Couch, Couch}*. For that action pattern, APD actually extracts two patterns: *{Frontdoor, Frontdoor, Frontdoor}* and *{Couch, Couch}*. PMM infers that the first pattern represents a 'Leave House' activity, whereas the second is 'Relax'. However, the ground truth shows that those actions are not representing any activity. This could possibly be inferred from the timestamps of the actions (between 11:14 and 11:18), specially discarding the 'Leave House' activity. However, HARS fails at distinguishing those subtle details.

### *b: PARAMETER CONFIGURATION TESTS*

As explained above, weight values do not have a big impact in the performance of HARS. In order to show that, Table 9 depicts some of the experiments we performed with different weight values for both House A and C. We changed the value of a single weight between experiments, to see clearly the impact of each concept (action, location, duration and starting time). For visualisation purposes, we depict only the F-Measure.

### *c: DATA QUANTITY INFLUENCE TESTS*

On the other hand, as we are using a pattern mining algorithm (APD), we wanted to see how the performance of HARS varies depending on the quantity of available data to find frequent patterns. That has a big impact when deciding how often HARS should be executed for a given use case. For that purpose, we designed another experiment. We use House A dataset in that new experiment. Specifically, we take 1, 5, 10, 15, 20 and 25 (all) days of the dataset and run HARS on those *reduced* datasets. For example, when testing with 1 day, we only use the first day of the dataset to find action patterns and recognise the activities. The evaluation of the results are also provided for that unique day. The same applies to other number of days. To assess the performance of HARS, we depict precision, recall and F-Measure. The obtained results can be seen in Figure 4.

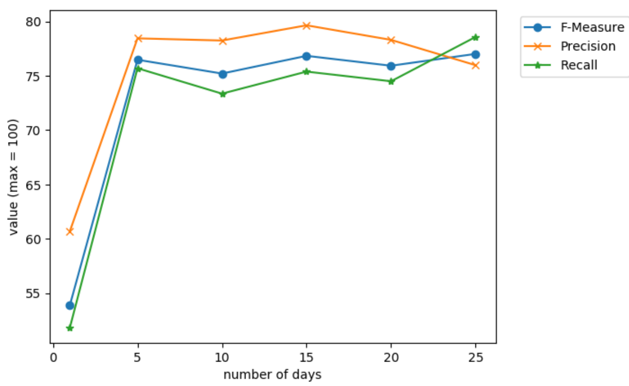


**FIGURE 4.** Results of the experiment to assess the performance of HARS depending on the quantity of available data. Several number of days are shown for House A dataset, with obtained precision, recall and F-Measure.

### *d: HARS APPLICABILITY TEST*

Finally, to show the applicability of HARS to different scenarios, we also show the obtained results in another dataset. More concretely, we run HARS on Tapia dataset, which has no relation with House A and C datasets. Thus it is a good dataset to combine with the previous two and have diverse

**TABLE 10.** **Comparison of the performance of the Naive Bayes classifier and HARS on Tapia dataset. As the *idle* activity is not considered in the results of Naive Bayes, the comparison has to be used only as a reference.**

| Activity | Naive Bayes | HARS |
|---|---|---|
| Preparing lunch | 29.0% | 35.0% |
| Toileting | 31.0% | 23.5% |
| Preparing breakfast | 6.0% | 45.8% |
| Bathing | 29.0% | 34.6% |
| Dressing | 3.0% | 35.4% |
| Grooming | 26.0% | 53.6% |
| Preparing a beverage | 13.0% | 42.9% |
| Doing laundry | 7.0% | 55.9% |
| **Average** | 17.99% | 40.84% |

application scenarios. As a result of this experiment, we obtained an average precision of 56.33%, recall of 40.47% and F-Measure of 43.81%. Given the number of activities, this performance is in line with the results obtained in House C. Tapia *et al.* [10] use Naive Bayes classifiers for AR, but they do not show the results in terms of precision, recall and F-Measure. They show the accuracy obtained for a given set of activities; concretely, those activities that have more than six instances in the dataset. The evaluation methodology which is closest to ours is the so called *percentage of time activity is detected*. However, as they do not consider the *idle* activity, both results cannot be compared on equal basis. Anyway, Table 10 shows a comparison between both approaches, which should only be used as a reference.

