Detecting Indicators of Cognitive Impairment via Graph Convolutional Networks

Damla Arifoglu, Hammadi Nait Charif and Abdelhamed Bouchachia

Bournemouth University
United Kingdom

Abstract

While the life expectancy is on the rise all over the world, more people face health related problems such as cognitive decline. Dementia is a name used to describe progressive brain syndromes affecting memory, thinking, behaviour and emotion. People suffering from dementia may lose their abilities to perform daily life activities and they become on their caregivers. Hence, detecting the indicators of cognitive decline and warning the caregivers and medical doctors for further diagnosis would be helpful. In this study, we tackle the problem of activity recognition and abnormal behaviour detection in the context of dementia by observing daily life patterns of elderly people. Since there is no real-world data available, firstly a method is presented to simulate abnormal behaviour that can be observed in daily activity patterns of dementia sufferers. Secondly, Graph Convolutional Networks (GCNs) are exploited to recognise activities based on their granular-level sensor activations. Thirdly, abnormal behaviour related to dementia is detected using activity recognition confidence probabilities. Lastly, GCNs are

Email address: `{darifoglu,hncharif,abouchachia}@bournemouth.ac.uk` (Damla Arifoglu, Hammadi Nait Charif and Abdelhamed Bouchachia)

compared against the state-of-the-art methods. The results obtained indicate that GCNs are able to recognise activities and flag abnormal behaviour related to dementia.
Keywords: Sensor-based Activity Recognition, Smart Homes, Abnormal Behaviour Detection, Graph Convolutional Networks, Cognitive Decline

1. Introduction

In recent years, cognitive decline and mental diseases in ageing people have become a major public health problem all over the world. Alzheimer's Disease International estimated that there were 50 million people suffering from dementia worldwide in 2018 and the number will increase to 82 million by 2030 and to 150 million by 2050 [1]. Cognitive impairment is a mental health disorder that causes a decline in cognitive abilities, including memory and thinking skills. Elderly people with cognitive decline experience a gradual loss of their ability to perform daily life activities due to memory impairments. Thus, they need special care and help from their caregivers which represent a social, psychological, physical and economic burden on families, caregivers and the society as a whole.

The assessment of cognitive decline is useful to identify care needs during Activities of Daily Living (ADLs), and to monitor the evolution of the condition. Most dementia patients have a strong desire to live autonomously in their known environments [2]. The use of smart home technologies can substantially support the lives of people with dementia. This kind of technology would also be helpful to in detecting the symptoms/signs of dementia and to warn the caregivers and medical doctors for further diagnosis.

Recent studies show that deviations in activity patterns can be indicators of cognitive decline [2, 3, 4, 5]. Although people don't perform the same activities in the same way every time, these activities still follow a set of patterns in terms of locations, time and frequency in each day. Deviations from these emerging patterns may be indicators of cognitive decline worth detecting. Behavioural changes such as sleep disturbance, night-time waking and inability to complete tasks can be indicators of cognitive impairment [6, 7, 8]. For example, an elderly person with cognitive decline may forget to have her lunch, take multiple lunches instead, wake up in the middle of the night, or go to the toilet frequently. She may also get confused and do mistakes during the execution of daily life activities. Unfortunately, currently there are no smart homes specially designed for elderly people suffering from cognitive impairment.

Although there are a few promising methods developed to assess behaviour change unobtrusively in real-time [9, 10, 11], the translation of the current knowledge into assisted living technologies still needs more work. In this paper, a novel approach is proposed to detect the indicators of cognitive impairment. The main idea of our work is that daily life activity patterns are indicators of cognitive decline [2]. This study aims to detect these indicators and assist the diagnosis and decision making of medical doctors and clinicians.

In image recognition, the pixels in images form edges and edges construct different shapes and then shapes form objects in images. Similar to image recognition, there are granular-level patterns in daily life activities. Daily life activities are often composed of several sub-activities [12, 13] or granular actions [14] and have hierarchical structure [15]. For example; the activity *wash clothes* implies the following actions: *get clothes from basket, fill up washing machine, turn on washing machine*. When we think about sensor based activities, some daily life activities such as *sleeping* or *wash dishes* may not have explicit sub-activities involved in, but we can exploit motion sensors replaced at home and their relative location and relationship with each other as sub-activities. As depicted in 1, the activity *wash dishes* consists of sub-activities *move to the kitchen area*, *move to the kitchen sink* and *use water*. The ordered sequence of motion sensors M_3, M_4, M_6 form these sub-activities hierarchically and then they result in *wash dishes* activity. An occupant in a house may mainly move around the kitchen sink in the *wash dishes* activity and the movement between kitchen range and the sink is observed and this leads to triggering of sensors next to the sink and kitchen in some order, or a person may stay around the bedroom area during *sleeping* activity.

Sub-activities and their relations with each other are important clues to understand and assess the cognitive status of an elderly person. The anomalies related to dementia may be reflected in the repetition frequency of granular level actions and their relation with each other. For example, when an elderly person wants to make a phone call, he/she may check the phonebook many times and perform this step more than once, even though they can successfully complete this activity. Thus building activities from their granular units hierarchically would be helpful to understand the internal dynamics of the activities. In the present study, we model sub-activities and their relationship exploiting raw sensor measurements in a graph structure. Modelling these sub-activities and constructing upper level activities based on a hierarchical relationship of these granular level structures would be helpful to better understand and model abnormal behaviour related to dementia.

Existing studies represent sensor activations in a fixed feature vector which can be thought as a bag that collects the sensors which are triggered in a given time [16, 17]. This representation ignores the fine-grained details such as frequency and the order of activations. This study addresses this shortcoming by focusing on raw sensor activations to recognise activities and then flag deviations related to dementia. Using raw sensor activations coming as a data stream allows us to capture the interaction between sensor activations as well as their intensity and ordering. As mentioned above, the granular level details (sub-activities, sensor activations and their relationship with each other) of activities are important in terms of understanding anomalies stemming from cognitive status in the context of dementia. However, current traditional features lose this information coming from raw sensor activation data. Thus, in this study our aim is to make use of this information to model activities better for the detection of abnormal behaviour. Secondly, this unstructured raw sensor activation data cannot be modelled by using fixed length features. Therefore, in this study, the problem of activity recognition is emulated as a graph labelling problem where each sensor activation is represented as a node in a graph. Modelling activities based on their sensor activations in a graph gives the opportunity to build the hierarchical relationship of sub-activities. Inspired by work on graph labelling [18, 19, 20], we explore the use of Graph Convolutional Networks (GCNs) to model activities from their low-level units. Moreover, there exists no publicly available datasets on abnormal behaviour of elderly people with dementia. Producing such datasets requires time and adequate experimental environment. A method is proposed to artificially produce abnormal activities reflecting on typical behaviour of elderly people with dementia. Similar studies [21, 22] have also adopted simulation.

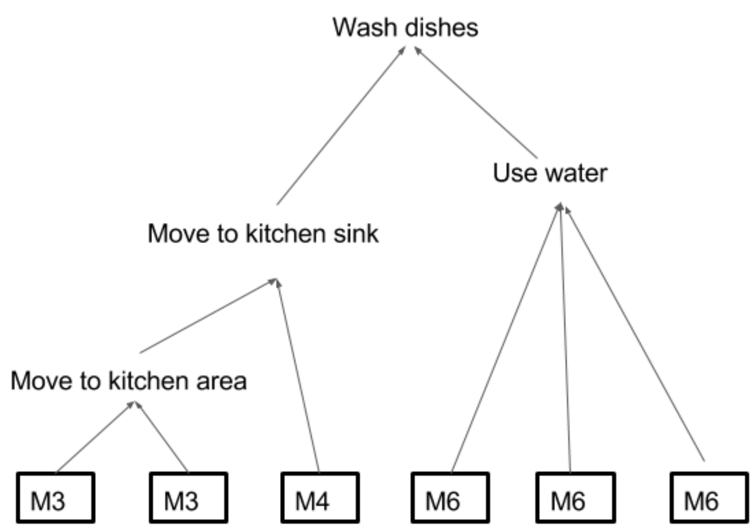


Figure 1: Activity *washing dishes* and its sub-activities.

The main contributions of the present paper are as follows:

1. Given the difficulty of collecting a real-world dataset, a method is proposed to simulate the abnormal behaviour of elderly people with dementia.
2. Instead of relying on features which ignore the granular level details of sensor activations, we exploit raw sensor activations to encode sub-activities. This representation is helpful in encoding activation frequency and ordering of sensors better.

