



**HAL**  
open science

# Human Activity Discovery and Recognition using Probabilistic Finite-State Automata

K Viard, M. P. Fanti, G. Faraut, J-J Lesage

► **To cite this version:**

K Viard, M. P. Fanti, G. Faraut, J-J Lesage. Human Activity Discovery and Recognition using Probabilistic Finite-State Automata. *IEEE Transactions on Automation Science and Engineering*, In press, 17 (4), pp. 2085-2096. hal-02557589

**HAL Id: hal-02557589**

**<https://hal.science/hal-02557589>**

Submitted on 28 Apr 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Human Activity Discovery and Recognition Using Probabilistic Finite-State Automata

K. Viard, M. P. Fanti, *Fellow, IEEE*, G. Faraut, *Member, IEEE*, and J-J. Lesage, *Member, IEEE*

**Abstract**— Ambient assisted living and smart home technologies are a good way to take care of dependant people whose number will increase in the future. They allow the discovery and the recognition of human’s Activities of Daily Living (ADLs) in order to take care of people by keeping them in their home. In order to consider the human behaviour non-determinism, probabilistic approaches are used despite difficulties encountered in model generation and probabilistic indicators computing. In this paper, a global method, based on Probabilistic Finite-State Automata and the definition of the normalised likelihood and perplexity is proposed to manage ADLs discovery and recognition. In order to reduce the computational complexity, some results about a simplified normalised likelihood computation are proved. A real case study showing the efficiency of the proposed method is discussed.

**Note to Practitioners**— This paper is motivated by the problem of the automatic recognition of activities that are daily performed by elderly or disabled people in a smart dwelling. The set of activities to be recognized is defined by a medical staff (e.g. to prepare meal, to do housework, to take leisure, etc.) and correspond to pathologies that have to be monitored by doctors (e.g. loss of memory, loss of mobility, etc.). The proposed method is based on a systematic procedure of offline construction of a model for each activity to be monitored (the activity discovering step). The online recognition of activities actually performed (the activity recognition step) is afterwards based on these models of activities. Since the human behaviour is non-deterministic, and may even be irrational, probabilistic activity models are built from a learning database. In the same way, probabilistic indicators are used for determining online the most probable activities actually performed. The efficiency of the proposed approach is illustrated through a case study performed in a smart living lab.

**Index Terms**— Smart Home, Activity of Daily Living, Activity Discovery, Activity Recognition, Probabilistic Finite-State Automata, Normalised Likelihood.

Manuscript received March 5, 2018.

K. Viard, J-J. Lesage and G. Faraut are with the Université Paris-Saclay, ENS Paris-Saclay, LURPA, 94235, Cachan, France (e-mail: {kevin.viard, jean-jacques.lesage, gregory.faraut}@ens-paris-saclay.fr)

M. P. Fanti is with Dipartimento di Ingegneria Elettrica e dell’Informazione, Politecnico di Bari, Via Orabona 4, 70125 Bari, Italy (phone: 39-080-5963643; fax: 39-080-5963410; e-mail: [mariafia.fanti@poliba.it](mailto:mariafia.fanti@poliba.it)).

## I. INTRODUCTION

ACCORDING to several demography studies, the dependency rate of the world population is continuously increasing since 2010. In 2050, the part of the population aged 60 or more will rise to 30% in the majority of countries [1]. This societal evolution is becoming an important human and economic issue for next years. In fact, current health and well-fare institutions will not be sufficient to treat this proportion of elderly people. Hence, severe pressure on the public healthcare sector and lack of adequate facilities are driving the way in which health services are delivered to the patients [2, 3]. Therefore, alternative solutions have to be found and rapidly developed in order to supply help and independence to people suffering from not too severe pathologies.

Smart homes, which integrate medical equipment and other ambient assisted living technologies, can play a lead role in revolutionizing the way in which healthcare services are being provided to the elderly people [4, 5]. Health at home systems is an efficient possible solution that consists of keeping old people at home as long as possible, thanks to an automatic monitoring of their daily life.

The Activities of Daily Living (ADLs) analysis is one of the main investigation fields in health-assistive smart homes and smart environments [6-15]. To live independently at home, individuals need to be able to complete ADLs such as eating, dressing, cooking, drinking, and taking medicine. Current studies about ADLs mainly treat two problems: Activity Discovery (AD) and Activity Recognition (AR) [6]. In particular, AD is a technique employed to reduce the need for expert knowledge by using learning algorithms to discover activities in sensor events raw sequences [6-8]. In addition, the objective of AR is to detect the activity actually performed by the inhabitant [9-15]. The generally accepted approach to activity recognition is to design and/or use machine learning techniques to map a sequence of sensor events to a corresponding activity label.

In [9, 12] the authors describe all inhabitant activities by only one Hidden Markov Model (HMM), one of the most frequent models used in literature for AD and recognise each activity by applying the Viterbi algorithm [11]. Unfortunately, the complexity of the model drastically increases with the number of activities and sensors. Furthermore, the used model has no intermediary semantic levels between activities and sensors and the precision of the recognition is not guaranteed. In a previous work [8], each activity is modelled by only one

HMM for discovery purpose and in [10] a new probabilistic indicator for AR, called normalised likelihood, is proposed.

In [13-15], the authors present a system that recognises a set of activities modelled by HMMs. Moreover, activities are classified by a probability that allows recognising the performed activity as being the one, which is represented by the most probable model. Typically, the works focusing on recognition give few details about how the probabilistic models are built. By using a similar model, paper [16] addresses the problem of recognising ADLs in smart homes in a Hidden Semi-Markov Model (HSMM) framework. In addition, comparing different kinds of HMMs, the authors show that the proposed novel form of HSMM, called Coaxian Hidden Semi-Markov Model, performs online activity classification and segmentation from a segmented training data. The performance of the model is compared with various counterparts by a likelihood computation that can be used only if the systems exhibit the same event sets. On the contrary, typically activities are linked to different events (i.e., sensors) because they occur in different home areas and are detected by using equipment located in different areas. A common drawback of these approaches is that they do not explain how the probabilistic model of scenarios is built and, typically, they are manually built by an expert.

This paper proposes a novel approach for AR and AD in order to fill the gap that is present in the works of the related literature. Indeed, most contributions suggest AR methodologies, but they do not give precise information on the way activities are discovered and whether the recognition works well when there are variations during the performance of the activities.

The novelty of this paper is threefold.

First, the activities are modelled by using Probabilistic Finite State Automata (PFA), a superclass of HMMs [17] that are powerful modelling techniques when the system states are partially or completely unknown. On the other hand, PFA models are chosen for three main reasons: i) the structure of the PFA can be automatically deducted from the input data of AD, hence, the use of HMMs is not necessary; ii) the proposed framework allows automatically building the models of scenarios by overcoming the drawback of the Markov models; iii) the PFA allows exploiting existing tools and consolidated theoretical results that give the possibility of automatically building the activity models (see for example the algorithms presented in [30-31]).

Second, in order to identify ADLs linked to different sets of events and having some non-deterministic variations, a new method is developed to allow distinguishing activities. To this aim, we adopt the perplexity evaluation by introducing the normalized likelihood to select the most probable activity. This methodology allows overcoming the limit of considering models sharing the same set of events. Moreover, in order to reduce the computational complexity some results simplifying the normalised likelihood computation are proved.

Third, the presented methods are applied to a real smart flat provided by the ENS Paris-Saclay (France) and experimental results are discussed for both AD and AR methodologies.

Finally, an important benefit of the proposed approach concerns the collection of the needed expert data in comparison

with the methods of the related literature. Indeed, the proposed method allows building the AD module and performing the recognition by using short strings of data: hence, the approach can be suitably applied in real situations.

In this paper, assumptions and related works are first presented in Section II and Section III proposes a formal statement of the problem. Section IV and V present the AD and AR methods, respectively, and Section VI discusses a real case study. Finally, Section VII draws the conclusions.

## II. LITERATURE REVIEW AND ASSUMPTIONS

In this section we start by listing the main assumptions of the proposed framework, in relation with the results and the contributions of the related literature. The methodology proposed for solving the AR and AD problems are based on the following assumptions:

- 1) only binary sensors are used for observing patient activities;
- 2) human behaviour is non-deterministic and may even be irrational;
- 3) the smart dwelling is supposed to be occupied by a single inhabitant;
- 4) the database which has been recorded during the learning period, and which is the input data of AD, does not include the knowledge of activities which have been actually performed.

The rationality of considering such assumptions is based on the following justifications that are discussed on the basis of the literature review.

### *Assumption 1:*

In most works of the related literature, the use of cameras is needed in at least one step of ADLs monitoring [18-20]. Indeed, cameras provide information of very high level of semantics for discovering activities. Nevertheless, they are often considered too intrusive and raise problems of acceptance by monitored people [21]. This is the main reason why cameras are not used in this work, but only binary sensors, such as motion detectors or door barriers. Such sensors are furthermore low cost, what is very interesting in the context of health at home, which is becoming a mass problem.

### *Assumption 2:*

Human behaviour is non-deterministic and characterized by small changes every day. Therefore, ADLs models which are not robust to minor variations, like data mining approach [7, 22], are non-considered to maximise the robustness and applicability of the proposed method.

### *Assumption 3:*

In order to assume that more inhabitants are in the dwelling, it is necessary to consider that each inhabitant wears a sensor that allows identifying himself (e.g. a RFID sensor) and therefore knowing who has generated which event. Such wearable sensors are very often used for AD and AR [20], but the efficiency of this kind of sensors strongly depends on the ability and the willingness of the patients to wear them every day, and sometimes during the night. As in the case of

cameras, this sensor technology also raises some problems of acceptance and it is sometimes not compatible with the pathology of patients to be monitored (e.g. loss of memory).

Hence, according to Assumption 1, only binary sensors are used for both AD and AR and it is necessary to assume that a single inhabitant is living in the smart home. This assumption is quite restrictive but allows proposing a complete solution for AD and AR that is based on the use of binary sensors only.