## V. DISCUSSION

### A. PERFORMANCE

The results obtained in the experiments of Section IV show that HARS has a performance comparable to supervised learning approaches. More concretely, for House A HARS obtains the best metrics, scoring for F-Measure almost 5 points more than the best supervised learning approach (HSMM). However, for House C, HARS is 4 points below the HSMM, but still ahead of NB and CRF, two typical supervised learning approaches used for AR (look at Tables 5 and 6). We can also use the reference of Tapia dataset, where HARS outperforms the results of a Naive Bayes classifier, even though considering the *idle* activity, which makes the problem harder (Table 10). But due to the differences in evaluation methodologies, we will not use the results obtained in Tapia dataset to discuss the performance of HARS.

It is worth to comment that for a fair comparison of those results, the specific evaluation methodologies have to be taken into account. Kasteren et al. use a technique known as *leave one day out*, which means that they train the model using all the available days except one, to test the performance in that discarded day [37]. Thus, the results they obtain are an average of applying this technique for all the combinations. In their paper, Kasteren et al. also show the standard deviation of their results. For House C, HSMM obtains an F-Measure of 47.9%, with a standard deviation of 11.3 points.

For HARS there is no training process, hence we test the performance for the whole dataset. Our results are not average values. It has to be highlighted that for both houses, HARS is inside the performance margins of the best supervised learning approach. This fact supports our claim of having a performance comparable to supervised learning techniques.

Notice the importance of such a result, specially when considering the useful information for each of the techniques. Fully unsupervised learning uses only sensor information, while supervised learning approaches have the activity label for each sensor event, from where very detailed relations can be learnt, such as the order in which sensors fire, time distances between sensors, which sensors are representative for a given activity and so on. In the case of HARS, the useful information level is between the other two techniques, but still very far from the information level of supervised learning techniques. EAMs are generic and rough representations of activities, but they are easy to obtain and model. Results show that such prior knowledge is enough to perform as well as supervised learning approaches, avoiding any annotation effort from users or technicians.

Following these reasons, it is particularly interesting to compare HARS with USMART [35], because both approaches adopt a similar philosophy. Unfortunately, even though both approaches are tested on the same datasets, obtained results cannot be fairly compared. If we look at Tables 5 and 6, we might conclude that HARS and USMART have very similar performances. However, it has to be noted that USMART does not consider the 'None' activity to calculate the average F-Measure. We believe that is a positive feature of HARS compared to USMART. We believe that for an AR system to work in real world scenarios, *idle* or 'None' activities have to be handled, since there are a lot of time periods where monitored people may not perform any target activity, even though sensors are reporting activations. Detecting and recognising *idle* periods where sensor activations are registered is a very hard problem. Supervised learning approaches and HARS can deal with those situations - with different success rates -, whereas USMART cannot yet. Looking at the difficulties of detecting 'None' activities, which show the worst recognition rates for HARS, we presume that the results of USMART would degrade significantly, but we cannot make such a claim without experimentation.

Focusing on HARS, confusion matrices in Tables 7 and 8 show that the main problem of our approach is related to those actions that are not part of any activity, i.e. actions tagged as 'None'. Even though HARS can detect *idle* activities, this is the weakest point of the system. Actions tagged as 'None' are the main source of false positives and negatives. This is quite normal though, since the PMM algorithm tries to explain every action pattern found with a set of activities. So, if an action pattern is observed, which can be due to erratic behaviour or even an unknown activity, the algorithm will try to *fit* EAMs to it. Furthermore, we observed that due to the APD algorithm, long activities like 'Prepare Dinner' and 'Prepare Breakfast' are very difficult to catch in a single pattern - here long refers to the number of actions -. It was quite common to see the same activity split into different patterns, where some intermediate actions were already tagged as 'None' activities by the pattern finder. Hence, the PMM step is not applied to those actions, because it is interpreted that they are not part of any activity. Those observations suggest that further work is required in the APD step. We believe that to improve the detection rates of *idle* activities, we have to re-work the APD algorithm rather than the PMM, trying to assure that the patterns processed by PMM are *real* activities, not *idle* activities.

