



A new approach based on temporal sub-windows for online sensor-based activity recognition

Macarena Espinilla¹ · Javier Medina¹ · Josef Hallberg² · Chris Nugent³

Received: 30 October 2017 / Accepted: 6 March 2018 / Published online: 10 March 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

Usually, approaches driven by data proposed in literature for sensor-based activity recognition use the begin label and the end label of each activity in the dataset, fixing a temporal window with sensor data events to identify the activity carried out in this window. This type of approach cannot be carried out in real time because it is not possible to predict the start time of an activity, i.e., the class of the future activity that an inhabitant will perform, neither when he/she will begin to carry out this activity. However, an activity can be marked as finished in real time only with the previous observations. Therefore, there is a need of online activity recognition approaches that classify activities using only the end label of the activity. In this paper, we propose and evaluate a new approach for online activity recognition with three temporal sub-windows that uses only the end label of the activity. The advantage of our approach is that the temporal sub-windows keep a partial order in the sensor data stream from the end time of the activity in a short-term, medium-term, long-term. The experiments conducted to evaluate our approach suggest the importance of the use of temporal sub-windows versus a single temporal window in terms of accuracy, using only the end time of the activity. The use of temporal sub-windows has improved the accuracy in the 98.95% of experiments carried out.

Keywords Activity recognition · Data sensor stream · Fuzzy linguistic modelling · Sensor data stream processing · Smart environments

1 Introduction

The sensor-based activity recognition is particularly suitable to deal with Activities of Daily Living (ADLs) that involve a number of objects within an environment (Kruger et al. 2014; Alemdar and Erosy 2017; Gutiérrez et al. 2017;

Calvaresi et al. 2017). This kind of activity recognition is a useful tool for unobtrusive home environment monitoring (López et al. 2017). Approaches used for sensor-based activity recognition are divided into two main categories: data-driven approaches (DDAs) and knowledge-driven approaches (KDAs) (Chen et al. 2012).

DDAs is based on machine learning techniques in which a pre-existent dataset of user behaviours is required. A training process is carried out to build an activity model, which is followed by a testing process to evaluate the generalization of the model in classifying unseen activities (Gu et al. 2011; Li et al. 2014; San Mateo et al. 2010). With KDAs, an activity model is built through the incorporation of rich prior domain knowledge gleaned from the application domain. Knowledge engineering and knowledge management techniques are used to carry out the classification process (Espinilla et al. 2017; Chen and Nugent 2009; Zhang et al. 2017). KDAs have the advantages of being semantically clear and logically elegant. Nonetheless, this kind of approach is weak to deal with uncertainty and temporal information.

✉ Macarena Espinilla
mestevez@ujaen.es

Javier Medina
jmquero@ujaen.es

Josef Hallberg
Josef.Hallberg@ltu.se

Chris Nugent
cd.nugent@ulster.ac.uk

¹ Department of Computer Science, University of Jaén, Jaén, Spain

² Department of Computer Science, Lulea tekniska Universitet, Luleå, Sweden

³ School of Computing and Mathematics, Ulster University, Coleraine, UK

Due to the advantages of the DDAs in order to manage the uncertainty and temporal information, this paper is focused on this kind of approaches in which a dataset is required. There are repositories that contain several ADL datasets in smart environments. One the most well-known repositories is CASAS¹ (Schmitter-Edgecombe et al. 2009) and, in this context, it is noteworthy the Open Data Initiative (ODI) (Nugent et al. 2016) for activity recognition consortium that aims to create a structured approach to provide annotated datasets in an accessible format.

In the literature, there is a wide range of DDAs that works extremely well adopting a pre-segmentation process for activity recognition (Ordoñez et al. 2013a, b; Banos et al. 2014). In the pre-segmentation process, the sensor data stream is divided into segments by using the begin label and the end label of each activity. The begin label indicates the start time of the activity and the end label indicates the end time of the activity. Therefore, each segment of the sensor data stream corresponds to an activity and this segmentation is trained and evaluation by DDAs. Therefore, DDAs offer excellent results in offline activity recognition where the begin label and the end label of each activity in the dataset is known.

However, the successful results that these DDAs provide cannot be transferred to real-time activity recognition since the begin label cannot be predicted. This is a relevant problem when an inhabitant is generating a dataset in a smart environment and an external person is monitoring the inhabitant. The external person does not know previously the future activity that will be started by the inhabitant. Nevertheless, it does know when the activity is ended according to the previous observations. Due to this fact, there is a need of online activity recognition approaches that classify activities using exclusively the end label of the set of activities.

This paper is focused on the process of online activity recognition in a smart environment in which a single person performs ADLs by using the end label of each activity. The proposed approach is composed of two learning layers that consider three temporal sub-windows. The first layer is considered for learning each activity model using the sensor data stream generated by binary sensors in three temporal sub-windows. In the second layer, a general classifier among activities is trained, considering also three temporal sub-windows, to obtain the classified activity among the set of activities. Therefore, there is a first layer that is trained and then, the output of the first layer is provided to the second layer in order to train the second layer. This approach is selected based on the philosophy of deep learning (Bengio 2009) that achieves promising performance in the activity

recognition process and many fields (Wang et al. 2017), among them image processing (Chan et al. 2015) in order to interpret the image by using a greyscale or a RGB values. The main advantage of deep learning is the search of a deep high-level knowledge from the learning of multiples levels or layers in where each one is responsible for extracting one or more features.

Therefore, the main contribution of this paper is an approach that uses only the end label of the activity for the online activity recognition process. To do so, temporal sub-windows are proposed, which keep a partial order in the sensor data stream from the end time of the activity in a short-term, medium-term, long-term. The size of these temporal sub-windows are computed by statistical measures. For the sake of simplicity, the number of three temporal sub-windows has been selected as a compromise between accuracy and computations.

In order to show the effectiveness of the proposed approach, an evaluation is carried out using two popular ADL datasets in smart environments. The online activity recognition process is carried out using exclusively the end label of each activity with three popular classifiers. The comparative study is done using a single time window versus our proposal of three temporal sub-windows.

The rest of the paper is organized as follows. Section 2 highlights the key points of related works as well as the classical method to process a sensor data stream. Section 3 presents the proposed approach that consider three temporal sub-windows for sensor-based activity recognition by using only the end label. Section 4 presents the description of the used datasets and the set of experiments to evaluate the proposed approach. Section 5 presents and discusses the computed results. Finally, Sect. 6 summarizes the work presented in the paper and addresses the future works.

2 Background

In this section, we reviews works related to our proposal as well as the key points of them. Then, we review the traditional method to process a sensor data stream.

2.1 Related works

In Ordoñez et al. (2013a, b) several classifiers and probabilistic method were evaluated in online recognition, providing notable results between 50 and 70% of f-measure. In these papers, several techniques related to the representation of sensor data streams, such as change point and last activated sensor, are used as features of several classifiers (hidden Markov model, multi layer perceptron, support vector machine, decision trees, k-nearest neighbor or rules).

¹ <https://ailab.wsu.edu/casas/datasets> (last checked on February 28, 2018)

In addition, the use of a sliding window mechanism (Dietrich 2002) was introduced in this work. Similar results were obtained in Cook (2010) where naive Bayes, hidden Markov models (HMM) and conditional random fields were analyzed. In other works, Krishnan and Cook (2014) studied support vector machine with a full description of representation of sensors and contextual information in activity recognition.