#### Assumption 4:

In the main part of the works based on assumptions 1 to 3, a perfect knowledge of activities performed by the inhabitant during the learning period is needed [12, 23]. This information is in practice very difficult to obtain. In [23], the monitored patient indicates what activity is performed. Of course, the efficiency of this approach is compared with the ability and the willingness of the person to declare his activity: in general, numerous reported activities errors are introduced in the database [23]. In other works [19], experts are in charge of enriching the database by studying sensor logs or using cameras exclusively during the learning phase. This approach is expensive, intrusive and not accurate. In both cases, the labelling step is difficult and unreliable: in this paper the AD does not require the knowledge of actually performed activities during the learning phase.

### III. FRAMEWORK AND NOTATIONS FOR THE ACTIVITY DISCOVERY AND RECOGNITION

In this section we describe the main components of the proposed discovery and recognition framework that allows automatically building models representing activities, models that are used to recognize the ADL performed by the inhabitant. This framework has to consider the privacy of the inhabitant, the non-determinism of the human behaviour, and the different ways to perform the same ADL, i.e. the slight variations to perform an activity has to lead to a recognition of the activity. Hence, we describe the structure of the AD and AR tasks and the used notations for the definition of the PFA.

#### A. Framework of the Activity Discovery and Recognition

ADLs are regularly performed by a person and can be decomposed into several actions [18, 24, 25]. For instance, “cooking” can be decomposed in “preparing pasta”, “preparing a ready-cook dish”, “ordering meal on the net”, etc. Moreover, actions can be described as a sequence of elementary moves. The hierarchical decomposition of activities in actions and observable moves is represented in Fig. 1. Note that an observable move can be linked to several actions and can be observed by binary sensors.

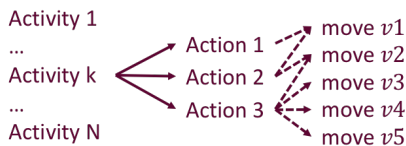


Fig. 1 Hierarchical decomposition of activities in actions and moves.

The structure of the framework developed to perform AD and AR procedures is represented in Fig. 2 showing the main modules of the strategy: the AD and AR modules.

In some preliminary operations, the set of activities to be monitored are determined and the related actions and moves are singled out. On the basis of the considered moves, the sensors are chosen and positioned in the house.

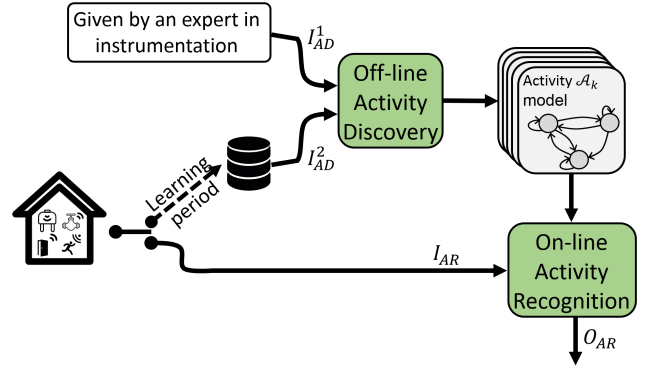


Fig. 2 Structure of the activity discovery and recognition.

The AD module is applied off-line for a learning period with the objective of generating formal models of ADLs. To this aim, in coherence with assumptions presented in Section II, two sets of inputs denoted  $I_{AD}^1$  and  $I_{AD}^2$  are employed:

- $I_{AD}^1$  symbolizes the items that are provided by an expert: 1) the set of activities to be monitored; 2) the set of actions composing each activity; 3) the moves connected with the actions and linked with the sensors in the dwelling;
- $I_{AD}^2$  is a log obtained by observing an inhabitant life during a learning period, i.e., streaming data represented by sequences of events detected by the sensors.

Starting from such inputs, the AD module provides the set  $A = \{\mathcal{A}_k\}$  of PFA models of the activities. Note that in this paper, symbol  $\mathcal{A}_k \in A$  is used to represent both the activity and the PFA modelling this activity.

The AR module works on-line to identify in real time the activity actually performed by the inhabitant. The inputs of this module are the following:

- the observation  $I_{AR}$  provided by the binary sensors of the real-time behaviour of the monitored person;
- the set of the activity models  $A = \{\mathcal{A}_k\}$  obtained by the AD.

The output  $O_{AR}$  of the AR module is the recognition of the activity performed by the inhabitant.

#### B. Probabilistic Finite-State Automata: Notations and Definitions

In order to describe the non-deterministic behaviour of the house inhabitant, the activities are modelled in a PFA framework as follows [17]:

*Definition 1* : A PFA  $\mathcal{A}_k$  is a tuple

$\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$ , where:

- $Q_{\mathcal{A}_k}$  is a finite non-empty set of states  $q$ ;

- $\Sigma_{\mathcal{A}_k}$  is a non-empty alphabet of events  $e$ ;
- $\delta_{\mathcal{A}_k} \subseteq Q_{\mathcal{A}_k} \times \Sigma_{\mathcal{A}_k} \times Q_{\mathcal{A}_k}$  is a set of transitions;
- $I_{\mathcal{A}_k} : Q_{\mathcal{A}_k} \rightarrow [0,1]$ : the initial-state probabilities;
- $P_{\mathcal{A}_k} : \delta_{\mathcal{A}_k} \rightarrow [0,1]$ : the transition probabilities;
- $F_{\mathcal{A}_k} : Q_{\mathcal{A}_k} \rightarrow [0,1]$ : the final-state probabilities.

Note that  $I_{\mathcal{A}_k}$ ,  $P_{\mathcal{A}_k}$  and  $F_{\mathcal{A}_k}$  are functions such that [17]:

$$\sum_{q \in Q_{\mathcal{A}_k}} I_{\mathcal{A}_k}(q) = 1, \quad (1)$$

and

$$\forall q \in Q_{\mathcal{A}_k}, F_{\mathcal{A}_k}(q) + \sum_{e \in \Sigma_{\mathcal{A}_k}, q' \in Q_{\mathcal{A}_k}} P_{\mathcal{A}_k}(q, e, q') = 1. \quad (2)$$

When used in subscript of a symbol,  $\mathcal{A}_k$  represent the PFA to which the symbol is linked.

Now, according to Definition 1, the PFA model of activity  $\mathcal{A}_k$  is specified as follows:

- $q_l \in Q_{\mathcal{A}_k}$  represents an action performed during the activity; transitions starting from this state represent the probabilities to switch from this action to another one;
- $q_0 \in Q_{\mathcal{A}_k}$  is the initial dummy state that we consider as the initial condition of the activities where no action is performed;
- an event  $e_i \in \Sigma_{\mathcal{A}_k}$  describes a move that may occur in a state of activity  $\mathcal{A}_k$ ;
- a transition  $(q_m, e_j, q_s) \in \delta_{\mathcal{A}_k}$  if starting from state  $q_m$  event  $e_j$  may occur and activity  $\mathcal{A}_k$  reaches state  $q_s$ ;
- each activity starts in the initial dummy state  $q_0$  with probability equal to 1, it holds  $I_{\mathcal{A}_k}(q_0) = 1$  and  $I_{\mathcal{A}_k}(q_l) = 0$  for each  $q_l \in Q_{\mathcal{A}_k}$ ;
- $P_{\mathcal{A}_k}(q_m, e_j, q_s)$  is the probability, in activity  $\mathcal{A}_k$ , of observing event  $e_j$  and destination state  $q_s$  starting from state  $q_m$ ;
- $F_{\mathcal{A}_k}(q_m) = 0$  for each  $q_m \in Q_{\mathcal{A}_k}$ .

An example of PFA is illustrated on Fig. 3.

On the basis of Definition 1, several objects can be defined:

- $w = e_i e_j \dots e_n$  is a sequence of observed events and the length of  $w$  (denoted  $|w|$ ) corresponds to the number of events in the sequence;
- $\theta = (q_l, e_i, q_m, e_j, q_s, \dots, q_p, e_n, q_r)$  is a path of transitions for  $w$  in  $\mathcal{A}_k$ , i.e., the sequence of transitions  $(q_l, e_i, q_m), (q_m, e_j, q_s), \dots, (q_p, e_n, q_r) \in \delta_{\mathcal{A}_k}$  consistent with  $w = e_i e_j \dots e_n$ ;
- $\mathcal{L}(w)$  denotes the language, i.e., a set of sequences of events generated from the observed sequence  $w$ ;
- $\Sigma_{\mathcal{A}_k}^m$  is the set of all possible sequences of length  $m$  which can be generated with symbols of the alphabet  $\Sigma_{\mathcal{A}_k}$ . For example, let  $\Sigma_{\mathcal{A}_k} = \{e_1, e_2, e_3\}$  be an alphabet, then  $\Sigma_{\mathcal{A}_k}^3 = \{e_1 e_1 e_1, e_1 e_1 e_2, e_1 e_1 e_3, e_1 e_2 e_1, \dots, e_3 e_3 e_3\}$  is the set of all possible sequences of length 3;

- $w_{p_k} \subset w$  is the projection of sequence  $w$  on alphabet  $\Sigma_{\mathcal{A}_k}$  of activity  $\mathcal{A}_k$ , where the projection of  $w \in \Sigma^*$  on alphabet  $\Sigma_{\mathcal{A}_k}$  is defined as follows [29]:

$$Proj : \Sigma^* \rightarrow \Sigma_{\mathcal{A}_k}^* \quad (3)$$

with:

$$Proj(\epsilon) := \epsilon$$

$$Proj(e) := \begin{cases} e & \text{if } e \in \Sigma_{\mathcal{A}_k} \\ \epsilon & \text{if } e \in \Sigma \setminus \Sigma_{\mathcal{A}_k} \end{cases}$$

$$Proj(w) := Proj(w)Proj(e) \text{ for } w \in \Sigma^*, e \in \Sigma_{\mathcal{A}_k}$$

where  $\Sigma$  is the set of system events,  $\Sigma^*$  and  $\Sigma_{\mathcal{A}_k}^*$  represent the Kleene-closure [29] of  $\Sigma$  and  $\Sigma_{\mathcal{A}_k}$ , respectively, and  $\epsilon$  is the empty symbol, i.e., the sequence of length 0. In other words, the mathematical projection function allows obtaining a sequence including only events of a specific activity  $\mathcal{A}_k$ . For instance, let  $w = e_1 e_2 e_1 e_3 e_2 e_4 e_1 e_5 e_2 e_3$  be an observed sequence: the projection of  $w$  on  $\Sigma_{\mathcal{A}_k} = \{e_1, e_3, e_5\}$  is the sequence  $w_{p_k} = e_1 e_1 e_3 e_1 e_5 e_3$ .

