On multi-resident activity recognition in ambient smart-homes

Son N. Tran¹ · Dung Nguyen² · Tung-Son Ngo³ · Xuan-Son Vu⁴ · Long Hoang⁵ · Qing Zhang⁶ · Mohan Karunanithi⁶

Published online: 12 November 2019
© Springer Nature B.V. 2019

Abstract
Increasing attention to the research on activity monitoring in smart homes has motivated the employment of ambient intelligence to reduce the deployment cost and solve the privacy issue. Several approaches have been proposed for multi-resident activity recognition, however, there still lacks a comprehensive benchmark for future research and practical selection of models. In this paper, we study different methods for multi-resident activity recognition and evaluate them on the same sets of data. In particular, we explore the effectiveness and efficiency of temporal learning algorithms using sequential data and non-temporal learning algorithms using temporally-manipulated features. In the experiments we compare and analyse the results of the studied methods using datasets from three smart homes.

Keywords Multiresident activity · Pervasive computing · Smart homes

1 Introduction
In intelligent environments such as smart homes, activity recognition plays an important role, especially when applying to health monitoring and assistance (Das and Cook 2004). Many efforts have been made to model the activities of residents in order to facilitate reasoning of their behaviour. The success of such models would result in reducing cost of traditional health care, a smarter and safer home for eldercare, and better assistance for patients. Classifying human activities has been studied intensively within the computer vision domain (Poppe 2010; Saini et al. 2018). This, however, may raise an issue on the privacy of residents due

¹ University of Tasmania, Newnham Campus, Launceston, TAS 7250, Australia
² Duy Tan University, Da Nang, Vietnam
³ Department of Computer Science, FPT University, Hanoi, Vietnam
⁴ Department of Computing Science, Umeå University, Umeå, Sweden
⁵ Posts and Telecommunications Institute of Technology, Ho Chi Minh City, Vietnam
⁶ The Australian E-Health Research Centre, CSIRO, Brisbane, QLD 4029, Australia

© Son N. Tran
sn.tran@utas.edu.au
to the use of unwelcome devices, i.e. cameras. Alternatively, many other approaches rely on wearables sensors (Plötz et al. 2011), which seems less intrusive but require users to wear an electronic device everywhere and every time. Recent attention is aiming at intelligent environments where residents can live their own way, without being disturbed by the presence of a device on their bodies. This is an important research topic that would shape the future of smart homes. With the advances in pervasive sensing technologies one can install a set of non-intrusive sensors in the environment with respect to residents’ privacy (Wilson and Atkeson 2005; Singla et al. 2010). However, in contrast to the development of ambient hardware, the reality of intelligent algorithms for such modern smart homes is still challenging.

Activity recognition in ambient environments has been studied for years, most of that focus on single resident, aiming to support independent living (van Kasteren et al. 2008). However, in practice, this is not always the case since modern smart environments should be able to support multiple occupants. As a result, there is a growing desire for a model that is capable of capturing the complex nature of both independent and joint activities. This is a challenging task because different from the case of single resident where the sensors’ states reflect directly the activity of a specific person, in multi-resident case that information, as known as data association is not commonly known. In recent work, temporal approaches have been widely employed to model activities in smart homes [see the survey Benmansour et al. (2015)]. However, there still lacks a comprehensive study on how different sequence models perform in this application domain. Furthermore, there also lacks a comparison between temporal approaches and non-temporal approaches which use temporal features as input.

In this paper, first we investigate the use of hidden Markov model (HMM), conditional random field (CRF), and recurrent neural network (RNN) for multi-resident activity recognition in ambient smart homes. The study also focuses on two different methods of encoding activities of multiple residents: combined labels and separate labels. Second, we investigate two different strategies for representing temporal features as inputs of non-temporal models such as K-Nearest Neighbour (KNN), Random Forest (RF) and Feed-forward Neural Network (FNN). We expect that the work would serve as a benchmark and guideline for those who seek for a solution in this domain. The techniques used in this paper are not novel but our study and findings are novel which would serve as baselines for other research. They are also valuable for practical users to decide what approaches should be used when certain circumstances are met.

In general, for temporal learning approaches, we model the activity sequences of multiple residents as an input of a function with the input is the states of a smart home. Such states, in this case, are defined as the values of the sensors in the smart home. The output can be divided into different types of representation. First, we consider the output as a single variable which is the combined activities of all residents. With this, we can apply sequence models such as HMMs, CRFs, and RNNs for classification task straightforwardly. The second type of output is encoded by separating it into multiple variables, with each variable represents the activities of a resident while sharing the same input. In order to deal with multiple residents, we need more complex models to capture the interactive and collaborative behaviours. For HMMs, we employ the factorial variant (Ghahramani and Jordan 1997) add more cross dependencies. In the case CRFs, cross dependencies would be computationally expensive especially for the data of very long sequences, so that factorial CRFs (Sutton et al. 2007) are used. With RNNs, we share the hidden layer for all different label variables.

For temporal features processing, we use two strategies: MAX and CONCAT. In MAX, the input for non-temporal model is represented by a vector of maximum values of the sensors within a period of time. In CONCAT, the input is a concatenation of vector representations of sensors’ states at all time points of that period.
The contributions of this work are:

– A comprehensive empirical survey of different approaches for multi-resident activity recognition in ambient smart homes.
– A study and comparison of the effect of output processing: combined labels versus separate labels.
– A study and comparison of the effect of temporal approaches for activity data: sequence modelling versus temporal features processing.

In the experiments, we conduct evaluations of the models on three smart homes from two benchmark datasets, namely CASAS, ARAS House A, ARAS House B. The results show that recurrent neural networks with gated recurrent units is better than other models and also considerably efficient. We found that using combined labels is more effective than separate labels. The temporal features processing approaches also achieve good performance, i.e. they outperform the temporal learning approaches in ARAS House B and also are better than most of the temporal learning approaches (except CRF) in ARAS House A.

