Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 110 (2017) 86–93

Procedia Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017)

Activity Recognition and Abnormal Behaviour Detection with Recurrent Neural Networks

Damla Arifoglu*, Abdelhamid Bouchachia

Department of Computing and Informatics
Bournemouth University, UK

Abstract

In this paper, we study the problem of activity recognition and abnormal behaviour detection for elderly people with dementia. Very few studies have attempted to address this problem presumably because of the lack of experimental data in the context of dementia care. In particular, the paper investigates three variants of Recurrent Neural Networks (RNNs): Vanilla RNNs (VRNN), Long Short Term RNNs (LSTM) and Gated Recurrent Unit RNNs (GRU). Here activity recognition is considered as a sequence labelling problem, while abnormal behaviour is flagged based on the deviation from normal patterns. To provide an adequate discussion of the performance of RNNs in this context, we compare them against the state-of-art methods such as Support Vector Machines (SVMs), Naïve Bayes (NB), Hidden Markov Models (HMMs), Hidden Semi-Markov Models (HSMM) and Conditional Random Fields (CRFs). The results obtained indicate that RNNs are competitive with those state-of-art methods. Moreover, the paper presents a methodology for generating synthetic data reflecting on some behaviours of people with dementia given the difficulty of obtaining real-world data.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Smart Homes; Sensor based Activity Recognition; Recurrent Neural Networks; Dementia; Abnormal Behaviour Detection

1. Introduction

Studies indicate that by year 2030, 19% of people will be aged 74 to 84 and nearly half of people who are older than 84 will have dementia[1]. Elderly people may suffer from the consequences of dementia, which is a condition that causes problems with mobility, physical and mental abilities such as memory and thinking[2]. It also may cause decrease in the ability of speaking, writing, distinguishing objects, performing motor activities and performing complex functional tasks (paying bills, preparing a meal, shopping, managing medication, etc.)[3]. An elderly person having such cognitive decline loses independence in daily life and requires care and support from caregivers.

Cognitive diseases like dementia need to be detected at an early stage so that early treatment will be possible. However, research shows that 75% of dementia and early dementia cases go unnoticed[4] and many such cases are only

* Corresponding author. Tel.: +44 (0)1202 524111 ; fax: +44 (0)1202 962736.
E-mail address: darifoglu@bournemouth.ac.uk

1877-0509
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.06.121

diagnosed when such impairment reaches moderate or advanced stage. The detection of early signs of motion and cognitive impairment (MCI) via activity recognition will be useful to track motion and cognitive capabilities of the elderly, thus improving their life quality and financial saving. Unfortunately, currently there are no dementia friendly smart homes addressing these people's special needs.

Most common types of dementia (Alzheimer, Parkinsons disease) can be identified by behavioural changes like sleep disturbances, difficulty of walking and inability to complete tasks. Such changes can provide key information about memory, mobility and cognition of a person. For instance, an inhabitant suffering from Alzheimer may forget his lunch, take multiple lunches instead, wake up in the middle of the night, go to the toilet frequently, or have dehydration problems because of forgetting to drink daily amount of water.

Recent studies suggest that changes in complex daily life tasks can be indicators of early decline[5]. The best markers of cognitive decline may not necessarily be detected based on a person's performance at any single point in time, but rather by monitoring the trend over time and the variability of change in a duration[5]. Thus, tracking an elderly person's life over time in a specially designed smart home, doing in-home health assessment and detecting the indicators of dementia at an early step would be beneficial.

The identification of early onsets of dementia using non-medical diagnosis methods requires the development of new diagnostic tools. Although a few promising methods have been experimentally validated[6,7,8,9,10], the translation of the current knowledge into smart homes still requires more dedication and work. Current assessment methods mostly rely on queries from questionnaires or in-person examinations, which depend on recall of events or brief snapshots of function that may poorly represent a person's typical state of function. Moreover, these studies include some pre-defined tasks given to the patients in order to do automatic assessment of cognitive decline by trained experts.

The main motivation for our work is that cognitive decline can be observed in daily activities and routines of an elderly. Real-time monitoring of activities performed by elderly in a smart home would be beneficial for the early detection of such decline. In this study, we firstly recognise activities by variants of RNNs, namely VRNNs, LSTMs and GRUs and model the daily behaviour routines of a person. Whenever a new sequence is introduced, any abnormality deviating from these regular behaviours are detected and could be used for further investigation by formal or informal carer.

Unfortunately, there exists no publicly available dataset on abnormal behaviour of people with dementia. Producing such a dataset require time and adequate experimental environment. Thus we propose in this paper, a way to artificially produce data on abnormal activities reflecting on typical behaviour of elderly people with dementia. We believe that this an important contribution.