- $\mathcal{L}(w)$  is a language generated from sequence  $w$  and composed by all substrings of  $w$  such that  $|w| \geq 2$ .

#### IV. ACTIVITY DISCOVERY

In this section the AD method to model ADLs in a PFA framework is developed by following three steps: 1) generating the PFA structure; 2) analysing and processing a set of streaming data; 3) computing the probabilities of the PFA model.

##### A. Generation of the PFA Structure

Starting from the hierarchical description of each activity (given in Fig. 1), the PFA structure of each activity  $\mathcal{A}_k$  is determined by building the associated transition digraph (direct graph)  $D(\mathcal{A}_k) = (Q_{\mathcal{A}_k}, \delta_{\mathcal{A}_k})$ . More precisely, the set of nodes of  $D(\mathcal{A}_k)$  corresponds to the states set  $Q_{\mathcal{A}_k}$  and the set of direct arcs is associated with the transitions in  $\delta_{\mathcal{A}_k}$ , i.e., there exists a direct arc starting from node  $q_g$  and ending to node  $q_h$ , if  $(q_m, e_j, q_s) \in \delta_{\mathcal{A}_k}$ . Fig. 3 shows an example of digraph describing the structure of an activity.

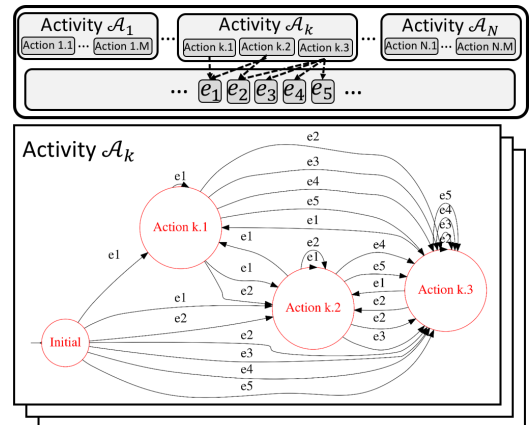


Fig. 3 Generation of model structure from the activities semantic description.

### B. Analysis and Process of Streaming Data

This section briefly describes the approach for processing the streaming data in the context of the smart home dataset. The sensors embedded in the considered smart apartments are binary sensors that are in two states - ‘ON’ and ‘OFF’. Then they can generate two events: one is linked to the rising edge of the binary information (sensor|1), the other one is linked to the falling edge (sensor|0) (see as an example Fig. 4).

In the related literature different approaches are proposed for processing streaming data represented by sequences of the sensor firings [26, 27]. The *explicit windowing* is not adapted to our hypotheses because it requires an expert to segment sequence. The *time windowing* could be interesting in case of timed approach, but our approach is event-based. Furthermore, with time windowing, some windows could have no event. By consequence, we choose the sensor *event-based windowing* approach that consists in dividing the sequence into windows containing a fixed number of sensor events. In this case, the windows vary in their duration and this is fine considering that during the performance of activities, multiple sensors could be triggered, while during silent periods, there will not be many sensor firings. Hence, we consider a long observation that we divide in windows  $w$  containing a fixed number of events and we denote by  $w_{p_k}$  the projection of  $w$  on activity alphabet  $\Sigma_{\mathcal{A}_k}$ .

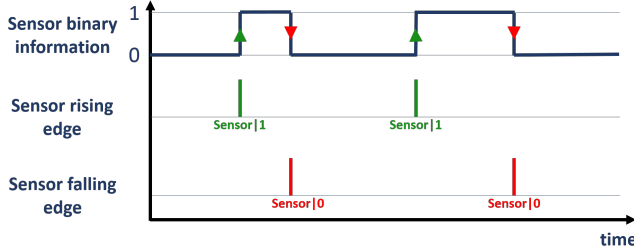


Fig. 4. Event emission from sensor binary information.

### C. Probabilities Computation

The goal of this subsection is to show how to compute the probabilities associated to each transition of the PFA model, that is the output of the AD module.

Let  $\mathcal{A}_k$  be the PFA of the activity  $\mathcal{A}_k$ ; let  $e_i, e_j \in \Sigma_{\mathcal{A}_k}$  be two events of the activity  $\mathcal{A}_k$ ; let  $w_{p_k}$  be a sequence of events of  $\Sigma_{\mathcal{A}_k}$  obtained by projection function of the sequence  $w$  on alphabet  $\Sigma_{\mathcal{A}_k}$ ; let  $q_l, q_m \in Q_{\mathcal{A}_k}$  be two actions performed in the activity  $\mathcal{A}_k$ . The probability to move from action  $q_l$  into action  $q_m$  by the event  $e_i$  is defined by:

$$P(q_l, e_i, q_m) = P(q_l \rightarrow q_m | q_l) \times P(e_i | q_l \rightarrow q_m, q_l). \quad (4)$$

In words, the probability to move from  $q_l$  to  $q_m$  by event  $e_i$  is defined by the probability to move from  $q_l$  to  $q_m$  multiplied by the probability to generate  $e_i$  during this move.

Moreover, the probabilities of the equation (4) are computed by a standard approach: the likelihood of one or more events happening divided by the number of possible outcomes. More precisely, it holds:

$$P(q_l \rightarrow q_m | q_l) = \frac{\tilde{N}(q_l \rightarrow q_m | q_l)}{\sum_{q_n \in Q_{\mathcal{A}_k}} \tilde{N}(q_l \rightarrow q_n | q_l)} \quad (5)$$

$$P(e_i | q_l \rightarrow q_m, q_l) = \frac{\tilde{N}(e_i | q_l \rightarrow q_m)}{\sum_{e_j \in \Sigma_{\mathcal{A}_k}} \tilde{N}(e_j | q_l \rightarrow q_m)} \quad (6)$$

where  $\tilde{N}(q_l \rightarrow q_m | q_l)$  denotes the number of occurrences of transitions from  $q_l$  to  $q_m$  and  $\tilde{N}(e_i | q_l \rightarrow q_m)$  the occurrence of event  $e_i$  conditioned to the transition  $q_l \rightarrow q_m$  during the run period.

Unfortunately, the number of the performed actions in each activity is unknown. By consequence, we can only consider the sequence  $w_{p_k}$  to evaluate probabilities (5) and (6). Then, in order to compute  $\tilde{N}(q_l \rightarrow q_m | q_l)$  and  $\tilde{N}(e_i | q_l \rightarrow q_m)$ , we use observable occurrences or successions of events by the following indicators:

- $N_{e_i}^k$ : number of times event  $e_i$  is observed in the sequences  $w_{p_k}$ ;
- $N_{init\ e_i}^k$ : number of times event  $e_i$  is observed as first event in the sequences  $w_{p_k}$ ;
- $N_{e_i \rightarrow e_j}^k$ : number of times event  $e_j$  follows event  $e_i$  in the sequences  $w_{p_k}$ .

Let  $C_{e_i}$  (resp.  $C_{e_j}$ ) be the number of actions (states) having the event  $e_i$  (resp.  $e_j$ ) as input, and let  $\Sigma_{q_l}$  (resp.  $\Sigma_{q_m}$ ) be the set of events that are in input of state  $q_l$  (resp.  $q_m$ ). Now,  $\tilde{N}(q_l \rightarrow q_m | q_l)$  and  $\tilde{N}(e_i | q_l \rightarrow q_m)$  are determined as follows:

$$\tilde{N}(q_l \rightarrow q_m | q_l) = \begin{cases} \sum_{e_i \in \Sigma_{q_m}} \frac{1}{C_{e_i}} \times N_{init\ e_i}^k & \text{if } q_l = q_0 \\ \sum_{e_i, e_j \in \Sigma_{q_l}} N_{e_i \rightarrow e_j}^k & \text{if } q_l = q_m \neq q_0 \\ \sum_{e_i \in \Sigma_{q_l}; e_j \in \Sigma_{q_m}; e_j \notin \Sigma_{q_l}} \frac{1}{C_{e_j}} \times N_{e_i \rightarrow e_j}^k & \text{if } q_m \neq q_l \neq q_0 \end{cases} \quad (7)$$

and

$$\tilde{N}(e_i | q_l \rightarrow q_m) = N_{e_i}^k \text{ if } e_i \in \Sigma_{q_m}; 0 \text{ otherwise.} \quad (8)$$

In equations (7) and (8) we consider the fact that when a human starts a new action, the performed move is not dependent on the past action. In equation (7), the number of occurrences of transitions from  $q_l$  to  $q_m$  is determined considering 3 cases: i) if  $q_l = q_0$  we consider the number of times event  $e_i$  that is in input of  $q_m$  is observed as first event, averaged by the number of states having the event  $e_i$  as input; ii) if  $q_l = q_m \neq q_0$  we count the number of times event  $e_j$  follows event  $e_i$  where  $e_j$  and  $e_i$  are in input of  $q_l$ ; iii) if  $q_m \neq q_l \neq q_0$  then the count is similar to case i) but for the computation of  $N_{e_i \rightarrow e_j}^k$  that represents number of times  $e_j$  follows  $e_i$  in the considered sequence.

Moreover, in equation (8), we consider the occurrence of  $e_i$  independent of the starting state  $q_l$  but it depends only on the reached state  $q_m$ . Hence, the occurrence of event  $e_i$  conditioned to the transition  $q_l \rightarrow q_m$  during the run period is equal to

number of times  $e_i$ , which is in input of  $q_m$ , is observed during the run period.

#### D. Complexity of the AD algorithm

The complexity of the three steps of the AD algorithm is the following:

$$C_{AD} = O\left(\left(|LearningDB| - |w|\right) \left(\max_{\mathcal{A}_k} \left(\text{card}(Q_{\mathcal{A}_k})\right)\right)^2 + \max_{\mathcal{A}_k} \left(\text{card}(\Sigma_{\mathcal{A}_k})\right) \times \text{card}(A)\right)$$

where:

- symbol  $\text{card}(\cdot)$  stands for “cardinality of the set ( $\cdot$ );
- $|LearningDB| - |w|$  is the number of windows obtained by sliding;
- $\max_{\mathcal{A}_k} \left(\text{card}(Q_{\mathcal{A}_k})\right)$  is the maximum number of actions linked to the same activity;
- $\max_{\mathcal{A}_k} \left(\text{card}(\Sigma_{\mathcal{A}_k})\right)$  is the maximum number of events linked to the same activity.

