# Efficient Activity Recognition in Smart Homes Using Delayed Fuzzy Temporal Windows on Binary Sensors

Rebeen Ali Hamad, Alberto Salguero Hidalgo, Mohamed-Rafik Bouguelia, Macarena Espinilla Estevez, and Javier Medina Quero

***Abstract*—Human activity recognition has become an active research field over the past few years due to its wide application in various fields such as health-care, smart home monitoring, and surveillance. Existing approaches for activity recognition in smart homes have achieved promising results. Most of these approaches evaluate real-time recognition of activities using only sensor activations that precede the evaluation time (where the decision is made). However, in several critical situations, such as diagnosing people with dementia, "preceding sensor activations" are not always sufficient to accurately recognize the inhabitant's daily activities in each evaluated time. To improve performance, we propose a method that delays the recognition process in order to include some sensor activations that occur after the point in time where the decision needs to be made. For this, the proposed method uses multiple incremental fuzzy temporal windows to extract features from both preceding and some oncoming sensor activations. The proposed method is evaluated with two temporal deep learning models (convolutional neural network and long short-term memory), on a binary sensor dataset of real daily living activities. The experimental evaluation shows that the proposed method achieves significantly better results than the real-time approach, and that the representation with fuzzy temporal windows enhances performance within deep learning models.**

***Index Terms*—Activity recognition, fuzzy temporal windows, deep learning, temporal evaluation.**

## I. Introduction

SMART homes make use of distributed sensor networks with high processing capabilities and low power consumption, and have the ability to record information about the behavior of an inhabitant who interacts with the environment [1], [2]. These environments are made to perceive the user's context in order to help people in their daily living activities and provide smart solutions to address some of the problems associated with the growing size of the population [3], [4].

Activity Recognition (AR) in smart homes is an active research topic as well as a useful means to assess the circumstances of a person's daily living. It aims to predict human activities within the smart environment by continuously observing a series of actions and environmental conditions. In the health context, AR plays an important role in assessing the patient's condition. Examples of this include the identification of abnormal behaviours for elderly dementia sufferers [5], and the early detection of Alzheimer disease [6].

In data-driven AR, an activity model is built by training machine learning models on sensor data corresponding to various activities of the inhabitant. Then, a testing stage is carried out to evaluate the built model on a stream of sensor activations [7]. In particular, binary sensors are deemed to be one of the useful means for unobtrusive monitoring within the home environment. One of the advantages of such data-driven AR approaches consists in the capability of handling uncertainty and temporal information. Among these approaches, Deep Learning methods have been used for activity recognition, and evaluated on binary sensors [8] by using Convolutional Neural Networks (CNN) [9] and Long Short-Term Memories (LSTM) [10].

Existing approaches for activity recognition in smart homes perform predictions in real time based on sensor activations that precede the evaluation time. Due to only relying on previous activations, such real-time approaches may lack precision in recognizing some daily life activities. To overcome this problem, it becomes necessary to know which sensor activations are generated later, since the activity to be recognized will depend on the subsequent sensors. In order to illustrate the limitation of the real-time approaches for activity recognition, let's consider the following two scenarios. In the first scenario, if a binary sensor deployed on the front door generates an open activation, the real-time approach will recognize the activity as "the inhabitant has left the house". However, it can happen that the inhabitant opens the main door only to talk to another person in the entrance of the house and returns to the house again without leaving it. In this scenario, it is important to consider the activations generated after opening the door. Indeed, if nothing happens after a short time (e.g., two minutes), then the inhabitant has performed the activity "leaving the house"; however, if sensor activations

Manuscript received February 14, 2019; revised April 23, 2019; accepted May 15, 2019. Date of publication May 22, 2019; date of current version February 6, 2020. This work was supported by the REMIND project Marie Sklodowska-Curie EU Framework for Research and Innovation Horizon 2020 under Grant 734355. *(Corresponding author: Rebeen Ali Hamad.)*

R. A. Hamad and M.-R. Bouguelia are with the Department of Intelligent Systems and Digital Design, Halmstad University, Halmstad 30118, Sweden (e-mail: rebali@hh.se; mohbou@hh.se).

A. S. Hidalgo is with the Department of Computer Science, University of Cádiz, Cádiz 11001, Spain (e-mail: alberto.salguero@uca.es).

M. E. Estevez and J. M. Quero are with the Department of Computer Science, University of Jaén, Jaén 23071, Spain (e-mail: mestevez@ujaen.es; jmquero@ujaen.es).

Digital Object Identifier 10.1109/JBHI.2019.2918412

are generated during the next two minutes after opening the door, then the inhabitant has only opened the door without leaving the house. The second scenario that illustrates the problem of real-time activity recognition approaches is as follows. An inhabitant goes to the kitchen and opens the refrigerator. Until this action is followed up with interactions with other objects (and therefore the activation of other sensors) it is not possible to recognize exactly whether the inhabitant has gone to open the refrigerator to have a snack and eat, or to perform some other activity, for example, to take out certain products from the refrigerator for cooking. Therefore, the accuracy of a real-time (or online) activity recognition approach is confronted with the use of oncoming sensor activations with a slight delay.

Although real-time approaches enable permanent interaction with users in smart environments, it has a lower recognition rate [11], [12]. This is not admissible in some critical cases such as diagnosing dementia, as it requires more accurate activity recognition to correctly detect abnormal behaviours in the inhabitant, [13]. These cases, although less frequent, prevent AR from being a high-precision tool for assessing the conditions of inhabitants in smart homes.

The contribution of this paper takes the above-mentioned issues into consideration. We propose a data-driven approach that aims to increase precision and sensitivity in daily activity recognition by means of i) delaying the activity recognition, ii) extracting representations of binary sensor activations that occur before and after the time where the prediction is made, iii) evaluating Deep Learning methods for classification, and iv) analyzing the impact of the delayed AR process on precision and sensitivity.

The remainder of this paper is structured as follows: Section II describes previous work in activity recognition related to our proposal, highlighting the main novelty of the proposed methodology. Section III presents the proposed methodology for activity recognition as a high-precision tool. In Section IV, the proposed methodology is evaluated on a popular dataset with daily activities developed in a real smart environment [4]. Finally, a conclusion and future works are proposed in Section V.

## II. Related Works

Activity recognition based on the use of binary sensors is a useful approach to evaluate the conditions of daily living within a sensorized environment in an unobtrusive manner. Binary sensors are small and light devices which are installed in everyday objects to register human interaction, such as, passive infrared sensors, motion detectors, contact switches, break-beam sensors, and pressure mats [14]. They have been proposed as suitable devices for describing daily human activities within a smart environment setting. Their main advantages are that they are: i) easy to install, ii) small in size, iii) low-cost and iv) minimally invasive in comparison to videos and microphones [15].