The rest of the paper is organised as follows. Section 2 provides a brief overview of the related research to both activity recognition and abnormal behaviour detection. Section 3 presents the details of the proposed methodology together with the datasets and models used. Section 4 describes the experimental set-up and results of the experiments followed by a discussion. Finally, Section 5 concludes the paper.

2. Literature Review

Activity recognition has been addressed using methods such as decision trees, Bayesian methods (Naïve Bayes and Bayesian Networks), k-Nearest Neighbours, Neural Networks (Multilayer perceptron), SVMs, Fuzzy logic, Regression models, Markov models (Hidden Markov Models, Conditional Random Fields) and classifier ensembles (Boosting and bagging)[11]. Recently, there has been growing interest in deep convolutional neural networks[12,13,14,15], Deep Belief Networks[16], Restricked Boltzman Machines (RBMs)[17,18,16,19] and RNNs[14,15,20]. Previous work shows that RNNs are useful, but leaves a lot of room for improvement. It is worthwhile to stress that to the best of our knowledge, this study is the first applying RNNs to detect abnormalities related to dementia in the daily life routines of an elderly person.

In[17], RBMs are used for feature extraction and selection from sequential data. In[14], the authors use a combination of deep convolutional networks and LSTM to do multi-modal wearable activity recognition by showing that their approach outperforms some of the previously reported results by up to 9% on OPPORTUNITY dataset. In[21], the authors utilised convolutional networks to classify activities using time-series data collected from smart phone sensors. Experiments show that increasing the number of convolutional layers increases the performance, but the complexity of the derived features decreases with every additional layer. In[15], the authors explore deep, convolutional and recurrent approaches across three representative datasets that contain movement data captured with wearable sensors. Moreover, they describe how to train recurrent approaches in this setting and introduce a novel regularisation approach, showing better results over OPPORTUNITY, PAMAP2 and Daphnet Gait datasets. In[19], results with RBM on CASAS dataset outperformed HMM and Naïve Bayes Classifier (NBC) in most of the cases. In[22], the authors use RNNs to predict the future values (start time, duration) of the activities.

Most of the aforementioned studies use movement data such as OPPORTUNITY, SKODA[17,12,13,14] or UCI HAR smart phone dataset, MIT home dataset[21,16], which are obtained through body worn sensors. Except the work by Fang et al.[19,20], none of these studies focus on daily activity datasets collected by sensors placed at home. In this work, we investigate RNNs on daily activities data obtained by van Kasteren[23] using various environment sensors (see Sec. 3.1 for more details).

In-home automatic assessment of cognitive decline has been the subject of some studies dedicated [24,6,25,18]. For instance, in[24], machine learning approaches such as SVMs and Naïve Bayes are used. In[18], Parkinson's Disease state assessment in home is explored by means of RBMs using data from body worn sensors. In[25], the authors use Markov Logic Network, which is a probabilistic logic that unifies statistical and symbolic reasoning to detect anomalies. In[24], some instructions to perform some tasks (e.g., sweeping the kitchen, dusting the floor, etc.) are given to the patients who then receive scores after completing those tasks. These scores are calculated based on the time spent, the frequency of the sensor triggered, etc. One disadvantage of this scenario is that some pre-selected activities are performed and instructions are given to the elderly who might not be able to cope with such tasks at all. Moreover, using rule-based systems, an expert is needed to manually integrate resident-specific rules to the system since every person has her/his own daily life routines. For example waking up and drinking water in the middle of the night might be normal for a person, while abnormal for some other person. However, our approach does not require any expert knowledge, since it learns what is normal and abnormal from the training data automatically. Specifically, we aim in this study to detect anomalies in the natural flow of daily living without giving any instruction and considering not only some time interval, but everyday living scenario. Continuous assessment of the person is more valid, since activities are performed in the person's own home setting.

3. Proposed Method

To assess RNNs in activity recognition and abnormal activity detection, we propose the following steps: Firstly, raw dataset is segmented into slices by using a sliding window approach. The window size is 60 seconds time of sensor readings as described in[23]. Secondly, sensor-based features are extracted from these slices. These features are *binary*, *change-point* and *last-fired* representations which are used also in[23]. Thirdly, RNNs (Vanilla, GRU and LSTM) are trained to recognise daily activities and encode daily-life behaviour routines. Lastly, the trained model is used to detect anomalies deviating from the normal daily-life sequences.

In the following we describe the dataset as well as the methodology used to generate artificial dataset that reflects on the typical behaviour of a person with dementia.