Note that the complexity of the AD algorithm is polynomial and mainly depends on the maximum number of actions linked to the same activity. Since the number of actions is not large, the algorithm is easily computable.

## V. ACTIVITY RECOGNITION

### A. Recognition protocol

The AR consists in recognising on-line (i.e., in reasonable time), which activity is actually performed by the inhabitant.

Recognition is done by computing an indicator evaluating the probability that a PFA generates an observed sequence  $w$ . The activity having the maximum probability of generating sequence  $w$  can be considered as being the activity currently performed. In [17], perplexity is presented as a useful indicator to compute distances between a language  $\mathcal{L}(w)$  generated from the observed sequence  $w$  and a PFA  $\mathcal{A}_k$ .

#### Definition 2 : Likelihood

Let  $\Theta_{\mathcal{A}_k}(w)$  be the set of paths for  $w$  in  $\mathcal{A}_k$ . The probability of generating  $w$  with  $\mathcal{A}_k$  is the *likelihood* of  $w$  considering  $\mathcal{A}_k$  and can be computed by:

$$P_{\mathcal{A}_k}(w) = \sum_{\theta \in \Theta_{\mathcal{A}_k}(w)} P_{\mathcal{A}_k}(\theta). \quad (9)$$

Note that  $P_{\mathcal{A}_k}(\theta)$  is the probability of the sequence of transitions  $\theta = (q_l, e_i, q_m), (q_m, e_j, q_s), \dots, (q_p, e_n, q_r)$ , consistent with  $w = e_i e_j \dots e_n$  in  $\mathcal{A}_k$  and calculated as  $P_{\mathcal{A}_k}(\theta) = P(q_l, e_i, q_m) P(q_m, e_j, q_s) \dots P(q_p, e_n, q_r)$ , assuming the independence among transitions.

Two definitions of the perplexity are reported [17]: perplexity “per string” and perplexity “per symbols”. Here, we consider the perplexity “per string”.

#### Definition 3 : Perplexity

The perplexity “per string” is defined as the inverse of the geometric mean of the likelihood:

$$PP(\mathcal{L}(w)|\mathcal{A}_k) = \left[ \prod_{v \in \mathcal{L}(w)} P_{\mathcal{A}_k}(v) \right]^{-\frac{1}{\text{card}(\mathcal{L}(w))}}. \quad (10)$$

The perplexity is based on the computation of likelihood  $P_{\mathcal{A}_k}(v)$  for  $v \in \mathcal{L}(w)$  and can be used to distinguish between two automata  $\mathcal{A}_1$  and  $\mathcal{A}_2$  only if the same sequence  $v$  is used for the two models; i.e., if  $v \in \Sigma_{\mathcal{A}_1}^*$  and  $v \in \Sigma_{\mathcal{A}_2}^*$ . However, the values of the perplexity are not significant if the alphabet of the sequence is not included in the alphabet of automaton.

In the considered problem two issues prevent applying the standard likelihood-based approaches. First, all the PFAs  $\mathcal{A}_k$  do not share the same alphabet  $\Sigma_{\mathcal{A}_k}$ . Second, it is necessary to project the observed sequence on the PFA’s alphabet in order to filter the events not belonging to the PFA alphabet.

In order to overcome these two issues, we propose an AR protocol that is composed by four steps:

1 – the windowing of observed events considers a sequence  $w$  composed with a fix number of events;

2 – for each PFA  $\mathcal{A}_k$ , the projection  $w_{p_{\mathcal{A}_k}}$  of the considered sequence  $w$  is obtained;

3 – a language  $\mathcal{L}(w_{p_{\mathcal{A}_k}})$  is generated for each projected sequence  $w_{p_{\mathcal{A}_k}}$  as described in subsection III.B;

4 – the probability for each model  $\mathcal{A}_k$  to generate the language  $\mathcal{L}(w_{p_{\mathcal{A}_k}})$  is computed by likelihood algorithm.

### B. Normalised likelihood and perplexity

Note that methods based on the likelihood computation are not pertinent to the projected sequences since it can lead to compare likelihood or perplexity of sequences having different lengths. The risk is the shorter sequences will always more probable than longer ones as a consequence of the projection of the observed sequences. In order to be able to compare sequences having different length, we propose a normalisation of the classical likelihood computation.

#### Definition 4 : The normalised likelihood

Let us consider the PFA  $\mathcal{A}_k$  and a given sequence  $w \in \Sigma_{\mathcal{A}_k}^*$ . The *normalised likelihood* of sequence  $w$  in  $\mathcal{A}_k$ , is defined as:

$$\|P_{\mathcal{A}_k}(w)\| = \frac{P_{\mathcal{A}_k}(w)}{\max_{v \in \Sigma_{\mathcal{A}_k}^*} [P_{\mathcal{A}_k}(v)]}. \quad (11)$$

In other words, for a given length of the sequences, the normalised likelihood normalises the probability of a sequence with regards to the probability of the sequence having the highest probability to be generated by the PFA. This is to address the problem of the probability which decreases with the sequence size.

Analogously, the normalised perplexity is defined as follows.

*Definition 5 : The normalised perplexity*

Let us consider the PFA  $\mathcal{A}_k$ , a language  $\mathcal{L}(w)$  generated from the observed sequence  $w$  and the normalised likelihood  $\|P_{\mathcal{A}_k}(v)\|$  of  $v \in \mathcal{L}(w)$ . We define the normalised perplexity “per string” as the inverse of the geometric mean of the normalised likelihood:

$$\|PP(\mathcal{L}(w)|\mathcal{A}_k)\| = \left[ \prod_{v \in \mathcal{L}(w)} \|P_{\mathcal{A}_k}(v)\| \right]^{\frac{1}{\text{card}(\mathcal{L}(w))}}. \quad (12)$$

Since the perplexity (resp. normalised perplexity) represents the distance between a language and the probability that it is generated by a model, we impose minimizing its value. Moreover, the inverse of the perplexity (resp. normalised perplexity) represents the probability, for the considered model, to generate the considered sequence and has to be maximised. In order to simplify the computation, maximizing the inverse of the normalised perplexity  $\left(\frac{1}{\|PP(\mathcal{L}(w)|\mathcal{A}_k)\|}\right)$  is preferred.

*C. Normalised likelihood computation*

The computation of the normalised likelihood has a high complexity. In particular, the complexity of the computation of  $P_{\mathcal{A}_k}(w)$  is polynomial with  $|w|$  thanks to the optimised forward algorithm [28]:  $C_L = O(\text{card}(Q_{\mathcal{A}_k})^2 \times |w|)$ .

On the other hand, the complexity of  $\max_{v \in \Sigma^{|w|}} [P_{\mathcal{A}_k}(v)]$  is exponential with  $|w|$  and polynomial with the event set cardinality:

$$C_M = O(\text{card}(\Sigma_{\mathcal{A}_k})^{|w|} \times \text{card}(Q_{\mathcal{A}_k})^2 \times |w|). \quad (13)$$

However, the normalised likelihood computation can be decomposed in two parts corresponding to two different steps of the computation:

1.  $P_{\mathcal{A}_k}(w)$  can be computed on-line when the estimation is done;
2.  $\max_{v \in \Sigma^{|w|}} [P_{\mathcal{A}_k}(v)]$  can be computed off-line for each  $\mathcal{A}_k$  and for all possible values of  $|w|$ .

Even if the maximum likelihood is computed off-line, the computational effort can be too high, and the complexity of this step has to be reduced.

Two complementary methods of complexity reduction are presented: model reduction and dynamic computation.

*1) Complexity Reduction by Model Reduction*

In order to reduce the computational complexity, a reduction of the model can be envisaged. Indeed, by keeping, for each couple of states, only the transitions with higher probabilities, it is possible to obtain an abstraction of the considered PFA having the same maximum of likelihood.

In the following, we define the reduced PFA and the rules to obtain it.

*Definition 6 : The reduced PFA*

Let  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  be a PFA, we denote by  $\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle$  the *reduced PFA* associated with  $\mathcal{A}_k$  where  $\Sigma_{\mathcal{A}_k}^r$ ,  $\delta_{\mathcal{A}_k}^r$  and  $P_{\mathcal{A}_k}^r$  are obtained by the *Reduction Procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ .

In order to specify the *Reduction Procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ , the following definitions are necessary.

First, we denote by  $Geq_{q_l, q_m}(e_i)$  (resp.  $Equ_{q_l, q_m}(e_i)$ ) the set of events  $e_j \in \Sigma_{\mathcal{A}_k}$  having probability  $P(q_l, e_j, q_m)$  to occur from state  $q_l$  to state  $q_m$  greater than or equal to (resp. equal to)  $e_i \in \Sigma_{\mathcal{A}_k}$  having a probability  $P(q_l, e_i, q_m)$  to occur. More formally, it holds:

$$Geq_{q_l, q_m}(e_i) = \{e_j | e_j \in \Sigma_{\mathcal{A}_k} \text{ and } P(q_l, e_j, q_m) \geq P(q_l, e_i, q_m)\}, \quad (14)$$

$$Equ_{q_l, q_m}(e_i) = \{e_j | e_j \in \Sigma_{\mathcal{A}_k} \text{ and } P(q_l, e_j, q_m) = P(q_l, e_i, q_m)\}. \quad (15)$$

In the following, the *reduction procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$  is presented.

*Reduction procedure  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$*

*Step 1: Selection of candidate events*

We first keep events having the same intersection sets of  $Geq$  and  $Equ$  are selected:

$$\Sigma_{\mathcal{A}_k}^r = \left\{ e_j \left| \begin{array}{l} e_j \in \Sigma_{\mathcal{A}_k} \text{ and } \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Geq_{q_l, q_m}(e_j) \\ = \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Equ_{q_l, q_m}(e_j) \end{array} \right. \right\} \quad (16)$$

*Step 2: Deletion of equivalent events*

For all the event sets  $\bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Equ_{q_l, q_m}(e_j)$ , only one event is kept: a new  $\Sigma_{\mathcal{A}_k}^r$  is thus obtained.