3. We consider the problem of activity recognition as a graph labelling problem and exploit GCNs to model activities based on their fine-grained sensor activations. Modelling activities in a graph makes it possible to encode the activities' intrinsic sub-structures. Then abnormal behaviour related to dementia are detected exploiting the nodes and their relationships in the graph.

The rest of the paper is organised as follows (see Figure 2). Section 2 summarises the related work in the area. Section 3 describes the dataset used and explains the simulation of dementia related abnormal behaviour. Section 4 presents sensor representation and Graph Convolutional Networks (GCNs) to detect abnormal activities. Section 5 describes the experimental set-up, the experimental evaluations along with a discussion. Finally, Section 6 concludes the paper.

2. Literature Review

Currently, expert knowledge is required to assess the cognitive status of elderly people. In-person examinations of experts include questionnaires, recall of events or brief snap-shots of function which have some limitations such as their episodic nature, and possible biased reporting. For example, in [23], elderly people are asked to complete a sequence of scripted actions while being monitored via Web camera. In [24], sensor based features such as the duration of the activity and the number of sensors triggered are fed into some machine learning algorithms to do cognitive status assessment. However, a brief description for each task is provided to the participants. Our method does not impose aid, so activities are done on the fly without any intervention.

In [10], a Markov Logic network based hybrid technique is used to detect abnormal behaviour of elderly people. The method includes supervised learning, rule-based reasoning and probabilistic reasoning. However, steps of each action are defined prior to the construction of the model. The inference engine evaluates the rules such as taking a medicine that was not prescribed. These rules are extracted from Mild Cognitive Impairment (MCI) indicators. These rules however depend on the home environment, sensors and the particular habits of the person. Rule-based systems rely on experts to manually add specific rules to the system, since daily life routines change for each person.

In [25], the authors introduce *activity curves* which model daily activity routines into individuals. Abnormal behaviour is detected by comparing the *activity curves* against the actual behaviour. In [26], the authors use an abstraction layer to create a common ground for home sensor configurations. Then, a probabilistic spatio-temporal model is built to summarise daily patterns. The probabilistic model defines a likelihood of the person's activity based on the location at each hour of the day. Abnormal behaviour, such as not leaving the bed for a long time or not going to the bedroom for sleeping during the night, is detected using a cross-entropy measure. In [27, 5], the authors exploit Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) to detect abnormal behaviour stemming from dementia in a daily living scenario. However, their study fails to capture the intrinsic structure of activities and cannot detect anomalies occurring at low-level since they rely on fixed length features ignoring the sensor activation relationship and ordering.

In some studies, the assessment is done by attaching motion sensors on kitchen utilities and observing their usage frequency and time [28, 29]. In [21], the authors exploit Hidden Markov Model (HMM) and fuzzy rules to detect duration, time and frequency related anomalies. In [30], behavioural patterns of the residents are extracted using Bayesian statistics. These patterns are used to detect abnormal behaviour that potentially indicates deviations in cognitive status of the person. In [31], a Markov chain model is used to model the daily routines based on historical data. An entropy rate is calculated to detect the unusual patterns of people with dementia in their day-to-day life.

In [32], 3D Convolutional Neural Networks (CNNs) are applied on brain structural MRI scan data to predict the individual diagnosis of Alzheimer's disease (AD) and mild cognitive impairment. Moreover, data augmentation methods, such as rotation, flipping and scaling, deformation and cropping, are used to reduce overfitting in CNNs by providing more training and validation examples. In [33], the limitation of training data on brain MRI data is tackled by introducing dense connections to 3D-CNN. Also a probability-based fusion method was used to combine the base classifiers of 3D-CNNs.

When there is no real world data available, data simulation can be an alternative [21, 22, 28, 30]. In [21], the authors modified a real-world dataset to synthesise health related abnormal behaviour for their experiments. Eight daily activities such as sleeping, waking up, walking, eating are chosen and health related abnormal behaviour like frequent toilet visit, no exercise, slept without dinner are synthesised. In [22], more data is synthesised using Hidden Markov Models (HMMs) based on a small set of real data collected. To increase the realism of data simulation, the sensor events were modelled by a combination of Markov chains and the Poisson distribution. However, in both [21, 22], it is not mentioned in detail how the data synthesis was done. In [28], the authors modified a real-life data set of an older adult converting basically the rooms into activities. The authors focused on walking and eating in conjunction with the sleeping activity and samples of these activities are manually inserted in the XML data set. In [30], abnormal sensor readings are manually identified by a trained expert. A real dataset is taken as a base and synthetic errors were generated. In [4], Recursive Auto-Encoders (RAE) are used to cope with the scarcity of data. The authors demonstrate the idea of using transfer learning When there is limited data available. They learn "normal" behaviour in a source household, and then transfer the parameters of a RAE to another house (source) to detect abnormal behaviour of dementia sufferers.

Graph-structured data can be found in different domains such as chemistry, natural language semantics, social networks, and knowledge bases [18, 19, 20]. Graph convolutions have been widely used to learn high-level features by considering spatio-temporal relationships among nodes of a graph. In [34], graph kernels are used to embed meaningful local neighbourhoods of the graphs in a continuous vector space. In [35], the authors represent graphs as multi-channel image-like structures that allow them to be handled by 2D Convolutional Neural Networks (CNNs). In [36], molecules are represented as an undirected Graph Convolution Networks (GCNs) to predict molecular properties. Moreover, in anomaly detection literature, graph-based methods are preferred when there is inter-dependent data since graphs can offer pow-

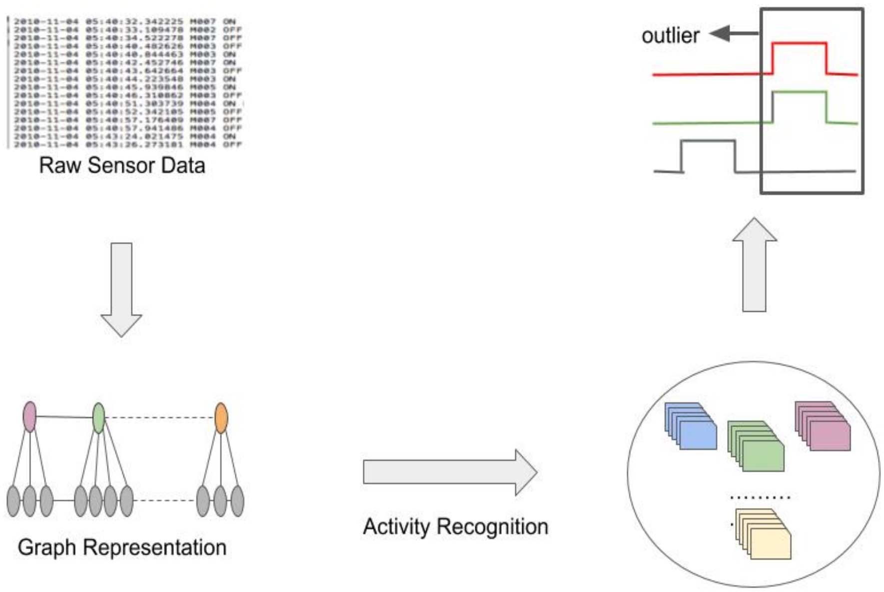


Figure 2: Overview of the proposed method.

erful representation abilities [37, 13, 38]. The most similar approach to ours is [38], where motion sensors in a smart home are represented as nodes in a graph and resident's movements are encoded as edges. Then graph-based features are extracted and used as input for an SVM. If sensor A is activated after sensor B, then there is an edge between the corresponding nodes. Each triggered edge is considered in the data as a feature. If an edge exists in the graph for a current activity, then corresponding edge attributes are represented as a value in the feature vector. However, in our case instead of taking sensors as nodes, we encode each activation as a node, which allows us to capture the ordering of sensor activations.