The real-time capabilities have become a key challenge in activity recognition to offer a tool that meets real-world conditions, and enable *adequate assistance services* which can be offered within Ambient Assisted Living [16]. Previously, a combination of human-defined binary sensor features, such as the last activation within a fixed time period, or the current raw activation, were proposed as suitable representations in real-time AR using windowing approaches [17]. In [18], the combination of multiple and incremental fuzzy temporal windows has shown an increase in the performance of real-time AR compared to previous binary sensor representations. Most of the algorithms carry out successful analyses on stored datasets, but in real time, data should be quickly pre-processed for proper action to be taken if needed [19]. Using smart home settings, activity recognition systems attempt to recognize daily routine activities. [20] recognizes a set of activities such as *Breakfast, Brushing teeth, Drinking, Showering*, using machine learning algorithms including Hidden Markov mode (HMM), condition random field (CRF), hidden semi-Markov models (HSMM) and semi-Markov conditional random field (SMCRF). [21] recognizes daily living activities *bedtoilet, transition, leaving, eating, cooking, relaxing, sleeping and working* using Naive Bays, HMM, SVM, CRF, SVMS, which consistently obtains 84% average weighed accuracy. The increasingly large amounts of smart home datasets have lead to the use of deep learning approaches. [22] and [8] applied Long Short Term Memory (LSTM) on three real world smart home datasets where the number of activities are 10, 13, 16 for the first, second and third houses respectively. The results of LSTM outperforms Naive Bays, HMM, HSMM, CRF in terms of accuracy and performance. However, the features of the datasets for the input models were computed using only equally-sized temporal windows and the techniques of separating data for training and testing phases to avoid over-fitting are not described. Moreover, the accuracy of the individual classes are not computed, rather only average accuracy of the models is shown and the training time of each model is not computed. Hence, the recognition of which activities are improved is not clear.

As explained previously, since real-time activity recognition approaches use only the preceding sensor activations, they can perform poorly in recognizing some activities [11]. From an operational perspective, there are situations where one needs to achieve higher precision while tolerating a short time delay (i.e. real time is not necessary). For example, when monitoring a person with early dementia using binary sensors, it is necessary to provide high accuracy even if it means introducing a short delay in recognizing the activity. The proposed data-driven approach differs from the previously proposed ones by taking into consideration the aforementioned issues, and aiming to further increase the precision and sensitivity of the activity recognition by means of:

- Including multiple and incremental fuzzy temporal windows (FTWs) to compute the features from a given sensor in both preceding activations, as well as oncoming activations. In order to avoid the human configuration of FTWs we have modeled their shapes with the Fibonacci sequence, which has been defined to model incremental sequences in a harmonic way under the fields of mathematics, science, and engineering [23].
- Evaluating the temporal models using a Deep Learning approach has presented significant differences compared to other contexts of activity recognition [24]. Hybrid models combining CNN and LSTM are evaluated as Deep Learning architectures. This is compared with other models

under delayed representation of sensor activations. Moreover, we have evaluated the configuration for LSTM and CNN described in [8] based on equal-sized temporal windows for extracting features, showing the impact of FTWs in extracting features to represent suitable long-term performance of sensor activations with a low number of features and achieving better F1-score.

- Analyzing the impact of delaying the activity recognition by using oncoming sensor activations in real daily activities. Specifically, showing the impact of including oncoming sensor activations in activity recognition to improve the recognition of some rare activities such as *leaving, snack, grooming and dinner* that have been poorly recognized using only real-time activity recognition.

Consequently, to the best of our knowledge, including and evaluating both preceding and oncoming sensor activations for activity recognition has not been attempted before. Human activity recognition considering oncoming sensors activation plays a key role in recognizing activities properly for the purpose of enhancing elderly care alert systems. In concrete terms, in building robust AR for diagnosing patients based on their user activities in smart homes in a highly accurate way.

## III. Methodology

In order to increase the precision and sensitivity of activity recognition in smart homes, in this paper we propose a new data-driven methodology that applies multiple and incremental fuzzy temporal windows to extract features from both preceding and partial oncoming sensor activations. This results in a sequence of temporal features that boosts learning using LSTM sequence models as well as CNN.

### A. Representation of Activities and Binary Sensors

A set of binary sensors is represented by $S = \{S_1, \ldots, S_{|S|}\}$ whose related set of daily activities is represented by $A = \{A_1, \ldots, A_{|A|}\}$, where $|S|$ and $|A|$ are the number of sensors and daily activities respectively. Each binary sensor and each daily activity are described by a set of binary activations within a range of time, defined by a starting and ending point of time as shown by Eq. (1):

$$\begin{aligned} S_i &= \{S_{i_0}, \ldots, S_{|S_i|}\}, S_{i_j} = \{S^0_{i_j}, S^+_{i_j}\} \\ A_i &= \{A_{i_0}, \ldots, A_{|A_i|}\}, A_{i_j} = \{A^0_{i_j}, A^+_{i_j}\} \end{aligned} \tag{1}$$

where $i$ in $|S_i|$ and $|A_i|$ is the total number of activations for a given binary sensor $S_i$ and a daily activity respectively, and ii) $S^0_{i_j}, S^+_{i_j}$ the starting and ending point of a given time of activation respectively. Finally, the timeline is determined by the range of time $T = \{min(S^0_{i_j}), \max(S^+_{i_j})\}$.

### B. Feature Sequence of Binary Sensors With Incremental Fuzzy Temporal Windows

In this Section, a binary-sensor representation by means of multiple and incremental FTWs is described. A FTW describes a membership degree from a given time $t^*$ to a point of time $t_i$ by means of their temporal distance $\Delta t^*_i = t^* - t_i, t^* > t_i$. In case $\Delta t^*_i > 0$, we are evaluating a preceding point of time, and in case $\Delta t^*_i < 0$ an oncoming point of time. A FTW $T_k$ is defined by a fuzzy set $T_k(\Delta t^*_i)$, which is characterized by a membership function $\mu_{T_k}(\Delta t_i)$. In sake of simplicity we note $T_k(\Delta t^*_i)$ instead of $\mu_{T_k}(\Delta t^*_i)$.

In this work, each FTW $T_k$ is described by a fuzzy set characterized with a membership function and its shape corresponds to a trapezoidal function $T_k(\Delta t^*_i)[l_1, l_2, l_3, l_4]$. The well-known trapezoidal membership functions are defined by a lower limit $l_1$, an upper limit $l_4$, a lower support limit $l_2$, and an upper support limit $l_3$ according to Eq. (2):

$$TS(x)[l_1, l_2, l_3, l_4] = \begin{cases} 0 & x \leq l_1 \\ (x - l_1)/(l_2 - l_1) & l_1 < x < l_2 \\ 1 & l_2 \leq x \leq l_3 \\ (l_4 - x)/(l_4 - l_3) & l_3 < x < l_4 \\ 0 & l_4 \leq x \end{cases} \tag{2}$$

Once a FTW $T_k$ is defined, the activation degree of a binary activation $S_{i_j}$ from a sensor $S_i$ at evaluated time $t^*$ is computed by Eq. (3).

$$T_k(S_{i_j}, t^*) = \begin{cases} \max(T_k(\Delta t^*_i)) \forall t_i \in S_{i_j} & \exists t_i \in [S^0_{i_j}, S^+_{i_j}] \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

So, in order to obtain the degree of the binary activation $S_{i_j}$ within the FTW $T_k$, we compute the maximal degree of $T_k$ for each point $t_i$ of time within the time interval from sensor activation.