3.1. Dataset and Features

We used the popular dataset collected by Van Kasteren[23] from 3 households which are denoted as dataset *A*, *B* and *C*. The data captures daily-life activities such as sleeping, cooking, leaving home, etc. using sensors placed at the homes in less than a month. Please see[23] for more details. We applied the same sliding window approach as in[23] to extract the sensor reading chunks. We also considered three feature representations: *binary*, *change-point* and *last-fired* which are described as follows:

- *Binary*: This representation gives 1 when the sensor is triggered and 0 when that sensor is not triggered.
- *Change-point*: This representation gives information when a sensor changes value. More specifically, it gives 1 when a sensor changes its current state (either from state 1 to state 0 or vice versa) and a 0 when its value remains the same.
- *Last-fired*: This representation indicates which sensor is fired last. The sensor that changed state last continues to give 1 and changes to 0 when another sensor changes state.

3.2. Generation of Abnormal Activities Related to Dementia

Since we do not have any available dataset related to abnormal behaviour of people with dementia, we artificially create some anomalies in the dataset. In order to show the applicability of the proposed work to detect these anomalies, we focus on two different kinds of anomalies that can be seen in daily-life routines of elderly people with dementia: 1) Forgetting or repeating activities 2) Dehydration and disruption in sleep.

1. **Forgetting and repeating activities:** Elderly people suffering from dementia may forget whether they performed a particular daily activity or not, so they may repeat that activity multiple times or they may skip that activity. For instance, an elderly person suffering from Alzheimer may forget to have lunch, take multiple lunches instead [26], to have dinner and start to prepare it in the middle of the night. To reflect on this, we generate this kind of abnormal activities by manually inserting a specific set of actions within the normal activity sequence. This will result in multiple occurrences of that activity, which will occur in some inadequate time of the day such as having dinner in the middle of the night. We inject the instances of the following activities: *brushing teeth, preparing dinner, eating, getting snack* into the normal activity sequences to generate abnormal activities related to the frequency.
2. **Dehydration and disruption in sleep:** Degeneration of the sleep-waking cycle, sleep disorders and night time wandering are among the most severe behavioural symptoms of dementia. For example, elderly people may wake up many times in the night to use the toilet and go back to sleep and may forget to take daily amount of water [26,27]. We simulate these anomalies by inserting some synthetic activities in the normal night-time activity sequences of a person. More specifically, we inject *getting drink, going to toilet* into the *sleeping* activity of normal daily activity sequences. This will emulate the activities of getting drink and going to the toilet frequently in the middle of the night.

We generate these abnormal activity instances on dataset A which has the following 9 activities: *Leave house, use toilet, take shower, brush teeth, go to bed, prepare breakfast, prepare dinner, get snack, get drink.* As a result, we have multiple instances of those injected instances in order to simulate the anomalies related to dementia. Here, please note that there is only one subject in the dataset. We take the lifestyle in the training data as a norm and then synthesise the abnormalities deviating from this norm and introduce these abnormalities in the test data. These activities are totally normal on their own but they become abnormal when they occur at a wrong time of the day and after or before a specific activity. Hence, capturing these abnormalities within the context is important. In all, we manually synthesise 135 abnormal activity slices.

3.3. Activity Recognition and Abnormal Behaviour Detection

We believe that the order of activities and their temporal and spatial information is important to encode an elderly person's daily life routines. This kind of information can provide important cues to understand the daily patterns and thus to detect any anomalies in those patterns. Sequence labelling methods such as HMMs and RNNs can capture temporal and spatial relationship between activities, which some generative methods like SVMs can not do. In this work, we investigate the adequacy of RNNs to this task.

In order to recognise daily activities, training instances of the datasets and their corresponding labels are fed into the RNNs. Then when a new test sequence is introduced, the trained model assigns labels to each activity instances of that sequence. Each model gives a confidence value about the assigned label for the new sequence. Firstly, we calculate the mean of confidence values of training instances that are assigned by the model. Then, when a new test sequence is introduced if the model assigns it to a class label with a confidence value which is bigger than the mean, the sequence is considered as a normal activity, otherwise it is abnormal activity.

3.4. RNN Architectures

In the following we give a summary of the RNN architectures used in this work, more specifically Vanilla RNNs, Long Short Term Memory RNNs, and Gated Recurrent Unit RNNs. Then, we describe how they are used in the context of daily activity recognition and abnormal activity detection tasks.