The paper is organised as follows. In the next section, we review the related literature of our work. Section 3 describes the multi-activity modelling framework and the models studied in this paper. In Sect. 5 we perform experiments and analyse the results. Section 6 discusses the findings and finally, in Sect. 7 we conclude the work.

2 Related work

Research on multi-resident activity recognition has been emerging recently due to the increasing demand for health monitoring in ambient intelligent environments. The task can be done by employing sequence models to perform prediction on the activity events over time. As pointed out in Benmansour et al. (2015), we focus on three main approaches that are achieving state-of-the-art results in temporal modeling to be used in this paper including (1) directed dynamic Bayesian networks (DDB), (2) undirected dynamic Bayesian networks (UDB), and (3) recurrent neural networks (RNN). Regarding DDB, we select Hidden Markov Model (HMM) (Rabiner 1990), factorial HMM (Brand et al. 1997). In multi-resident smart homes, HMMs have been studied intensively, as being showed in previous works (Alemdar et al. 2013; Chen and Tong 2014; Singla et al. 2010; Cook 2012). For UDB approach, Conditional Random Field (CRF) and factorial CRF are the most popular for smart home datasets (Crandall and Cook 2008; Hsu et al. 2010; Benmansour et al. 2015; Wang et al. 2011). Also multiple different settings of RNN models including different activation functions (e.g., tanh), different recurrent units (e.g., gated RNN) and so on, are tested in this work.

In previous works non-temporal approaches have also been employed for modelling activities of multi-residents in smart homes Alemdar et al. (2013). In this work, we also adopt such approaches and further study the effect of different temporal data processing strategies.

From learning perspective, the problem of multi-resident activity recognition can be seen as multi-tasks learning on sequence data. However, most of the work we found in the literature focus on modelling different tasks from different data sources by taking the advantage of recurrent neural networks in learning more generalised representation from larger amount of data combined. Different from that, in this work we do not have such augmentation since there exists only one dataset for activities of multiple residents.
3 Temporal modelling approaches

Let us denote $a_{m,t}$ and $o_t$ as the activity of resident $m$ and the sensors’ state at time $t$ respectively. For ease of presentation we denote $a_t = \{a_1^t, a_2^t, \ldots, a_{M}^t\}$ as the activities of all $M$ residents at time $t$. We use $t_1:t_2$ to denote a sequence of events/states from time $t_1$–$t_2$. For example, $a_{t_1:t_2} = \{a_1^{t_1}, \ldots, a_2^{t_2}\}$ is the sequence of activities performed by all residents from time $t_1$–$t_2$. In this paper, we evaluate two ways of modelling the activities of multiple residents. First, we combine the activities such that the activities of all resident at a time step is represented by a single variable. For that we need to predict $a_{1:T}$ given the states of sensors $o_{1:T}$. A combining label is a group of activities performed by different resident at a time. Suppose that each resident $m$ has $K_m$ labels, this would results in $\prod_m K_m$ combined labels to cover all possible activities of all residents without losing any information. Second, we model each resident’s activity as a separate variable.

3.1 HMM-based approaches

A HMM (Rabiner 1990) consists of a single hidden and an observation variable which assumes a Markov process. In the case of combined labels we can use a single HMM to model the activities as a joint distribution as:

$$p(a_{1:T}, o_{1:T}) = p(o_1^1 | a^1) p(a^1) \prod_{t=2}^{T} p(o_t | a_t) p(a_t | a_{t-1})$$

(1)

Inference of activities given a sequence of sensors’ states can be done efficiently using dynamic programming, i.e. Viterbi algorithm (Rabiner 1990). For the separate labels, different HMMs have been used such as parallel HMMs, coupled HMMs (Wang et al. 2011; Son et al. 2017). In this paper we use factorial HMM with cross dependency shown in Fig. 1d, as this variant achieves better performance than the other HMMs (Son et al. 2017). Factorial HMM (Ghahramani and Jordan 1997), is a HMM with multiple hidden variables. In order to represent the relations between activities among residents, we add cross connections from all hidden variables at time $t - 1$ to each hidden variable at time $t$. This results in a factorial HMM model with cross dependency as we introduce here in the paper. The joint distribution of this HMM is:

$$p(a_{1:T}, o_{1:T}) = p(o_1^1 | a^1) \prod_m p(a_{m,1}^1) \prod_{t=2}^{T} (p(o_t^t | a_t^t) \prod_m p(a_{m,t}^t | a_{m,t-1}^t))$$

(2)

Similar to a normal HMM, inference of activities can easily be done by dynamic programming. Here, only the transition and the prior probabilities are changed in comparison to the HMM above. Therefore, we can apply the Viterbi algorithm by replacing $p(a_t^t | a_{t-1}^t)$ with $\prod_m p(a_{m,t}^t | a_{m,t-1}^t)$ and $p(a_1^1)$ with $\prod_m p(a_{m,1}^1)$.