Let us focus on House C, where HARS obtains poorer results. We can observe in the confusion matrix (Table 8) that the worst performance is related to similar activities. More concretely, it can be seen clearly that HARS has several problems when distinguishing among activities 'Brush Teeth', 'Use Toilet Upstairs', 'Take Medication' and 'Shave'. Those four activities share almost the same actions, they are performed in the same location and even the starting time and duration do not differ a lot. Even for a human observer is very difficult to distinguish among those activities, based on the deployed sensors and their activations.

Finally, we would like to highlight the impact of annotation errors in the results. Both House A and C have annotation errors in the ground truth. Those errors are more notable in House C. For example, in the confusion matrix of Table 8 we can observe that 11.48% of times, 'Leave House' activity instances have been recognised as 'Relax'. If we go to the ground truth, we can see how a *Frontdoor* action is followed by a sequence of *Couch* actions. Those *Couch* actions are wrongly annotated as 'Leave House'. The same happens for 'Get Drink' and 'Use Toilet Upstairs' (18.8%), 'Go To Bed' and 'Take Medication' (10.59%) and 'Get Dressed' and 'Take Shower' (12.0%). The number of errors for those pairs of activities is quite high, but it is difficult to say that they are solely due to annotation errors. However, we can claim that the results shown in Section IV are a lower bound of the performance of HARS. It is also important to highlight that annotation errors do not affect the same way to supervised learning techniques and to HARS. In the case of HARS, every annotation error leads to a recognition error. However, depending on the nature of those errors, supervised learning techniques may learn to identify those (erroneous) instances as they are annotated.

### B. PARAMETER CONFIGURATION

HARS has some parameters that have to be externally provided, as explained in Section III. The most important ones are the weights of the cost function in Equation 1. A bad configuration of those weights may lead to bad performance. However, it is important to note that the configuration process does not require any technical knowledge of the PMM algorithm. In general, with a broad knowledge of the specific dataset, weights can be configured easily, as shown in Section IV. We saw during experiments that small variations of the weights do not change significantly the performance, thus specifying the relative importance of each concept is enough to have a good performance.

Table 9 shows some experiments carried out to show the effect of different weights. As can be seen, the performance changes, but the difference between the best and worst F-Measure values is about 2-2.5 points. Take into account that the depicted experiments do not consider weight values that are extremely different from the optimal configuration. The objective was to show that once the relative difference of weights is quite clear, varying some of them does not affect too much the performance.

Additionally, those weights provide a way to adapt the behaviour of HARS to specific persons. Supervised learning approaches rely on the training phase to learn personal models. HARS addresses this problem using the unsupervised learning step, which captures the frequent action sequences of a person, and tuning the weights of the cost function. In that sense, it has been curious to see that for the datasets we used, the same weights give the best performance. We believe that this can be explained with the nature of activities and sensors. In both datasets there are groups of activities that can be distinguished based on actions. But to distinguish the activities inside the same group, starting time plays a crucial role (the difference between 'Prepare Breakfast', 'Prepare Lunch' and 'Prepare Dinner', for example). That is why action and starting time weights are the highest for both datasets.

Due to the offline nature of HARS and its deployment, there is another key parameter; namely, the execution frequency of the system. As explained in Section III-C, depending on the execution frequency, HARS will have different quantities of data both for the action pattern mining (APD) and the pattern recognition (PMM). In principle, the quantity of data should affect the pattern mining process, since APD finds frequent action sequences. In consequence, the recognition performance of HARS should also be dependent on the quantity of data. To better assess the influence of the quantity of data in the performance of HARS, Figure 4 shows the results obtained for the designed experiment on House A, using several number of days. As can be seen, even with one day of data, the F-Measure is already around 55%. With 5 days of data, F-Measure increases to around 75%. Adding more days does not seem to affect much the performance of HARS. We can observe slight fluctuations of precision, recall and F-Measure. Looking more carefully at the results, those fluctuations seem to be more related with the specific execution of activities registered on those days, rather than with the quality of the action patterns extracted by APD. This suggests that APD needs few days (5 in this case) to be able to extract reliable action patterns. That is why we are currently running HARS once per week in the City4Age project.