3. Dementia-driven Data Generation

3.1. Dataset

Aruba test-bed of CASAS smart home [39] is chosen to evaluate the proposed method and compare it with state-of-the-art methods. The data in this test-bed is collected by using 3 door, 31 motion and 5 temperature sensors over 224 days. However, in this paper we use only door and motion sensors in our study since they are useful in the context of dementia. The data collected is represented as a list of sensor name and time-stamp measurements (see Table 1). The sensors used has only binary states, namely OFF and ON status. The activities in this dataset are the daily life activities that can be used to simulate abnormal behaviour in daily life routines of elderly people suffering from dementia. These 11 activities are, *Meal Preparation*, *Relax*, *Eating*, *Work*, *Sleeping*, *Wash dishes*, *Bed to toilet*, *Enter home*, *Leave home*, *Housekeeping* and *Respirate*. They are performed by an adult and they don't include any abnormal ones. Thus, we need to modify some of these activities to generate some artificial abnormal behaviour.

Table 1: Sample of sensor reading data.

Date	Time	Sensor ID	Sensor Status
2011-04-01	01:16:10.814699	M004	ON
2011-04-01	01:16:11.429192	M007	ON
2011-04-01	01:16:16.462383	M004	OFF
2011-04-01	01:16:16.599859	M005	ON
2011-04-01	01:16:19.899843	M003	ON
2011-04-01	01:16:22.102316	M005	OFF

3.2. Augmented Data

In this study, two types of abnormal behaviour of elderly people are tackled: 1) *activity* related anomalies and 2) *sub-activity* related anomalies. In *activity* related anomalies, an activity itself is totally normal while there is an anomaly related to its frequency or its timing in a day. On the other hand, *sub-activity* related anomaly is related to the context and the quality of activity performed such as frequency of sensor activations involved as well as their order and correlation. In the first one, activities as a whole are repeated or forgotten (e.g. having dinner); while in the second one, some steps (sensor activations) of activities are forgotten or repeated (e.g. adding salt to a dish).

3.2.1. Activity Related Abnormal Behaviour

To simulate this type of indicators, we insert a specific set of activities within a sequence of daily activities (see Algorithm 1). This will give multiple occurrences of the same activity. Moreover, insertion in some inadequate time of the day will generate time-related abnormality such as having dinner in the middle of the night. We insert the instances of the following activities: *preparing meal, eating, working, washing dishes, leaving home, entering home* into the normal activity sequences.

We simulate sleep disorders and night time wandering anomalies by inserting some activities in the normal night-time activity sequences of a person. More specifically, we insert *eating, bed to toilet, resperate* into the *sleeping* activity of normal activity sequences. This will reflect on the abnormal behaviour such as having a drink and going to the toilet frequently in during the night.

Input: A sequence S of sensor activations in a day such as

$S = < s_1, s_2, \ldots, s_n >$ where each s_i is a sensor activation.

An activity $A = < a_1, a_2, \ldots, a_m >$ where each a_j is a sensor activation. `/*` A `is chosen specially (e.g.` *eating*`) to reflect a dementia related abnormal behaviour. */`

Output: $S = < s_1, s_2, \ldots, s_l, a_1, a_2, \ldots, a_m, s_{l+1}, \ldots, s_n >$

while *true* **do**

    Choose a random position l in S;

    Insert A into S at position l;

end

Algorithm 1: Simulation of *activity* related abnormal behaviour

As described in Algorithm 1, all insertions are done randomly. First a random instance of a given activity type (for example, *meal preparation*) from whole dataset is chosen, and then it is injected in a random position. Note that these activities are totally normal on their own, but become abnormal when they occur at a wrong time of the day or after/before a specific activity. In all, we generate 77 abnormal activity instances. A set of modified abnormal behaviour is depicted in Table 2.

3.2.2. Sub-activity Related Abnormal Behaviour

This kind of abnormal behaviour is generated by repeating specific sensor activations in a given activity. For this purpose, given random instances of *working, eating, meal preparation, bed to toilet*, we randomly insert specific sensors ($M_{26}, M_{14}, M_{18}, M_4$ respectively) involved in these activities (see Al-

Table 2: Examples of abnormal behaviour. The following abbreviations are used for the activities. *S: Sleeping, M: Meal preparation, E: Eating, R: Respirate, W: Working, B: Bed to toilet.* The inserted activities are shown in bold. T refers to the type of abnormal behaviour. For *sub-activity* related abnormal behaviour, because of space problem, only a subset of sensor activations are shown.

T	Original	Modified	Abnormality
Activity	S - M - E	S - **B** - S - M - E	Sleep disorder
	S - M - E	S - **R**- S - M - E	Sleep disorder
	S - M - E	S - **R** - S - M - E	Sleep disorder
	S - M - E	S - **M** - S - M - E	Repetition
	S - M - E - H	S - M - E - H - **E** - **E**	Repetition
Sub-activity	W: M_{26}, M_{28}, M_{27}	M_{26}, M_{28}, $\boldsymbol{M_{26}}$, $\boldsymbol{M_{26}}$, M_{27}	Confusion
	B: M_4, M_5, M_7	M_4, $\boldsymbol{M_5}$, $\boldsymbol{M_5}$, M_7, $\boldsymbol{M_5}$	Confusion
	W: M_{18}, M_{20}, M_{15}	M_{18}, $\boldsymbol{M_{15}}$, $\boldsymbol{M_{15}}$, M_{20}, M_{15}	Confusion
	E: M_{14}, M_{19}, M_{18}	M_{14}, $\boldsymbol{M_{14}}$, $\boldsymbol{M_{14}}$, M_{19}, M_{18}, $\boldsymbol{M_{14}}$	Confusion
	M : M_{18}, M_{19}, M_{15}	M_{18}, $\boldsymbol{M_{18}}$, $\boldsymbol{M_{18}}$, M_{19}, M_{15}, $\boldsymbol{M_{18}}$	Confusion

gorithm 2). For example, for *working* activity, the sensor M_{26} is repeated more than usual to emulate the usage of a computer (see Figure 3).

```
2011-04-01 08:42:35.982779  M026    ON  Work
2011-04-01 08:42:37.732557  M028    OFF
2011-04-01 08:42:37.732557 M26 ONabn
2011-04-01 08:42:41.771143  M027    ON
2011-04-01 08:42:44.654344  M027    OFF
2011-04-01 08:42:49.308347  M026    OFF
2011-04-01 08:42:49.308347 M26 ONabn
2011-04-01 08:42:49.686528  M026    ON
2011-04-01 08:42:49.686528 M26 ONabn
2011-04-01 08:42:56.781314  M026    OFF
2011-04-01 08:42:56.781314 M26 ONabn
2011-04-01 08:42:59.231909  M026    ON
```

Figure 3: A snapshot for *sub-activity* related abnormal behaviour synthesis. $ONabn$ shows the inserted sensor activations.

Input: A sequence of S of sensor activations of an activity A such as

$S = < a_1, a_2, \ldots, a_n >$ where each a_i is a sensor activation

A sensor type M that occurs in activity A.

`/*` A `and` M `are chosen specially (e.g. sensor` M_6 `in` *working* `activity) to reflect a dementia related abnormal behaviour. */`

Output: $S = a_1, M, a_2, M, a_3, \ldots, M, a_n$

while *true* **do**

  Choose a random location l in S

  Insert M into S at l

end

Algorithm 2: Simulation of *sub-activity* related abnormal behaviour.

4. GCN-based Abnormal Behaviour Detection

In this paper, we aim to detect abnormal behaviour reflecting the cognitive status of elderly people exploiting GCNs. For this purpose, the following steps of Figure 2 are applied: 1) Raw sensor data is represented as a graph. 2) Graph convolution network (GCN) is trained to model activities. 3) Abnormal activities are detected using GCN.

4.1. Sensor Representation

Current studies take all sensor activations triggered at a given time and put them in a vector of fixed length N, where N is the total number of sensors in the dataset. Thus, each sensor is represented only once ignoring the frequency and the order of activation in that given time. This representation resembles to bag-of-words representation in textual document representation, thus we name it as Bag-Of-Sensors (BOS). In this study, each sensor activation coming from raw data will be represented as a node in the graph. Firstly, a graph is constructed as in Figure 4. There are two node types in our graph, namely *inner* nodes and *outer* nodes. Each *inner* node in the graph represents a sensor activation from input data. These nodes have one-hot encoding of their sensor activations. For example, if we consider the one-minute piece of raw sensor data in Table 1. It is mapped into raw sensor data of M_4, M_7, M_5, M_3, if we only consider ON status of each activation. Then there will be 4 inner nodes in this graph, where each inner node represents a sensor activation. For example, the first inner node M_4 will have one-hot encoding feature of 1×34, where the index at 4 is 1 while others are 0. These inner nodes will have dummy labels as they don't have any activity labels of their own. Considering that there are 11 activities in our dataset and 1 activity for *nil* activity, labels are represented by a vector of size 1×12. For example, the nodes having label 2 will have the label vector 0100000000000. For inner nodes, all values of label vector will be zero, meaning they don't have any labels. Inner nodes are connected to their subsequent ones as shown in Figure 4.