The key points of the aforementioned related works associated with our approach are: (i) temporal windows, which include the stream of state-change sensors generated by binary sensors, (ii) dynamic window sizes computed with statistical measures, (iii) multiple levels of learning.

These works have served as a motivation to the proposed approach to manage the process of online activity recognition by using only the end label of each activity. The relevant points are the following:

- (i) The use of temporal sub-windows, which increase the past temporal information of sensors in the feature vector. In our case, from the end label of each activity without considering the begin label of each activity in the dataset.
- (ii) A first layer of learning, which receives as input the set of sensor activations in the three temporal sub-windows, developing a specific classifier by each activity.
- (iii) A second layer of learning, which receives the output of the first layer, to classify the unseen activity using a general classifier. To do so, the current and past activation of the specific classifiers in the first layer are collected as an activity stream, which is represented as a feature vector using the same process of the three temporal sub-windows related to sensor streams in first layer.

2.2 Computing a classical feature matrix

The traditional process to compute a feature matrix from the sensor data stream is reviewed in this section.

2.2.1 Data sensor stream

Smart environments are residences in which a set of sensors is deployed and networked to a range of objects or locations. These are used to identify people in the environment and their actions (Chen et al. 2012). Previous approaches have demonstrated how binary sensors are a useful tool for solving the ADL recognition problem in smart homes (Tapia et al. 2004). In this paper, a set of annotated datasets in smart environments is used. Each dataset is composed by a binary data sensor stream from sensors that monitored the ADLs

Table 1 Example of a partial sensor data stream while activity A_1 is carried out

Date	Time	Sensor	Value	Activity
2015-01-20	18:22:32	D01	CLOSE	A_1 _Begin
2015-01-20	18:25:33	D01	OPEN	
2015-01-20	18:26:52	D01	CLOSE	
2015-01-20	18:27:47	D05	OPEN	
2015-01-20	18:28:21	D05	CLOSE	
2015-01-20	18:29:38	WT1	OPEN	
2015-01-20	18:32:33	WT1	CLOSE	
2015-01-20	18:32:12	D01	OPEN	
2015-01-20	18:32:57	D01	CLOSE	
				A_1 _End

performed in a smart environment setting by a single inhabitant. In Table 1, a partial sensor data stream of an activity is shown in a smart environment with the following binary sensors (Quesada et al. 2015):

- D01: Kitchen door sensor (open/close).
- D02: Living room door sensor (open/close).
- D03: Cutlery cupboard sensor (open/close).
- D04: Dishes cupboard sensor (open/close).
- D05: Glasses and cups cupboard sensor (open/close).
- D06: Pantry cupboard sensor (open/close).
- D07: Microwave sensor (open/close).
- D08: Fridge door sensor (open/close).
- M01: Chair sensor (absence/presence).
- M02: Sofa sensor (absence/presence).
- TV: Television sensor (on/off).
- PH: Phone sensor (pick up/hang up).
- WT1: Water sensor (open/close).
- KT: Kettle sensor (absent/present).

2.2.2 Segmentation process

This subsection briefly reviews the three common approaches in the literature for processing a sensor data stream (Chen et al. 2015), other kind of segmentation processes can be found in Triboan et al. (2017).

- (1) *Based on activity* This type of segmentation is also called explicit and it is the most popular segmentation adopted for sensor-based activity recognition, providing excellent results in a wide range of DDA approaches (Junker et al. 2008). Typically, a pre-segmented sensor data corresponds to an activity that will be used to train the activity model (Yala et al. 2017). Therefore, in the activity segmentation, the sensor data stream is divided into sensor data segments that are denoted by $\text{Seg} = \{C_1, \dots, C_A\}$, each segment possibly

Table 2 Example of a feature vector generated by the sensor data stream illustrated in Fig. 1

D01	D02	D03	D04	D05	D06	D07	D08	KT	M1	M2	PH	TV	WT	Act
1	0	0	0	1	0	0	0	0	0	0	0	0	1	A_1

corresponding to an activity and, in a second step, the classification of each segment is carried out. The main disadvantage of this kind of approach is that it is not feasible on real time activity recognition due to the need to pre-segment previously.

- (2) *Based on time* Many researchers also have used this second approach. In this approach, the sensor data stream is divided into segments with equal size time denoted by $\text{Seg} = \{T_1, \dots, T_T\}$. The main problem of this approach is to identify the optimal time length of each segment. This kind of approach provides good results in smart environments (Wang et al. 2012), being very appropriate.
- (3) *Based on events* The third approach to process the sensor data stream is to divide the sensor data stream into segments with equal number of sensor events denoted by $\text{Seg} = \{E_1, \dots, E_E\}$. The main problem with this approach is to separate sensor events, which may correspond to different activities that can be included in the same segment. This approach is adopted by some researches with continuous sensors like acceleration (Banos et al. 2014).

In our proposal, the segmentation based on time is adopted due to its benefits to manage binary sensors deployed in smart environments. In order to deal with the problem of selecting the optimal time length of each segment, we propose the use of statistical measures, mean and coefficient of variation of the length activities, which have provided good results in the process for activity recognition of a single activity (Carnevali et al. 2015).

2.2.3 Feature matrix

The sensor data stream is discretized into segments with a time window size that is defined by Δ_t , considering the segmentation process.

For each end label of an activity t_{end} in the dataset, a feature vector F^k is computed, considering the segment between t_{end} and $t_{\text{end}} - \Delta_t$. The set of activity classes are denoted by $A = \{A_1, \dots, A_i, \dots, A_{A_N}\}$, where A_N is the number of activity classes on the dataset. Each feature vector is denoted by F^k and has $S_N + 1$ components, where S_N is the number of sensors in the dataset that are identified by $S = \{s_1, \dots, s_j, \dots, s_{S_N}\}$.

Each computed feature vector in the dataset is defined by $F^k = (f_1^k, \dots, f_j^k, \dots, f_{S_N}^k, f_{S_N+1}^k)$, where f_j^k is a binary value

that indicates if the sensor s_j was fired at least once, 1, or was not fired 0 in each segment. The last component $f_{S_N+1}^k \in A$, indicates the activity carried out in each segment. An example of a feature vector computed from the partial sensor data stream illustrated in Table 1 is shown in Table 2 (Quesada et al. 2015).

In the first learning layer of the proposed approach, probabilistic measures, mean and standard deviation, will be consider to compute the temporal window size for each activity. Considering these sizes, each feature vector will be computed, describing the activations in the three temporal sub-windows.