### C. SCALABILITY, ACTIVITY GRANULARITY AND SEMANTIC MEANING

As noted in Section II, supervised learning approaches do not suffer from activity granularity and semantic meaning, but they are not scalable, i.e. deploying such algorithms in a new environment with a new user implies a lot of data collection and annotation time. Unsupervised learning approaches solve the scalability problem, but suffer from activity granularity and lack of semantic meaning. However, HARS is scalable, overcomes the problems of the semantic meaning of action patterns and partially handles activity granularity, as shown in the examples of Section IV. Those advantages come with a performance comparable to supervised learning approaches, which make HARS a good candidate for real world deployments. In scenarios and applications where several people have to be monitored under varying sensor infrastructures, supervised learning approaches do not scale well, since obtaining labelled datasets for every monitored person is unfeasible. In contrast, defining a set of EAMs is assumable, given that only incomplete knowledge is required about actions, locations, approximate duration and usual starting times. As a proof of this scalability, we have presented how HARS can be used in three different datasets, where three different persons are monitored in different environments.

In our experience, to deploy a supervised learning AR system, the following steps have to be completed: (i) collect enough data to train supervised models, (ii) annotate data with correct activity labels, and (iii) train the supervised model. Those steps usually involve several days of effort, although the exact number of time may vary a lot depending on the number of activities to be monitored, the number of sensors deployed and their nature. In contrast, for HARS we have to follow the steps described in Section III-C, where modelling EAMs is the most time-consuming step, following our experience. For example, in this paper we have shown experiments on three different datasets. We needed roughly around one hour to model the EAMs for each dataset. Afterwards, we did not need any training or annotation step, since the system was ready to start working. In consequence, we are comparing two processes that differ in the time scale, based on our experience: while supervised approaches typically need deployment times in the order of days, HARS can be deployed in hours. Take into account that as HARS is an offline algorithm, the data collection phase is already part of the functioning system, thus, it is not a required step for deployment. In our opinion, those considerations make HARS more scalable than supervised AR systems, following the definition given in Section I.

The only scalability concern may refer to the number of activities to be monitored. For every target activity, an EAM has to be provided. The PMM algorithm checks all the possible combinations of EAMs (see Equation 7), whose number grows exponentially on the number of EAMs. However, typical scenarios do not present more than 15-20 activities. Given that relatively low number of activities and the pruning strategies used in PMM (see Section III-B), our experiments do not show any scalability problem. In terms of computing resources, the most demanding dataset is Tapia dataset, which contains 21 activities and 76 sensors. Even with those numbers, using a laptop equipped with an Intel Core i5 CPU and 8 GB of RAM - not exclusively dedicated to the task -, a whole day of sensor activations was processed in average in around 3 minutes. Furthermore, for those scenarios where the PMM approach might be intractable due to the high number of activities, it would be straightforward to incorporate a genetic algorithm in the matching process. However, for the current scenarios we handle, it is not necessary to adopt further optimisation strategies.

Activity granularity and semantic meaning are also handled using EAMs. The APD algorithm frequently extracts action patterns composed by more than one activity, but PMM can deal with those situations and infer the best combination of activities to explain the action pattern and assign the suitable semantic labels. This is very important, specially to provide meaningful information to caregivers or assistive application developers. However, note that HARS cannot assign activity labels to each registered action, but to action patterns. So there is still some ground of improvement from the granularity point of view (see Section VI).

## VI. CONCLUSIONS AND FUTURE WORK

This paper has presented HARS, a novel scalable hybrid activity recognition system for intelligent environments. Due to the deployment requirements of the City4Age project, very close to real world conditions, we cannot use supervised learning approaches, since they do not scale well. In such scenarios, if supervised learning techniques want to be used, the problem of annotating datasets for hundreds of users under varying sensor deployments has to be faced. This poses a real challenge in terms of the invested effort. Therefore, we have adopted a combination of unsupervised learning and knowledge-driven techniques. The obtained results show that HARS has a comparable performance to supervised learning approaches, handling some of the most important problems of unsupervised learning approaches, such as activity granularity and lack of semantic meaning.