1. **Vanilla Recurrent Neural Networks:** In feed-forward neural network, it is assumed that all inputs and outputs are independent of each other, but RNNs have a recurrent hidden state whose activation at each time is dependent on that of the previous time. This architecture is recurrent as some of the connections within the network form a directed cycle, where the current time-step t considers the states of the network in the previous time-step $t-1$. They share parameters for different time-steps which enables them to be used in sequential data. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being dependent on the previous computations. Another way to think about RNNs is that they have a memory which captures information about what has been calculated so far. However, there is a drawback of Vanilla RNNs, as shown by Bengio et al.[28], Vanilla RNNs are not capable of capturing long term dependencies on sequences because of the vanishing gradient problem. In theory, RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps. Thus, the following two RNN architectures are exploited to solve this problem.
2. **Long Short Term Memory (LSTM) Recurrent Neural Networks:** LSTM cells are designed to counter the effect of diminishing gradients when error derivatives are backpropagated through many layers through time in recurrent networks[29]. Each LSTM unit keeps track of an internal state that represents its memory. Over time the cells learn to output, overwrite, or null their internal memory based on their current input and the history of past internal states, leading to a system capable of retaining information across hundreds of time-steps[29]. LSTM blocks have 3 gates to control the flow of information into or out of their memory. For example, an *input gate* controls the extent to which a new value flows into the memory. A *forget gate* controls the extent to which a value remains in memory while an *output gate* is used to compute the output activation of the block (see Figure 1).

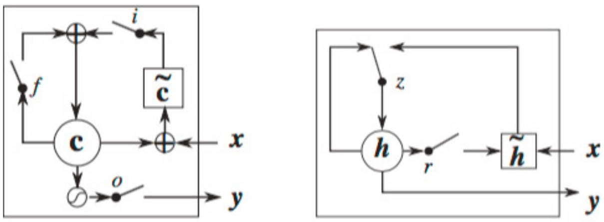

Fig. 1. Left: LSTM, Right: GRU. While LSTM can be described as the input signals x_t at time t, the output signals y_t, the forget gate f_t, and the input gate i_t, the output gate o_t,; GRU, on the other hand, can be described in terms of two internal variables, which retain the previous h and current h inner states respectively.

3. **Gated Recurrent Unit:** Cho et al.[28] recently proposed GRU, which is like LSTM but it has fewer parameters than LSTM, as GRUs lack an *output gate*. In GRU, each hidden unit has two gates, which are called *update* and *reset gates* (see Figure 1). GRU also controls the flow of information to prevent vanishing gradient problem, but without having to use a memory unit.

4. Experiments and Results

We used Keras Deep Learning library's[30] and Theano's[31] implementations of the RNNs (GRU, LSTM, Vanilla RNN) in this study. Moreover for the sake of comparison, we also used the One-class SVM from WEKA with default parameters, Naïve Bayes (NB), Hidden Markov Models (HMM), Hidden Semi-Markov Models (HSMM) and Conditional Random Fields (CRF) which are based on the implementation provided in[23].

We split the data (see Sec. 3.1) into a test and training set using the leave-one-day-out cross-validation approach. One full day of sensor readings is used for testing and the remaining days are used for training. Then we cycle over all days and report the average performance.

We evaluate metrics proposed in[23]: precision, recall, F-measure and accuracy. We calculate precision and recall for each class separately and then take the average over all classes. Note that precision and recall measures are used since these metrics give some idea about how well the models perform on imbalanced datasets like the one in this