In Fig. 1, some examples of activation degree from a FTW $T_k$ and three different binary activations $S_{i_j}$ are shown.

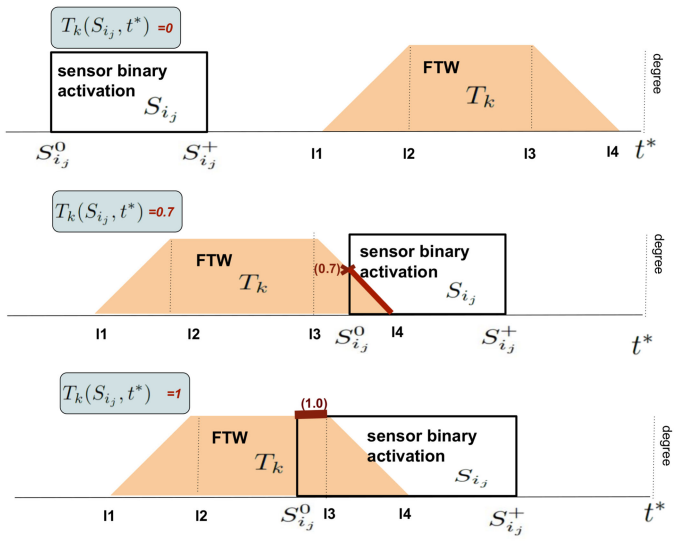



Fig. 1. Activation degree from a FTW $T_k$ and different binary activations $S_{i_j}$, which is computed as the maximal degree of $T_k$ for each point $t_i$ of time within the sensor activation. Top) No overlapping between $T_k$ and $S_{i_j}$, middle) partial overlapping between $T_k$ and $S_{i_j}$ and bottom) partial overlapping within support limits of $T_k$.

Next, in order to evaluate a FTW $T_k$ and a sensor $S_i$ in the complete timeline, we aggregate $T_k(S_i, t^*)$ computing the maximal value in the time-line according to Eq. (4).

$$T_k(S_i, t^*) = \max(T_k(S_{i_j}, t^*)), \forall S_{i_j} \in S_i \quad (4)$$

This representation includes an updated model from previous works [18], [25], where a fuzzy aggregation of the sensor activation in wearable and binary sensors was initially proposed. In this work, we have simplified and improved the computing of the activation degree by evaluating the sensor activation and FTW in a continuous way using Eq. (3). In Fig. 2, a representation of preceding and oncoming FTWs defined by the Fibonacci sequence in an sample timeline is shown.

### C. Preceding and Oncoming Sensor Activation With FTWs

In this work multiple and incremental FTWs are defined to collect i) long-term to short-term temporal sensor activations, and ii) the preceding and oncoming sensor activations. To do so, first, the maximal temporal size is defined for evaluating preceding and oncoming sensor activations, which are defined by $L^*_{-;}$ and $L^*_+$ respectively. We note:

- $L^*_{-;}$ must include a value which determines a wide range of preceding evaluation of sensors, as it has been demonstrated that long-term activations of sensors are key to improve AR [18], [26]. For example, the difference between some activities carried out in the kitchen, such as breakfast, lunch, dinner or snack, is determined by the evaluation of the bed sensor activation in short, middle and long term. In addition, due to temporal aggregation with incremental FTWs, defining hours of sensor evaluation are reduced to a short number of features [18]. The definition of preceding FTWs is determined by $T_k(\Delta t^*_i)$ where $\Delta t^*_i > 0, t^* - L^*_{-;} \leq t_i \leq t^*$.
- $L^*_+$ represents the maximal time to evaluate oncoming sensor activations. A long-term evaluation will improve the results. For this reason, in the same way as $L^*_{-;}$, with preceding sensor activations, the results also improved. It generates a delay in comparison with real-time AR. However, the results obtained will be more precise. The more time is spent to evaluate oncoming sensor activations, the more time is needed to obtain the recognized activity. So, $t^+$ defines the evaluating time which is defined as the point of time where the AR evaluates the time $t^*$. The definition of oncoming FTWs is determined by $T_k(\Delta t^*_i)$ where $\Delta t^*_i < 0, t^* \leq t_i \leq t^+$.

Second, in order to generate multiple and incremental FTWs from $L^*_{-;}$ and $L^*_+$ in a simple manner, the following process is proposed:

- A set of incrementally ordered evaluation times $L = \{L_1, \ldots, L_{|L|}\}, L_{i-1} < L_i$, which define the limits of the trapezoidal functions, are calculated according to the temporal window index $T_k = T_k(\Delta t^*_i)[L_k, L_{k-1}, L_{k-2}, L_{k-3}]$.
- $L$ is defined by the Fibonacci sequence [27] $L = \{1, 2, 3, 5, 8, \ldots\}$, which has been previously shown as a successful sequence for defining FTWs without requiring expert knowledge definition [26].
- $L_+$ and $L_{-;}$ are established from the maximal temporal size $L^*_{-;}$ and $L^*_+$ respectively. In both cases, the number of FTWs defined by Fibonacci is a set to the closer value of $L^*_{-;}$ and $L^*_+$ in the sequence $L$. For example, if $L^*_+ = 240m$, $L_+ = \{0m, 1m, 1m, 2m, 3m, 5m, \ldots, 144m, 240m\}$.

### D. Configuring Sequence Features With FTWs, Time Segmentation and Labeling

The use of FTWs $T_k$ defines a sequence representation of preceding and oncoming features for each sensor $S_i$ in a given point of time $t_i$. For evaluation purposes of the AR model, the activity is estimated for each evaluation time step of the timeline [17].

So, the timeline $T = \{min(S^0_{i_j}), max(S^+_{i_j})\}$ is divided in time steps $t^*$ defined by a temporal granularity $\Delta T$ where $t^{*-1} = t^* - \Delta T, t^{*+1} = t^* + \Delta T, \forall t^*$, which configure the points of time where the AR is evaluated.

For each evaluated time step $t^*$ and sensor $S_i$, the preceding and oncoming FTWs are computed to determine the sequence features. It is worth noting that the FTW parameters are defined according to distance to the evaluated time $t^*$, so a sliding window configuration [28] is developed in the timeline $T$ to compute the features.

Finally, in order to label each evaluated time step $t^*$, the activity $A_i$ is selected, which is developed if and only if $\exists t^* \in [A^0_{i_j}, A^+_{i_j}] \forall A_{i_j}$.