3.2 CRF-based approaches

Conditional random field is a probabilistic graphical model which can be used for modelling sequences, similar as HMMs. The difference here is that a CRF is a discriminative model representing a conditional distribution:
Fig. 1 Sequence models for multi-resident activity recognition. \textit{hmm} hidden Markov model, \textit{crf} conditional random field, \textit{rnn} recurrent neural networks, \textit{fhmm} factorial hidden markov model (with cross dependencies), \textit{fcrf} factorial conditional random field, \textit{mrnn} multi-labels recurrent neural networks. The top row depicts the models for combined labels and the bottom row depicts the models for separate labels.

\[
p(a^{1:T} | o^{1:T}) = \frac{1}{Z(o^{1:T})} \prod_t \Psi(a^t, a^{t-1}, o^t) \tag{3}
\]

where \(\Psi(a^t, a^{t-1}, o^t) = \exp(\sum_k \theta_k f_k(o^t, a^t, a^{t-1}))\) with \(f_k\) are the feature functions, \(\theta_k\) are parameters of the features, and \(Z(o^{1:T})\) is the partition function:

\[
Z(o^{1:T}) = \sum_{a'} \prod_t \Psi(o^t, a^t, a'^{t-1}) \tag{4}
\]

This is a CRF for combined labels, for the case of separate labels, we split the hidden unit to have a variant as known as factorial CRF (Sutton et al. 2007). In Fig. 1e we illustrate the graphical presentation of this model.

3.3 RNN-based approaches

A recurrent neural network is constructed by rolling a feed-forward neural network over time where the hidden layer is connected to itself by a recurrent weight. As shown in Fig. 1c, we can use the output layer to represent the combined activities of multiple residents. For example, at time \(t\) the probability of a joint activity \(a^t\) is:

\[
p(a^t | o^{1:T}) = \text{softmax}(h^t U + b) \tag{5}
\]

where \(U\) is the weight matrix connecting the hidden layer and the output layer; \(b\) is the biases of the output units. We will show how hidden state \(h^t\) is computed later in this section. For the other case where activities of residents are modelled separately we can split the output layer into multiple layers, as shown in Fig. 1f. Let us suppose that there are \(M\) residents, the probability of a resident \(m\) performs an activity \(a^{m,t}\) at time \(t\) is: \(p(a^{m,t}) = \)
softmax($h'_t U_m + b_m$). Here each output layer is connected with a shared hidden layer by a weight matrix $U_m$. The hidden state in both cases (combined labels and separate labels) is computed as $h'_t = \tanh(o'_t W + h'_{t-1} V + c)$. This is the simplest form of hidden unit which is said not very useful to capture long-term information and suffer the problem of vanishing/exploding gradient (Hochreiter and Schmidhuber 1997). This is also shown that such problems can be ameliorated by using complex gates in the hidden units. In this work, we will empirically investigate three types of hidden units for RNNs, including the original “tanh” and two others with more complex gates as known as long-short term memory (LSTM) (Hochreiter and Schmidhuber 1997) and gated recurrent units (GRU) (Cho et al. 2014).

4 Temporal data processing

In this section we discuss two data processing approaches to incorporate temporal information in the inputs for learning an activity recognition model using standard (non-sequential) classification algorithms. In order to predict the activity at time $t$, sensors’ states within a temporal window of length $\tau$ prior to $t$ are used as input. Here, $\tau$ is called “context-length”. Formally, let us consider the sensors’ states $f(o'_{t-\tau+1:t})$ as an input for activities of multiple residents $a^r$, where $f$ can be either MAX or CONCAT function. The MAX function takes into account the sensors’ states and returns a vector where each element representing the maximum value of a sensor within an interval $t - \tau + 1 : t$. The CONCAT function, on the other hand, presents the input as a concatenation of vectors $o'_{t-\tau+1}, ..., o'_{t-1}$. An illustration of MAX and CONCAT functions with context length $\tau = 5$ are shown in Fig. 2.

In this work we use three classifiers for evaluation, namely K-nearest Neighbour (KNN), Random Forest (RF), and Feed-forward Neural Networks (FNN). For simplicity, with these models we only employ combined labels.

5 Experiments

5.1 Datasets

In the experiments we evaluate the effectiveness of all the models above on three smart homes data in two benchmark datasets.

The CASAS data was collected in the WSU smart department Testbed with different pairs of participants playing the roles of two residents where each resident performing 15 unique activities (Cook et al. 2010). The data is collected in 26 days in a smart home equipped with 37 ambient sensors. Data in CASAS is presented in “Date Time Sensor_ID Value Resident_ID Activity” format. For example, “2008-11-10 14:28:17.986759 M22 ON 2 2” shows that resident 2 is hanging up clothes at 14:28:17.986759 on 2008-11-10 when motion...
sensor M22 is triggered. Similarly, “2008-11-10 14:38:47.974299 M13 OFF 2 8 1 9” means at 14:38:47.974299 on 2008-11-10 when motion sensor M13 is off resident 1 is setting dining room table for dinner while resident 2 is setting out ingredients for dinner in the kitchen. At each time step we create a vector which consists of the values of all sensors, where ON, PRESENT, OPEN are set as 1 s and OFF, ABSENT, CLOSE are set as 0 s. Each sensor is represented as an element in the vector whose value will be updated according to the value of that sensor. This type of representation was used in Singla et al. (2010). The IDs and names of activities are shown in Table 1. This data can be downloaded at http://casas.wsu.edu/datasets/adlmr.zip.

The ARAS data (Alemdar et al. 2013) is collected in two different houses, denoted as House A and House B, in 30 days. In these environments, there are 20 sensors for two residents in each house where each resident is asked to perform 27 different activities. The format of ARAS data is presented as: “Sensor_1 Sensor_2 …Sensor_20 R1_Activity R2_Activity”. This can be seen as a vector of sensors’ values and activities of residents. For example, “0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 6” indicates that resident 1 is watching TV and resident 2 is using the Internet when the force sensor in the couch (sensor 4) and distance sensor in the chair (sensor 6) are triggered. The IDs and names of activities are shown in Table 2. This data is available at http://www.cmpe.boun.edu.tr/aras/, or it can be reached at https://tinyurl.com/ya6odxcx in the case the first link is down.

