UiO : **University of Oslo**

Flávia Dias Casagrande

# Sensor Event and Activity Prediction using Binary Sensors in Real Homes with Older Adults

**Thesis submitted for the degree of Philosophiae Doctor**

Department of Informatics
Faculty of Mathematics and Natural Sciences

Department of Mechanical, Electronics, and Chemical Engineering
OsloMet – Oslo Metropolitan University

**2019**

*Series of dissertations submitted to the*
*Faculty of Mathematics and Natural Sciences, University of Oslo*
*No.* 1234

ISSN 1234-5678



Cover: Hanne Baadsgaard Utigard.
Print production: Reprosentralen, University of Oslo.

*To my boyfriend, my mom, and my dad. This achievement is ours.*

*Ao meu namorado, à minha mãe, e ao meu pai. Essa conquista é nossa.*

# Preface

This thesis is written for the Faculty of Mathematics and Natural Sciences at the University of Oslo for the degree of Philosophiae Doctor (Ph.D.). The research presented here has been conducted under the supervision of Associate Professor Evi Zouganeli and the co-supervision of Professor Jim Tørresen. The work has been funded by the Research Council of Norway, under the SAMANSVAR programme (247620/O70) and is a part of the interdisciplinary "Assisted Living Project – *responsible innovations for dignified lives at home for people with mild cognitive impairment or dementia*". The project involves researchers from the Department of Nursing and Occupational Therapy (OsloMet), Work Research Institute (Oslomet), Department of Mechanical, Electrical and Chemistry Engineering (Oslomet), the Norwegian Board of Technology, Oslo Municipality, the commercial partners Sensio AS and RoomMate AS, and external partners from the Karlsruhe Institute of Technology, the University of Exeter and the University of Bristol.

The thesis is a collection of six papers. They are presented here in chronological order. The common theme of the papers is sensor event and activity prediction using binary sensor data collected from homes of older adults. The papers are preceded by introductory chapters that relate them to each other and describe the motivation and background information for the work. The papers are jointly published with Associate Professor Evi Zouganeli and Professor Jim Tørresen.

## Acknowledgement

There are several people I would like to thank for their help and support during the period I have been working on my Ph.D.

To my supervisors Evi Zouganeli and Jim Tørresen for valuable advice and supervision during the past years, and for the opportunity of realising this project. A special thanks to Evi for the guidance during these years of assistance and endurance during very challenging moments working in an interdisciplinary team. Thank you also Jim for finding the time in your busy calendar and for the support, especially in these last months.

To the residents participating in the project and the housekeepers at the seniors' care unit Skøyen Omsorg+ for the amazing patience, goodwill, and friendliness. Your participation in this project was crucial for its success. A special thanks to Tune, who helped me arrange visits to the participants several times.

To the researches of the Assisted Living Project for a fruitful interdisciplinary collaboration. We had both very fun and very rough moments, and the outcome

of it is a lot of learning and respect to each other's expertise. A special thanks to Torhild Holthe and Dr. Anne Lund for the amazing effort in recruiting participants and for the help with communication with the residents throughout the trial.

To Tore and Amir, for the great support with the lab and other technical matters. Thank you also Tore for the fun conversations and great positive energy.

To my co-workers and especially friends at OsloMet – Laura, Ashish, Desta, Ramtin, and Jaroslav – for listening to me and offering comforting and encouraging words. Having you guys around everyday was a real pleasure and – really! – essential. My days were way better and funnier. I will miss you and our fun meals at work.

To my far-away friends – especially Bruna, Jheniffer, Nuno, Victor, Marcel, Roberto, Lívia, Shweta, Fay, Taner, Natália – who were always available to listen to me when I needed and sent me good thoughts. I miss having you around and really appreciate your friendship.

To all the amazing friends in Oslo who supported me through these very challenging past three years – Solvejg, Daniele, Laura, Ashish, Ramtin, Desta, Jonas, Alice, Rie, Drago, and Jaroslav. I could not ask for better friends and more unforgettable great moments. Thank you so much for listening to my fears and stories over and over during these years. Your understanding and encouragement were very meaningful for the success of my Ph.D and my sanity.

To my wonderful family. A gigantic thanks to my parents, Margareth and José Sérgio, and my siblings, Wagner and Andréa, who greatly supported and believed in me. Your support was essential for the accomplishment of this Ph.D. A big special thanks to my mom, who listened to me and my thoughts basically every day the last years and sent positive thoughts pretty much non-stop. I love you all. *À minha família maravilhosa. Muito obrigada aos meus pais, Margareth e José Sérgio, e meus irmãos, Wagner e Andréa, que me apoiaram imensamente e acreditaram em mim. Seu apoio foi essencial para o meu sucesso nesse doutorado. Amo todos vocês. Um obrigado especial à minha mãe que ouviu à mim e aos meus pensamentos praticamente todos os dias nos últimos anos e me mandou energia positiva praticamente o tempo todo.*

To the most marvellous boyfriend in the universe, Simon Simonsson. Thank you so much for the endless support, encouragement, patience, motivating words, and funny and caring moments in these past two very tough years. I can say that without a doubt having you around was one of the most essential requirements for me to complete this Ph.D. Thank you for being in my life. I love you.