To sum up, HARS possesses the desired features for a real world AR system:

1) Its performance is comparable to state-of-the-art supervised learning approaches, which are the most successful AR systems nowadays.
2) It is scalable. Only EAMs have to be provided to the system, which are rough and generic activity models. EAMs can be obtained from everyday commonsense knowledge and the description of a given sensor infrastructure.
3) It provides semantic activity labels, making easier the monitoring step for caregivers and the development of assistive applications.
4) In contrast to unsupervised learning approaches, it can handle, at some extent, activity granularity. It can distinguish user-defined activities in sequences of chaining activities.

It is also important to highlight that HARS is an offline algorithm. At its current state, it cannot provide AR on streaming data. However, that is not the objective of the designed system, which is not thought for scenarios where immediate assistance is needed. On the contrary, HARS is designed to support further analysis techniques to detect behaviour changes and correlate those changes with MCI and frailty. In consequence, HARS should not be used in scenarios that require online AR.

In order to make our results reproducible and contribute to the scientific community, the software implementation of HARS, as well as all the files (datasets and EAMs) and scripts for the experiments performed in this paper, are publicly available at GitHub.[5]

In the short-time, we plan to explore some strategies to deepen in the activity granularity problem. We want to use the information contained in EAMs to be able to label each action of an action pattern with the corresponding activity, whenever is needed. The actions defined in the EAMs could easily be identified in a given pattern. However, those actions that are not in any EAM of the recognised activities are harder to classify. For those cases, we are planning to use location, time and action similarity metrics to assess the proper activity.

For the longer time, the careful analysis of the experiments carried out in Section IV has shown that there is still space for improvements, specially regarding the APD algorithm (Section V). We plan to follow two different strategies to improve APD. First of all, Rashidi and Cook have already presented a better mining algorithm called COM [29]. To the best of our knowledge, COM has not been made available, so we are planning to implement it and test whether we obtain better results.

On the other hand, we also consider a very different approach for APD. More concretely, we are planning to use neural embeddings to represent sensor events as vectors in a continuous space where similar sensor events would appear close to each other [42]. That would allow us to span those *action vectors* in the time axis and apply clustering approaches with typical vector distance metrics. Furthermore, the PMM algorithm would also be different, since the similarity between EAMs and action patterns can be defined in a totally different way. We believe that the envisioned new approach will be able to mine activities and address activity variations better than the current approaches.

Finally, due to some design decisions we took to ensure the simplicity of EAMs, HARS has limitations regarding the chronological order of actions. As can be seen in Section III, neither EAMs nor PMM take into account the order of actions when recognising activities. People tend to perform activities in many varying ways. In consequence, the order of actions varies among people. The modelling effort of incorporating those order-varying actions in EAMs does not contribute to make EAMs simple models. However, in some scenarios the order in which actions are executed can be key to recognise different activities. That is not the case in the datasets we used for the experiments (Section IV), but examples of such cases exist. In the current state of HARS it is not trivial to incorporate the order of actions without making EAMs complex. We believe the order of actions for an activity is a personal feature, and as such, it is better suited to be taken into account in the learning step rather than in the modelling and matching steps. For that purpose an interesting line of research could be designing more order-dependent pattern matching algorithms which can be combined with simple activity models.

[5] https://github.com/aitoralmeida/c4a_activity_recognition

## ACKNOWLEDGMENTS

Thanks to CASAS team in Washington State University for making their excellent software available to researchers and to Tim Van Kasteren and Emmanuel Munguia Tapia for publishing the activity dataset for benchmarking purposes.

## REFERENCES

[1] G. Onder, B. W. J. H. Penninx, L. Ferrucci, L. P. Fried, J. M. Guralnik, and M. Pahor, "Measures of physical performance and risk for progressive and catastrophic disability: Results from the women's health and aging study," *J. Gerontol. A, Biol. Sci. Med. Sci.*, vol. 60, no. 1, pp. 74–79, Jan. 2005.