In order to understand the behaviour of residents, we visualise their activities in three environments, as being shown in Fig. 3. The x and y axes indicate the activities of resident 1 and resident 2 respectively. The hotter color of a cell indicates that the activities of two residents occur more often (see the heat map bar in each figure). In CASAS data, there exist many cases that only individual’s activities are labelled, i.e. only activity of one resident is known at a time. Here we use activity ID = 0 to present unknown activity of the other resident. As we can see in Fig. 3a, there are only 5 cases where two residents perform different activities at the same time. Notably, there is only one case where they are doing the same activity which is a cooperative task where a resident retrieves dishes from a kitchen cabinet and resident 2 requests help from resident 1 to identify cabinet in which the dishes are located.

In ARAS house A and ARAS house B the distributions of activities are much more diverse than in the CASAS with many different activities have been performed. Also, in these environments more cooperative tasks can be seen too, such as “using internet”, “sleeping”, “going out”, “having dinner”, “having breakfast”, etc.
Table 1  Activities of each resident in CASAS dataset

| ID | Activity                              | Description                                                                                           |
|----|---------------------------------------|-------------------------------------------------------------------------------------------------------|
| 0  | Other/unknown                         |                                                                                                       |
| 1  | Filling medication dispenser          | Fill medication dispenser in the kitchen using items obtained from the cabinet. Return items to the cabinet when done |
| 2  | Hanging up clothes                    | Hang up clothes in the hallway closet. The clothes are laid out on the couch in the living room       |
| 3  | Moving furniture                      | Move the couch and coffee table to the other side of the living room. Request help from another person |
| 4  | Reading magazine 1                    | Sit on the couch and read a magazine                                                                   |
| 5  | Watering plants                       | Water plants located around the apartment. Use the watering can located in the hallway closet. Return the watering can to the closet when finished |
| 6  | Sweeping floor                        | Sweep the kitchen floor using the broom and dust pan located in the kitchen closet. Return the tools to the closet when finished |
| 7  | Playing checker                        | Play a game of checkers for a maximum of five minutes                                                  |
| 8  | Preparing dinner                      | Set out ingredients for dinner in the kitchen                                                          |
| 9  | Setting table                         | Set dining room table for dinner                                                                      |
| 10 | Reading magazine 2                    | Read a magazine on the living room couch                                                               |
| 11 | Paying bill                           | Simulate paying electric bill                                                                         |
| 12 | Gathering food                        | Gather food for a picnic from the kitchen cupboard and pack them in a picnic basket                    |
| 13 | Retrieving dishes                     | Retrieve dishes from a kitchen cabinet                                                                |
| 14 | Packing supplies                      | Pack supplies in the picnic basket                                                                    |
| 15 | Packing food                          | Pack food in the picnic basket and bring the basket to the front door of the apartment                 |

Table 2  Activities of each resident in ARAS dataset

| ID | Activity       | ID | Activity       | ID | Activity       |
|----|----------------|----|----------------|----|----------------|
| 0  | Other          | 1  | Going out      | 2  | Preparing breakfast |
| 3  | Having breakfast| 4  | Preparing lunch| 5  | Having lunch    |
| 6  | Preparing dinner| 7  | Having dinner  | 8  | Washing dishes |
| 9  | Having snack   | 10 | Sleeping       | 11 | Watching TV    |
| 12 | Studying       | 13 | Having Shower  | 14 | Toileting      |
| 15 | Napping        | 16 | Using internet | 17 | Reading book   |
| 18 | Laundry        | 19 | Shaving        | 20 | Brushing teeth |
| 21 | Talking on the phone | 22 | Listening to music | 23 | Cleaning    |
| 24 | Having conversation | 25 | Having guest   | 26 | Changing clothes |

5.2 Evaluation

We denote $\hat{a}^{1:T}$ with $\hat{a}^i = \{\hat{a}^{1,i}, \hat{a}^{2,i}, \ldots, \hat{a}^{M,i}\}$ as the predicted activities of all residents in the house for each instance in the test set $D_{test}$. We also denote a ground truth is $a^{1:T}$. The performance of a model is measured by the accuracy of each resident’s activities and the accuracy of all residents’ activities. The former is computed as:
\[
\text{accuracy}_m = \frac{1}{|D_{\text{test}}|} \sum_{a^{m,t} \in D_{\text{test}}} \frac{1}{T} \sum_t (a^{m,t} = \hat{a}^{m,t})
\]

(6)

where \(a^{m,t}\) and \(\hat{a}^{m,t}\) are ground truth and predicted activity of resident \(m\) at time \(t\) in each instance of the test set \(D_{\text{test}}\). Similarly, the accuracy for activities of all residents is:

\[
\text{accuracy}_{\text{all}} = \frac{1}{|D_{\text{test}}|} \sum_{a^t \in D_{\text{test}}} \frac{1}{T} \sum_t (a^t = \hat{a}^t).
\]

(7)

5.3 Results

We partition the CASAS data into 24 days for training, 1 day for validation and 1 day for testing. The ARASA and ARASB are the data from ARAS House A and ARAS House B each consist of 7 days for training, 2 days for validation and 2 days for testing.

The models are selected as follows. For HMM and fHMM we selected the best models based on the Laplacian smoothing factor. The smoothing factor is chosen from \(10^{-6}\) to \(10^{-2}\) in log-space. For the CRFs, we do not use any hyper-parameters and set the maximum iteration is 1000. We use MALLET to implement fCRFs and set the penalty hyper-parameters to zeros. For RNN-based models and FNN, we perform model selection by using grid-like search on number of hidden units and learning rate. For KNN and RF we perform grid-search on the number of neighbours and on the number of trees. We denote RNN and mRNN are recurrent neural networks for combined labels and separate labels respectively. We also use subscripts \(\text{tanh}, \text{gru}, \text{lstm}\) to denote the type of hidden units in RNNs. Finally, we use \(\text{max}\) and \(\text{cc}\) to denote the temporal processing methods, namely \(\text{MAX}\) and \(\text{CONCAT}\) respectively.