3 Method

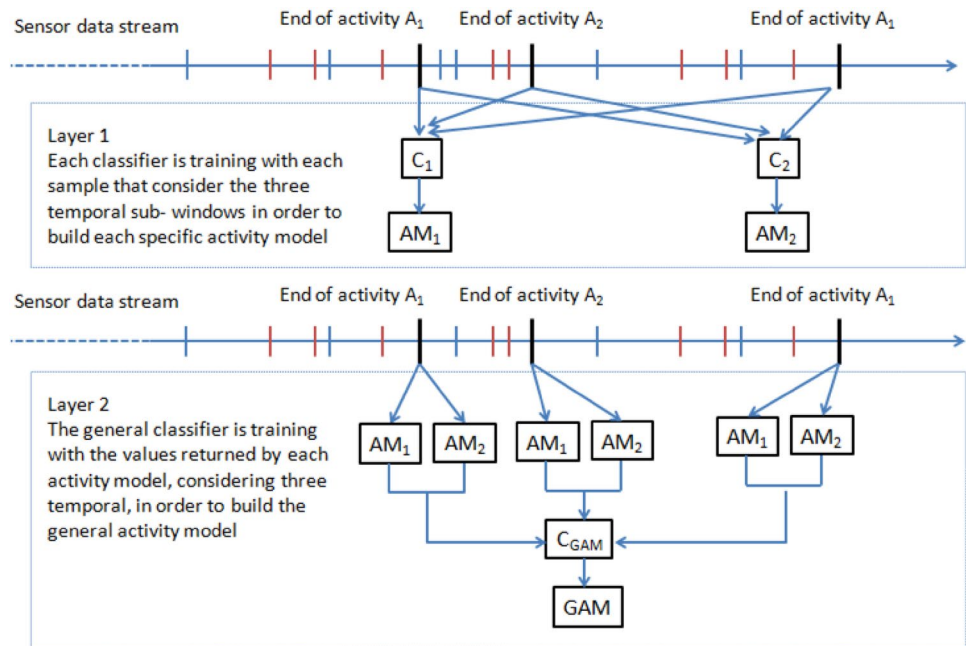
In our approach, the main goal is to identify the activity that is carried out by using only the end label in the sensor data stream. To do so, the proposed approach considers the definition of three temporal sub-windows in two learning layers. This section, first, presents the scheme of the proposed approach and then presents each layer in detail.

3.1 Scheme of the proposed approach with temporal sub-windows

In the proposed approach, the start time (begin label) of an activity is not used, only is used the end time (end label) in the activity recognition process.

From the end time of an activity, the changes in a sensor are represented in three times by using three temporal sub-windows. So, the proposed approach allows a more detailed representation of the sensor data stream versus a single temporal window, offering a partial order among sensor data events. The aim of the first layer is to train a classifier per each activity class, using the data sensor stream generated by binary sensors in three temporal sub-windows. In this first layer, a specific classifier for each kind of activity is built.

Once the specific activity models have been trained in the first layer, a general classifier among kind of activities is trained in the second layer, considering also three temporal sub-windows, in order to build a general activity model to classify a kind of activity among the set of potential activity classes. In this second layer, the input is the set of predicted values from each specific classifier in the three sub-windows (short-term, medium-term and long-term).

Fig. 1 Scheme of the proposed approach

In Fig. 1 is illustrated the proposed approach based on two learning layers with three sub-windows. In the first layer, C_1 and C_2 are classifiers, which are trained with the sensor data stream in order to build specific activity models, in this case AM_1 and AM_2 . In the second layer, the specific activity models, AM_1 and AM_2 , are used to compute the prediction of each respective activity class. The set of computed values is used to train the C_{GAM} to build the general activity model GAM .

In the following sections, each learning layer is presented, describing in detail the role of the three temporal sub-windows and how feature vectors are computed to train each activity model.

3.2 Layer 1: learning specific activity models

In the first layer, a specific activity model for each activity is built. These specific activity models consider the activity lengths computed by the statistical measures to compute the feature vectors that will be used to train each classifier per each specific activity model.

3.2.1 Segmentation based on time

The set of specific activity models are denoted by $AM = \{AM_1, \dots, AM_i, \dots, AM_{A_N}\}$ where A_N is the number of kind of activities and $A = \{A_1, \dots, A_i, \dots, A_{A_N}\}$ the set of kind of activities. Each activity length size is computed by the mean x_{A_i} and the standard deviation CV_{A_i} of each activity A_i . Furthermore, a factor w_1 is used to weigh the importance of the standard deviation in the window size. So, the time

window size of each activity is denoted by Δ_{A_i} and defined by $\Delta_{A_i} = x_{A_i} + w_1 \times CV_{A_i}$.

The activity model AM_i for each activity A_i is built to recognize when this kind of activity is carried out and when not, using the temporal window size Δ_{A_i} , considering the partial sensor data stream when an activity is ended by means three temporal sub-windows. The trained classifier to build the activity model is noted by C_i and it can be the same or different for each kind of activity (Quinlan et al. 1993). The input of each classifier C_i is shaped by the set of feature vectors that considers each segment between t_{end} and $t_{end} - \Delta_{A_i}$.

It is noteworthy that this window size is fixed in the training of each classifier without to take into account the real beginning of the activity (begin label).

3.2.2 Feature matrix computed with three temporal sub-windows

Our proposed approach is focused on three temporal sub-windows: short-term, medium-term and long-term, which keep a partial order among data sensor events from the end time of each activity.

In Figs. 2 and 3, the scheme of the segmentation to compute two feature vectors from the sensor data stream is illustrated. There are two sensors, which are represented by the color red and by the color blue as well as three sub-windows: Long term (W_{LT}), Medium-term (W_{MT}) and Short-term (W_{ST}). The first and second components correspond to activations of the red and blue sensors respectively in the long-term sub-window. The third and fourth components correspond to the red and blue sensors in the medium-term

Fig. 2 Scheme of the segmentation to compute two feature vectors with the temporal window size of the activity A1 and the temporal window size of the activity A2 in the sensor data stream when the end of A1 is identified

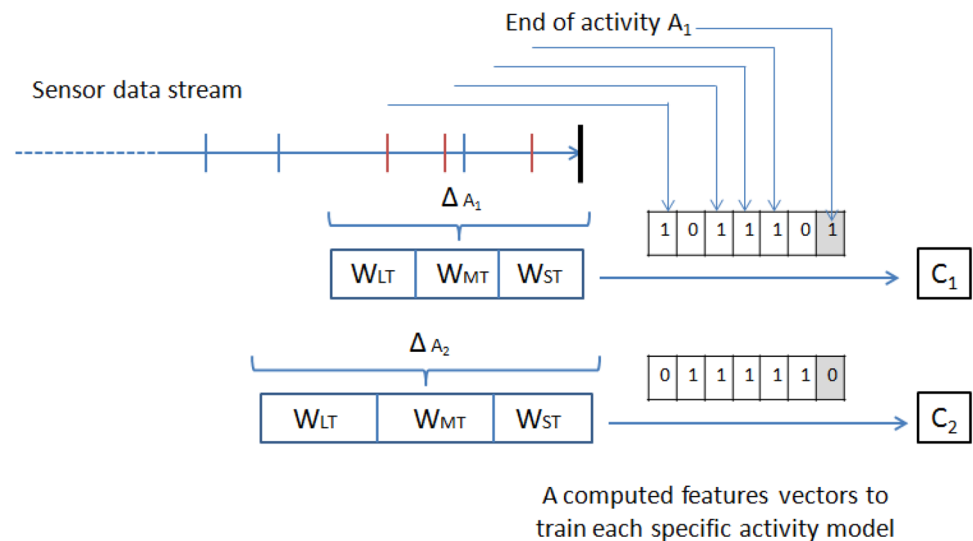


Fig. 3 Scheme of the process to build each activity model, AM1 and AM2, in order to compute the set of feature vectors to train each classifier, C1 and C2