## IV. Experimental Setup and Evaluation

### A. Data

Data from two real intelligent homes A and B in which participants perform their daily routine were acquired for this work [4]. These two homes are equipped with sensors that are able to capture the movements and interactions of the resident. In home A, 9 human daily activities that were performed in 14 days over a period of 19,932 minutes were described by an incoming stream of binary events from 12 sensors in the home. In home B, 10 human daily activities that were performed in 22 days over a period of 30,495 minutes were described by 12 binary sensors. The timeline of the activities is segmented in time slots using the window size $\Delta$ t=60 s, based on the standard reference from [8], [17], [24]. The activities of homes A and B are *Breakfast, Grooming, Leaving, Lunch, Showering, Sleeping, Snack, Spare Time, Toileting*; in addition to these, home B has the activity *Dinner*.

The leave-one-day-out cross-validation is used for evaluation, where a single day of the activities is used for the test set and the rest of the days are used for the training set. This process is repeated until the data from all the days is involved in the training set and the testing set [15]. The average F-score is computed from the results of the cross-validation as done in [8], [24].

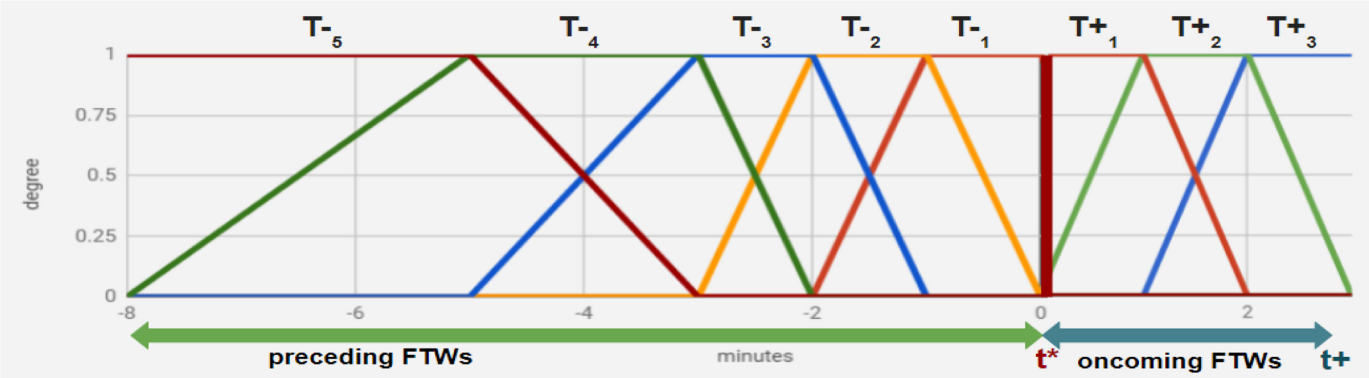



Fig. 2. Example of preceding and oncoming FTWs $L_+ = \{0, 1, 1, 2, 3\}$ and $L_{-;} = \{0, 1, 1, 2, 3, 5, 8\}$. Evaluation time $t^+$ and the evaluated time $t^*$ in the timeline are shown.

Since the classes of the datasets are imbalanced, the sampling approach is used by randomly oversampling observations of the minority class and undersampling observations of the majority class to have an equal number of samples (5000 samples) for each class. This allows us to create a more balanced dataset and avoid having models biased toward one class or the other [18], [29].

### B. Feature Vector

Features are computed by applying 15 FTWs on the raw data from all 12 binary sensors in each minute for both datasets. The datasets A and B have 19,932 and 30,495 samples respectively, where each sample represents one minute of data with $12 \times 15 = 180$ features. The resulting datasets are used for real-time activity recognition. Feature extraction based on FTWs is evaluated and compared with Equally-sized (1 minute) temporal windows (ESTWs) [8] and Raw Last sensor Activation (RAW) in one-minute windows [26]. Regarding the feature extraction with delays in time, different delays in time are considered, which are 5 minutes, 20 minutes, 1 hour, and 4 hours. For example, when a 5-minutes delay is used over real time, 4 FTWs in addition to 15 FTWs are computed, where the 4 FTWs represent the 5-minute delay in time, hence, the number of features becomes $12sensors \times 19FTWs$. For other delays in time, 20 minutes, 1 hour, and 4 hours, the number of features for each sensor becomes 22, 24, and 27 respectively.

### C. Model Selection and Architecture

The models that have been used in this study are described below.

- Long-Short Term Memory (LSTM): a type of recurrent neural network (RNN) that includes a memory and is designated to learn from sequence data, such as sequences of observations over time. LSTM is most widely used in natural language processing and speech recognition that can model temporal dependence between observations [30]. LSTM has obtained satisfying results in activity recognition [31], [32]. Hence, in this study LSTM is designed to be used in the experiments by stacking a LSTM layer with 40% dropout rate and 0.001 learning rate followed by a fully-connected, i.e., a dense layer and a soft-max layer. For all the models in this study the batch size and training epochs are equal to 10, which is a total of 100 batches during the entire training process. Commonly, large batch sizes result in quicker progress in training but mostly do not always converge as fast. On the other hand, smaller batch sizes train slower but could converge faster, therefore it is mostly an independent problem [33]. Regarding the 40% dropout, which is a regularization technique for preventing deep learning models from overfitting [34], the dropout ignores randomly selected neurons during the training phase. Those ignored neurons are temporally removed on the forward pass and their weights are not updated on the backward pass.
- Convolutional Neural Network (CNN): used in the experiments because CNN is competent in extracting features from signals. CNN has obtained promising results in image classification, text analysis and speech recognition [32]. CNN has two advantages for human activity recognition which are local dependency and scale invariance. Local dependency refers to the nearby observations in human activity recognition that are likely to be correlated, while scale invariance means the scale is invariant for different paces or frequencies. CNN can learn hierarchical data representations which leads to rendering promising results in human activity recognition [32]. In this study, a one dimensional (1D) CNN architecture is developed which can extract local 1D sub-sequences from the sequence data. The 1D CNN could be competitive with RNNs on some sequence-processing applications such as audio generation and machine translation with a cheaper computation cost compared to RNN. The model is designed by stacking two convolutional layers with 40% dropout rate and 0.001 learning rate followed by a max-pooling layer and followed by a fully-connected, i.e., a dense layer and a soft-max layer.
- Hybrid model: since input sub-sequences are processed by 1D CNN independently, unlike LSTM, the sub-sequences are not sensitive to the time step order. However, many

convolution layers and pooling layers could be stacked to recognize longer-term patterns. This leads to long chunks of the original inputs to be considered by the upper layers. Yet this is not a fairly strong way to induce order sensitivity, when order sensitivity is key to activity recognition, since RNNs for processing very long sequences are highly expensive, while 1D CNNs are cheap. Hence, designing a hybrid model by combining the speed and lightness of CNNs with the order-sensitivity of LSTM consists in using a 1D CNN as a pre-processing step before an LSTM. This strategy is particularly important because the 1D CNN turns the long input sequence data into much shorter sequences of high-level features. The sequence data of extracted features by 1D CNN then becomes the input to the LSTM part of the network. The hybrid model is the combination of two deep learning models, particularly Convolutional neural network and Long-Short Term Memory (CNN LSTM). CNN LSTM is able to systematically learn feature representation and model the temporal tendencies between their activations [24]. The model is designed by stacking a convolutional layer that was followed by a max-pooling layer. Then, a LSTM Layer with 40% dropout rate and 0.001 learning rate is followed by a fully-connected, i.e., a dense layer and a soft-max layer.