Table 3 shows the results of all models on three datasets. In CASAS data, RNN_{gru} outperforms other models with 83.66% accuracy and 0.8227 F1 score while KNN_{cc} achieves the lowest accuracy (36.84%) and F1 score (0.3149). In ARASA, HMM-based models achieve lower performance. It seems that the simplicity of HMM cannot capture the complexity of this data as we learn that the number of observed sensors’ values in ARASA is 10 times more than CASAS and 3 times more than ARASB. In this dataset, CRF has the best results. In ARASB the temporal learning approaches have similar results ranging from 74.44% accuracy and 0.7246 F1 for fCRF to 76.86% accuracy and 0.7588 F1 for mRNN_{tanh}. Interestingly, the temporal data processing approaches perform better in this case with RF_{max} achieve the highest accuracy and F1 score.

The confusion matrices for RNN_{gru} on three datasets are presented in Fig. 4. Due to the large number of classes in ARAS House A and ARAS House B we only show the most occurred ones for a better view. Here, we create indices of the activities starting from 0 for the most popular activities and in a descending order. We also normalise the matrices to deal with the display problem from unbalanced data where labels with fewer samples will be overshadowed by those having much more samples. We can see from Fig. 4a that in CASAS most activities are predicted with good accuracy except the combined activity 14 (resident A and resident B both are retrieving dishes from a kitchen cabinet) where the majority of inputs for this class are predicted as activity 19 (resident A is setting dining room table for a dinner while resident B is preparing dinner). In ARAS House-A and ARAS House-B the confusion matrices in Fig. 4b and c show that most of the incorrect predictions are below the diagonal lines, close to the left. This means that inputs from less common activities are more likely to be classified as activities that appear more often.
Table 3 Prediction accuracy for all models on three datasets

|          | R1   | R2   | All   |
|----------|------|------|-------|
|          | Acc  | F1   | Acc   | F1   | Acc   | F1   |
| CASAS    |      |      |       |
| RNN_tanh | 66.62| 0.6435|64.88  | 0.6407| 58.08 | 0.5256|
| mRNN_tanh| 69.14| 0.7011|60.61  | 0.5894| 47.94 | 0.4689|
| RNN_gru  | 92.26| 0.9197|87.68  | 0.8529| 83.66 | 0.8227|
| mRNN_gru | 90.89| 0.9010|83.97  | 0.8288| 77.91 | 0.7603|
| RNN_lstm | 89.40| 0.8879|87.45  | 0.8701| 82.14 | 0.8119|
| mRNN_lstm| 69.05| 0.6903|84.30  | 0.8421| 77.49 | 0.7599|
| HMM      | 65.24| 0.6434|65.82  | 0.6615| 56.58 | 0.5382|
| fHMM     | 73.55| 0.7192|67.44  | 0.6692| 55.43 | 0.5320|
| CRF      | 76.40| 0.7615|66.07  | 0.6624| 64.32 | 0.6219|
| fCRF     | 58.21| 0.5904|56.76  | 0.5442| 45.84 | 0.4404|
| KNNmax   | 53.81| 0.5400|47.46  | 0.4592| 37.07 | 0.3324|
| KNNcc    | 53.23| 0.5298|46.91  | 0.4510| 36.84 | 0.3149|
| RFmax    | 51.39| 0.5085|49.88  | 0.4882| 40.88 | 0.3927|
| RFcc     | 50.78| 0.5002|48.99  | 0.4755| 40.72 | 0.3897|
| FNNmax   | 49.54| 0.4913|50.12  | 0.4935| 43.30 | 0.4216|
| FNNcc    | 48.76| 0.4882|50.07  | 0.4866| 43.18 | 0.4201|
| ARAS House-A | | | | | | |
| RNN_tanh | 67.02| 0.6619|73.09  | 0.7184| 53.07 | 0.5221|
| mRNN_tanh| 68.12| 0.6229|74.74  | 0.7249| 53.26 | 0.5209|
| RNN_gru  | 69.79| 0.6901|74.20  | 0.7284| 56.23 | 0.5513|
| mRNN_gru | 70.03| 0.6912|75.37  | 0.7399| 56.65 | 0.5606|
| RNN_lstm | 69.94| 0.6721|73.59  | 0.7200| 56.44 | 0.5547|
| mRNN_lstm| 69.97| 0.6828|75.39  | 0.7453| 56.25 | 0.5521|
| HMM      | 43.95| 0.4298|54.93  | 0.5316| 19.13 | 0.1905|
| fHMM     | 44.19| 0.4274|54.58  | 0.5317| 19.13 | 0.1835|
| CRF      | 70.73| 0.6836|78.17  | 0.7642| 61.72 | 0.5987|
| fCRF     | 69.50| 0.6783|69.50  | 0.6811| 55.95 | 0.5462|
| KNNmax   | 49.52| 0.4704|58.47  | 0.5523| 37.38 | 0.3372|
| KNNcc    | 49.74| 0.4669|72.29  | 0.6779| 36.94 | 0.3235|
| RFmax    | 69.91| 0.6556|79.95  | 0.7427| 60.18 | 0.5707|
| RFcc     | 70.05| 0.6531|79.60  | 0.7374| 60.39 | 0.5682|
| FNNmax   | 68.58| 0.6520|79.45  | 0.7453| 60.57 | 0.5784|
| FNNcc    | 69.87| 0.6529|79.62  | 0.7373| 60.17 | 0.5668|
| ARAS House-B | | | | | | |
| RNN_tanh | 81.09| 0.8111|78.16  | 0.7663| 76.15 | 0.7503|
| mRNN_tanh| 91.93| 0.9077|78.99  | 0.7712| 76.86 | 0.7588|
| RNN_gru  | 81.69| 0.7999|78.95  | 0.7912| 76.83 | 0.7526|
| mRNN_gru | 82.04| 0.8152|78.90  | 0.7701| 76.83 | 0.7536|
| RNN_lstm | 82.15| 0.8157|77.84  | 0.7712| 76.72 | 0.7511|
| mRNN_lstm| 81.66| 0.8002|78.92  | 0.7727| 76.60 | 0.7511|
| HMM      | 79.29| 0.7752|76.98  | 0.7528| 75.07 | 0.7399|
Table 3 continued