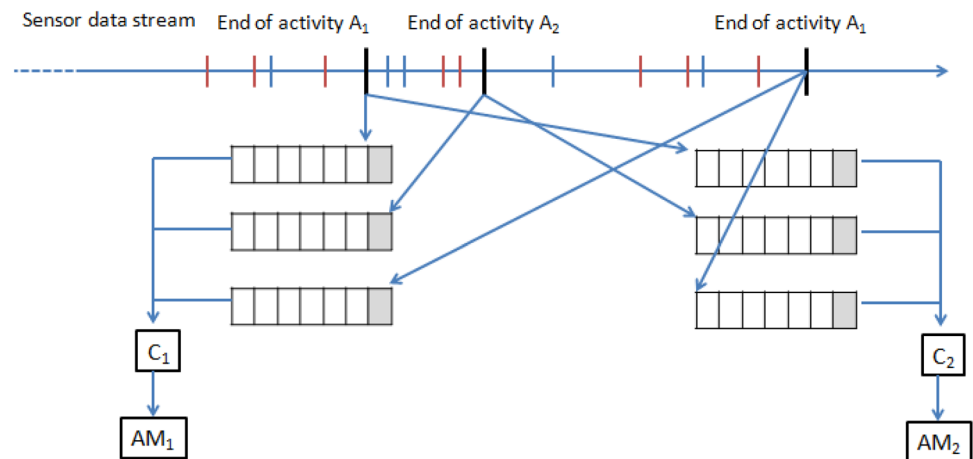


Table 3 Examples of set of features vectors for the activity A_i

	Sensors sub-window long-term (W_{LT})					Sensors sub-window medium-term (W_{MT})					Sensors sub-window short-term (W_{ST})					Act
	S_1	...	S_j	...	S_{S_N}	S_1	...	S_j	...	S_{S_N}	S_1	...	S_j	...	S_{S_N}	A_i
$F^{t1_{end}}$	1	...	0	...	0	0	...	1	...	0	0	...	0	...	1	1
$F^{t2_{end}}$	0	...	0	...	0	0	...	1	...	0	0	...	0	...	1	1
...																
$F^{t'_{end}}$	0	...	1	...	1	1	...	0	...	1	1	...	1	...	0	0
...																

sub-window, respectively. The fifth and sixth components correspond to the red and blue sensor in the sub-window in the short-term, respectively. If the sensor is activated in each temporal sub-window the value is 1, otherwise 0. The last component of the feature vector indicates the activity. If the activity corresponds with the activity learned by each classifier (C_1 and C_2), the value of this component is 1, and 0 otherwise.

The training process computes a set of computed feature vectors (feature matrix); the number of rows of this matrix will be the number of activities in the training dataset.

Each feature vector is defined by $F^{t'_{end}} = (f^{t'_{end}}_1, \dots, f^{t'_{end}}_{3 \times S_N}, f^{t'_{end}}_{3 \times S_N + 1})$. In the Table 3 an example of feature matrix by the activity A_i is illustrated. This set of feature vectors is provided to the classifier C_i in order to

build the activity model AM_i . Each feature vector is computed when the end label of an activity is detected in order to obtain the training set. If the end label of the activity is A_i , the last component is 1, otherwise the last component is 0.

The long term sub-window is represented by the S_{Nth} components in the feature vector $(f_{1'}^{t'_{end}}, \dots, f_{j'}^{t'_{end}}, \dots, f_{S_N}^{t'_{end}})$ that describes the changes in the set of sensors $S = \{S_1, \dots, S_j, \dots, S_{S_N}\}$. This long-term sub-window is computed in the time interval defined by Eq. (1).

$$W_{LT} = \left[t'_{end} - \Delta_{A_i}, t'_{end} - \frac{t'_{end} - \Delta_{A_i}}{3} \right) \quad (1)$$

The medium-term sub-window is represented by the following S_{Nth} components in the feature vector $(f_{S_N+1}^{t'_{end}}, \dots, f_{S_N+j}^{t'_{end}}, \dots, f_{2 \times S_N}^{t'_{end}})$ that represents the changes in the set of sensors $S = \{S_1, \dots, S_j, \dots, S_{S_N}\}$. This medium-term sub-window is computed in the interval defined by Eq. (2).

$$W_{MT} = \left[t'_{end} - 2 \times \frac{t'_{end} - \Delta_{A_i}}{3}, t'_{end} - \frac{t'_{end} - \Delta_{A_i}}{3} \right) \quad (2)$$

The short-term sub-window is represented by the following S_{Nth} components in the feature vector $(f_{(2 \times S_N + 1)}^{t'_{end}}, \dots, f_{(2 \times S_N + j)}^{t'_{end}}, \dots, f_{(3 \times S_N)}^{t'_{end}})$ that indicate the changes in the set of sensors $S = \{S_1, \dots, S_j, \dots, S_{S_N}\}$. This short-term window is computed in the time interval defined by Eq. (3).

$$W_{ST} = \left[t'_{end} - \frac{t'_{end} - \Delta_{A_i}}{3}, t'_{end} \right] \quad (3)$$

Finally, the last component of each feature vector $f_{3 \times S_N + 1}^{t'_{end}} \in \{0, 1\}$ indicates if the end label of the activity completed is A_i , 1, or it is other activity of the dataset, 0.

3.2.3 Building the specific activity models

In the last step of the first learning layer, the set of specific activity models $AM = \{AM_1, \dots, AM_i, \dots, AM_{A_N}\}$ are built by means of a training process. For each end label t'_{end} in the sensor data stream, A_N feature vectors are built. So, a feature vector $F_{A_i}^{t'_{end}}$ for each activity A_i is built, taking into account its proper temporal window size Δ_{A_i} .

Finally, the set of feature vectors of each activity A_i in dataset is used to build the activity model AM_i by means of the training process with the classifier C_i . As mentioned, the classifier can be the same or different for each activity, the

only requirement is that it can receive feature vectors with binary values.

Once the learning of the first layer has been completed, the learning of the second layer is performed, which is presented in the following subsection.

3.3 Layer 2: learning general activity model

In the previous layer, a set of specific activity models is computed. The aim of this second layer is to train a general activity model to obtain the classified activity among the set of potential activities. To do so, the values computed in each specific activity models are used, which correspond with the prediction of each respective activity class in three sub-windows. So, the second layer keeps the partial order among activities in three temporal sub-windows.

3.3.1 Segmentation based on time

In this step, an extension of the temporal window size Δ_{A_i} for each activity is proposed with the aim of expanding the partial data stream considering the previous sensor events of activities that will be used in the general activity. Hence, the temporal window size in the second layer of each activity $\Delta_{A_i'}$ considers a second factor w_2 , to weight the temporal window size in the first layer. So, the temporal window size in the second layer of each activity $\Delta_{A_i'}$ is defined by $\Delta_{A_i'} = \Delta_{A_i} \times w_2$.

3.3.2 Feature vectors of activities that are computed with temporal three sub-windows

A feature vector of this second layer describes the computed values of each activity model in three time points for each end label in the dataset, considering three temporal sub-windows: short-term, medium-term and long-term. So, each computed value with each specific activity model corresponds with the prediction of its activity class.

In Fig. 4, the scheme of the segmentation to compute the components related to the activity model AM_i is illustrated. Each returned value corresponds to the predicted value of the specific activity model AM_i .

In the Table 4, examples of computed feature vectors for the set of specific activity models are illustrated. For example, it is assumed that the value returned by each specific activity model in the first layer is within the unit interval. In this case, the first value, 0.8, represents the computed value in the AM_i for the A_i activity in the long-term sub-window, considering the activity length $\Delta_{A_i'}$.