Table 1. Activity recognition results on dataset *A*

Model	Feature	Precision	Recall	F-Measure	Accuracy
NB	Binary	48.3 ± 17.7	42.6 ± 16.6	45.1 ± 16.9	77.1 ± 20.8
	Change-point	52.7 ± 17.5	43.2 ± 18.0	47.1 ± 17.2	55.9 ± 18.8
	Last-fired	67.3 ± 17.2	64.8 ± 14.6	65.8 ± 15.5	95.3 ± 2.8
HMM	Binary	37.9 ± 19.8	45.5 ± 19.5	41.0 ± 19.5	59.1 ± 28.7
	Change-point	70.3 ± 16.0	74.3 ± 13.3	72.0 ± 14.2	92.3 ± 5.8
	Last-fired	54.6 ± 17.0	69.5 ± 12.7	60.8 ± 14.9	89.5 ± 8.4
HSMM	Binary	39.5 ± 18.9	48.5 ± 19.5	43.2 ± 19.1	59.5 ± 29.0
	Change-point	70.5 ± 16.0	75.0 ± 12.1	72.4 ± 13.7	91.8 ± 5.9
	Last-fired	60.2 ± 15.4	73.8 ± 12.5	66.0 ± 13.7	91.0 ± 7.2
CRF	Binary	59.2 ± 18.3	56.1 ± 17.3	57.2 ± 17.3	89.8 ± 8.5
	Change-point	73.5 ± 16.6	68.0 ± 16.0	70.4 ± 15.9	91.4 ± 5.6
	Last-fired	66.2 ± 15.8	65.8 ± 14.0	65.9 ± 14.6	96.4 ± 2.4
Vanilla	Binary	46.5 ± 17.7	64.8 ± 16.2	53.5 ± 16.3	86.8 ± 10.6
	Change-point	46.3 ± 19.5	63.8 ± 16.4	53.2 ± 17.9	61.4 ± 16.4
	Last-fired	61.9 ± 19.1	74.3 ± 12.8	67.2 ± 16.4	95.5 ± 3.4
LSTM	Binary	50.8 ± 18.4	63.9 ± 16.5	56.2 ± 17.1	86.7 ± 10.5
	Change-point	46.8 ± 18.7	63.6 ± 14	53.5 ± 16.7	61.4 ± 16.4
	Last-fired	63.7 ± 19.9	73.9 ± 16.8	68.1 ± 18.2	96.7 ± 2.6
GRU	Binary	47.3 ± 18.7	69.1 ± 14.9	55.4 ± 16.5	86.6 ± 10.7
	Change-point	42.9 ± 19	65.0 ± 15.3	51.0 ± 17.1	61.4 ± 16.4
	Last-fired	61.8 ± 16.3	80.6 ± 11.5	69.5 ± 14.0	96.1 ± 2.5
SVM	Binary	45.6 ± 17.9	69.1 ± 15.9	54.2 ± 15.9	85.4 ± 10.4
	Change-point	40.3 ± 19.1	63.4 ± 14.6	48.6 ± 17.0	55.9 ± 18.7
	Last-fired	58.6 ± 16.2	77.2 ± 14.0	66.3 ± 14.9	96.1 ± 2.4

Table 2. Activity recognition results on dataset *B*.

Model	Feature	Precision	Recall	F-Measure	Accuracy
NB	Binary	33.6 ± 10.9	32.5 ± 8.4	32.4 ± 8.9	80.4 ± 18.9
	Change-point	40.9 ± 7.2	38.9 ± 5.7	39.5 ± 5.9	67.8 ± 18.6
	Last-fired	43.7 ± 8.7	44.6 ± 7.2	43.3 ± 4.8	86.2 ± 13.8
HMM	Binary	38.8 ± 14.7	44.7 ± 13.4	40.7 ± 12.4	63.2 ± 24.7
	Change-point	48.2 ± 17.2	63.1 ± 14.1	53.6 ± 16.5	81.0 ± 14.2
	Last-fired	38.5 ± 15.8	46.6 ± 19.5	41.8 ± 17.1	48.4 ± 26.9
HSMM	Binary	37.4 ± 16.9	44.6 ± 14.3	39.9 ± 14.3	63.8 ± 24.2
	Change-point	49.8 ± 15.8	65.2 ± 13.4	55.7 ± 14.6	82.3 ± 13.5
	Last-fired	40.8 ± 11.6	53.3 ± 10.9	45.8 ± 11.2	67.1 ± 24.8
CRF	Binary	35.7 ± 15.2	40.6 ± 12.0	37.5 ± 13.7	78.0 ± 25.9
	Change-point	48.3 ± 8.3	51.5 ± 8.5	49.7 ± 7.9	92.9 ± 6.2
	Last-fired	46.9 ± 12.5	47.8 ± 12.1	46.6 ± 12.9	89.2 ± 13.9
Vanilla	Binary	26.7 ± 13.5	46.9 ± 24.8	32.5 ± 17.9	65.2 ± 34.7
	Change-point	39.6 ± 8	62.4 ± 15.3	48.3 ± 10.2	76.9 ± 13.9
	Last-fired	41.2 ± 12.3	64.4 ± 17.8	49.7 ± 13.6	87.9 ± 13.1
LSTM	Binary	29.1 ± 12.0	44.0 ± 22.0	33.9 ± 16.2	63.5 ± 32.7
	Change-point	40.0 ± 11.2	59.0 ± 16.4	47.5 ± 12.9	76.8 ± 14.2
	Last-fired	40.8 ± 10.7	60.1 ± 16.3	48.2 ± 12.3	87.2 ± 13.2
GRU	Binary	28.5 ± 15.9	36.3 ± 17.2	31.4 ± 16.2	64.5 ± 32.1
	Change-point	37.7 ± 7.6	53.5 ± 9.2	44.9 ± 7.1	76.4 ± 14.5
	Last-fired	41.7 ± 13.2	56.9 ± 17.9	47.5 ± 14.6	87.0 ± 12.9
SVM	Binary	39.6 ± 10.9	58.5 ± 17.4	46.7 ± 12.9	81.6 ± 18.5
	Change-point	32.3 ± 6.5	53.6 ± 7.5	40.0 ± 6.2	67.9 ± 28.5
	Last-fired	36.4 ± 5.4	54.6 ± 10.4	43.5 ± 6.6	86.2 ± 14.9

study. On the other hand, the accuracy represents the percentage of correctly classified time slices, therefore more frequently occurring classes have a larger weight in this measure.