Outer nodes represent time-slice activity labels of sensor activities. *Inner* nodes which fall within a certain t-minutes time-slice (e.g. 1 minute) are merged with their *outer* node. Thus, there is only one *outer* node for each t-minutes sensor activations. Here, note that the number of sensor activations (the number of *inner* nodes) is arbitrary. An *outer* node will have its corre-

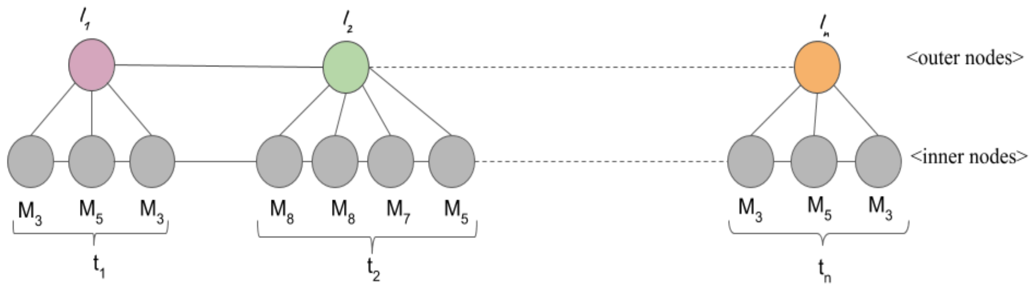


Figure 4: Graph structure constructed for sensor activations. The nodes in the upper line represent *outer* nodes, while the lower nodes are *inner* nodes.

sponding time-slice's activity label. However, an *outer* node has an empty feature vector of the same size $1 \times N$. *Outer* nodes are connected to the subsequent *outer* nodes and to their corresponding *inner* nodes. If whole data is split into n time-slices, then there are n *outer* nodes in the constructed graph. The data (*outer* nodes) is split into 3 subsets: training, testing and validation. We group *inner* nodes within one-minute chunks which is chosen experimentally.

4.2. Graph Convolutional Networks

CNNs can extract their own features from the raw data. CNNs are currently the state-of-the-art method for many problems in the literature. However, irregular data such as graphs cannot be handled by CNNs since CNNs require a fixed dimension of input. Recently, there has been a growing interest in GCNs to apply the same convolution idea on graph-structured data [40, 18, 19, 20]. The convolution is done on the spatial neighbourhood of a graph network.

Inspired by the solutions offered by GCNs, in this study, we use the GCN model proposed by [40], which obtains a linear-time graph-CNNs of spectral graph convolutions by using a convolutional architecture via a localised first-order approximation in Fourier-domain. In GCNs, the hidden layers serve as a kernel and calculate feature maps. They encode graph structure and features coming from nodes and their neighbours to extract fruitful features.

In a graph structured neural network, $f(X, A, L)$, $f(.)$ is a differentiable function like a neural network, X is a matrix of feature vectors and A is an adjacency matrix and L is one-hot encoding labels of instances. $A \in \mathbb{R}^{N \times N}$ where N is the number of nodes in the graph, $X \in \mathbb{R}^{N \times F}$ where F is the number of features. Before calculating the hidden layer activations, a degree matrix D is calculated where $D_{ii} = \sum_j A_{ij}$. Then, self-connections of nodes are added to the graph adjacency matrix A so that $\tilde{A} = A + I_N$, where I_N is the identity matrix. Then the new degree matrix $\tilde{D}$ becomes $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. After these calculations, given $W^{(l)}$ and $H^{(l)}$, a layer-wise propagation rule is applied using activation function $\sigma(.)$. This activation function is Rectified Liner Unit (ReLu), where $ReLU(x) = max(0, x)$. Here, $W^{(l)}$ is a trainable weight matrix specific layer l and $H^{(l)} \in \mathbb{R}^{N \times M}$ is the matrix of activations of the l^{th} layer, where M is the number of hidden neurons in that layer. In the first layer, $H^{(0)} = X$ since the first layer is the input layer. Then the activations of hidden layer $l + 1$, $H^{(l+1)}$, is calculated in Equation 1.

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)}) \tag{1}$$

$$Z = f(X, A) = softmax(\hat{A}\ ReLu(\hat{A} X W^{(0)})\ W^{(1)}) \tag{2}$$

After calculating $\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$, a forward propagation rule is applied for a GCN which has two layers in Equation 2. Here, $W^{(0)} \in \mathbb{R}^{C \times H}$ is a weight matrix from input to hidden layer for a hidden layer with H feature maps (hidden neurons) and C input channels. $W^{(1)} \in \mathbb{R}^{H \times F}$ is a weight matrix from hidden to output layer.

The GCN weights $W^{(0)}$ and $W^{(1)}$ are optimised exploiting the gradient descent where full batch training is used in every iteration. As a loss function of the model, cross-entropy error is used as in Equation 3, where Y_L is the set of training nodes that have labels.

$$E = -\sum_{l \in Y_L} \sum_{f=1}^{F} Y_{lf} \, lnZ_{\,lf} \tag{3}$$

4.3. Activity Recognition and Abnormal Behaviour Detection

To recognise activities for *outer* nodes, firstly, a graph is constructed as shown in Figure 4. The sensor activations are represented as *inner* nodes, while the labels are represented as *outer* nodes (Section 4.1). The adjacency matrix A and feature matrix X are constructed based on the graph structure used to train the GCN on *normal* activities. To detect abnormal activities, we use the confidence probability values of assigned labels and then by using a threshold value, we decide if they are abnormal or not (see Algorithm 3).

Input: Time-slices in test set $< n_1, n_2, \ldots, n_j >$

where each n_i is represented as an outer-node in the graph.

GCN Classifier G

Threshold th

foreach n_i **do**

Classify n_i with G

Obtain classifier confidence probability p_i

if $p_i \geq th$ **then**

n_i is normal

else

n_i is abnormal

end

end

Algorithm 3: Abnormal behaviour detection

5. Experiments

5.1. Experimental Set-up

In order to evaluate the proposed GCN model, Aruba dataset (Section 3) is split into training and testing sets. The training dataset consists of the first 139 days. Next 15 days are used for validation and the remaining 70 days are used for testing. The split is done based on days to use the information coming from time-series data [29, 5]. Since there is no abnormal activities in the original data, the test set is modified as described in Section 3. The modifications are done separately for the abnormal behaviour types, *activity* and *sub-activity*, which result in two different test sets. We analyse the two anomalies separately to see the affect of GCNs on both anomalies individually.

Moreover, GCNs are compared with the following state-of-the-art supervised methods: LSTMs (Long Short Term variants of RNNs), Naïve Bayes (NB) classifier, Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs). In these classifiers, BOS representation is adopted since these methods require fixed-length input vectors ($1 \times N$), for N being the number of sensors in the dataset. Each sensor activated in 1 minute is represented as 1, while others are given 0.

Keras Deep Learning library's [41] and Theano's [42] implementation of LSTM and CNN is used in this study. NB, HMM and CRF are based on the implementation provided in [43]. In the LSTM and CNN related experiments, Adam optimiser [44] is used. The models and the parameters are decided experimentally as follows. In the LSTM related experiments, the instances are fed into the system with a batch size of 20. In LSTM, two hidden layers of 50 and 100 nodes are used. Then, dense layers of size 100, 128 and 50 are added to the network, followed by a softmax layer. There are drop-out layers with a probability of 0.5 between each two layers in LSTM models. As for CNN, time-series window of length 10 seconds is extracted from the raw sensor readings. The CNN model has the following layers: A 2D convolutional layer (with 20 kernels of size 5×10), a Max Pooling layer (with a pooling size of 2×2), a 2D convolutional layer (with 10 kernels of size 10×15), a Max Pooling layer (with a pooling size of 2×2), a flatten layer, and two dense layers of size 128 and 50, followed by a softmax layer to do the classification.