In this second layer, a feature vector with three sub-windows is defined by $F2_{t'_{end}} = (f2_1^{t'_{end}}, \dots, f2_{3 \times A_N}^{t'_{end}}, f2_{3 \times A_N + 1}^{t'_{end}})$.

Fig. 4 Scheme of the process to compute the value returned by the specific activity model AM1

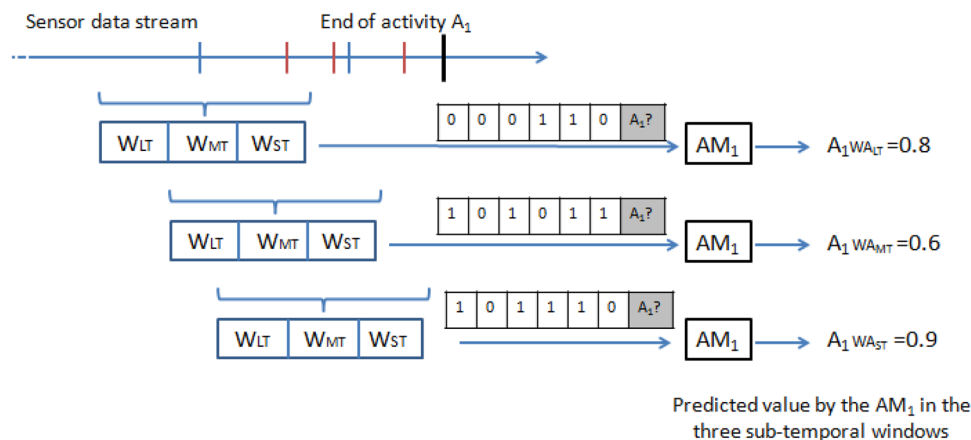


Table 4 Example of a matrix feature for the general activity model

Activities	Activity—sub-window long-term (WA _{LT})					Activity—sub-window medium-term (WA _{MT})					Activity—sub-window short-term (WA _{ST})					Real activity
	A ₁	...	A _i	...	A _{A_N}	A ₁	...	A _i	...	A _{A_N}	A ₁	...	A _i	...	A _{A_N}	
F2 ^{t₁end}	0.8	...	0.2	...	0.1	0.6	...	0.3	...	0.2	0.9	...	0.4	...	0.1	A ₂
F2 ^{t₂end}	0.7	...	0.3	...	0.9	0.5	...	0.4	...	0.1	0.8	...	0.5	...	0.3	A ₁
...																
F2 ^{t_{end}}	0.2	...	0.6	...	0.8	0.4	...	0.8	...	0.8	0.1	...	0.6	...	0.9	A _{A_N}
...																

The long term sub-window is represented by the first A_{Nth} components in the feature vector $(f2_1^{t'_{end}}, \dots, f2_{A_N}^{t'_{end}})$

that describe the computed values of the set of activities $A = \{A_1, \dots, A_i, \dots, A_{A_N}\}$ with their specific activity models $AM = \{AM_1, \dots, AM_i, \dots, AM_{A_N}\}$, respectively, at the moment defined by Eq. (4).

$$WA_{LT} = t'_{end} - 2 \times \frac{\Delta_{A'_i}}{3} \quad (4)$$

The medium term sub-window is represented by the following A_{Nth} components in the feature vector $(f2_{A_N+1}^{t'_{end}}, \dots, f2_{A_N+i}^{t'_{end}}, \dots, f2_{2 \times A_N}^{t'_{end}})$ that describe the computed values of the set of activities $A = \{A_1, \dots, A_i, \dots, A_{A_N}\}$ with their specific activity models $AM = \{AM_1, \dots, AM_i, \dots, AM_{A_N}\}$, respectively, at the moment defined by Eq. (5).

$$WA_{MT} = t'_{end} - \frac{\Delta_{A'_i}}{3} \quad (5)$$

The short term sub-window is represented by the following A_{Nth} components of the feature vector $(f2_{2 \times A_N+1}^{t'_{end}}, \dots, f2_{2 \times A_N+i}^{t'_{end}}, \dots, f2_{3 \times A_N}^{t'_{end}})$ that describe the computed values of the set of activities $A = \{A_i, \dots, A_j, \dots, A_{A_N}\}$ with their specific activity models

$AM = \{AM_1, \dots, AM_i, \dots, AM_{A_N}\}$, respectively, at the moment defined by Eq. (6).

$$WA_{LT} = t'_{end} \quad (6)$$

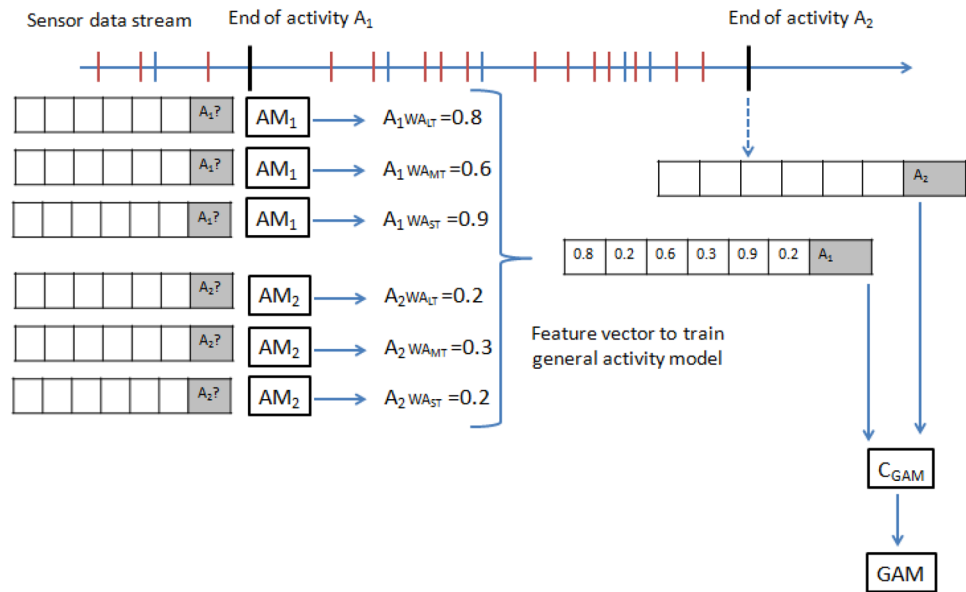
Finally, the last component of each feature vector $f2_{3 \times A_N}^{t'_{end}} \in A = \{A_1, \dots, A_i, \dots, A_{A_N}\}$ indicates the real activity completed at time t'_{end} .

3.3.3 Building the general activity model

In the last step of the second learning layer, the general activity model GAM is built. For each t'_{end} in the sensor data stream, A_N feature vectors are generated. So, a vector feature $F2_{t'_{end}}$ for each activity A_i is built, taking into account its proper temporal window size $\Delta_{A'_i}$ in the three temporal sub-windows.