### D. Measurement

The following well-known metrics are used to evaluate the models.

- Precision represents the proportion of correct observations of a class to the entire observations classified as that class, with high precision indicating low false positives. Precision is equal to $\frac{TP}{TP+FP}$:
- Sensitivity (recall) shows the proportion of observations correctly classified as a given class to the actual total observations in that class calculated using the formula $\frac{TP}{TP+FN}$:
- F1-score (F1-sc) shows an insight into the balanced between precision ($\frac{TP}{TP+FP}$) and sensitivity (recall) ($\frac{TP}{TP+FN}$). This metric is also used in Activity recognition [17].

### E. Results and Discussion

In this section, results of different models are shown and discussed based on different feature extraction approaches. In a real-time activity recognition scenario, only past sensor data are considered for each evaluation time. Evaluation time is the time when the decision-making of each activity has been developed. Here, two different feature extraction approaches based on several machine algorithms have been compared with FTWs. First approaches described in [4] included an ad-hoc feature vector with Raw and Last sensor Activation (RLA) within a one-minute window as a suitable configuration for learning AR with C4.5 and SVM for the homes A and B. Second, we include the results from approach [8], where equally-sized temporal windows (ESTWs) of one minute sensor activation are described as a suitable feature extraction approach for AR binary sensors with LSTM and CNN models. Table I shows the results of the F1-score and training time for real-time recognition of different models based on extracting features with ESTWs, RLA and FTWs. The training time is the average of 10 runs of the training model. The F1-score results of the models based on FTWs are significantly higher than the results of the models based on ESTWs and RLA. Therefore, FTWs have been used for this paper as a contribution to AR research, with different delays in time in addition to real time to improve the recognition process. To predict what activity has been performed in a specific time *T*, different time delays of oncoming sensors after the time *T* in addition to the preceding sensor activations of the time *T* are included in the feature extraction based on FTWs. For example, now (Evaluation time), we evaluate what activity was developed 4 hours ago (Evaluated time). Evaluated time. It is the time which is evaluated by the classifier to recognize which activity has been developed in this point of time based on the preceding and oncoming sensor data. In the case of real time, evaluation time is equal to evaluated time. In the case of delays in time that consider oncoming sensor data, evaluation time is higher (delayed) from evaluated time.

In the scenario where the activity recognition is delayed, preceding sensor activations with different time delays, particularly 5 minutes, 20 minutes, 1 hour, and 4 hours, are tested to improve the recognition process. The results show that delaying time with LSTM in decision-making leads to building more accurate models. The results are significantly improved when considering oncoming sensor activations and increasing delay in the evaluated time. In house A, for example, the total results of F1-score of the model in real time is 89.05, while the results of the model are improved notably, up to 96.44, when considering oncoming sensor activations. Tables II and III show the results of the F1-score and training time of LSTM, CNN, and the hybrid CNN LSTM based on FTWs from home A and B respectively.

The results indicate that the F1-score of the models improves substantially by increasing time delays with a slight increase of training time. This means that delaying the decision-making of human activity recognition yields better and more accurate models. In addition to the F1-score, a precision summary of the models is also shown in Fig. 3 for homes A and B respectively. The figures show that the precision of the models is increased when considering oncoming sensor activations over real time.

Finally, the results of the models with 4-hour time delays are based on extracting features using three approaches. Firstly, Raw and Last Next Activation (RLNA) for learning AR with C4.5 and SVM. Secondly, Equally-sized Temporal Windows (ESTWs) [8] for learning AR with LSTM. Thirdly, Fuzzy Temporal Windows (FTWs) for learning AR with LSTM. Table IV shows the results of the F1-score and training time of the models based on extracting features with ESTWs, RLNA and FTWs. The results indicate that LSTM based on FTWs has obtained the highest F1-score with suitable and reasonable training time compared to LSTM based on ESTWs or C4.5 and CVM based on RLNA.

In summary, the proposed method of this paper has enhanced the models for recognizing all the activities performed in homes A and B while maintaining a low time cost. We highlight that

TABLE I

F1-SCORE AND TRAINING TIME FOR REAL-TIME RECOGNITION OF DIFFERENT MODELS BASED ON EXTRACTING FEATURES WITH EQUAL-SIZED TEMPORAL WINDOWS (ESTWS), RAW AND LAST ACTIVATION (RLA) AND FUZZY TEMPORAL WINDOWS (FTWS)

(a) Home A

| Activity | ESTWs | | RLA | | FTWs | | |
|---|---|---|---|---|---|---|---|
| | LSTM | CNN | C4.5 | SVM | LSTM | CNN | CNN LSTM |
| Breakfast | 72.35 | 69.75 | 82.71 | 81.75 | 95.08 | 92.69 | 94.85 |
| Grooming | 52.10 | 49.13 | 75.21 | 75.22 | 65.96 | 56.25 | 69.31 |
| Leaving | 88.66 | 87.98 | 98.56 | 98.56 | 88.97 | 94.31 | 87.18 |
| Lunch | 77.77 | 84.00 | 86.49 | 82.26 | 96.20 | 95.76 | 95.27 |
| Showering | 91.59 | 79.50 | 96.43 | 96.42 | 94.47 | 93.23 | 94.47 |
| Sleeping | 99.82 | 99.66 | 99.99 | 99.98 | 99.68 | 95.42 | 99.62 |
| Snack | 0.0 | 0.0 | 00.00 | 40.00 | 90.23 | 89.62 | 91.63 |
| Spare Time | 96.24 | 96.19 | 98.53 | 98.54 | 100.00 | 100.00 | 100.00 |
| Toileting | 32.93 | 29.26 | 35.19 | 38.09 | 70.37 | 64.62 | 63.39 |
| **Total** | **67.94** | **66.16** | **74.79** | **79.43** | **89.05** | **85.38** | **88.41** |
| **Train-time** | 5.1m | 3.9m | 4.1s | 1.92s | 3.65m | 2.1m | 3.1m |

(b) Home B

| Activity | ESTWs | | RLA | | FTWs | | |
|---|---|---|---|---|---|---|---|
| | LSTM | CNN | C4.5 | SVM | LSTM | CNN | CNN LSTM |
| Breakfast | 41.83 | 46.34 | 41.64 | 29.36 | 92.32 | 89.43 | 91.08 |
| Grooming | 61.65 | 59.92 | 90.79 | 90.31 | 74.33 | 69.89 | 72.61 |
| Leaving | 70.08 | 71.16 | 98.66 | 98.68 | 88.59 | 88.25 | 89.11 |
| Lunch | 0.0 | 01.38 | 13.20 | 13.28 | 94.66 | 92.51 | 94.34 |
| Showering | 74.94 | 52.79 | 95.54 | 96.00 | 88.6 | 86.86 | 88.20 |
| Sleeping | 96.78 | 96.68 | 98.41 | 98.34 | 99.38 | 99.45 | 99.39 |
| Snack | 0.05 | 09.00 | 13.54 | 17.26 | 83.57 | 79.54 | 84.72 |
| Spare Time | 88.06 | 87.48 | 90.97 | 91.15 | 96.98 | 96.00 | 96.82 |
| Toileting | 27.70 | 23.25 | 88.00 | 84.78 | 60.82 | 56.37 | 59.68 |
| Dinner | 25.30 | 22.50 | 11.11 | 00.00 | 79.15 | 74.64 | 75.28 |
| **Total** | **51.84** | **49.83** | **64.19** | **61.99** | **85.83** | **83.29** | **85.12** |
| **Train-time** | 6.4m | 5.1m | 5.8s | 3.2s | 4.8m1 | 2.9m | 4.2m |