*Step 3: Conservation of transitions linked to the kept events*

Only transitions implying kept events are conserved, all the other are deleted. Probability of those transitions are not changed.

$$\delta_{\mathcal{A}_k}^r = \{(q_l, e_j, q_m) | (q_l, e_j, q_m) \in \delta_{\mathcal{A}_k} \text{ and } e_j \in \Sigma_{\mathcal{A}_k}^r\} \quad (17)$$

$$P_{\mathcal{A}_k}^r = \{P(q_l, e_j, q_m) | (q_l, e_j, q_m) \in \delta_{\mathcal{A}_k}^r \text{ and } P(q_l, e_j, q_m) \in P_{\mathcal{A}_k}\} \quad (18)$$

The problem reduction leads to a new model with a lower number of events than the original one. Thus, the number of combinations to compare in the determination of the normalised perplexity is reduced. The following proposition proves that the likelihood maximum value is conserved after the reduction.

**Proposition 1:**

Let  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  be a PFA and



$\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle$  be the *reduced PFA* obtained by the *Reduction Procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ . Then  $\forall w \in \Sigma_{\mathcal{A}}^*$  of length  $|w|$  it holds:

$$\max_{u \in \Sigma_{\mathcal{A}_k^r}^{|w|}} [P_{\mathcal{A}_k^r}(u)] = \max_{v \in \Sigma_{\mathcal{A}_k}^{|w|}} [P_{\mathcal{A}_k}(v)]. \quad (19)$$

The proof is in Appendix.

By applying the *Reduction Procedure* and thanks to Proposition 1, the computational complexity of the normalised likelihood is reduced by substituting  $\Sigma_{\mathcal{A}_k}$  with  $\Sigma_{\mathcal{A}_k^r}$  in equation (13).

Now, the following proposition shows that the complexity can be further reduced by the *Reduction Procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ .

**Proposition 2:**

Let  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  be a PFA and  $\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle$  be the *reduced PFA* obtained by the *Reduction Procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ . Then, the computational complexity of the normalised likelihood is the following:

$$C_M = O(2^{[card(Q_{\mathcal{A}_k)}-1]^{|w|}} \times card(Q_{\mathcal{A}_k})^2 \times |w|). \quad (20)$$

The proof is in Appendix.

2) *Complexity reduction by dynamic computation*

In addition to the model reduction, a computational simplification can be employed. In fact, it is necessary to find the  $\max_{v \in \Sigma_{\mathcal{A}_k}^{|w|}} [P_{\mathcal{A}_k}(v)]$ .

To this aim, the probabilities  $P[v_2, q_{final} = q_i]$  to generate a subsequence  $v_2$  of the considered  $v$  to reach state  $q_{final} = q_i$  can be computed only one time for all sequences  $v \in \Sigma_{\mathcal{A}_k}^{|w|}$  sharing  $v_2$ . This dynamic reduction removes the linear component  $|w|$  in equation (20) as follows:

$$C_M = O(2^{[card(Q_{\mathcal{A}_k)}-1]^{|w|}} \times card(Q_{\mathcal{A}_k})^2). \quad (21)$$

**Remark 1:**

It is important to note that equation (21) represents the complexity of the AR algorithm that is reduced by the proposed model reduction and dynamic computation. Moreover, also in this case the complexity is function of the number of actions connected with an activity. However, since a complex activity may be decomposed in no more than ten/twelve actions and the sliding window length is selected by the observer, the proposed AR algorithm is suitable for real-time computation.

## VI. CASE STUDY

In this section, a case study is discussed by considering a real flat and some experiments performed by experts in order to show the application of the presented AD and AR approaches. To this aim, we describe in detail the considered flat, the choice and the location of the sensors, the semantic classification of the activities and the used database. The experiments are

implemented by considering the real life of the flat not elderly inhabitant.

### A. Smart Flat Description

The studied flat is provided by the ENS Paris-Saclay in France and is composed of two rooms that can be divided into four zones: entrance, bathroom, kitchen and sleeping zone (see Fig. 5 and 6). The flat is equipped with 21 binary sensors positioned as shown in Fig. 6.

Each one is denoted by an explicit name and can generate two events: one linked to rising edge and one linked to the falling edge (see Fig. 4). The list of the sensor events used in this case study is given in Table I

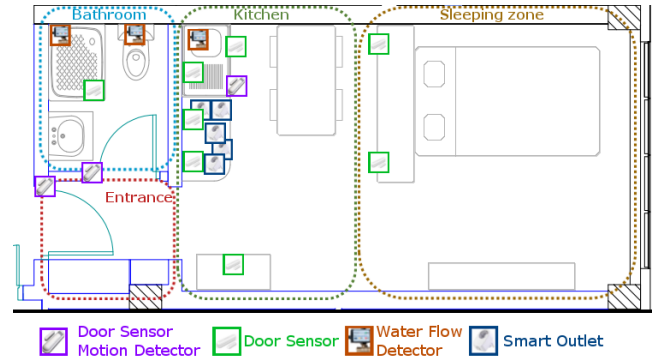


Fig. 5. The smart flat and the positioned binary sensors.

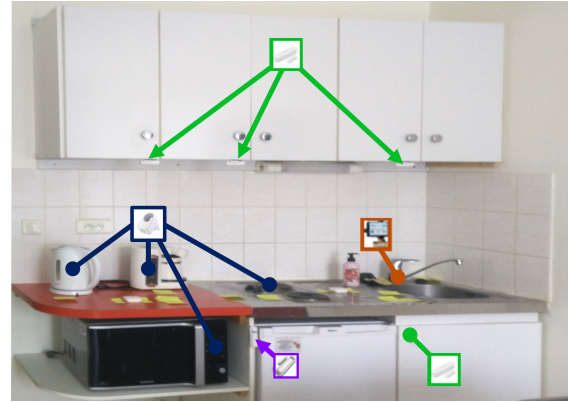


Fig. 6. Kitchen view of the smart flat.

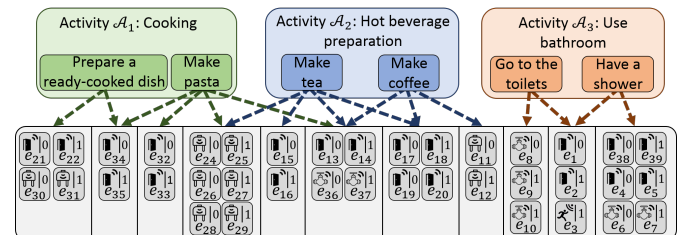


Fig. 7. Semantic decomposition of activities in actions and moves.

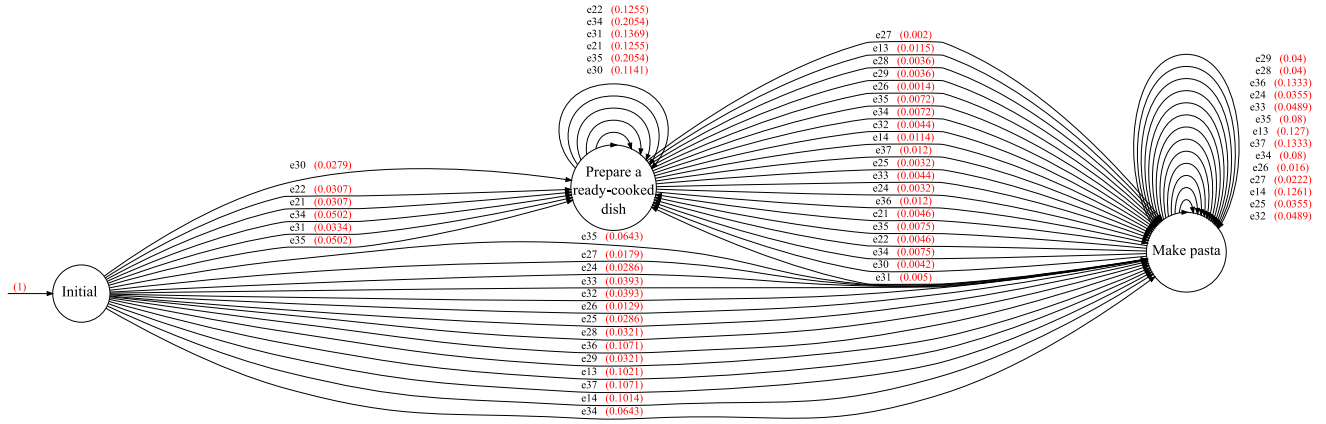


Fig. 9 Digraph  $D(\mathcal{A}_1)$ : Cooking modelled by a PFA. In black: the structure resulting from AD: step 1, in red: probabilities computed in AD: step 3.

### B. Activities to be monitored

In the case study, we consider 3 activities, 6 actions and 42 observable moves (events). The activities analysed for the ADL analysis are  $\mathcal{A}_1$ ="Cooking",  $\mathcal{A}_2$ ="Hot beverage preparation" and  $\mathcal{A}_3$ ="Use bathroom" as Fig. 7 shows. Moreover, each activity is described by two states (actions) and each state is connected with a set of events (Fig. 7).

Note that activities  $\mathcal{A}_1$  and  $\mathcal{A}_2$  share some sensors detecting the actions "Make pasta" and "Make tea": for instance, the events "boil water" and "using hotplates" are connected with both actions. However, since activity  $\mathcal{A}_3$  is carried out in a different area of the flat (i.e., the bathroom), the events linked to  $\mathcal{A}_3$  are fully independent from other activities. The list of events connected with moves and sensors is given in Table I.

In order to estimate the robustness of the approach, activities  $\mathcal{A}_k$  with  $k = 1, 2, 3$  are observed a large number of times by introducing the following variations:

- the insertion of noisy events (i.e., events not included in  $\Sigma_{\mathcal{A}_k}$ ) during their realisation, for instance by wandering in the flat,
- some actions are interrupted;
- the execution order of elementary moves composing actions is changed;
- the action "make tea" is realised by two different ways: using the kettle or boiling water with hotplates.