In Fig. 5 is illustrated this process. For each end label of an activity, a feature vector is computed that is used to train the classifier in the general activity model. The first and the second components correspond to the value predicted by the specific activity model AM_1 and the specific activity model AM_2 in a long-term sub-window. The third and fourth components correspond to the value predicted by the specific activity model AM_1 and the specific activity model AM_2 in a middle-term sub-window, respectively. The fifth and sixth components correspond to the value predicted by the specific

Fig. 5 Scheme of the process to compute feature vectors to train the classifier of the general activity model



activity model AM_1 and the specific activity model AM_2 in a short-term sub-window. The last component corresponds to the real activity of the end label. The components of the rest of features vectors have been removed for simplicity.

Finally, the set of feature vectors $F2^{t_{end}}$ is used to train the classifier C_{GAM} in order to build the general activity model GAM . The classifier used in this second learning layer may be the same or different from the classifiers used in the first learning layer. The only requirement is that it can receive feature vectors with the computed values in the first layer.

4 Experiments

In this section, first, the two datasets used in the experimentation to evaluate our proposal are reviewed. Then, the description of the experimental setup, which is adopted in the evaluation of our approach, is presented.

4.1 Datasets

In the evaluation of our proposal, the following two datasets have been used.

- (1) *VanKasteren* This dataset is a popular activity recognition dataset (Van Kasteren et al. 2008) of a smart environment. This dataset is composed by binary temporal data from a number of sensors, which monitored the ADLs carried out in a home setting by a single inhabitant. This dataset was collected in the house of a 26-year-old male who lived alone in a three-room apartment. This dataset contains 245 activities that are annotated in the stream of state-change sensors gener-

ated by 14 binary sensors. In this dataset, eight activities are classified: enter house, exit house, use toilet, take shower, go to bed, wake up, prepare food and, finally, get drink. From the original dataset, the activity go to bed was divided into going to sleep and wake up as well as the activity leave house was divided into exit and enter. Furthermore, breakfast and dinner activities were merged into eating, which can be naively discerned by the time of the day.

- (2) *Ordonez* This dataset is located in the UC Irvine Machine Learning Repository (Ordoñez et al. 2013a, b). The dataset represents two participants performing ten ADL activities in their own homes. The activities were performed individually and this dataset is composed by two instances of data, each one corresponding to a different user and summing up to a total of 35 days. In this dataset, there are 12 sensors and 10 activities that are classified as: enter house, exit house, use toilet, take shower, go to bed, wake up, prepare food, get drink, spare time and grooming. From the original dataset, the sleeping activity was divided into going to sleep and wake up, and the leaving activity was divided into exit and enter. Furthermore, breakfast, lunch and dinner activities were merged into eating, which can be naively discerned by the time of the day.

Furthermore, a type of virtual time sensor has been included in both datasets, which provides a binary activation in a time interval. With this simple method, it is possible to know the daily time context in the feature vector automatically. Four virtual time sensors were integrated in the following hour ranges: [0–6 h), [6–12 h), [12–18 h) and [18–0 h).

Table 5 Activity lengths expressed in minutes of each activity for Van Kasteren dataset

	Shower	Exit	Toilet	To sleep	Snack	Wake up	Eating	Enter
×	573.39	120	104.62	120	53.25	120	819.83	120
CV	155.04	0	100.56	0	67.06	0	1094.5	0

Table 6 Activity lengths expressed in minutes of each activity for Ordoñez dataset

	Shower	Exit	Toilet	Grooming	Going to sleep	Wake up	Eating	Spare time	Enter	Snack
×	403.5	240	434.1	1953.1	240	240	4997.2	6661.5	240	32.8
CV	257.7	0	1604.1	12959.6	0	0	18039.2	5699.9	0	81.9

4.2 Experimental setup

In the experimental setup, a time window size for each activity in the first layer is computed by statistical measures (mean, which is represented by \times , and standard deviation, which is represented by CV). The Table 5 shows measures by each activity for the Van Kasteren dataset and Table 6 shows this information for the Ordoñez dataset.

In the first layer, five values of w_1 are proposed to weigh the importance of the standard deviation in the window size: $w_1 = \{0, 0.25, 0.5, 0.75, 1\}$. In the second layer, three values of w_2 are proposed to weigh the importance of the standard deviation in the window size $w_2 = \{0, 0.5, 1\}$.

In the evaluation shown in this paper, three experimentations are shown that combine three popular classifiers. The first classifier is the C4.5 (Ordoñez et al. 2013a, b) that builds decision trees using the concept of information entropy. We have integrated post and pre-pruning techniques, using the confidence factor 40%. The second classifier is the support vector machine (SVM) that is focused on a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyper plane. A hyperplane is a decision boundary, which separates the tuples of one activity from another. In our experiment, the SVM has implemented the John Platt's sequential minimal optimization algorithm for training a support vector classifier (Keerthi et al. 2001). Finally, we have used the Naive Bayes classifier (NB) (John and Langley 1995). The basic idea in NB classifier is to use the joint probabilities of sensors and activities to estimate the category probabilities given a new activity. This method is based on the assumption of sensor independence, i.e., the conditional probability of a sensor given an activity is assumed to be independent of the conditional probabilities of other sensors given that activity.

The classification accuracy is computed on the test set by means a leave-one-out that is a particular case of f -cross-validation when $f=1$. The main advantage of this validation is that all the activities in the dataset are used for training and testing, avoiding the problem of considering how the dataset is divided.

5 Results and discussions

In this section, the set of results obtained in the set of experiments are presented.

Each end label of each activity in each dataset is compared to the classified activity with the proposed approach. The results have been presented in terms of classification accuracy. Let N_{A_i} be the number of activities of the class A_i , and TP_{A_i} the number of activities correctly classified by the proposed approach. The classification accuracy is defined by Eq. (7):

$$\text{Accuracy} = \sum_{i=1}^{N_{A_i}} \frac{TP_{A_i}}{N_{A_i}}. \quad (7)$$

The set of accuracy values in the VanKasteren dataset is shown in Table 7, with the following classifiers NA, SVM and C4.5, according to the experimental setup.

The column named as 1W represents the accuracy with the approach with two learning layers with a single temporal window. The column named 3SW represents the accuracy values using our proposal with three temporal sub-windows. The column named Dif., which means difference, represents the difference between the 1W column and the 3SW column.

In the case of the NA classifier, there is an average percentage difference of 4.7% and the accuracy is improved in the 100% of experiments. In the case of the SVM classifier, there is an average percentage difference of 14.17% and the accuracy is also improved in the 100% of experiments. In the case of the C4.5 classifier, there is an average percentage difference of 9.6% and the accuracy is improved in the 93.33% of experiments.

Second, the set of accuracy values in the Ordonez dataset is shown in Table 8, with the NA, SVM and C4.5 classifiers, following the experimental setup.