[2] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[3] G. Azkune, A. Almeida, D. López-de-Ipiña, and L. Chen, "Extending knowledge-driven activity models through data-driven learning techniques," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3115–3128, Apr. 2015.

[4] D. A. Gold, "An examination of instrumental activities of daily living assessment in older adults and mild cognitive impairment," *J. Clin. Exp. Neuropsychol.*, vol. 34, no. 1, pp. 11–34, Jan. 2012.

[5] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput. Vis. Image Understand.*, vol. 115, no. 2, pp. 224–241, Feb. 2011.

[6] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[7] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing* (Lecture Notes in Computer Science). Vienna, Austria: Springer, 2004, pp. 1–17.

[8] O. Brdiczka, P. Reignier, and J. L. Crowley, "Detecting individual activities from video in a smart home," in *Knowledge-Based Intelligent Information and Engineering Systems*. Berlin, Germany: Springer, 2007, pp. 363–370.

[9] D. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods Inf. Med.*, vol. 48, no. 5, pp. 480–485, 2009.

[10] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Pervasive Computing* (Lecture Notes in Computer Science), vol. 3001, A. Ferscha and F. Mattern, Eds. Berlin, Germany: Springer, 2004.

[11] T. van Kasteren and B. Krose, "Bayesian activity recognition in residence for elders," in *Proc. Int. Conf. Intell. Environ.*, 2007, pp. 209–213.

[12] U. Maurer, A. Smailagic, D. Siewiorek, and A. Rowe, "Location and activity recognition using eWatch: A wearable sensor platform," in *Ambient Intelligence in Everyday Life*. Berlin, Germany: Springer, 2006, pp. 86–102.

[13] T. van Kasteren, G. Englebienne, B. Kröse, and A. Noulas, "Accurate activity recognition in a home setting," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, 2008, pp. 1–9.

[14] A. Galata, N. Johnson, and D. Hogg, "Learning structured behaviour models using variable length Markov models," in *Proc. IEEE Int. Workshop Modelling People*, Sep. 1999, pp. 95–102.

[15] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 90–126, 2006.

[16] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 994–999.

[17] N. Oliver, A. Garg, and E. Horvitz, "Layered representations for learning and inferring office activity from multiple sensory channels," *Comput. Vis. Image Understand.*, vol. 96, no. 2, pp. 163–180, 2004.