TABLE II

F1-SCORE AND TRAINING TIME (MINUTES) OF LSTM, CNN, AND CNN LSTM WITH DIFFERENT DELAYS IN TIME BASED ON FTWS FROM ORDONEZ HOME A

| Activity | LSTM | | | | CNN | | | | CNN LSTM | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 Minutes delay | 20 Minutes delay | 1 Hour delay | 4 Hours delay | 5 Minutes delay | 20 Minutes delay | 1 Hour delay | 4 Hours delay | 5 Minutes delay | 20 Minutes delay | 1 Hour delay | 4 Hours delay |
| Breakfast | 95.18 | 95.87 | 96.87 | 96.95 | 92.91 | 93.24 | 93.29 | 94.69 | 95.22 | 95.87 | 96.78 | 96.69 |
| Grooming | 66.85 | 67.96 | 71.24 | 86.19 | 66.71 | 71.28 | 76.81 | 83.68 | 72.76 | 75.28 | 81.26 | 86.19 |
| Leaving | 91.17 | 91.85 | 94.15 | 99.85 | 94.31 | 96.71 | 96.83 | 99.49 | 97.41 | 98.21 | 98.78 | 99.85 |
| Lunch | 96.88 | 97.96 | 97.82 | 99.00 | 95.76 | 96.73 | 97.22 | 98.92 | 96.11 | 96.42 | 96.57 | 99.07 |
| Showering | 95.21 | 95.53 | 96.89 | 98.21 | 93.23 | 93.53 | 95.11 | 96.49 | 94.52 | 95.62 | 95.79 | 99.98 |
| Sleeping | 99.68 | 99.72 | 99.33 | 99.85 | 95.42 | 96.38 | 96.87 | 99.73 | 99.73 | 99.57 | 99.86 | 99.85 |
| Snack | 93.72 | 94.99 | 96.82 | 99.31 | 89.62 | 92.21 | 95.14 | 98.42 | 97.99 | 98.36 | 98.72 | 99.34 |
| Spare Time | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Toileting | 73.52 | 73.68 | 74.73 | 90.76 | 65.23 | 67.27 | 71.62 | 86.97 | 78.75 | 82.51 | 82.98 | 90.76 |
| **Total** | **90.24** | **90.84** | **91.97** | **96.44** | **88.13** | **89.70** | **91.43** | **95.37** | **92.49** | **93.53** | **94.52** | **96.97** |
| **Train-time** | 3.91 | 4.1 | 4.6 | 5.1 | 2.3 | 2.7 | 3.1 | 4.01 | 3.2 | 3.5 | 3.9 | 4.6 |

TABLE III

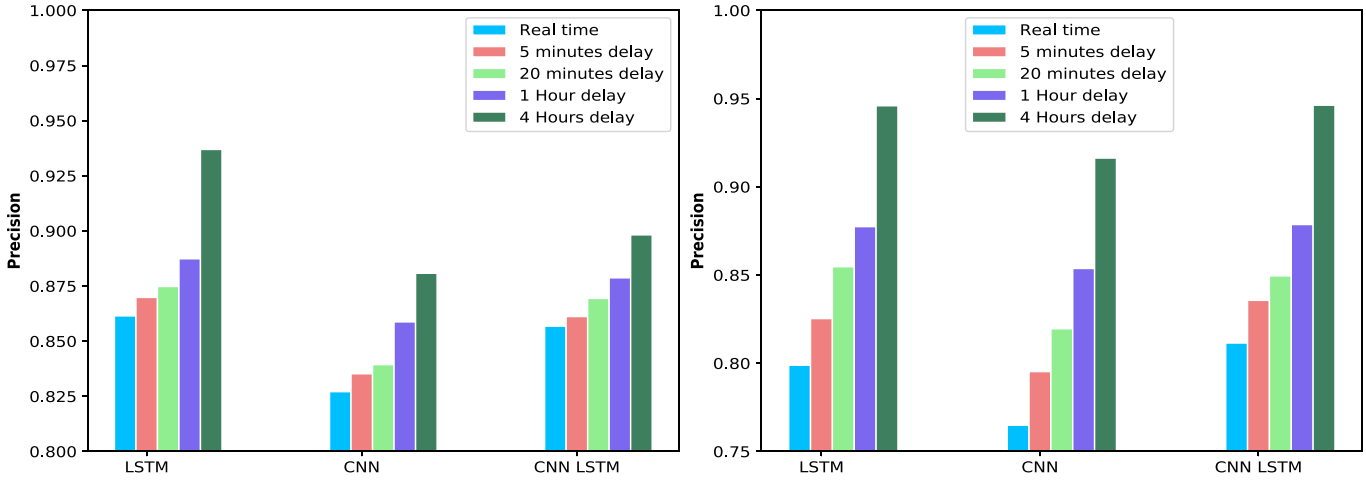
F1-SCORE AND TRAINING TIME (MINUTES) OF LSTM, CNN, AND CNN LSTM WITH DIFFERENT DELAYS IN TIME FROM ORDONEZ HOME B