In the case of the NA classifier, there is an average percentage difference of 7.3% and the accuracy is improved in the 100% of experiments. In the case of the SVM classifier, there is an average percentage difference of 11.8% and the accuracy is also improved in the 100% of experiments. In

Table 7 Accuracy obtained with the Van Kasteren dataset using the classifiers: NA, SVM and C4.5, following the approach with two learning layers and a single temporal window (1W) and our approach with three temporal sub-windows (3SW)

w_1	w_2	NA			SVM			C4.5		
		1W	3SW	Dif.	1W	3SW	Dif.	1W	3SW	Dif.
0	0	77.6	78.9	1.3	73.3	79.2	5.9	81.2	78.2	-3.0
0	0.25	75.9	77.9	2.0	70.3	81.8	11.6	75.6	81.5	5.9
0	0.5	75.2	77.2	2.0	65.0	78.2	13.2	75.2	80.9	5.6
0	0.75	72.9	79.2	6.3	73.6	80.5	6.9	72.3	80.5	8.3
0	1	75.9	78.2	2.3	65.0	81.5	16.5	71.9	80.9	8.9
0.5	0	74.3	80.9	6.6	71.6	78.5	6.9	76.6	79.2	2.6
0.5	0.25	71.0	80.5	9.6	66.3	81.5	15.2	73.3	80.5	7.3
0.5	0.5	73.9	77.9	4.0	66.0	78.2	12.2	68.6	79.5	10.9
0.5	0.75	65.7	74.9	9.2	54.8	78.5	23.8	63.4	80.5	17.2
0.5	1	69.3	73.6	4.3	63.0	80.2	17.2	66.0	79.2	13.2
1	0	70.6	75.6	5.0	66.3	80.5	14.2	76.9	81.2	4.3
1	0.25	70.0	76.6	6.6	58.4	83.2	24.8	74.6	82.2	7.6
1	0.5	71.3	74.9	3.6	54.1	80.2	26.1	62.7	79.9	17.2
1	0.75	70.0	71.9	2.0	63.7	64.0	0.3	63.7	80.9	17.2
1	1	68.0	73.3	5.3	61.7	79.5	17.8	58.4	79.2	20.8

Bold values indicate the main differences

Table 8 Accuracy obtained with the Ordoñez dataset using the classifiers: NA, SVM and C4.5, following the approach with two learning layers and a single temporal window (1W) and our approach with three temporal sub-windows (3SW)

w_1	w_2	NA			SVM			C4.5		
		1W	3SW	Dif.	1W	3SW	Dif.	1W	3SW	Dif.
0	0	66.7	80.4	13.8	81.5	86.6	12.0	79.0	89.1	10.1
0	0.25	68.8	72.8	4.0	69.6	81.5	4.3	74.3	81.5	7.2
0	0.5	64.5	70.3	5.8	72.1	76.4	3.6	71.0	74.3	3.3
0	0.75	59.4	65.2	5.8	70.7	74.3	8.0	68.1	69.6	1.4
0	1	56.2	62.0	5.8	65.9	73.9	5.1	67.8	68.5	0.7
0.5	0	67.0	71.4	4.3	81.5	86.6	21.7	79.7	89.1	9.4
0.5	0.25	62.7	68.8	6.2	59.8	81.5	15.2	73.2	82.6	9.4
0.5	0.5	54.3	65.2	10.9	63.8	79.0	17.8	62.7	81.5	18.8
0.5	0.75	50.0	61.6	11.6	62.3	80.1	14.9	60.1	77.2	17.0
0.5	1	51.1	57.6	6.5	58.0	72.8	10.9	56.2	76.4	20.3
1	0	65.6	68.8	3.3	74.3	85.1	21.7	75.0	86.6	11.6
1	0.25	56.5	65.2	8.7	58.3	80.1	14.5	67.8	84.4	16.7
1	0.5	48.6	55.8	7.2	56.9	71.4	14.1	57.2	74.3	17.0
1	0.75	48.9	56.9	8.0	57.2	71.4	7.6	57.6	68.5	10.9
1	1	47.1	54.3	7.2	54.0	61.6	12.0	56.5	62.0	5.4

Bold values indicate the main differences

the case of the C4.5 classifier, there is an average percentage difference of 10.6% and the accuracy is improved in the 100% of experiments.

As additional comments, we highlight that the C4.5 and SVM classifiers have provided a higher performance than NA. It is because NA is weak when detecting the interactions among features (Yala et al. 2017), which is key in the activity recognition process. Furthermore, we note that on adjusting of the size of the temporary windows in the first and second layers, w_1 and w_2 respectively; we can observe that the influence of w_1 is mostly sensitive to w_2 because of

handling with lower-level information. In the first layer, the input from sensor data streams contains more attributes and variability of the samples over time, in regards to the stream of prediction of activities input in second layer, which is based on more stable and higher-level information.

In light of the results, we can indicate the evaluation of distinct values for window sizes shows a stable accuracy, describing the flexibility of our approach for handling different sizes. The experiments conducted to evaluate our approach demonstrate the relevance of the use of temporal sub-windows versus a single temporal window in terms

of accuracy. So, 98.95% of experiments have improved the accuracy by using three temporal sub-windows.

Finally, one of the main strengths of our approach is that it is not closely related to the specific classifiers in the specific activity models neither in the first learning layer nor in the global activity models in the second learning layer.

6 Conclusions and future works

Activity recognition from sensor data stream for assisted living is based on unobtrusive sensors that do not disturb people in their daily life. There is a need of online activity recognition approaches driven by data that classify activities using only the end label of each activity in the dataset. This paper has been focused on the definition of three temporal sub-windows from the end label of each activity to compute the set of feature vectors to represent the information in a partial order in an approach for activity recognition with two learning layers. The first layer is considered for learning each activity model, using the sensor data stream generated by sensors in the three temporal sub-windows. Once the specific activity models have been trained, the set of obtained values from specific classifiers composes the input of the second layer. In the second layer, a general classifier among activities is trained, considering also three temporal sub-windows in order to obtain the classified activity among the set of potential activities.

In this paper, three temporal sub-windows have been selected to prove the concept and benefits of using sub-windows compared to using just a single time-window, while providing balance between accuracy gain and computation time.

The main strengths of our approach can be summarized in four key points. (i) A main advantage of our proposed approach, which differs from the other proposed approaches in the literature, is that our approach uses only the end label of each activity in the dataset in the training and testing processes. This is valuable for in-the-wild trials when the start of an unscripted activity is unknown at real-time. (ii) The success of our approach is based on the partial order of the sensor events as well as the activities that are represented in three temporal sub-windows. (iii) The proposed approach considers two learning layers to search a deep high-level knowledge from the learning. The first layer is focused on the sensor events stream and the second layer is focused on the activities streams. (iv) The proposed approach is not tied to a specific classifier; the approach is open to use different classifiers in each specific activity model in the first learning layer as well as in the general activity model in the second learning layer.

Our future work is focused on studying the optimum number of temporal sub-windows for the activity recognition

process to shed light on the recognition of overlapping activities. Furthermore, another more ambitious plan for future work is focused on performing activity recognition without needing to know when the activity has started or ended. To do so, the sensor data stream will be processed in periods, following the proposed approach and fixing a threshold to identify when an activity is performed.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant agreement No 734355 together the Spanish government by research project TIN2015-66524-P.