The test database is generated using recorded activity instances placed in a random time order and separated by a random number of random noisy events not belonging to the performed activities. The resulting sequence is composed of 2087 events corresponding to twenty realizations of each activity (Fig. 8).

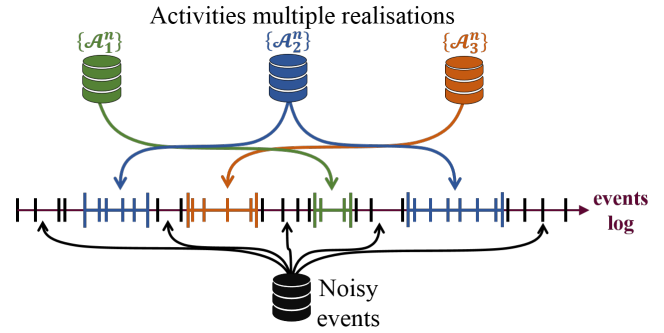


Fig. 8. Structure of the test sequence of observed events.

### C. Activity Discovery

In this section, the presented AD method is applied to the case study and in particular to  $\mathcal{A}_1$ ="Cooking". The learning database is composed of 20 occurrences of each activity.

TABLE I  
LIST OF EVENTS LOGGED IN THE SMART FLAT

Id	Event name	Id	Event name
$e_1$	Bathroom Door_PIR Open 0	$e_{22}$	Kitchen Fridge_PIR Open 1
$e_2$	Bathroom Door_PIR Open 1	$e_{23}$	Kitchen Fridge_PIR Presence 1
$e_3$	Bathroom Door_PIR Presence 1	$e_{24}$	Kitchen Hotplate_bottom Power 0
$e_4$	Bathroom Shower_Door Open 0	$e_{25}$	Kitchen Hotplate_bottom Power 1
$e_5$	Bathroom Shower_Door Open 1	$e_{26}$	Kitchen Hotplate_top Power 0
$e_6$	Bathroom Shower_Waterflow Flow 0	$e_{27}$	Kitchen Hotplate_top Power 1
$e_7$	Bathroom Shower_Waterflow Flow 1	$e_{28}$	Kitchen Kettle Power 0
$e_8$	Bathroom Sink_Waterflow Flow 0	$e_{29}$	Kitchen Kettle Power 1
$e_9$	Bathroom Sink_Waterflow Flow 1	$e_{30}$	Kitchen Microwave_oven Power 0
$e_{10}$	Bathroom Toilets_Waterflow Flow 1	$e_{31}$	Kitchen Microwave_oven Power 1
$e_{11}$	Kitchen Coffee_Machine Power 0	$e_{32}$	Kitchen Sideboard_Left Open 0
$e_{12}$	Kitchen Coffee_Machine Power 1	$e_{33}$	Kitchen Sideboard_Left Open 1
$e_{13}$	Kitchen Cupboard_Bottom Open 0	$e_{34}$	Kitchen Sideboard_Right Open 0
$e_{14}$	Kitchen Cupboard_Bottom Open 1	$e_{35}$	Kitchen Sideboard_Right Open 1
$e_{15}$	Kitchen Cupboard_CenterLeft Open 0	$e_{36}$	Kitchen Sink_Waterflow Flow 0
$e_{16}$	Kitchen Cupboard_CenterLeft Open 1	$e_{37}$	Kitchen Sink_Waterflow Flow 1
$e_{17}$	Kitchen Cupboard_CenterRight Open 0	$e_{38}$	Kitchen Wardrobe Open 0
$e_{18}$	Kitchen Cupboard_CenterRight Open 1	$e_{39}$	Kitchen Wardrobe Open 1
$e_{19}$	Kitchen Cupboard_Left Open 0	$e_{40}$	Entrance Door_PIR Open 0
$e_{20}$	Kitchen Cupboard_Left Open 1	$e_{41}$	Entrance Door_PIR Open 1
$e_{21}$	Kitchen Fridge_PIR Open 0	$e_{42}$	Entrance Door_PIR Presence 1

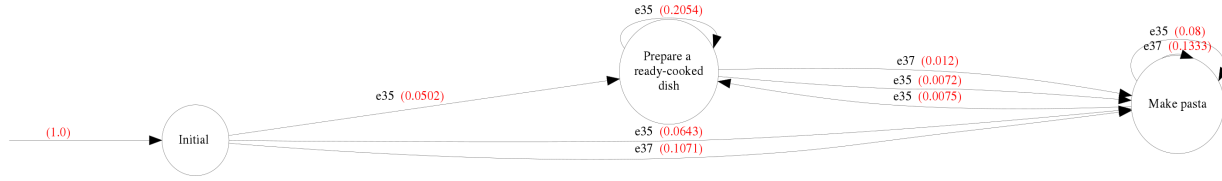


Fig. 11. Activity  $\mathcal{A}_1$ : **Cooking** reduced model for AR off-line maximum likelihood computation.

### Step 1: Generation of PFA Structure

From the semantic decomposition of Fig. 7, the model structure is generated and shown by the black digraph  $D(\mathcal{A}_1)$  of Fig. 9.

### Step 2: Analysis and Process of Streaming Data

In the considered case study, a length of 5 events is considered for the sliding window.

Indicators presented in subsection IV.B are computed for each activity by considering the projected sequence of each sliding window. For instance, in the considered database, some indicators involving sensors  $e_{14}$  and  $e_{35}$  for activity  $\mathcal{A}_1$  are the following:

$$\begin{aligned} N_{e_{14}}^1 &= 142, N_{e_{35}} = 90; \\ N_{init\ e_{14}}^1 &= 112, N_{init\ e_{35}} = 29; \\ N_{e_{14} \rightarrow e_{13}} &= 96, N_{e_{14} \rightarrow e_{26}} = 4, N_{e_{14} \rightarrow e_{27}} = 4, N_{e_{14} \rightarrow e_{24}} = 4, \\ N_{e_{32} \rightarrow e_{14}} &= 3, N_{e_{34} \rightarrow e_{14}} = 10, N_{e_{28} \rightarrow e_{14}} = 5, N_{e_{27} \rightarrow e_{14}} = 4, \\ N_{e_{29} \rightarrow e_{14}} &= 4, N_{e_{24} \rightarrow e_{14}} = 4, N_{e_{35} \rightarrow e_{34}} = 66. \end{aligned}$$

### Step 3: Probabilities Computation

In this step, the PFA probabilities are computed according to equations (4) to (8). At the end of this step, ADLs are fully modelled and the PFA obtained for activity  $\mathcal{A}_1$ ="Cooking" is depicted in Fig. 9.

Finally, the described three steps are applied for each activity  $\mathcal{A}_1$ ,  $\mathcal{A}_2$  and  $\mathcal{A}_3$  but all the resulting models are not presented for the sake of brevity.

### D. Activity Recognition

In this subsection, the proposed AR approach is applied to the case study. For the sake of brevity, only the results obtained during one realisation of the activity "Cooking" are shown.

The observed sequence is the following: ...  $e_{22}e_{21}e_{31}e_{23}e_{35}e_{34}e_{30}e_{23}$  ...

The described steps of Section V.A are applied to the case study.

1 – If a window of length 5 is considered, then the sequences can be the following:

$$\begin{cases} w_1 = e_{22}e_{21}e_{31}e_{23}e_{35} \\ w_2 = e_{21}e_{31}e_{23}e_{35}e_{34}. \\ \dots \end{cases}$$

2 – For each PFA  $\mathcal{A}_k$  with  $k = 1, 2, 3$  modelling the activities, a projection of sequences  $w_i$  for  $i=1, 2$  is obtained and the projected sequence is denoted by  $w_{p,\mathcal{A}_k}$ . For example, the obtained projected sequences of  $w_2$  are:

$$\begin{cases} w_{p,\mathcal{A}_1} = e_{21}e_{31}e_{35}e_{34} \\ w_{p,\mathcal{A}_2} = \emptyset \\ w_{p,\mathcal{A}_3} = \emptyset \end{cases} \quad (22)$$

3 – A language  $\mathcal{L}(w_{p,\mathcal{A}_k})$  is generated for each projected sequence  $w_{p,\mathcal{A}_k}$  with  $k = 1, 2, 3$ . Sequences (22) give:

$$\begin{cases} \mathcal{L}(w_{p,\mathcal{A}_1}) = \{e_{21}e_{31}, e_{31}e_{35}, e_{35}e_{34}, e_{21}e_{31}e_{35}, \\ e_{31}e_{35}e_{34}, e_{21}e_{31}e_{35}e_{34}\} \\ \mathcal{L}(w_{p,\mathcal{A}_2}) = \emptyset \\ \mathcal{L}(w_{p,\mathcal{A}_3}) = \emptyset \end{cases} \quad (23)$$

4 – The inverse of the normalised perplexity for each model  $\mathcal{A}_k$  to generate the language  $\mathcal{L}(w_{p,\mathcal{A}_k})$  is computed. Fig. 10 shows the evolution of this value during the realisation of the "Cooking" activity. Here, in order to validate the method, the log of actually performed activities is compared with the computed estimations. We enlighten that this log is for the validation procedure only and it is not required by the proposed method. The knowledge of performed activity is drawn with plain lines. The probability is equal to 1 when the activity is actually performed. The value of the presented estimator is drawn by the crossed lines.

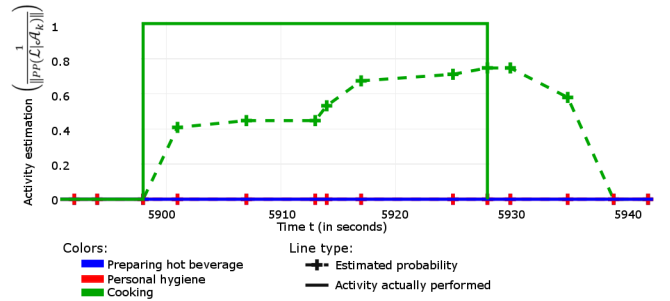


Fig. 10. Inverse of the normalised perplexity during the realisation of the activity "Cooking" using no shared events.

The example shows that for each new observed event, the estimation of probability is actualised. The language is empty if the projected sequence length is lower than 2 and an offset is systematically observed when the activity starts. Furthermore, another offset is present when an activity stops.