| Activity | LSTM | | | | CNN | | | | CNN LSTM | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 Minutes delay | 20 Minutes delay | 1 Hour delay | 4 Hours delay | 5 Minutes delay | 20 Minutes delay | 1 Hour delay | 4 Hours delay | 5 Minutes delay | 20 Minutes delay | 1 Hour delay | 4 Hours delay |
| Breakfast | 93.31 | 94.11 | 94.81 | 99.69 | 91.27 | 91.97 | 95.65 | 99.25 | 94.33 | 94.87 | 96.43 | 99.54 |
| Grooming | 78.37 | 78.89 | 79.23 | 91.97 | 73.14 | 76.36 | 82.85 | 87.07 | 78.91 | 83.22 | 85.23 | 90.67 |
| Leaving | 93.62 | 98.72 | 98.72 | 99.50 | 94.28 | 98.51 | 98.63 | 99.39 | 98.84 | 98.86 | 98.89 | 99.55 |
| Lunch | 95.41 | 95.54 | 96.22 | 99.02 | 92.84 | 93.14 | 95.87 | 98.43 | 96.96 | 96.94 | 97.12 | 99.14 |
| Showering | 89.42 | 90.1 | 90.32 | 98.85 | 87.12 | 87.98 | 92.65 | 99.42 | 89.31 | 94.84 | 96.42 | 99.42 |
| Sleeping | 99.41 | 99.48 | 99.67 | 99.80 | 99.48 | 99.51 | 99.63 | 99.68 | 99.64 | 99.65 | 99.68 | 99.73 |
| Snack | 84.67 | 85.42 | 85.67 | 96.39 | 81.57 | 83.63 | 89.71 | 94.49 | 86.53 | 88.94 | 92.74 | 97.63 |
| Spare Time | 96.98 | 96.93 | 96.98 | 99.33 | 97.14 | 97.52 | 97.58 | 98.95 | 96.92 | 97.12 | 97.54 | 99.29 |
| Toileting | 62.37 | 66.37 | 67.12 | 86.23 | 62.73 | 65.72 | 69.43 | 76.17 | 67.63 | 72.78 | 76.68 | 85.19 |
| Dinner | 81.24 | 84.24 | 86.24 | 97.39 | 79.16 | 83.16 | 88.83 | 95.62 | 78.83 | 83.74 | 87.68 | 97.03 |
| **Total** | **87.48** | **88.98** | **89.49** | **96.82** | **85.87** | **87.75** | **91.05** | **94.84** | **88.79** | **91.09** | **92.84** | **96.72** |
| **Train-time** | 4.9 | 5.23 | 5.41 | 5.93 | 3.17 | 3.33 | 3.84 | 4.31 | 4.36 | 4.51 | 4.82 | 5.71 |

the proposed model with FTWs and Deep Learning achieves encouraging performance regarding ad-hoc classical approaches and sensor representations, as well as previous approaches based on Deep Learning with Equally-sized temporal windows.

This is particularly so in the activities that real-time models have difficulty recognizing accurately, such as *Leaving*, *Snack*, *Grooming*, and *Toileting* from home A. Regarding home B, the results of the same activities in addition to *Dinner* are significantly improved. This refers to the fact that taking oncoming sensor activations into account is crucial in order to enhance the learning process of the models.

Finally, we note the learning time in Deep Learning is in the magnitude of minutes compared with classical approaches which develop learning in seconds. However, the evaluation of Deep Learning under this approach has been developed with low learning time requirements (10 epochs which take less than 6 minutes). In addition, one limitation of this work is the difficulty in handling interleaved activities due to a single classifier




(a) Home A (b) Home B

Fig. 3. Precision of LSTM, CNN and CNN LSTM for real time and different delays in time.

TABLE IV
F1-SCORE AND TRAINING TIME FROM 4 HOURS DELAYED IN ADDITION TO REAL-TIME RECOGNITION FOR DIFFERENT MODELS WITH EQUAL-SIZED TEMPORAL WINDOWS (ESTWS), RAW AND LAST NEXT ACTIVATION (RLNA) AND FUZZY TEMPORAL WINDOWS(FTWS)

(a) Home A

| Activity | RLNA | | FTWs | ESTWs |
|---|---|---|---|---|
| | C4.5 | SVM | LSTM | LSTM |
| Breakfast | 82.87 | 77.66 | 96.95 | 78.32 |
| Grooming | 75.65 | 75.14 | 86.19 | 59.17 |
| Leaving | 99.94 | 99.94 | 99.85 | 91.21 |
| Lunch | 70.90 | 78.00 | 99.00 | 81.31 |
| Showering | 96.42 | 96.42 | 98.21 | 94.53 |
| Sleeping | 99.98 | 99.98 | 99.85 | 99.41 |
| Snack | 00.00 | 38.46 | 99.31 | 48.37 |
| Spare Time | 98.51 | 98.55 | 100.00 | 97.41 |
| Toileting | 62.68 | 61.57 | 90.76 | 51.32 |
| **Total** | **78.33** | **80.63** | **96.44** | **77.89** |
| **Train-time** | 2.78 s | 2.21 s | 5.1 m | 13.3 m |

(b) Home B

| Activity | RLNA | | FTWs | ESTWs |
|---|---|---|---|---|
| | C4.5 | SVM | LSTM | LSTM |
| Breakfast | 49.28 | 47.33 | 99.69 | 48.78 |
| Grooming | 95.10 | 94.06 | 91.97 | 69.26 |
| Leaving | 98.88 | 98.83 | 99.50 | 81.75 |
| Lunch | 29.42 | 31.87 | 99.02 | 41.61 |
| Showering | 95.54 | 95.54 | 98.85 | 96.13 |
| Sleeping | 99.39 | 99.34 | 99.80 | 99.48 |
| Snack | 46.52 | 45.14 | 96.39 | 39.69 |
| Spare Time | 91.58 | 91.74 | 99.33 | 92.91 |
| Toileting | 91.71 | 92.69 | 86.23 | 51.32 |
| Dinner | 21.51 | 00.00 | 97.39 | 42.83 |
| **Total** | **71.89** | **69.65** | **96.82** | **66.37** |
| **Train-time** | 6.77 s | 12.18 s | 5.93 m | 18.42 m |

being developed for learning with multi-class labels. In order to enable this methodology to support interleaved activities, a configuration using an ensemble architecture (where each activity is represented by a classifier) is necessary [26]; however, we note that the learning time and learning data size required is multiplied by the number of classes.

## V. CONCLUSION AND FUTURE WORK

Human activity recognition is a highly dynamic and challenging research field that plays a crucial role in diverse applications such as health care, elderly care, emergencies, security, smart environments, surveillance and context-aware-systems. In this study, we have proposed a new data-driven approach that aims to increase precision and sensitivity in human activity recognition applied in a smart home setting. The proposed method considers the partial oncoming sensor activations in addition to preceding sensor activations. With the use of oncoming sensor activation, we can take the benefits of enhancing the learning process that leads to improved recognition performance compared with the approaches using only the preceding sensor activations in the intelligent environment. Multiple and incremental fuzzy temporal windows were used to extract features from both preceding and partial oncoming sensor activations. Defining multiple and incremental fuzzy temporal windows from long-term to short-term has provided suitable semantics to determine a sequence of temporal features that boosts learning using LSTM sequence models and CNN.

Experiments show that precision and sensitivity increase by magnitudes when using preceding as well as partial oncoming sensor activations compared to precision and sensitivity when using only preceding sensor activations. The results of the experiment indicate that the more partial oncoming sensors are included, the better results are achieved. However, results are also improved when considering partial oncoming in addition to preceding sensor activations within a short amount of time, for example 5 minutes. The proposed approach of this paper has enhanced the models for recognizing all the activities performed in

houses A and B. Particularly the activities that real-time models have difficulty to recognize accurately such as *Leaving*, *Snack*, *Grooming*, and *Toileting* from house A. Regarding house B, the results of the same activities in addition to *Dinner* are significantly improved. This refers to the fact that taking partial oncoming sensor activations into account is crucial in order to enhance the learning process of the models.