In activity recognition experiment, nodes are represented by two different features. In the first one, we only consider the ON status of activations, while in the second one we also consider the OFF status. The feature vector for ON representation is of size 1×34, where 34 is the total number of sensors in the dataset. The OFF status is represented by adding another 34 values to the feature vector, which results in a feature vector of size 1×68. For example, if the sensor M_3 has status ON, the one-hot encoding feature vector of 1×34 will have 1 at position 3, while if the same sensor has status OFF, then feature vector of 1×68 will have 1 at position 37 ($= 34 + 3$). Moreover, activations of GCN hidden neurons are fed into an LSTM network to carry temporal information further. Here, the same LSTM network is used as described above. Thus activity recognition experiments with GCN have 3 variants: 1) when only ON status of sensor activations are used, 2) when both ON and OFF are used and 3) when kernels' (hidden layers) activations of GCN are fed into an LSTM network.

5.2. Evaluation Metrics

Precision, recall, accuracy and F-measures are used to evaluate classifier performance. Precision and recall are calculated for each class separately and then the average is taken over all classes. It is important to use these particular measures because we are dealing with unbalanced datasets where some classes appear much more frequently than others. As classes are considered equally important, the average precision and recall are taken over all classes. The accuracy gives information about the percentage of correctly classified time-slices, resulting in a larger weight for more frequently occurring classes. Kasteren et. al. [43] gives a detailed description of these measures.

In order to assess the abnormal behaviour detection success, True Positive Rate (TPR) and False Positive Rate (FPR) are used. These values for different thresholds are depicted on a receiver operating characteristic (ROC) curve. Moreover, Area Under Curve (AUC) is calculated for each model to interpret the results in a better way. True Positive Rate (TPR) refers to the method's ability to correctly detect instances which are abnormal. FPR gives the percentage of mislabelled normal instances, thus reflects on the method's ability to differentiate between normal and abnormal activities.

GCN experiments are performed based on Kipf et. al.'s Python implementation [40] with raw data. Learning rate is set to 0.01, drop-out value to 0.5, weight decay to 0.00005 and the number of hidden neurons to 64. The training is done in 100 epochs with an early stopping of 10. The experiments are conducted in two parts: 1) Activity recognition and 2) Abnormal behaviour detection.

5.3. Activity Recognition Experiments

In these experiments, we only used the test set with *activity* related abnormal behaviours since our focus is on abnormal behaviour detection rather than activity recognition. Moreover, activity recognition results are very similar for both test sets.

Table 3 shows the activity recognition results. In particular, the best accuracy is retrieved by LSTM (85.95%), while GCN-LSTM comes after with a slight difference (85.67%) when ON and OFF activations are used. However, GCN model using ON and OFF without LSTM gives higher precision (52.25%) and recall values (50.86%). In terms of F-measure, LSTM-BOS achieves 43.29%, while GCN-LSTM achieves 43.11% and GCN with ON and OFF achieves 51.55%. This shows that GCN with ON and OFF status is good at differentiating classes. This comes from the ability of GCN to model activity slices by taking sensor activation relationships into account. Moreover, LSTM experiment is performed with BOS representation which ignores the relationship between sensor activations. Thus, BOS representation might affect the performance of LSTM as well. On the other hand, feeding GCN activations to LSTM causes a decrease in precision and recall (42.61% and 43.62%). The reason for this might be that LSTM learns the temporal information of the most frequent classes and gives more importance to them.

Table 3: Activity recognition results in percentages when *activity* related anomaly test set is used.

Model	**Precision**	**Recall**	**F-Measure**	**Accuracy**
Dense (ON)	3.86	9.09	5.42	42.54
Cheby (ON)	44.96	44.73	44.85	79.45
GCN (ON)	46.65	49.77	48.16	84.01
GCN (ON+OFF)	**52.25**	50.86	51.55	84.06
GCN-LSTM (ON+OFF)	42.61	43.62	43.11	85.67
NB-BOS	44.45	69.09	54.10	74.59
HMM-BOS	40.21	**77.47**	52.94	72.97
CRF-BOS	44.46	47.42	45.90	80.39
CNN-BOS	38.81	43.18	40.88	80.68
LSTM-BOS	41.93	44.73	43.29	**85.95**

Although GCN with ON and OFF achieves slightly the same accuracy (84.06%) with its ON version (84.01%), it ends up with better F-measure (51.55% compared to 48.16%). Since F-measure is averaged on each class, this means that adding OFF status helps to differentiate the classes better. When OFF status is used, the model can understand the ordering of the sensor activations better. This ordering provides an update for the location of the person since sensors are activated one after another based on the location of replaced sensors . This gives more insight to the activity being performed. For example, results with only ON status show that *leave home* activity is confused with *enter home* activity, while *eating* activity is confused with *meal preparation* since the same sensor types are involved. However, adding OFF status improves the accuracy for these activities since it reduces the confusion. No method can identify *Wash dishes* well and most of the test instances are recognised as the *Meal preparation* activity. The reason is that these activities occur in the kitchen and use very similar objects.

Despite the high ability of CNNs to capture spatial context, CNN model achieves relatively less accuracy (80.68%) because of the BOS representation. Moreover, CRF model outperforms NB and HMM in terms of accuracy (80.39%, 74.59% and 72.97% respectively) but NB and HMM perform better in terms of recall (69.09% and 77.47% respectively). The reason is that CRF favours the most frequent class in the dataset.

Also we compared different variants of per-layer propagation model [40], namely Chebyshev filter and Multi-layer perceptron. In Chebyshev filter, the propagation model is updated using a Chebyshev filter, while in multi-layer perceptron, the propagation model is changed with a dense perceptron [40]. As seen in Table 3, GCN gives the best accuracy compared to the others.

Moreover, a hyper-parameter sensitivity analysis is performed to show the effect of individual parameters. For this experiment, the default parameters are set to the ones in [40], and then a value of only one hyper-parameter (drop-out, weight decay, learning rate, number of hidden neurons) is changed. The following set of hyper-parameters are analysed; drop-out $0.1, 0.2, 0.3, 0.4, 0.5$, weight decay $0.05, 0.005, 0.0005, 0.00005, 0, 000005$, learning rate $0.1, 0.01, 0.001$, number of hidden neurons $20, 16, 12, 8, 4$. As

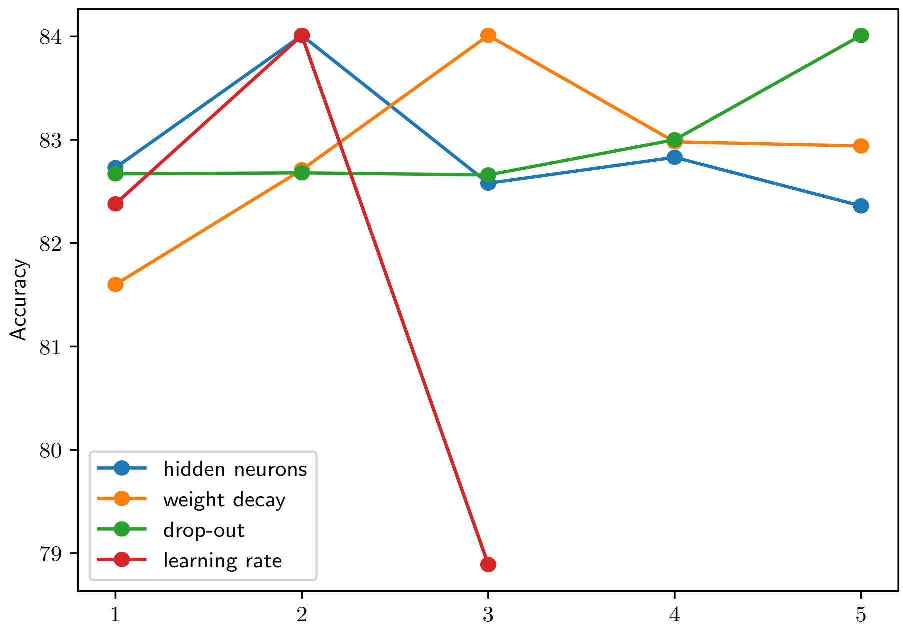


Figure 5: Hyper-parameter sensitivity analysis.y-axis shows accuracy while x-axis shows different parameters.

seen on Figure 5, the hyper-parameters set by [38] are the optimum ones for this task. Therefore, we have continued our experiments with this set of hyper-parameters.