To evaluate the performance of abnormal behaviour detection, we use the following evaluation metrics: True Positive Rate (TPR) and False Positive Rate (FPR). TPR is the percentage of correctly detected abnormal activities out of total abnormal activities, FPR is the percentage of normal activities that are detected falsely as abnormal activities by the algorithm (out of total number of normal activities).

To run experiments on RNNs, we left out 10% of the training data for validation and we used drop-out with a value of 0.2. We also set the batch size to 10 instances and the epoch to 500 iterations. The internal architecture of RNNs (2 layers consisting of 30 and 50 nodes respectively) and time step of the sequences (25 activity slices) were empirically set.

Note that the results obtained by the models HMM, HSMM, CRF and NB (see Tab. 1 - 4) are taken from the study by Kasteren et al.[23].

Table 1 refers to the results obtained on dataset *A* and shows that there is no clear winner among the three different feature representations. Considering the accuracy, the results indicate that LSTM is the best method (with the accuracy of 96.7%) when *last-fired* feature is used, while HMM performs the worst. Using *change-point* feature, HMM outperforms all other methods. Using *binary* feature on the other hand shows that CRF (accuracy of 89.8%) is the best. Also all RNNs, NB and SVM do not perform well when adopting *change-point* feature. HMM and HSMM are not good when using *binary* feature representation. In a nutshell, for the majority of the methods, except HMM and HSMM, *last-fired* representation is the best one. In terms of recall which reflects better on performance in the presence of imbalanced data, the highest value is obtained by GRU (80.6%). This potentially indicate that RNNs are good to detect relevant class instances. CRF, for instance, score higher on precision, because the most frequent-class instances are favoured, but then it is not so good at when it comes to the infrequent classes. Overall, there is a clear hint that that recurrent architectures perform better than HM, NB and HSMM for most of the cases, while CRF is slightly better than these recurrent architectures on dataset *A*.

Table 2 refers to the results obtained on dataset *B* and shows that SVM is the best method when adopting *binary* representation achieving the accuracy of 81.6%. On the other hand, CRF is the best when using the *change-point* feature and *last-fired* representations with accuracy 92.9% and 89.2% respectively. It can be noted that HMM is not as good as the other methods achieving in the best case only 81.0% with the *change-point* representation. The closest successful model to CRF is Vanilla RNN and again overall RNNs deliver high recall rates compared to the other methods. *Change-point* and *last-fired* representations give the highest recall results except for CRF.

Table 4 reports the results on dataset *C* showing that CRF performs best for *change-point* and *binary* representations obtaining 82.2% and 89.7% respectively. Overall, none of the methods performs well when adopting *binary*