References

- Alemdar H, Ersoy C (2017) Multi-resident activity tracking and recognition in smart environments. *J Ambient Intell Humaniz Comput* 8(4):513–529. <https://doi.org/10.1007/s12652-016-0440-x>
- Banos O, Galvez J-M, Damas M, Pomares H, Rojas I (2014) Window size impact in human activity recognition. *Sensors (Basel Switzerland)* 14(4):6474–6499. <https://doi.org/10.3390/s140406474>
- Bengio Y (2009) Learning deep architectures for AI. *Found Trends Mach Learn* 2(1):1–27. <https://doi.org/10.1561/22000000006>
- Calvaresi D, Cesarini D, Sernani P et al (2017) J Ambient Intell Human Comput 8:239. <https://doi.org/10.1007/s12652-016-0374-3>
- Carnevali L, Nugent C, Patara F, Vicario E (2015) A continuous-time model-based approach to activity recognition for ambient assisted living. In: Campos J, Haverkort B (eds) *Quantitative evaluation of systems. QEST 2015. Lecture notes in computer science*, vol 9259. Springer, Cham. https://doi.org/10.1007/978-3-319-22264-6_3
- Chan T-H, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2015) PCANet: a simple deep learning baseline for image classification? *IEEE Trans Image Process* 24(12):5017–5032. <https://doi.org/10.1109/TIP.2015.2475625>
- Chen L, Nugent C (2009) Ontology-based activity recognition in intelligent pervasive environments. *Int J Web Inf Syst* 5:410–430. <https://doi.org/10.1108/17440080911006199>
- Chen L, Nugent C, Wang H (2012) A knowledge-driven approach to activity recognition in smart homes. *IEEE Trans Knowl Data Eng* 24:961–974. <https://doi.org/10.1109/TKDE.2011.51>
- Chen B, Fan Z, Cao F (2015) Activity recognition based on streaming sensor data for assisted living in smart homes. In: *Intelligent Environments (IE), 2015 International Conference*, pp 124–127. <https://doi.org/10.1109/IE.2015.25>
- Cook DJ (2010). Learning setting-generalized activity models for smart spaces. *IEEE Intell Syst*. <https://doi.org/10.1109/MIS.2010.112>
- Dietterich TG (2002) Machine learning for sequential data: a review. In: Caelli T, Amin A, Duin R.P.W., de Ridder D, Kamel M (eds) *Structural, syntactic, and statistical pattern recognition. SSPR / SPR 2002. Lecture notes in computer science*, vol 2396. Springer, Berlin
- Espinilla M, Medina J, Calzada A, Liu J, Martínez L, Nugent C (2017) Optimizing the configuration of and heterogeneous architecture of sensors for activity recognition, using the extended belief rule-based inference methodology. *Microprocess Microsyst* 52:381–390. <https://doi.org/10.1016/j.micpro.2016.10.007>
- Gu T, Wang L, Wu Z, Tao X, Lu J (2011) A pattern mining approach to sensor-based human activity recognition. *IEEE Trans Knowl Data Eng* 23:1359–1372. <https://doi.org/10.1109/TKDE.2010.184>
- Gutiérrez López de la Franca, C, Hervás R, Johnson E, Mondéjar T, Bravo J (2017) Extended Body-Angles

- Algorithm to recognize activities within intelligent environments. *J Ambient Intell Humaniz Comput* 8(4):531–549. <https://doi.org/10.1155/2017/7932529>
- John GH, Langley P, (1995) Estimating continuous distributions in Bayesian classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, pp 338–345
- Junker H, Amft O, Lukowicz P, Traster G (2008) Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recogn* 41:2010–2024. <https://doi.org/10.1016/j.patcog.2007.11.016>
- Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KKR (2001) Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Comput* 13(3):637–649. <https://doi.org/10.1162/089976601300014493>
- Krishnan NC, Cook DJ (2014) Activity recognition on streaming sensor data. *Pervasive Mobile Comput* 10:138–154. <https://doi.org/10.1016/j.pmcj.2012.07.003>
- Krüger F, Nyolt M, Yordanova K, Hein A, Kirste T (2014) Computational state space models for activity and intention recognition. A feasibility study. *PLoS ONE* 9(11):e109381. <https://doi.org/10.1371/journal.pone.0109381>
- Li C, Lin M, Yang L, Ding C (2014) Integrating the enriched feature with machine learning algorithms for human movement and fall detection. *J Supercomput* 67:854–865. <https://doi.org/10.1007/s11227-013-1056-y>
- López G, Marín G, Calderón M (2017) Human aspects of ubiquitous computing: a study addressing willingness to use it and privacy issues. *J Ambient Intell Humaniz Comput* 8(4):497–511. <https://doi.org/10.1007/s12652-016-0438-4>
- Nugent C, Synnott J, Santanna A, Espinilla M et al. (2016). An initiative for the creation of open datasets within the pervasive healthcare. In: 10th EAI International Conference on Pervasive Computing Technologies for Healthcare, Cancun, Mexico, 16–19 May 2016, pp 180–183. <https://doi.org/10.4108/eai.16-5-2016.2263830>
- Ordoñez F, de Toledo P, Sanchis A (2013) Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors (Switzerland)* 13:5460–5477. <https://doi.org/10.3390/s130505460>
- Ordóñez FJ, Iglesias JA, De Toledo P, Ledezma A, Sanchis A (2013) Online activity recognition using evolving classifiers. *Expert Syst Appl* 40(4):1248–1255. <https://doi.org/10.1016/j.eswa.2012.08.066>
- Quesada FJ, Moya F, Medina J, Martínez L, Nugent C, Espinilla M (2015) Generation of a partitioned dataset with single, interleave and multioccupancy daily living activities. In: García-Chamizo J, Fortino G, Ochoa S (eds) Ubiquitous computing and ambient intelligence. sensing, processing, and using environmental information. Lecture notes in computer science, vol 9454. Springer, Cham, pp. 60–71. https://doi.org/10.1007/978-3-319-26401-1_6
- Quinlan R (1993) 4.5: programs for machine learning. Morgan Kaufmann Publishers, Burlington
- San Mateo CA, San Martín L, Pelaez V, Gonzalez R, Campos A, Lobato V (2010) Environmental user-preference learning for smart homes: an autonomous approach. *J Ambient Intell Smart Environ* 2:327–342
- Schmitter-Edgecombe D, Crandall AM, Sanders C, Thomas B (2009) Collecting and disseminating smart home sensor data in the CASAS project. In: Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research
- Tapia EM, Intille SS, Larson K (2004) Activity recognition in the home using simple and ubiquitous sensors. In: Ferscha A, Mattern F (eds) Pervasive computing. Pervasive 2004. Lecture notes in computer science, vol 3001. Springer, Berlin. https://doi.org/10.1007/978-3-540-24646-6_10
- Triboan D, Chen L, Chen F, Wang Z (2017) Semantic segmentation of real-time sensor data stream for complex activity recognition. *Personal Ubiquitous Comput* 21(3):411–425. <https://doi.org/10.1007/s00779-017-1005-5>
- Van Kasteren T, Noulas A, Englebienne G, Krose B (2008) Accurate activity recognition in a home setting. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp 1–9. <https://doi.org/10.1145/1409635.1409637>
- Wang L, Gu T, Tao X, Lu J (2012) A hierarchical approach to real-time activity recognition in body sensor networks. *Pervasive Mobile Comput* 8:115–130. <https://doi.org/10.1016/j.pmcj.2010.12.001>
- Wang J, Chen Y, Hao S, Peng X, Hu L (2017) Deep learning for sensor-based activity recognition: a survey. ArXiv e-print. arXiv:1707.03502v2
- Yala N, Fergani B, Fleury A (2017) Towards improving feature extraction and classification for activity recognition on streaming data. *J Ambient Intell Humaniz Comput* 8(2):177–189. <https://doi.org/10.1007/s12652-016-0412-1>
- Zhang T, Fu W, Ye J, Fischer M (2017) Learning movement patterns of the occupant in smart home environments: an unsupervised learning approach. *J Ambient Intell Humaniz Comput* 8(1):133–146. <https://doi.org/10.1007/s12652-016-0367-2>