5.3.1. Sensor Pattern Extraction

A quantitative analysis is provided for each activity class to show which sensors are involved the most during the classification decision of GCN. For this purpose, the average of all GCN hidden layer activations is calculated for each feature of each activity. The normalised sensor activations are visualised in Figure 6. The following abbreviations are used for the activities. *N: Nil, S:*

Sleeping, M: Meal preparation, E: Eating, R: Respirate, W: Working, B: Bed to toilet. For example the *outer* nodes of *meal preparation* activity correlate more with sensor M_{19}, while nodes of *eat* activity are affected by M_{14}, and *resperate* nodes are affected by M_{25}. *Wash dishes* nodes are affected by M_{15}, M_{18}, M_{19}, *bed to toilet* nodes by sensors M_3, M_4, M_5, M_6 and *leave home* and *enter home* by M_{30} and D_4 while *work* nodes gets excited by M_{26} and M_{27}. These are the sensors which are involved at most in these activities when we look at the sensor lay-out and raw sensor measurements.

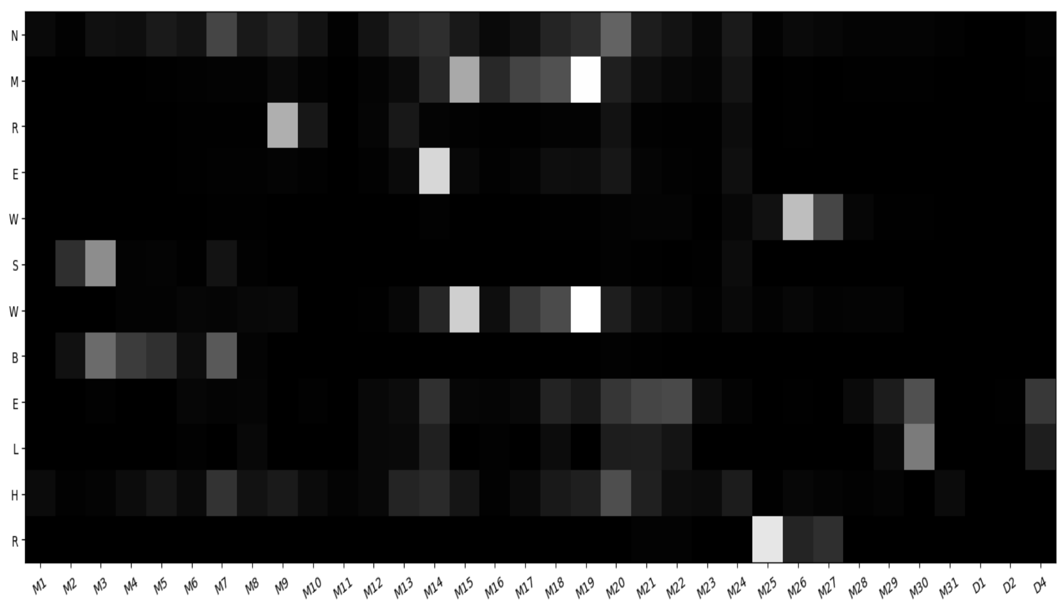


Figure 6: Sensor activation map for each activity. The lighter cell means that that activity nodes get more excited by that sensor.

Figure 7 presents a set of example sub-graphs from each activity category. For example, *meal preparation outer* node is affected by sensors M_{15}, M_{17}, M_{18}. Moreover, we extract the most common and important patterns for each activity class in the graph. This is similar to frequent sub-

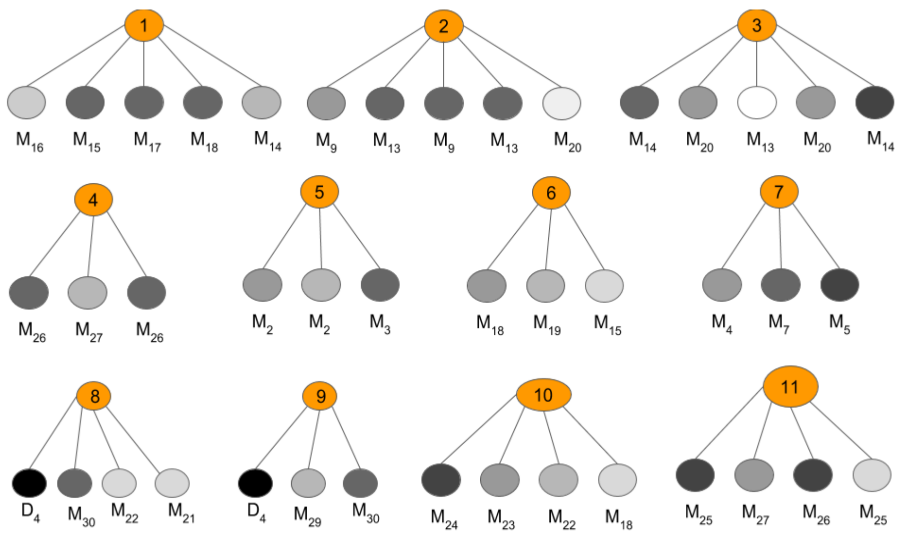


Figure 7: Sub-graph from each class showing the level of sensor activations. The darker colour means more excitement. The numbers on *outer* nodes show the class label of that time-slice (*outer* node). The classes are numbered in the following order: *Meal Preparation, Relax, Eat, Work, Sleep, Wash Dishes, Bed to toilet, Enter Home, Leave Home, Housekeeping, Resperate.*

graph mining, which is about discovering interesting patterns in graphs. In our case, sensor readings which are triggered frequently in an activity represent a pattern. If activation scores of a group of sensors are high, this means that there is an *outer* node which gets excited by that combination of sensors. For this purpose, all n-grams are calculated for *inner* nodes and their average sensor activation scores are calculated. Then all activations are sorted, and the $top - 500$ nodes, which is decided empirically, are taken for each activity class to calculate n-gram patterns. We calculate n-grams with only $n = 2$ and $n = 3$, which is enough to see the patterns in the dataset. A sample set of extracted patterns is shown in Table 4. For example, the sensors M_{19} and M_{15} are grouped in *meal preparation* activity. In the sensor layout, these sensors are placed close to each other and when the resident

Table 4: N-gram patterns.

Activity	**2-gram**	**3-gram**
Bed to Toilet	M_3, M_3 M_4, M_7 M_7, M_5	M_4, M_75, M_5 M_7, M_5, M_3
Meal Preparation	M_{19}, M_{15} M_{18}, M_{19} M_{19}, M_{17}	M_{15}, M_{19}, M_{19} M_{18}, M_{19}, M_{15} M_{17}, M_{19}, M_{15}
Relax	M_9, M_{20} M_9, M_9 M_9, M_{13}	M_9, M_9, M_{13} M_{10}, M_{10}, M_{10} M_9, M_{13}, M_{20}
Eating	M_{14}, M_{18} M_{14}, M_{14} M_{14}, M_{20}	M_{14}, M_{14}, M_{20} M_{24}, M_{14}, M_{14} M_{14}, M_{14}, M_{18}
Work	M_{26}, M_{26} M_{26}, M_{27} M_{22}, M_{28}	M_{26}, M_{27}, M_{26} M_{27}, M_{27}, M_{26}
Sleeping	M_3, M_2 M_3, M_3 M_2, M_3	M_3, M_2, M_3 M_7, M_3, M_7
Wash Dishes	M_{19}, M_{15} M_{18}, M_{19} M_{19}, M_{17}	M_{19}, M_{15}, M_{19} M_{15}, M_{15}, M_{15} M_{18}, M_{19}, M_{15}
Housekeeping	M_{20}, M_{20} M_7, M_5 M_{20}, M_8	M_{24}, M_{24}, M_{24} M_7, M_7, M_7
Leave Home	D_{24}, M_{30} M_{30}, M_{30} M_{22}, M_{30}	D_4, M_{30}, M_{30} M_{21}, M_{22}, M_{30}
Enter Home	D_4, M_{30} M_{22}, M_{21} M_{30}, M_{22}	D_4, M_{30}, M_{29} M_{30}, M_{22}, M_{21}
Resperate	M_{27}, M_{25} M_{25}, M_{26}	M_{25}, M_{25}, M_{26} M_{25}, M_{25}, M_{25}

performs *meal preparation* activity, these sensors are triggered one after another. The pattern constructed by these two sensors are identified in [13] as *near the kitchen range and sink*. Another grouping of sensors, namely M_{18}, M_{19}, M_{15} shows another pattern in this activity, which is again (M_{18} and M_{19}) found as a movement pattern in [13]. In *eating* activity, we see that M_{14} and M_{18} represent a pattern and M_{14}, M_{13}, M_{15} represent another pattern which is constructed by the sub-pattern M_{13}, M_{15} and the sensor