**Flávia Dias Casagrande**
Oslo, September 2019

# Abstract

Activity recognition and prediction are prerequisites for the realisation of intelligent support functions in smart homes, including functions that help older adults with mild cognitive impairment or dementia (MCI/D) live a safe and independent life at home. A fair amount of research on smart home functions has been conducted into assisting older adults with MCI/D in their everyday life, for example by prompting with reminders or encouragement, diagnosis tools, and prediction, anticipation, and prevention of hazardous situations. A number of algorithms for activity recognition and prediction have been reported in the literature. Most of the work in the literature, however, uses data collected in *controlled environments*, e.g. lab or testbeds, based on scripted activities. There is also no *comprehensive* comparative study that investigates state-of-the-art algorithms with respect to different input data configurations, required amount of data, limitations, and suitable applications. Moreover, there is no such comparative study applied to data collected in real homes.

The aim of this thesis is to identify, apply, and evaluate state-of-the-art prediction methods using data collected from *real homes* of older adults. The work includes the following main tasks. Definition of the sensor network system to be used, recruitment of participants for data collection from real homes, survey and selection of well-performing prediction methods in the literature, and finally their implementation and evaluation using real data.

A *limited* number of *non-intrusive* sensors were installed, in total around fifteen binary sensors (magnetic, power, and motion). The type and number of sensors were decided based on requirements and constraints imposed by the project, e.g. user privacy, budget, industrial partner products and discussions with the users and researchers participating in the project. The data were collected for between one month up to a year, depending on the apartment.

Two *probabilistic* algorithms and *recurrent neural networks* were identified as state-of-the-art methods for sequence prediction tasks. They were firstly applied to the real data to predict the *next sensor event* that is activated/deactivated in the home. A comparative analysis of the probabilistic methods and the recurrent neural network was then performed for several factors (prediction accuracy, required amount of data for convergence, execution time, number and type of sensors). The recurrent neural network with long short-term memory provided the best performance. It was therefore developed further by including the *time* component. Several ways of including time information were investigated. Finally, rule-based algorithms were implemented to identify activities from the binary sensors' data. *Activity prediction* was then carried out.

The thesis concludes that good accuracy can be achieved on the prediction of the next sensor event and the next activity using a limited number of non-intrusive binary sensors in real homes. It was also established in this work that only a relatively small dataset is required to converge. The achieved peak accuracy is nevertheless not yet sufficient for implementing intelligent assistive functions. It is, however, presumably sufficient for supporting health personnel and caregivers in their work by indicating the current activity status, alert a hazardous situation, measure activity levels, and similar . More sensors or alternative sensors that provide more information would enable improved automation functions by predicting the next sensor event or activity, its time of occurrence and its duration with greater accuracy. The thesis sheds light on sensor event and activity prediction methods, input configurations, predictability of these events in a home, and the amount of data required for convergence.

# List of Papers

## Paper I

Casagrande, F.D. and Zouganeli, E. ‘Occupancy and Daily Activity Event Modelling in Smart Homes for Older Adults with Mild Cognitive Impairment or Dementia’. In: *Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59)* **153** (2018), pp. 236–242. DOI: 10.3384/ecp18153236.

## Paper II

Casagrande, F.D. , Tørresen, J. and Zouganeli, E. ‘Sensor Event Prediction using Recurrent Neural Network in Smart Homes for Older Adults’. In: *2018 IEEE International Conference on Intelligent Systems (IS)* (2019), pp. 662–668. DOI: 10.1109/IS.2018.8710467.

## Paper III

Casagrande, F.D. , Tørresen, J. and Zouganeli, E. ‘Comparison of Probabilistic Models and Neural Networks on Prediction of Home Sensor Events’. To be published at *Proceedings of the 2019 IEEE International Joint Conference on Neural Networks (IJCNN).*

## Paper IV

Casagrande, F.D. , Tørresen, J. and Zouganeli, E. ‘Prediction of the Next Sensor Event and its Time of Occurrence in Smart Homes’. In: *Proceedings of 28th International Conference on Artificial Neural Networks – Springer-Verlag Lecture Notes in Computer Science (LNCS)*, (2019), pp. 462–242. DOI: 10.1007/978-3-030-30490-4_37.

## Paper V

Casagrande, F.D. and Zouganeli, E. ‘Prediction of Next Sensor Event and its Time of Occurrence using Transfer Learning across Homes’. To be published at *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA).*

## Paper VI

Casagrande, F.D. , Tørresen, J. and Zouganeli, E. 'Predicting Sensor Events, Activities, and Time of Occurrence Using Binary Sensor Data from Homes with Older Adults'. In: *IEEE Access Journal* (2019), pp. 111012–111029. DOI: 10.1109/ACCESS.2019.2933994.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This chapter introduces the motivation and foundation for the research contained in the thesis. The research question, sub-research questions and thesis outline are presented in the second half of this chapter.