Future work will explore the proposed approach in two main directions. First, we will work on boosting learning over different smart homes aiming to perform robust recognition of dangerous situations and detect behaviour deviations in order to enhance elderly-care alert systems. The long-term goal of our project based on the proposed approach will be key to transferring knowledge over different smart homes in terms of layout, resident and sensor configuration. Second, we will integrate heterogeneous sensors from wearable and location sources using fuzzy logic and scales in learning AR of patients in smart homes, which have been previously demonstrated as suitable fusion representation [35].

## REFERENCES


[1] R. A. Hamad, E. Järpe, and J. Lundström, "Stability analysis of the t-SNE algorithm for human activity pattern data," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2018, pp. 1839–1845.

[2] J. Bravo, R. Hervas, J. Fontecha, and I. Gonzalez, "m-health: Lessons learned by m-experiences," *Sensors*, vol. 18, no. 5, p. 1569, 2018.

[3] K. Bouchard *et al.*, "The cornerstones of smart home research for healthcare," *Smart Innov., Syst. Technol.*, vol. 93, pp. 185–200, 2018.

[4] F. J. Ordónez, P. de Toledo, and A. Sanchis, "Activity recognition using hybrid generative/discriminative models on home environments using binary sensors," *Sensors*, vol. 13, no. 5, pp. 5460–5477, 2013.

[5] A. Lotfi, C. Langensiepen, S. M. Mahmoud, and M. J. Akhlaghinia, "Smart homes for the elderly dementia sufferers: Identification and prediction of abnormal behaviour," *J. Ambient Intell. Humanized Comput.*, vol. 3, no. 3, pp. 205–218, 2012.

[6] R. Varatharajan, G. Manogaran, M. K. Priyan, and R. Sundarasekar, "Wearable sensor devices for early detection of alzheimer disease using dynamic time warping algorithm," *Cluster Comput.*, vol. 21, pp. 681–690, 2018.

[7] L. Á. San Martín, V. M. Peláez, R. González, A. Campos, and V. Lobato, "Environmental user-preference learning for smart homes: An autonomous approach," *J. Ambient Intell. Smart Environ.*, vol. 2, no. 3, pp. 327–342, 2010.

[8] D. Singh, E. Merdivan, S. Hanke, J. Kropf, M. Geist, and A. Holzinger, "Convolutional and recurrent neural networks for activity recognition in smart environment," in *Proc. Towards Integrative Mach. Learn. Knowl. Extraction*, Springer, 2017, pp. 194–205.

[9] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Joint Conf. Artif. Intell. ,* 2015, vol. 15, pp. 3995–4001.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] J. Suto, S. Oniga, C. Lung, and I. Orha, "Comparison of offline and real-time human activity recognition results using machine learning techniques," *Neural Comput. Appl.*, pp. 1–14, Mar. 2018, doi: 10.1007/s00521-018-3437-x.

[12] J. Medina, M. Fernandez-Olmo, M. D. Peláez, and M. Espinilla, "Real-time monitoring in home-based cardiac rehabilitation using wrist-worn heart rate devices," *Sensors*, vol. 17, 2017, Art. no. 2892.

[13] K. S. Gayathri, S. Elias, and B. Ravindran, "Hierarchical activity recognition for dementia care using Markov logic network," *Pers. Ubiquitous Comput.*, vol. 19, no. 2, pp. 271–285, 2015.

[14] D. H. Wilson and C. Atkeson, "Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors," in *Proc. Int. Conf. Pervasive Comput.*, Springer, 2005, pp. 62–79.

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

[16] H. Storf, T. Kleinberger, M. Becker, M. Schmitt, F. Bomarius, and S. Prueckner, "An event-driven approach to activity recognition in ambient assisted living," in *Proc. Eur. Conf. Ambient Intell.*, Springer, 2009, pp. 123–132.

[17] T. L. M. Van Kasteren, G. Englebienne, and B. J. A. Kröse, "An activity monitoring system for elderly care using generative and discriminative models," *Pers. Ubiquitous Comput.*, vol. 14, no. 6, pp. 489–498, 2010.

[18] J. Medina-Quero, S. Zhang, C. Nugent, and M. Espinilla, "Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition," *Expert Syst. Appl.*, vol. 114, pp. 441–453, 2018.

[19] P. Skocir, P. Krivic, M. Tomeljak, M. Kusek, and G. Jezic, "Activity detection in smart home environment," *Procedia Comput. Sci.*, vol. 96, pp. 672–681, 2016.

[20] T. L. M. Van Kasteren, G. Englebienne, and B. J. A. Kröse, "Activity recognition using semi-Markov models on real world smart home datasets," *J. Ambient Intell. Smart Environ.*, vol. 2, no. 3, pp. 311–325, 2010.

[21] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, 2013.

[22] D. Singh *et al.*, "Human activity recognition using recurrent neural networks," in *Proc. Int. Cross-Domain Conf. Mach. Learn. Knowl. Extraction*, Springer, 2017, pp. 267–274.

[23] A. P. Stakhov, "The generalized principle of the golden section and its applications in mathematics, science, and engineering," *Chaos, Solitons Fractals*, vol. 26, no. 2, pp. 263–289, 2005.

[24] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[25] J. Medina, L. Martinez, and M. Espinilla, "Subscribing to fuzzy temporal aggregation of heterogeneous sensor streams in real-time distributed environments," *Int. J. Commun. Syst.*, vol. 30, no. 5, p. e3238, 2017.

[26] J. Medina-Quero, C. Orr, S. Zang, C. Nugent, A. Salguero, and M. Espinilla, "Real-time recognition of interleaved activities based on ensemble classifier of long short-term memory with fuzzy temporal windows," *Multidisciplinary Digit. Publishing Inst. Proc.*, vol. 2, p. 1225, 2018.

[27] P. F. Byrd, "Expansion of analytic functions in polynomials associated with Fibonacci numbers," *Fibonacci Quart.*, vol. 1, no. 1, pp. 16–27, 1963.

[28] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive Mobile Comput.*, vol. 10, pp. 138–154, 2014.

[29] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM Sigkdd Explorations Newslett.*, vol. 6, no. 1, pp. 20–29, 2004.

[30] J. Collins, J. Sohl-Dickstein, and D. Sussillo, "Capacity and trainability in recurrent neural networks," *CoRR*, vol. abs/1611.09913, 2017.

[31] A. Murad and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, 2017, Art. no. 2556.

[32] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proc. 25th Int. Joint Conf. Art. Intell.*, New York, NY, USA, 2016, pp. 1533–1540.

[33] A. Devarakonda, M. Naumov, and M. Garland, "Adabatch: Adaptive batch sizes for training deep neural networks," *CoRR*, vol. abs/1712.02029, 2017.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[35] J. M. Quero, M. Á. L. Medina, A. S. Hidalgo, and M. Espinilla, "Predicting the urgency demand of copd patients from environmental sensors within smart cities with high-environmental sensitivity," *IEEE Access*, vol. 6, pp. 25081–25089, 2018.

[36] L. Wang, T. Gu, X. Tao, and J. Lu, "A hierarchical approach to real-time activity recognition in body sensor networks," *Pervasive Mobile Comput.*, vol. 8, no. 1, pp. 115–130, 2012.