This second offset is due to the use of a sliding window storing the last five observed events.

Finally, Fig. 11 shows the reduced PFA for activity  $\mathcal{A}_1$ . By applying the two complexity reduction strategies of the off-line computation, by a window of length  $|w|=5$ , we use  $2^5 \times 2^2 = 128$  operations instead of  $20^5 \times 2^2 \times 5 = 64.000.000$ .

### E. Case study discussion

The presented methods to model and recognise ADLs are efficient also to activities sharing events. In fact, Fig. 12 shows a case of shared events between the two activities “Cooking” and “Preparing Hot Beverage”. As expected, it is not possible to distinguish which activity is performed if a move that belongs to only one of the two activities does not occur. Hence, if two activities have many events in common, it is impossible to recognise the activity.

On the other hand, if a sensor is linked with too many activities, it is observed a lot of times during the learning period: it becomes predominant in all linked ADL models making it non-discriminant and noisy. Therefore, the events detected by such sensors are not useful.

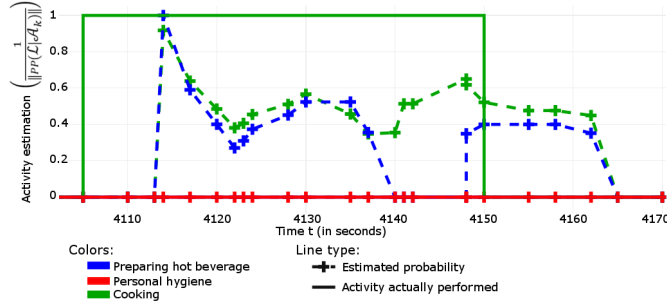


Fig. 12. Inverse of the normalised perplexity during the realisation of the activity “Cooking” based on shared moves.

In order to evaluate the efficiency of the proposed method, activity sequences are considered, and the results are shown in Fig. 13. Once again, we can observe indecision during the transition between two activities sharing the same events. Since the presented estimator (projection and normalised perplexity) allows finding the performed activity, it is possible to conclude that the presented method allows discovering and recognising activities performed by an inhabitant of a smart home without declaring the performed activity during the learning period.

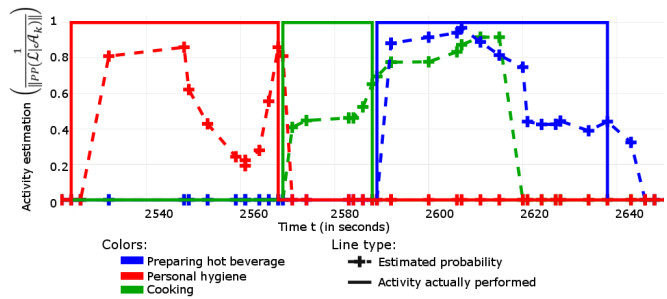


Fig. 13. Inverse of the normalised perplexity during the succession of three activities: “Personal Hygiene”, “Cooking” and “Preparing Hot Beverage”.

## VII. CONCLUSION

In this paper, a global approach for Activity of Daily Living Discovery and Recognition is proposed. To this aim, a procedure to automatically obtain the activities model in a Probabilistic Finite-State Automata framework is developed on the basis of the knowledge of training event logs database and a hierarchical decomposition of activities to monitor actions and moves. Then, an activity recognition method is presented on the basis of a newly defined distance estimator called

normalised likelihood and the extension to perplexity. In order to face the complexity of the activity recognition algorithm, we propose a complexity reduction method. Moreover, we prove that the normalised likelihood can be efficiently computed without loss of accuracy.

Finally, the proposed approach for Activity Discovery and Recognition is applied to a real living lab and the good quality of the obtained results is discussed.

Future works will be dedicated to the quantization of how valuable each sensor is in enabling the activity recognition and on the detection of the drifts in the activity realisation. Indeed, such drifts are often indicators of the evolution of numerous pathologies and the related detection will notably increase the performance of health at home technologies.

## APPENDIX

In order to prove Propositions 1 and 2, the following notation is defined:

- $e_i \in \Sigma_{\mathcal{A}_k}$ ,
- $e_{eq} \in \Sigma_{\mathcal{A}_k}$  such as  $e_{eq} \in \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Equ_{q_l, q_m}(e_i)$ ;
- $\{e_{eq}\}_{e_i} = \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Equ_{q_l, q_m}(e_i)$ ;
- $e_{sup} \in \Sigma_{\mathcal{A}_k}$  such as  $e_{sup} \in \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Geq_{q_l, q_m}(e_i)$  and  $e_{sup} \notin \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Equ_{q_l, q_m}(e_i)$ ;
- $\{e_{sup}\}_{e_i}$  is the set of all possible  $e_{sup}$  associated to  $e_i$ .

Thus, by definition it holds:

$$\{e_{eq}\}_{e_i} + \{e_{sup}\}_{e_i} = \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Geq_{q_l, q_m}(e_i).$$

### Proof of Proposition 1

In equation (16), only event  $e_i$  with  $\{e_{sup}\}_{e_i} = \emptyset$  are kept in the reduced PFA  $\mathcal{A}_k^r$ . We prove that the rejection of events having  $\{e_{sup}\}_{e_i} \neq \emptyset$  does not change the value of the maximum likelihood.

Let  $w = w'_1 \dots w'_k \dots e_i \dots w'_{|w|-1} w'_{|w|}$  be a sequence of events and  $e_i$  is one of the events in the sequence.

Let  $v = w'_1 \dots w'_k \dots e_{sup} \dots w'_{|w|-1} w'_{|w|}$  be a sequence of events that equals sequence  $w$  but for event  $e_i$ , which is changed by  $e_{sup} \in \{e_{sup}\}_{e_i}$ .

At this point, two cases exist  $\forall q_l, q_m \in Q_{\mathcal{A}_k}^2$ :

$$\begin{cases} P_{\mathcal{A}_k}[(q_l, e_{sup}, q_m)] = P_{\mathcal{A}_k}[(q_l, e_i, q_m)] \text{ if } e_{sup} \in Equ_{q_l, q_m}(e_i) \\ P_{\mathcal{A}_k}[(q_l, e_{sup}, q_m)] > P_{\mathcal{A}_k}[(q_l, e_i, q_m)] \text{ if } e_{sup} \notin Equ_{q_l, q_m}(e_i) \end{cases}$$

Thus, for each path  $\theta = (s_0, w'_1, s_1 \dots s_{j-1}, e_i, s_j \dots w'_{|w|}, s_{|w|})$  generating  $w$ , it exists a path  $\theta' = (s_0, w'_1, s_1 \dots s_{j-1}, e_{sup}, s_j \dots w'_{|w|}, s_{|w|})$  such as:

if  $e_{sup} \in Equ_{s_{j-1}, s_j}(e_i)$  then

$$\begin{aligned} I(s_0) \times \left( \prod_{h=1}^{|v|} P(s_{h-1}, v'_h, s_h) \right) &= I(s_0) \times \left( \prod_{h=1}^{|w|} P(s_{h-1}, w'_h, s_h) \right) \\ &\rightarrow P_{\mathcal{A}_k}(\theta') = P_{\mathcal{A}_k}(\theta) \end{aligned}$$

else if  $e_{sup} \notin Equ_{s_{j-1}, s_j}(e_i)$  then

$$I(s_0) \times \left( \prod_{h=1}^{|v|} P(s_{h-1}, v'_h, s_h) \right) > I(s_0) \times \left( \prod_{h=1}^{|w|} P(s_{h-1}, w'_h, s_h) \right) \\ \rightarrow P_{\mathcal{A}_k}(\theta') > P_{\mathcal{A}_k}(\theta).$$

Since  $e_{sup} \notin \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}^2} Equ_{q_l, q_m}(e_i)$ , the case  $e_{sup} \notin Equ_{s_{j-1}, s_j}(e_i)$  occurs at least one time, thus:

$$P_{\mathcal{A}_k}(v) = \sum_{\theta' \in \theta_{\mathcal{A}_k}(v)} P_{\mathcal{A}_k}(\theta') > P_{\mathcal{A}_k}(w) = \sum_{\theta \in \theta_{\mathcal{A}_k}(w)} P_{\mathcal{A}_k}(\theta). \quad (24)$$

Hence, (24) proves that, for all sequences  $w$  including an event  $e_i$  with  $\{e_{sup}\}_{e_i} \neq \emptyset$ , it exists another sequence  $v$  having the same length with a greater likelihood. Therefore, event  $e_i$  with  $\{e_{sup}\}_{e_i} \neq \emptyset$  can be excluded for the maximum likelihood computation.

In the same way, we can prove that the likelihood does not change by changing an event  $e_i$  by another event  $e_{eq} \in \{e_{eq}\}_{e_i}$ , then only one of them can be kept in  $\mathcal{A}_k^r$ . Consequently, it holds

$$\max_{u \in \Sigma_{\mathcal{A}_k}^{|w|}} [P_{\mathcal{A}_k}^r(u)] = \max_{v \in \Sigma_{\mathcal{A}_k}^{|w|}} [P_{\mathcal{A}_k}(v)] \text{ and the thesis is proved.}$$

■

#### Proof of Proposition 2

In order to prove Proposition 2, we recall that the following properties are direct consequences of equation (16).