|        | R1     |        | R2     |        | All    |        |
|--------|--------|--------|--------|--------|--------|--------|
|        | Acc    | F1     | Acc    | F1     | Acc    | F1     |
| fHMM   | 79.17  | 0.7746 | 77.10  | 0.7552 | 75.07  | 0.7258 |
| CRF    | 88.36  | 0.8725 | 89.27  | 0.8792 | 76.23  | 0.7488 |
| fCRF   | 76.01  | 0.7436 | 76.01  | 0.7473 | 74.44  | 0.7264 |
| KNNmax | 51.34  | 0.4901 | 60.98  | 0.5722 | 40.15  | 0.4036 |
| KNNcc  | 42.05  | 0.4355 | 40.35  | 0.4175 | 39.48  | 0.3972 |
| RFmax  | 95.97  | 0.9566 | 90.68  | 0.9214 | 89.41  | 0.8796 |
| RFcc   | 94.77  | 0.9407 | 90.71  | 0.9198 | 88.75  | 0.8652 |
| FNNmax | 96.05  | 0.9585 | 90.89  | 0.9219 | 89.35  | 0.8784 |
| FNNcc  | 94.62  | 0.9408 | 90.80  | 0.9137 | 88.54  | 0.8580 |

R1, R2, All are accuracy of predicted activities of resident 1, resident 2 and the joint activities. The bold values represent the best results with 95% confidence interval.

Fig. 4 Confusion matrices for CASAS, ARAS House A and ARAS House B
5.4 Combined labels versus separate labels

We now analyse the effectiveness of combined labels versus separate labels where temporal approaches are used. We compute the average accuracy of all temporal models that use combined labels approach on each data and compare it with the separate labels approach. The results are demonstrated in Fig. 5. It is consistent that models with combined labels have better prediction accuracy than models with separate labels. An interesting finding from the results here is that the combined label approaches not only have higher accuracy for joint activities but also outperform the separate label approach in predicting individual activities.

5.5 Effect of context-length

In this section, we will show the effect of context-length with three methods used in this work. In order to incorporate temporal information to such models, we apply a sliding window to the input within a number of time steps called “context length”. A context length $t$ of the window at time $t$ defines an (input, output) pair $(o^{t-t+1}, a^t)$.

In the experiments, we test the models with different context lengths: $\{5, 10, 50, 100, 200\}$ as shown in Figs. 6 and 7. We can see that, in general, the context length of 10 achieves the best results in most of the cases. It is worth noticing that using CONCAT function would result in high input dimension where long context-length did not fit the memory of our computers. Meanwhile, MAX function keeps the input dimension unchanged regardless of the context-length and even achieves slightly better performance than CONCAT.
5.6 Efficiency

Finally, we analyse the efficiency of the models in this application. Theoretically, the combined labels may need more parameters than the separate labels as the former require variables to represent $K_1 \times K_2 \ldots K_M$ activities while the latter use $M$ variables each represent $K_m$ activity values. However, this might be different in practice where smaller combined label model may have better results than bigger separate-label models. It would depend on how models are selected by validation sets. In Table 4 we report the running time of the models which give the best results in Sect. 5.3. Note that this comparison is not about the theoretical analysis of the models in general, instead it shows the empirical results about the efficiency of their implementation we used in this work. Overall, among all models, RNN_{gru} would be the best choice since it has the best performance while being considerably efficient. It is not as fast as RNN_{tanh} but it is quicker than the other RNN based models in most cases. Especially, it outperforms HMM based models and fCRF models with large margin. CRF is also a good choice since its best models are even faster than RNN_{gru} in CASAS and ARAS House B. However, in these two datasets it has lower accuracy than RNN_{gru} while in ARASA it is much slower than RNN_{gru}. Non-temporal models including KNN, RF and FNN are very efficient, although their performance in CASAS and ARAS House A datasets are not as good as RNN-based models, they outperform the other models in ARAS House B.

5.7 Compare with other methods

In this section, we compare with other methods from previous work, as shown in Table 5. For CASAS dataset, Hsu et al. (2010) and Chiang et al. (2010) achieve 64.16% and 61.78% respectively, using leave-one-out cross-validation. Another work by Singla et al. achieves 60.60% accuracy using three-fold cross-validation (Singla et al. 2010). For comparison, we select HMM with combined labels among our studied models and evaluate it with leave-one-out cross-validation. The results showed that the HMM outperforms the approaches in Hsu et al. (2010) and Chiang et al. (2010) with 69.127% accuracy.