[18] O. Brdiczka, J. L. Crowley, and P. Reignier, ''Learning situation models in a smart home,'' *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 39, no. 1, pp. 56–63, Feb. 2009.
[19] F. J. Ordóñez, J. A. Iglesias, A. Ledezma, A. Sanchis, and P. de Toledo, ''Online activity recognition using evolving classifiers,'' *Expert Syst. Appl.*, vol. 40, no. 4, pp. 1248–1255, 2013.
[20] L. Liu, Y. Peng, S. Wang, M. Liu, and Z. Huang, ''Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors,'' *Inf. Sci.*, vols. 340–341, pp. 41–57, May 2016.
[21] Y. Kwon, K. Kang, and C. Bae, ''Unsupervised learning for human activity recognition using smartphone sensors,'' *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6067–6074, Oct. 2014.
[22] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu, ''epSICAR: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition,'' in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2009, pp. 1–9.
[23] P. Rashidi and D. J. Cook, ''Keeping the resident in the loop: Adapting the smart home to the user,'' *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 5, pp. 949–959, Sep. 2009.
[24] E. O. Heierman and D. J. Cook, ''Improving home automation by discovering regularly occurring device usage patterns,'' in *Proc. 3rd IEEE Int. Conf. Data Mining*, Nov. 2003, pp. 537–540.
[25] T. Huỳnh and B. Schiele, ''Unsupervised discovery of structure in activity data using multiple eigenspaces,'' in *Location- and Context-Awareness*. Berlin, Germany: Springer, 2006, pp. 151–167.
[26] T. S. Barger, D. E. Brown, and M. Alwan, ''Health-status monitoring through analysis of behavioral patterns,'' *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 35, no. 1, pp. 22–27, Jan. 2005.
[27] T. Dimitrov, J. Pauli, and E. Naroska, ''Unsupervised recognition of ADLs,'' in *Artificial Intelligence: Theories, Models and Applications*. Berlin, Germany: Springer, 2010, pp. 71–80.
[28] P. Palmes, H. K. Pung, T. Gu, W. Xue, and S. Chen, ''Object relevance weight pattern mining for activity recognition and segmentation,'' *Pervasive Mobile Comput.*, vol. 6, no. 1, pp. 43–57, Feb. 2010.
[29] P. Rashidi and D. J. Cook, ''COM: A method for mining and monitoring human activity patterns in home-based health monitoring systems,'' *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, pp. 64-1–64-20, 2013.
[30] L. Chen and C. Nugent, ''A logical framework for behaviour reasoning and assistance in a smart home,'' *Int. J. Assistive Robot. Mechatron.*, vol. 9, no. 4, pp. 20–34, 2008.
[31] D. Riboni and C. Bettini, ''COSAR: Hybrid reasoning for context-aware activity recognition,'' *Pers. Ubiquitous Comput.*, vol. 15, no. 3, pp. 271–289, Mar. 2010.
[32] L. Chen, C. D. Nugent, and H. Wang, ''A knowledge-driven approach to activity recognition in smart homes,'' *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 961–974, Jun. 2012.
[33] M. H. M. Noor, Z. Salcic, and K. I.-K. Wang, ''Enhancing ontological reasoning with uncertainty handling for activity recognition,'' *Knowl.-Based Syst.*, vol. 114, pp. 47–60, Dec. 2016.
[34] D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt, ''Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning,'' in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*. New York, NY, USA: ACM Press, 2016, pp. 1–12.
[35] J. Ye, G. Stevenson, and S. Dobson, ''USMART: An unsupervised semantic mining activity recognition technique,'' *ACM Trans. Interact. Intell. Syst.*, vol. 4, no. 4, Nov. 2014, Art. no. 16.
[36] D. J. Cook, N. C. Krishnan, and P. Rashidi, ''Activity discovery and activity recognition: A new partnership,'' *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 820–828, Jun. 2013.
[37] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, ''Human activity recognition from wireless sensor network data: Benchmark and software,'' in *Activity Recognition in Pervasive Intelligent Environments*. Atlantis Press, 2011, pp. 165–186. [Online]. Available: https://link.springer.com/chapter/10.2991/978-94-91216-05-3_8#citeas
[38] R. Agrawal and R. Srikant, ''Mining sequential patterns,'' in *Proc. 7th Int. Conf. Data Eng.*, 1995, pp. 3–14.
[39] T. Gu, S. Chen, X. Tao, and J. Lu, ''An unsupervised approach to activity recognition and segmentation based on object-use fingerprints,'' *Data Knowl. Eng.*, vol. 69, no. 6, pp. 533–544, Jun. 2010.
[40] J. Pei, J. Han, and W. Wang, ''Constraint-based sequential pattern mining: The pattern-growth methods,'' *J. Intell. Inf. Syst.*, vol. 28, no. 2, pp. 133–160, Mar. 2007.
[41] A. Aztiria, J. C. Augusto, R. Basagoiti, A. Izaguirre, and D. J. Cook, ''Discovering frequent user–environment interactions in intelligent environments,'' *Pers. Ubiquitous Comput.*, vol. 16, no. 1, pp. 91–103, Jan. 2012.
[42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ''Distributed Representations of Words and Phrases and their Compositionality,'' in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2013, pp. 3111–3119.

**GORKA AZKUNE** received the Ph.D. degree in computer science from the University of Deusto. He is currently a Lecturer with the University of Deusto and a Researcher with the DeustoTech–Internet Research Group. His research interests include artificial intelligence, machine learning, and intelligent environments.

**AITOR ALMEIDA** received the Ph.D. degree in computer science from the University of Deusto. He is currently a Researcher and the Project Manager with the DeustoTech–Internet Research Group, Deusto Foundation. His research interests include the analysis of the behavior of the users in intelligent environments and the study the users activity and discourse on social networks.