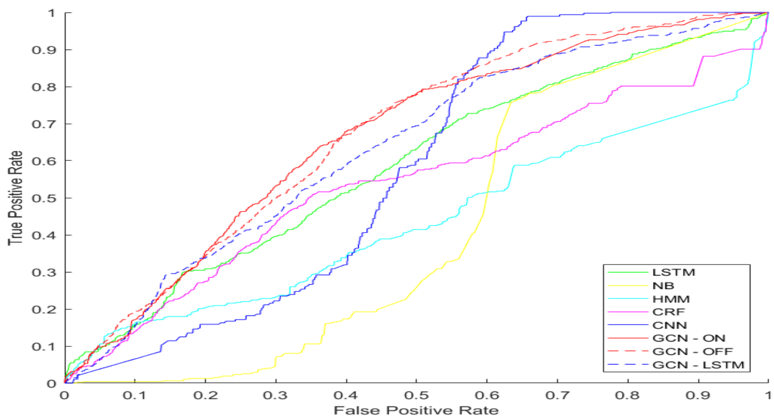
M_{14}. For the activity *sleeping*, the most frequent pattern is M_2, M_3. This makes sense because these sensors are on the bed and they will be triggered one after another during *sleeping* activity.

5.4. Abnormal Behaviour Detection

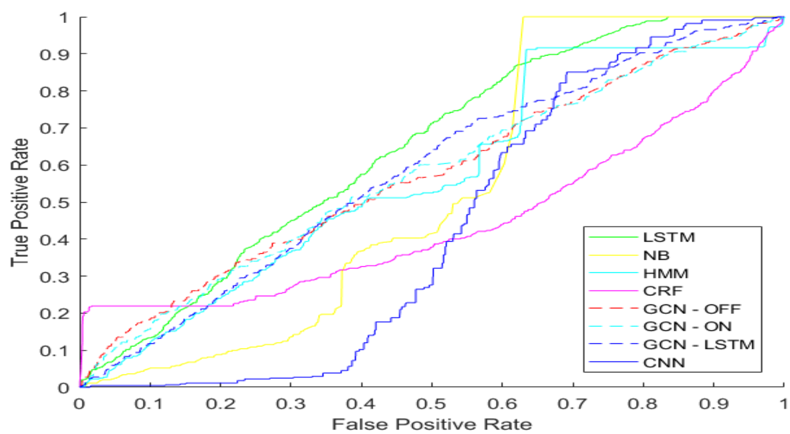
The abnormal behaviour detection results are visualised for *activity* and *sub-activity* related modified test sets separately on ROC curves in Figure 8(a). Moreover, AUC bars of ROC curves are shown in Figure 9(a). For *activity* related abnormal behaviours, the results show that GCN with only ON and with ON and OFF achieves the best with AUC of 66% and 67% respectively. Adding LSTM layer to the activations of GCN (with ON and OFF) reduces the AUC rate slightly to 63%. The reason for this might be that LSTM encodes temporal information further and tolerates small variations in the sequence. Adding OFF status to GCN model doesn't improve the result that much (around 1%), since *activity* related abnormality doesn't occur at sensor activation level.

Although CNN and LSTM are powerful models, they perform slightly worse than GCN models (with AUC of 57% and 59% respectively), since they rely on BOS representation. However, CNN catches GCN models at TPR (82%) and FPR (56%) on ROC curve. NB performs the worst because of its simple modelling capabilities. HMM and CRF, with AUC of 42% and 54%, doesn't perform well because of BOS representation.

Results for anomaly detection with *sub-activity* related test set are shown in Figures 8(b) and 9(b). LSTM performs the best AUC (63%), since inserting additional sensor activations (see Section 3), changes BOS representations. Thus, the activations mistakenly appear and LSTM can detect these



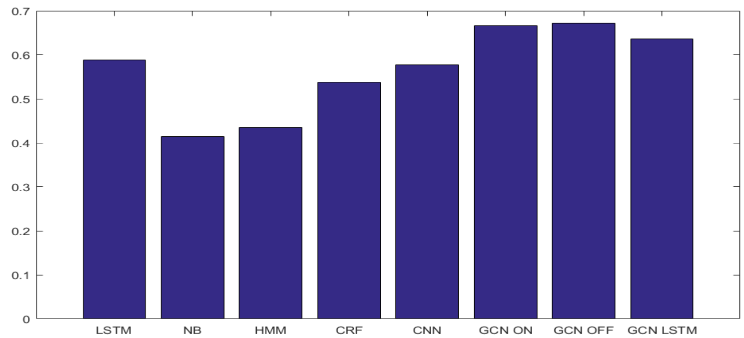
(a) ROC for anomaly detection when *activity related* test is used.



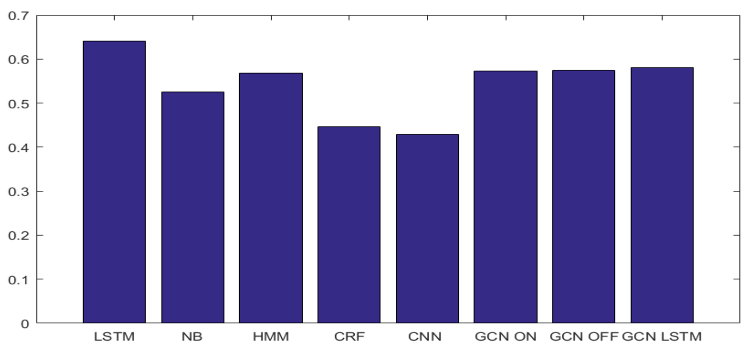
(b) ROC for anomaly detection when *sub-activity related* test is used.

Figure 8: ROC curves for anomaly detection.

changes by encoding temporal information. GCN-LSTM with ON and OFF produce AUC rate of 57% while GCN with only ON and GCN with both ON and OFF have similar AUC rate 56%. Thus, adding LSTM to GCN with ON and OFF improves the results reaching an FPR of 73% and TPR of 58% on ROC. Although, adding OFF helps differentiating between classes



(a) AUC for abnormal behaviour detection when *activity related* test is used.



(b) AUC for abnormal behaviour detection when *sub-activity related* test is used.

Figure 9: AUC bars for anomaly detection.

in activity recognition, it also increases the noise to detect abnormal activities. NB, with AUC of 52%, doesn't perform well since it is a simple model and can relate neither temporal nor spatial information. Although HMM can relate previous input to the current one, it achieves an AUC of 54%. As a discriminate model CRF performs only an AUC of 54%.

Moreover, in health-care problems, missing an anomaly - a true positive (TP)- may cause more serious problems than retrieving a high number of false positives (FP). Although our model fails to prune a high amount of false positives, it is good at detecting abnormal behaviour (high TPR), which is more important in the context of health-care. Moreover, the imbalanced data issue is the nature of anomaly detection tasks, where the number of normal instances dominates the number of abnormal instances. Thus, the model learns the normal behaviour better than abnormal behaviour, and fails to prune a good amount of FPs.

6. Conclusion

This paper presented a method to detect abnormal behaviour that stem from cognitive difficulties of elderly people. To cope with the scarcity of data, we proposed a data generation method to simulate abnormal behaviour reflecting cognitive status of elderly people with dementia. In contrast to the state-of-the-art, instead of relying on fixed length feature vectors, we exploited raw sensor measurements to encode information derived from sensor activations such as frequency and order of activations. We represented raw sensor activations in a graph and used Graph Convolutional Networks (GCN) to recognise activities and detect abnormal ones. The results show that the proposed GCN-based method can flag abnormal behaviour, not only related to activity patterns, but also the ones related to the intrinsic structure of activities in the context of cognitive decline. However, this method cannot relate one instance to another and neglects temporal information between time-slices. In the future, we will combine Recurrent Neural Networks (RNNs) with GCNs so that we can take both temporal and granular level information into account. Unfortunately, current simulation method does not reflect on the gradual decline in cognitive status. Moreover, our current approach may fail to detect abnormal behaviour arising from gradual deterioration regarding the cognitive status of an elderly person. As a future work, we are planning to collect real-world data reflecting gradual deterioration in cognitive status of an elderly person and then adapt our method to detect this kind of abnormal behaviour.

References

[1] The state of the art of dementia research: New frontiers 2018 (2018) 34–35.