For the ARAS, although the dataset consists of 27 unique activities many works group the activities into a smaller set of classes for evaluation. For example, in Alemdar et al. (2013) the activities are grouped into 6 classes, and in Tunca et al. (2014) they are grouped into 7 classes. However, we argue that this is not complete and not systematically sound. In Prossegger and Bouchachia (2014) the authors another approach called iterative decision tree (IDT) to model all 27 activities. Therefore, we compare one of the methods in this study
Table 4 (Average) Computational time for best models in Sect. 5.3 to train and predict activities in three datasets

| Model       | Data       | CASAS (s) | ARAS House-A (s) | ARAS House-B (s) | Avg. (s) |
|-------------|------------|-----------|-------------------|------------------|--------|
| RNN         |            | 96.23     | 3064.50           | 3547.74          | 2236.16 |
| mRNN        |            | 406.31    | 3.39 h            | 3.44 h           | 2031.44 |
| GRU         |            | 203.58    | 1.31 h            | 1.43 h           | 3355.85 |
| mGRU        |            | 545.39    | 1.47 h            | 1.38 h           | 1 h     |
| LSTM        |            | 183.64    | 7.87 h            | 8.61 h           | 5.51 h  |
| mLSTM       |            | 638.35    | 6.54 h            | 1.71 h           | 2.81 h  |
| HMM         |            | 0.17      | 95.23 s           | 50.74 s          | 48.71 s |
| fHMM        |            | 0.23      | 97.52 s           | 50.77 s          | 49.51 s |
| CRF         |            | 64.32 s   | ~ 8 h             | 289.17 s         | 2.70 h  |
| fCRF        |            | 3.6 h     | ~ 419 h           | ~ 220 h          | 214.2 h |
| KNNor       |            | 1.56 s    | 43.93 s           | 55.04 s          | 33.51 s |
| KNNcc       |            | 14.95 s   | 396.92 s          | 425.95 s         | 279.27 s|
| RFor        |            | 0.84 s    | 28.93 s           | 34.74 s          | 21.50 s |
| RFcc        |            | 1.73 s    | 42.84 s           | 47.25 s          | 30.61 s |
| FNNor       |            | 35.83 s   | 1252.73 s         | 1622.72 s        | 970.43 s|
| FNNcc       |            | 51.94 s   | 1553.56 s         | 1802.39 s        | 1135.96 s|

We denote the time in hours (h) if it is more than 3600 s otherwise we denote it in seconds (s)

Table 5 Compare with other methods from previous work

| Dataset      | This work | Other methods                                                                 |
|--------------|-----------|-------------------------------------------------------------------------------|
| CASAS        | 69.127 (HMM w. combined labels) | 64.16 (Hsu et al. 2010), 61.78 (Chiang et al. 2010)                           |
| ARAS House A | 55.73 (RNNgru w. combined labels) | 48.36 (Prossegger and Bouchacha 2014)                                        |
| ARAS House B | 77.33 (RNNgru w. combined labels) | 64.19 (Prossegger and Bouchacha 2014)                                        |

with IDT using days 1–7 for training and days 22–28 for testing. Here we chose the RNN with gated units and train it with 1000 hidden units in 30 epochs which achieved 55.73% and 77.33% accuracy for ARAS house A and ARAS house B respectively, which are better than 48.36% for house A, 64.19% for house B in Prossegger and Bouchacha (2014).

6 Discussion

Multi-resident activity recognition using ambient sensors is an important topic since it can reduce deployment cost and solve the privacy issue. Based on the complexity of 3 datasets (i.e., CASAS, ARASB, ARASA), we have some remarks related to the performance according to the complexity of dataset as following:

- Top five models which achieve the highest averaged accuracy (and F1) for all datasets RNNgru, RNNlstm, mRNNgru, CRF, and mRNNlstm, as shown in Fig. 8. However, readers should take into account the different nature of CASAS and ARAS datasets to make their own judgment. Although the former is a laboratory set-up and consists of smaller number
of activities, the way it is annotated allows the natural transitions of the activities. The latter (ARAS) dataset is collected in unscripted environments which reflects better a real-world situation. However, the use of GUIs for instant annotation makes it somehow disruptive and difficult to apply, especially to the case of aged care.

- Simple models, such as HMM and KNN, did not perform well on ARAS House-A but got decent performance on CASAS. Additionally, due to the simplicity of the model, the computation time of HMM is the shortest among all methods. Thus, HMM seems to be suitable to use in a place where prediction time is critical while does not have complex environments (e.g., fewer residents and fewer activities such as in senior house to detect regular/danger situations).

- CRF model is generally a good option if one wants to balance the trade-off between accuracy and computational time. In fact, CRF got the best performance on the ARAS House-A.

- Among temporal modelling approaches, recurrent neural networks with gated recurrent unit (i.e., GRU) got the best performance in CASAS and ARAS House-A, therefore, it is suitable to be applied in different environments including those with multiple activities. However, due to the complexity of the model, it requires more computational power and hence, might be suitable to be deployed in a well-equipped environment with high requirement on accuracy (e.g., hospital or office complex).

- For temporal data processing approaches, their advantage is efficiency while still having comparable or better performance to sequence models. In particular, RF and FNN outperform RNNs in ARAS House-A and ARAS House-B.

- In terms of temporal dependencies, we compared the effectiveness of separate labels and combined labels. The former method for encoding of labels assumes that activities of each resident should be inferred independently, while the latter assumes that the activities of all residents should be inferred at the same time. The results showed that combining labels performs better than separating labels.
Data association is one of the most important issues in multi-resident activity recognition which is about relating each resident with the sensors he or she interacts with. Previous approaches introduce an additional hidden variable to represent data association and Wilson and Atkeson (2005), Hsu et al. (2010). Although it helps improve the activity recognition performance, there are no ground-truths to validate the accuracy of that data association. Moreover, modelling data association as a hidden variable is risky as it has been shown that low accuracy of data association prediction can negatively affect the performance of activity recognition (Hsu et al. 2010). In this work, we focus on modelling temporal dependencies of residents’ activities instead of data association and showed in Sect. 5.7 that even with simple HMMs we can achieve better performance than iterative method predicting both data association and activities in CASAS dataset (Hsu et al. 2010). However, the idea of learning data association at the same time with activities is promising and we would like to investigate further in future work.