Table 3. Activity recognition results on dataset C

Model	Feature	Precision	Recall	F-Measure	Accuracy
NB	Binary	19.6 ± 11.4	16.8 ± 7.5	17.8 ± 9.1	46.5 ± 22.6
	Change-point	39.9 ± 6.9	30.8 ± 4.8	34.5 ± 4.6	57.6 ± 15.4
	Last-fired	40.5 ± 7.4	46.4 ± 14.8	42.3 ± 6.8	87.0 ± 12.2
HMM	Binary	15.2 ± 9.2	17.2 ± 9.3	15.7 ± 8.8	26.5 ± 22.7
	Change-point	41.4 ± 8.8	50.0 ± 11.4	44.9 ± 8.8	77.2 ± 14.6
	Last-fired	40.7 ± 9.7	53.7 ± 16.2	45.9 ± 11.2	83.9 ± 13.9
HSMM	Binary	15.6 ± 9.2	20.4 ± 10.9	17.3 ± 9.6	31.2 ± 24.6
	Change-point	43.8 ± 10.0	52.3 ± 12.8	47.4 ± 10.5	77.5 ± 15.3
	Last-fired	42.5 ± 10.8	56.0 ± 15.4	47.9 ± 11.3	84.5 ± 13.2
CRF	Binary	17.8 ± 22.1	21.8 ± 20.9	19.0 ± 21.8	46.3 ± 25.5 ‘
	Change-point	36.7 ± 18.0	39.6 ± 17.4	38.0 ± 17.6	82.2 ± 13.9
	Last-fired	37.7 ± 17.1	40.4 ± 16.0	38.9 ± 16.5	89.7 ± 8.4
Vanilla	Binary	15.4 ± 5.3	43.1 ± 18.1	22.2 ± 7.3	50.2 ± 22.4
	Change-point	31.3 ± 7.1	54.9 ± 11.3	39.5 ± 8.3	72.2 ± 13.0
	Last-fired	38.3 ± 16.3	59.6 ± 15.1	45.8 ± 14.8	86.7 ± 12.5
LSTM	Binary	16.8 ± 6.2	34.8 ± 12.5	22.1 ± 7.4	45.3 ± 21.2
	Change-point	31.0 ± 5.1	53.3 ± 6.5	38.9 ± 5.0	72.0 ± 13.0
	Last-fired	41.3 ± 17.2	57.3 ± 15.9	47.5 ± 16.1	87.4 ± 12.4
GRU	Binary	18.7 ± 8.3	33.2 ± 12.7	23.9 ± 9.6	46.7 ± 23.4
	Change-point	31.2 ± 8.3	47. ± 10.9	31.2 ± 8.5	71.6 ± 12.6
	Last-fired	40.4 ± 16.5	52.7 ± 16.4	45.4 ± 16.9	86.6 ± 12.3
SVM	Binary	19.4 ± 9.0	35.2 ± 12.7	24.0 ± 9.2	37.4 ± 19.0
	Change-point	25.6 ± 6.2	51.4 ± 9.5	34.0 ± 7.2	57.8 ± 15.5
	Last-fired	37.0 ± 7.9	55.5 ± 11.6	44.1 ± 8.5	87.5 ± 12.1

representation. The results are slightly better with *change-point* but clearly better when applying the *last-fired* representation. RNNs again give the highest recall values for all representations. Overall, the results show that RNNs perform better than HMM, NB and HSMM in all cases, while CRF is slightly better than RNNs. But in terms of recall, these later outperform all methods for all feature representations. The reason behind this is that RNNs perform better for imbalanced data compared to CRF. RNNs variants generally perform equally well.

For abnormal activity detection, we considered LSTM only and compared against NB, HSMM, HMM, SVM and CRF. TPR and FPR accuracy percentages are correspondingly; 40.40% and 43.50% for NB, 58.36% and 96.20% for HMM, 68.85% and 32.2% for HSMM, 66.22% and 40.50% for CRF, 72.11% and 44.0% for One-class SVM and 91.43% and 40.96% for LSTM. We used only *last-fired* feature in this experiment. The results indicate that LSTM is the best to prune false negatives compared to the other methods. Methods like NB, One-class SVM which do not capture the data order performs the worst. The models ignore the frequency of the activity, but apply the temporal and contextual information to make a decision. Results show that LSTM is capable of encoding the order of activities. Hence, when an activity is introduced in a different context or in a different order, LSTM can detect such anomalies.

Our current approach may fail to detect abnormalities, when there is gradual deterioration regarding the health of an elderly. We are planning to deal with this issue in the future while collecting real-world data in which gradual deterioration can be observed.

5. Conclusion

In this paper, we showed that RNNs perform well on the problem of activity recognition. They are also able to cope quite well with imbalanced data as well as anomaly detection which is very important in the context of dementia. Compared to a number of traditional and popular techniques used for activity recognition such as SVM, NB, HMM and HSMM, they perform much better, while remained very competitive with CRF. Furthermore, the empirical experiments showed that the three variants of RNNs generally perform equally well, but LSTM seems to be slightly better across all datasets used in this study. Moreover, in terms of representation, there is no clear preference, but *last-fired* feature seems to be better, at least on the datasets A and C, compared to the *change-point* and *binary* representations. Overall the study allowed to confirm that RNNs are very appropriate for activity recognition and abnormal activity detection. In our future investigations, we will extend RNNs to deep neural networks. We will also aim at collecting a dataset from a smart home dedicated to elderly people with dementia to further study behaviour anomalies related to dementia.

Acknowledgments

We thank to Tim Van Kasteren for providing the code and datasets.