[2] K. Wild, Aging changes, in: Geraotechnology, Vol. 9 No 2, 2010, pp. 121–125.

[3] P. Urwyler, R. Stucki, L. Rampa, R. Muri, U. Mosimann, T. Nef, Cognitive impairment categorized in community-dwelling older adults with and without dementia using in-home sensors that recognise activities of daily living, Scientific Reports 7:42084.

[4] D. Arifoglu, A. Bouchachia, Abnormal behaviour detection for dementia sufferers via transfer learning and recursive auto-encoders, in: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2019, pp. 529–534.

[5] D. Arifoglu, A. Bouchachia, Detection of abnormal behaviour for dementia sufferers using convolutional neural networks, Artificial Intelligence in Medicine 94 (2019) 88 – 95.

[6] M. Amiribesheli, A. Bouchachia, Smart homes design for people with dementia, in: 2015 International Conference on Intelligent Environments, 2015, pp. 156–159.

[7] S. Bhat, U. R. Acharya, Y. Hagiwara, N. Dadmehr, H. Adeli, Parkinson's disease: Cause factors, measurable indicators, and early diagnosis, Computers in Biology and Medicine 102 (2018) 234 – 241.

[8] C. Dhiman, D. K. Vishwakarma, A review of state-of-the-art techniques for abnormal human activity recognition, Engineering Applications of Artificial Intelligence 77 (2019) 21 – 45.

[9] P. Dawadi, D. Cook, M. Schmitter-Edgecombe, Smart home-based longitudinal functional assessment, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, 2014, pp. 1217–1224.

[10] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, R. Helaoui, Fine-grained recognition of abnormal behaviours for early detection of mild cognitive impairment, in: 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2015, pp. 149–154.

[11] T. Kirste, A. Hoffmeyer, P. Koldrack, A. Bauer, S. Schubert, S. Schröder, S. Teipel, Detecting the effect of Alzheimer's disease on everyday motion behaviour, Journal of Alzheimer's Disease (2014) 121–132.

[12] U. Naeem, R. Bashroush, R. Anthony, M. Azam, A. Tawil, S. Lee, M. Wong, Activities of daily life recognition using process represen-

tation modelling to support intention analysis, International Journal of Pervasive Computing and Communications 11 (3) (2015) 347–371.

[13] T. Zhang, W. Fu, J. Ye, M. Fischer, Learning movement patterns of the occupant in smart home environments: an unsupervised learning approach, Journal of Ambient Intelligence and Humanized Computing 8 (1) (2017) 133–146.

[14] N. D. Rodríguez, M. P. Cuéllar, J. Lilius, M. D. Calvo-Flores, A fuzzy ontology for semantic modelling and recognition of human behaviour, Know.-Based Syst. 66 (1) (2014) 46–60.

[15] J. Rafferty, L. Chen, C. Nugent, Ontological goal modelling for proactive assistive living in smart environments, in: Ubiquitous Computing and Ambient Intelligence. Context-Awareness and Context-Driven Interaction, Springer International Publishing, 2013, pp. 262–269.

[16] T. V. Kasteren, B. Krose, Bayesian activity recognition in residence for elders, in: 3rd International Conference on Intelligent Environments, 2007, pp. 209–212.

[17] D. Cook, M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment, Methods Inf. Med. 48 (2009) 480–485.

[18] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R. P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, 2015, pp. 2224–2232.

[19] Y. Seo, M. Defferrard, P. Vandergheynst, X. Bresson, Structured sequence modeling with Graph Convolutional Recurrent Networks., CoRR abs/1612.07659.

[20] S. Vashishth, S. S. Dasgupta, S. N. Ray, P. Talukdar, Dating documents using graph convolution networks, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 1605–1615.

[21] M. Forkan, I. Khalil, Z. Tari, S. Foufou, A. Bouras, A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living, Pattern Recognition 48 (3) (2015) 628 – 641.

[22] J. Lundström, E. Järpe, A. Verikas, Detecting and exploring deviating behaviour of smart home residents, Expert Systems with Applications 55 (2016) 429–440.

[23] A. Seelye, M. Schmitter-Edgecombe, D. J. Cook, A. Crandall, Naturalistic assessment of everyday activities and prompting technologies in mild cognitive impairment, Journal International Neuropsychologly Soc. (4): (2013) 442–52.

[24] P. N. Dawadi, D. J. Cook, M. Schmitter-Edgecombe, Automated cognitive health assessment using smart home monitoring of complex tasks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 43 (6) (2013) 1302–1313.

[25] P. N. Dawadi, D. J. Cook, M. Schmitter-Edgecombe, Modeling patterns of activities using activity curves, Pervasive Mob. Comput. 28 (C) (2016) 51–68.

[26] O. Aran, D. Sanchez-Cortes, M.-T. Do, D. Gatica-Perez, Anomaly detection in elderly daily behaviour in ambient sensing environments, Human Behaviour Understanding: 7^{th} International Workshop (2016) 51–67.

[27] D. Arifoglu, A. Bouchachia, Activity recognition and abnormal behaviour detection with Recurrent Neural Networks, in: 14^{th} International Conference on Mobile Systems and Pervasive Computing, 2017, pp. 86–93.

[28] G. Virone, Assessing everyday life behavioural rhythms for the older generation, Pervasive Mob. Comput. 5 (5) (2009) 606–622.

[29] N. Suryadevara, S. Mukhopadhyay, R. Wang, R. Rayudu, Forecasting the behavior of an elderly using wireless sensors data in a smart home, Engineering Applications of Artificial Intelligence 26 (10) (2013) 2641–2652.

[30] F. J. Ordóñez, P. de Toledo, A. Sanchis, Sensor-based bayesian detection of anomalous living patterns in a home setting, Personal and Ubiquitous Computing 19 (2) (2015) 259–270.

[31] S. Enshaeifar, A. Zoha, A. Markides, S. Skillman, S. T. Acton, T. Elsaleh, M. Hassanpour, A. Ahrabian, M. Kenny, S. Klein, H. Rostill, R. Nilforooshan, P. Barnaghi, Health management and pattern analysis

of daily living activities of people with dementia using in-home sensors and machine learning techniques, PLOS ONE 13 (5) (2018) 1–20.

[32] S. Basaia, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo, M. Filippi, Automated classification of alzheimer's disease and mild cognitive impairment using a single mri and deep neural networks, NeuroImage: Clinical 21 (2019) 101645.

[33] H. Wang, Y. Shen, S. Wang, T. Xiao, L. Deng, X. Wang, X. Zhao, Ensemble of 3d densely connected convolutional network for diagnosis of mild cognitive impairment and alzheimer's disease, Neurocomputing 333 (2019) 145 – 156.

[34] G. Nikolentzos, P. Meladianos, A. J. Tixier, K. Skianis, M. Vazirgiannis, Kernel graph convolutional neural networks, CoRR abs/1710.10689. URL `http://arxiv.org/abs/1710.10689`

[35] A. J. Tixier, G. Nikolentzos, P. Meladianos, M. Vazirgiannis, Classifying graphs as images with convolutional neural networks, CoRR abs/1708.02218.

[36] J. Li, D. Cai, X. He, Learning graph-level representation for drug discovery, CoRR abs/1709.03741. `arXiv:1709.03741`. URL `http://arxiv.org/abs/1709.03741`

[37] L. Akoglu, H. Tong, D. Koutra, Graph based anomaly detection and description: A survey, Data Min. Knowl. Discov. 29 (3) (2015) 626–688.

[38] S. S. Akter, L. B. Holder, Activity recognition using graphical features,

in: 2014 13th International Conference on Machine Learning and Applications, 2014, pp. 165–170.

[39] D. J. Cook, A. S. Crandall, B. L. Thomas, N. C. Krishnan, CASAS: A smart home in a box, Computer 46 (7) (2013) 62–69.

[40] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, CoRR abs/1609.02907. `arXiv:1609.02907`.
URL `http://arxiv.org/abs/1609.02907`

[41] C. François, Keras, https://github.com/fchollet/keras (2015).

[42] T. D. Team, Theano: A Python framework for fast computation of mathematical expressions, arXiv e-prints abs/1605.02688.

[43] T. Van Kasteren, G. Englebienne, B. J. A. Kröse, Human activity recognition from wireless sensor network data: Benchmark and software, Activity Recognition in Pervasive Intelligent Environments (2011) 165–186.

[44] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980.