*Property 1:* If event  $e_i$  is kept using equation (16), it exists a set of  $n_1$  origin and destination states  $C_{\mathcal{A}_k}^{n_1} = \{(q_{l_1}, q_{m_1}) \dots (q_{l_p}, q_{m_p}) \dots (q_{l_{n_1}}, q_{m_{n_1}})\}$  such that  $\forall e_j \in \Sigma_{\mathcal{A}_k}, \forall p \in [1, n_1]$  it holds:

$$P[(q_{l_p}, e_i, q_{m_p})] \geq P[(q_{l_p}, e_j, q_{m_p})].$$

Furthermore, according to (8),  $\tilde{N}(e_i | q_l \rightarrow q_m)$  and  $P_{\mathcal{A}_k}[(q_l, e_i, q_m)]$  do not depend on  $q_l$ . Thus,  $Geq_{q_l, q_m}(e_i)$  and  $Equ_{q_l, q_m}(e_i)$  depend only on  $e_i$  and  $q_m$ . It is possible to rewrite Property 1 as follow:

*Property 2:* If event  $e_i$  is kept using equation (16), it exists a set of  $n_2$  destination states  $D_{\mathcal{A}_k}^{n_2} = \{q_{m_1} \dots q_{m_p} \dots q_{m_{n_2}}\}$  such that  $\forall e_j \in \Sigma_{\mathcal{A}_k}, \forall q_l \in Q_{\mathcal{A}_k}, \forall p \in [1, n_2]$  it holds:

$$P[(q_l, e_i, q_{m_p})] \geq P[(q_l, e_j, q_{m_p})].$$

Moreover, for each possible set  $D_{\mathcal{A}_k}^{n_2}$ , of destination states, only one event is kept by the equivalent events deletion performed by step 2 of the *Reduction Procedure*. Thus, the number of kept events  $N_{\mathcal{A}_k} = Card(\Sigma_{\mathcal{A}_k}^r)$  is bounded by the number of possible sets  $D_{\mathcal{A}_k}^{n_2}$  that it is necessary to evaluate.

For a PFA with  $m = card(Q_{\mathcal{A}_k})$  states, sets composed with  $n_2 \in [1, m-1]$  destination states can be created. For each  $n_2$ , it exists  $\binom{m-1}{n_2}$  different possible sets  $D_{\mathcal{A}_k}^{n_2}$ . Thus, we have:

$$N_{\mathcal{A}_k} \leq \sum_{i=1}^{m-1} \binom{m-1}{i} = 2^{m-1} - 1.$$

Thus, according to equation (17) the complexity of the normalised likelihood is the following:

$$C_M = O(2^{[card(Q_{\mathcal{A}_k)}-1]|w|} \times card(Q_{\mathcal{A}_k})^2 \times |w|).$$

This proves Proposition 2. ■

#### REFERENCES

- [1] E. EUROSTAT, "Population structure and ageing," 2010.
- [2] D. Pal, Funilkul, N. Charoenkitkarn, and P. Kanthamanon "Internet-of-Things and Smart Homes for Elderly Healthcare: An End User Perspective," *IEEE Access*, vol. 6, pp. 10483-10496, 2018.
- [3] D. Bhatt and S. Kumar "Internet of Things: Smart Device for Smart City," *International Journal of Computer Applications* vol. 181, no. 7, pp. 1-4, 2018.
- [4] M. J. Deen, "Information and communications technologies for elderly ubiquitous healthcare in a smart home," *Pers. Ubiquitous Comput.*, vol. 19, no. 3-4, pp. 573-599, 2015.
- [5] J. van Hoof, G. Demiris, and E. Wouters, Eds., *Handbook of Smart Homes, Health Care and Well-Being*. Basel, Switzerland: Springer, Cham, 2017.
- [6] D. J. Cook, N. C. Krishnan, P. Rashidi, "Activity Discovery and Activity Recognition: A New Partnership", *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp.820-832,2013.
- [7] J. Saives, C. Pianon, and G. Faraut, "Activity discovery and detection of behavioral deviations of an inhabitant from binary sensors," *Automation Science and Engineering, IEEE Transactions on*, vol. 12, no. 4, pp. 1211-1224, 2015.
- [8] K. Viard, M. Fanti, G. Faraut, and J.-J. Lesage, "An Event-Based Approach for Discovering Activities of Daily Living by Hidden Markov Models," in *15th IEEE Int. Conf. on Ubiquitous Computing and Communications, (IUCC 2016)*, Granada, Spain, 2016
- [9] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 48-53, 2010.
- [10] K. Viard, M. P. Fanti, G. Faraut, and J.-J. Lesage, "Recognition of Human Activity Based on Probabilistic Finite-State Automata," in *22nd IEEE International Conference on Emerging Technologies and Factory Automation*, Limassol, Cyprus, 2017.
- [11] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.
- [12] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008.
- [13] V. Kellokumpu, M. Pietikäinen, and J. Heikkilä, "Human activity recognition using sequences of postures," *MVA2005 IAPR Conference on Machine Vision Applications*, pp. 570-573, 2005
- [14] S. Hongeng, F. Brémond and R. Nevatia, "Representation and optimal recognition of human activities", *Proceedings IEEE*

*Conference on Computer Vision and Pattern Recognition, CVPR 2000 (Cat. No. PR00662)*, Hilton Head Island, SC, pp. 818-825 vol.1, 2000

- [15] S. Hongeng, F. Brémond and R. Nevatia, "Video-Based event recognition: activity representation and probabilistic recognition methods", *Computer Vision and Image Understanding*, vol.96, no.2, pp. 129-162, 2015.
- [16] T. Duong, D. Phung, H. Bui, and S. Venkatesh, "Efficient duration and hierarchical modeling for human activity recognition," *Artificial intelligence*, vol. 173, no. 7-8, pp. 830–856, 2009.
- [17] E. Vidal, F. Thollard, C. De La Higuera, F. Casacuberta, and R. C. Carrasco, "Probabilistic finite-state machines - Part II." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1026–1039, 2005.
- [18] A. A. Chaaoui, P. Climent-Pérez, and F. Flórez-Revuelta, "A review on vision techniques applied to human behaviour analysis for ambient-assisted living," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10873–10888, 2012.
- [19] S. Gaglio, G. L. Re, and M. Morana, "Human activity recognition process using 3-d posture data," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 5, pp. 586–597, 2015.
- [20] C. H. Lu, Y. C. Ho, Y. H. Chen, and L. C. Fu, "Hybrid user-assisted incremental model adaptation for activity recognition in a dynamic smart-home environment," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 5, pp. 421–436, 2013.
- [21] S. Himmel, M. Ziefle, and K. Arning, *From Living Space to Urban Quarter: Acceptance of ICT Monitoring Solutions in an Ageing Society*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 49–58, 2013.
- [22] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Data Engineering, 1995. IEEE Proceedings of the Eleventh International Conference on*, pp. 3–14, 1995.
- [23] E. M. Tapia, S. S. Intille, and K. Larson, *Activity recognition in the home using simple and ubiquitous sensors*. Vol. 3001, Springer, chapter 10, 2004.
- [24] D. J. Cook and N. C. Krishnan, *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*. John Wiley & Sons, 2015.
- [25] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 2, pp. 274–283, 2010.
- [26] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and Mobile Computing*, vol. 10, Part B, pp. 138 – 154, 2014.
- [27] A. Galata, N. Johnson, D. Hogg, "Learning Variable-Length Markov Models of Behavior", *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 398-413, 2001.
- [28] D. Jurafsky and J. H. Martin, *Speech and language processing*. Pearson London, vol. 3, 2014.
- [29] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer US, Boston, MA, 2008.
- [30] K. Mukherjee, A Ray, "State Splitting and Merging in Probabilistic Finite State Automata for Signal Representation and Analysis" *Signal processing*. Vol. 104, Nov 2014, pp. 105-119.
- [31] P. Adenis, K Mukherjee, A Ray, "State Splitting and state in Probabilistic Finite State Automata" *Proc. of American Control Conference*, June 29 - July 01, 2011, pp. 5145-5150



**Kévin VIARD** received the M.S. degree from the Ecole Normale supérieure Paris-Saclay, Cachan, France in 2015 and a joint Ph.D. degree from the Ecole Normale supérieure Paris-Saclay, Cachan, France and from Politecnico di Bari, Bari, Italy in 2018. Since September 2018, he is teaching engineer sciences to French higher school preparatory classes. His research interests included smart environments, ambient assisted living, activity discovery and activity recognition and other topics related to automated systems.



**Maria Pia Fanti (M'92-SM'02-F'17)** received the Laurea degree in electronic engineering from the University of Pisa, Pisa, Italy, in 1983. She was a visiting researcher at the Rensselaer Polytechnic Institute of Troy, New York, in 1999. Since 1983, she has been with the Department of Electrical and Information Engineering of the Polytechnic of Bari, Italy, where she is currently a Full Professor of system and control engineering and Chair of the Laboratory of Automation and Control. Her research interests include management and modeling of complex systems, such as transportation and logistics systems; discrete-event systems; Petri nets; consensus protocols; fault detection.

Prof. Fanti has published more than +290 papers and two textbooks on these topics. She was editor of the IEEE Trans. on ASE and is Associate Editor of the IEEE Trans. on SMC: Systems. She was member at large of the Board of Governors of the IEEE Systems, Man, and Cybernetics Society, is member of the AdCom of the IEEE Robotics and Automaton Society, and chair of the Technical Committee on Automation in Logistics of the IEEE Robotics and Automation Society. She was General Chair of the 2011 IEEE Conference on Automation Science and Engineering, the 2017 IEEE International Conference on Service Operations and Logistics, and Informatics and the 2019 Systems, Man, and Cybernetics Conference.



**Gregory Faraut (M'07)** received the B.S. degree in electrical engineering and the M.S. degree in computer science from the University of Nice Sophia Antipolis, Nice, France, in 2004 and 2006, respectively, and the Ph.D. degree in automatic control from the Ampere Lab, INSA Lyon, Villeurbanne, France, in 2010. Since 2011, he has been an Associate Professor of Automatic Control with LURPA, ENS Paris-Saclay, Gif-Sur-Yvette, France. His research interests concern the field of

Discrete Event Systems with applications to manufacturing systems and production systems. More specifically, he is interested in identification techniques and Ambient Assisted Living (AAL) approaches based on Machine Learning. Dr. Faraut is Founding Co-Chair of the Technical Committee on Automation in Health Care Management of the IEEE Robotics and Automation Society.

The most recent work concerns, for AAL, the discovery, the detection and the evaluation of behavior's deviations of frailty people, and, in the area of manufacturing and production systems, the improvement of techniques for the use of Digital Twin.



**Jean-Jacques Lesage (M'07)** received the Ph.D. degree from the Ecole Centrale de Paris, France, and the Habilitation à Diriger des Recherches from the University of Nancy, France, in 1989 and 1994 respectively. Currently, he is Professor of Automatic Control with the Ecole Normale Supérieure Paris-Saclay, France.

His research interests include formal methods and models for identification, analysis and diagnosis of Discrete Event Systems, as well as applications to manufacturing systems, network automated systems, energy production, and more recently ambient assisted living.