Based on the above discussion, we hope that our remarks are useful for the future application to choose between different models based on their characteristics. It is also noted that, since we do not have more complex datasets, therefore, we cannot verify whether more complex models such as LSTM will be suitable to be applied because it faced overfitting issue in our datasets.

7 Conclusions

We have presented a benchmark study on activity recognition for multi-resident smart homes with ambient sensors. In this work we empirically compare different generic approaches including temporal models (with and without combined labels) and non-temporal models. The empirical results show that in the case of temporal models, using combined activities as single labels is more effective than separate labels. Among temporal models, recurrent neural networks with gated recurrent units achieve the highest average accuracy and F1 score. In the case of non-temporal models, two types of input representation are proposed, namely CONCAT and MAX. It showed that MAX function slightly better than CONCAT, and more efficient. The temporal approaches perform better than the non-temporal counterparts in CASAS and ARAS House-A datasets, but less effective in ARAS House-B.

References

Alemdar H, Ertan H, Incel OD, Ersoy C (2013) Aras human activity datasets in multiple homes with multiple residents. In: PervasiveHealth’13. ICST, Brussels, Belgium, pp 232–235. https://doi.org/10.4108/icst.pervasivehealth.2013.252120
Bennmansour A, Bouchachia A, Feham M (2015) Multioccupant activity recognition in pervasive smart home environments. ACM Comput Surv 48(3):34:1–34:36
Brand M, Oliver N, Pentland A (1997) Coupled hidden Markov models for complex action recognition. In: Proceedings of the 1997 conference on computer vision and pattern recognition, CVPR’97. IEEE Computer Society, Washington, DC, USA, p 994
Chen R, Tong Y (2014) A two-stage method for solving multi-resident activity recognition in smart environments. Entropy 16(4):2184
Chiang YT, Hsu KC, Lu CH, Fu LC, Hsu JYJ (2010) Interaction models for multiple-resident activity recognition in a smart home. In: IEEE/RSJ international conference on IROS, pp 3753–3758. https://doi.org/10.1109/IROS.2010.5650340
Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp 1724–1734

Cook DJ (2012) Learning setting-generalized activity models for smart spaces. IEEE Intell Syst 27:32–38. https://doi.org/10.1109/MIS.2010.112

Cook DJ, Crandall A, Singla G, Thomas B (2010) Detection of social interaction in smart spaces. Cybern Syst 41(2):90–104. https://doi.org/10.1080/01969720903584183

Crandall AS, Cook DJ (2008) Resident and caregiver: handling multiple people in a smart care facility. In: Proceedings of AAAI fall symposium—AI in eldercare—new solutions to old problems, AAAI technical report. vol FS-08-02. AAAI. http://www.aaai.org/Library/Symposia/Fall/fs08-02.php

Das SK, Cook DJ (2004) Health monitoring in an agent-based smart home. In: Proceedings of the international conference on smart homes and health telematics. ICOST, IOS Press, pp 3–14

Ghahramani Z, Jordan MI (1997) Factorial hidden Markov models. Mach Learn 29(2–3):245–273. https://doi.org/10.1023/A:1007425814087

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hsu KC, Chiang YT, Lin GY, Lu CH, Hsu JYJ, Fu LC (2010) Strategies for inference mechanism of conditional random fields for multiple-resident activity recognition in a smart home. Springer, Berlin, pp 417–426. https://doi.org/10.1007/978-3-642-13022-9_42

Plötz T, Hammerla NY, Olivier P (2011) Feature learning for activity recognition in ubiquitous computing. In: Proceedings of the twenty-second international joint conference on artificial intelligence—volume two, IJCAI'11. AAAI Press, pp 1729–1734

Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28(6):976–990

Prossegger M, Bouchachia A (2014) Multi-resident activity recognition using incremental decision trees. In: Adaptive and intelligent systems—third international conference, ICAIS 2014, Bournemouth, UK, September 8–10, 2014, pp 182–191

Rabiner LR (1990) Readings in speech recognition. A tutorial on hidden markov models and selected applications in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, pp 267–296

Saini R, Kumar P, PratinRoy P, Dogra DP (2018) A novel framework of continuous human-activity recognition using kinect. Neurocomputing 311:99–111

Singla G, Cook DJ, Schmitter-Edgecombe M (2010) Recognizing independent and joint activities among multiple residents in smart environments. J Ambient Intell Humaniz Comput 1(1):57–63. https://doi.org/10.1007/s12652-009-0007-1

Son NT, Qing Z, Mohan K (2017) Improving multi-resident activity recognition for smarter home. In: IJCAI 2017 WS on AI for IoT, IJCAI'17

Sutton C, McCallum A, Rohanimanesh K (2007) Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. J Mach Learn Res 8:693–723

Tunca C, Alemdar H, Ertan H, Incel OD, Ersoy C (2014) Multimodal wireless sensor network-based ambient assisted living in real homes with multiple residents. Sensors 14:9692–9719

van Kasteren T, Noulas A, Englebienne G, Kröse B (2008) Accurate activity recognition in a home setting. In: Proceedings of the 10th international conference on ubiquitous computing. UbiComp’08. ACM, New York, NY, USA, pp 1–9. https://doi.org/10.1145/1409635.1409637

Wang L, Gu T, Tao X, Chen H, Lu J (2011) Recognizing multi-user activities using wearable sensors in a smart home. Pervasive Mob Comput 7(3):287–298. https://doi.org/10.1016/j.pmcj.2010.11.008

Wilson DH, Atkeson C (2005) Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In: Proceedings of the third international conference on pervasive computing, PERVASIVE’05. Springer, Berlin, pp 62–79. https://doi.org/10.1007/11428572_5

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.