References

1. D. A. Umphred, R. T. Lazaro, M. Roller, and G. Burton. Neurological rehabilitation. In *Elsevier Health Sciences, vol. 27, no. 5*, 2013.
2. M. S. Albert et al. The diagnosis of mild cognitive impairment due to Alzheimer's disease: Recommendations from the national institute on aging-alzheimer's association workgroups on diagnostic guidelines for alzheimer's disease. *Alzheimer's & Dementia*, Volume 7-Issue 3:270–279, 2011.
3. W. Thies and L. Bleiler. 2013 Alzheimer's disease facts and figures. *Alzheimer's & dementia: the journal of the Alzheimer's Association vol. 9, no. 2*, page 208245, 2013.
4. M. R. Hodges, K. Kirsch, M. W. Newman, and M. E. Pollack. Automatic assessment of cognitive impairment through electronic observation of object usage. In *Pervasive*, pages 192–209, 2010.
5. K. Wild. Aging changes. In *Geraotechnology, Vol. 9 No 2*, pages 121–125, 2010.
6. P. Dawadi, D. Cook, and M. Schmitter-Edgecombe. Smart home-based longitudinal functional assessment. In *ACM UbiComp Workshop on Smart Health Systems and Applications*, 2014.
7. D. Riboni, C. Bettini, G. Civitarese, Z. Haider Janjua, and R. Helaoui. Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment. *IEEE International Conference on Pervasive Computing and Communications*, 2015.
8. O. D. Lara and M. A. Labrador. A mobile platform for real time human activity recognition. *IEEE Conference on Consumer Communications and Networks*, 2012.
9. T. Kirste, A. Hoffmeyer, P. Koldrack, A. Bauer, S. Schubert, S. Schrder, and S. Teipel. Detecting the effect of Alzheimer's disease on everyday motion behaviour. *Journal of Alzheimer's Disease*, Journal of Alzheimer's Disease:121–132, 2014.
10. M. Ermes, J. Parkka, and L. Cluitmans. Advancing from offline to online activity recognition with wearable sensors. *30thAnnual International Conference of the IEEE Engineering in Medicine and Biology Society*, page 44514454, 2008.
11. O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys Tutorials*, 15(3):1192–1209, 2013.
12. J. Yang, M. Nguyen, P. San, X. Li Li, and S. Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. pages 3995–4001, 2015.
13. M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*, pages 197–205, 2014.
14. F. J. Ordonez and D. Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.
15. N. Hammerla, S. Halloran, and T. Pltz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016.
16. S. Choi, E. Kim, and S. Oh. Human behavior prediction for smart homes using deep learning. In *2013 IEEE RO-MAN*, pages 173–179, 2013.
17. T. Plötz, N. Hammerla, and P. Olivier. Feature learning for activity recognition in ubiquitous computing. volume 2, pages 1729–1734, 2011.
18. N. Hammerla, J. Fisher, P. Andras, L. Rochester, R. Walker, and T. Plotz. PD disease state assessment in naturalistic environments using deep learning. pages 1742–1748, 2015.
19. H. Fang and C. Hu. Recognizing human activity in smart home using deep learning algorithm. In *33rd Chinese Control Conference*, pages 4716–4720, 2014.
20. H. Fang, H. Si, and L. Chen. Recurrent neural network for human activity recognition in smart home. *In Proceedings of 2013 Chinese Intelligent Automation Conference*, pages 341–348, 2013.
21. Charissa A. and Sung-Bae C. Evaluation of deep convolutional neural network architectures for human activity recognition with smartphone sensors. *In Proceedings of the KIISE Korea Computer Congress*, page 858860, 2015.
22. A. Lotfi, Caroline Langensiepen, Sawsan M Mahmoud, and M Javad Akhlaghinia. Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. *Journal of ambient intelligence and humanized computing*, 3(3):205–218, 2012.
23. T. Van Kasteren, G. Englebienne, and B. J. A. Kröse. Human activity recognition from wireless sensor network data: Benchmark and software. *Activity Recognition in Pervasive Intelligent Environments*, pages 165–186, 2011.
24. P. Dawadi, D. Cook, C. Parsey, M. Schmitter-Edgecombe, and M. Schneider. An approach to cognitive assessment in smart homes. In *Knowledge Discovery and Data Mining Workshop on Medicine and Healthcare*, 2011.
25. D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and V. Bulgari. From lab to life: Fine-grained behavior monitoring in the elderly's home. *In Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 344–349, 2015.
26. J. Saives, C. Pianon, and G. Faraut. Activity discovery and detection of behavioral deviations of an inhabitant from binary sensors. *IEEE Transactions on Automation Science and Engineering*, 12(4):1211–1224, 2015.
27. M. Amiribesheli and A. Bouchachia. Smart homes design for people with dementia. In *2015 International Conference on Intelligent Environments*, pages 156–159, 2015.
28. K. Cho, B. Merrienboer, D. Bahdanau, and Y. Bengio. *On the properties of neural machine translation: Encoder-decoder approaches*. 2014.
29. S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, Nov 1997.
30. C. François. Keras. https://github.com/fchollet/keras, 2015.
31. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, may 2016.