## 1.1 Motivation

The United Nations predicts that 22% of the world population will be aged 60 years old or over in 2050, as opposed to 12% in 2017 (Figure 1.1) [120]. The percentage will be even higher in Europe – 35%. Conditions that arise in older age include dementia (D), which affected 47 million people worldwide in 2015, a number that is expected to increase to 132 million by 2050 [127]. Mild cognitive impairment and dementia (MCI/D) are a cognitive decline that can affect attention, concentration, memory, comprehension, reasoning, and problem solving [129]. These interfere greatly with a person's ability to perform daily activities and therefore leads to the disability and dependency of older adults. It furthermore not only impacts the individuals involved, but also their carers, families, and societies. The World Health Organization reports that the consequences of dementia include a high increase in the cost of caring for those with dementia. This includes the large number of health professionals required, and loss in productivity due to the total costs related to dementia undermining social and economic development [127].

Ambient assisted living technologies (AALT) can be extremely helpful in minimising some of these foreseen consequences and current challenges. AALT can enable people to remain *longer* and *age in their homes* in many ways, e.g. by assisting individuals in daily activities, monitoring health and safety at home, and by improving the cost-effectiveness and quality of health and social services [11]. AALT usually is comprised of information and communication technologies (ICT), stand-alone assistive devices, and smart homes. A smart home can be defined as being a dwelling in which sensors and controllers are installed to enhance one or more aspects of the resident's life [30]. This can for example include comfort, energy efficiency, security, and safety. Smart homes originally required an interface which the resident could use to interact with the home system. Constant progress in artificial intelligence has, however, enabled a number of improvements in smart home systems, including systems that can make decisions itself based on previously gathered information and prior inputs from the residents [104]. Such a system could therefore potentially be advantageous for older adults with MCI/D.

A fair amount of research on smart home functions has been aimed at assisting older adults and older adults with MCI/D in their everyday life [11,

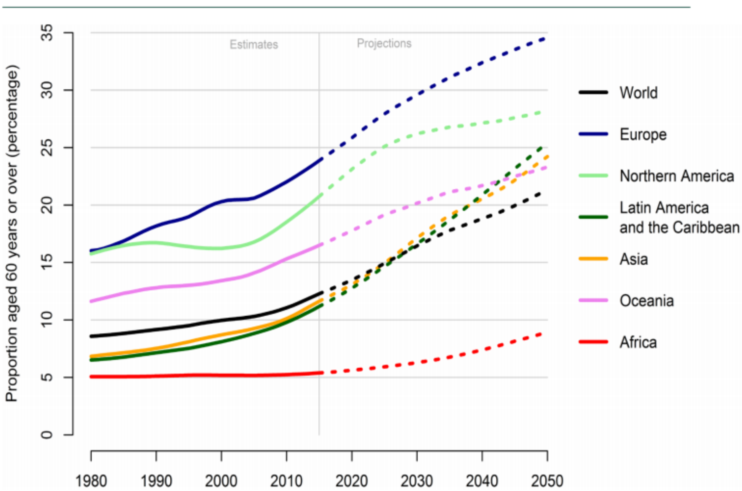



Figure 1.1: Percentage of population aged 60 years or over [120].

88, 102, 109]. Examples include prompting through reminders or encouragement [55, 81], diagnosis tools [5, 64], and prediction, anticipation, and prevention of hazardous situations [43, 76]. The operation of these functions is dependant on the performance and robustness of the activity recognition and prediction algorithms these functions are based on. The research literature contains a number of well-performing algorithms for activity recognition and prediction. Most of this work, however, uses data collected in controlled environments such as lab environments and testbeds, based on scripted activities [22]. This thesis aims to investigate the performance of state-of-the-art prediction algorithms in real home environments. Can these methods be applied reliably in a real home setup? How far are we from actually achieving smart homes that can assist older adults to age well, safely and independently at home? Only when such algorithms perform with sufficient accuracy can smart home functions be implemented and be useful to older adults, including potentially older adults with MCI/D.

## 1.2 Interdisciplinarity

This thesis was carried out as a part of the interdisciplinary "Assisted Living project (ALP) – Responsible innovations for dignified lives at home for persons with mild cognitive impairment or dementia", funded by the Norwegian Research Council. The project originally aimed to develop Assisted Living Technology (ALT) solutions for users with MCI/D by adopting a Responsible Research and Innovation (RRI) approach [44, 111]. Experts from the fields of health, ethics, and technology were involved in the project. In addition, the project cooperated with regional actors such as Oslo Municipality, the Norwegian Board of Technology (NBT), national and international academic partners (University of Bristol, University of Exeter, and Karlsruhe Institute of Technology), and industry partners (Sensio AS and RoomMate AS).

The interdisciplinary nature of the group, particularly the involvement of health experts, was a crucial element in recruiting users to participate in our field trial. Successful recruitment has enabled data collection from the smart homes of real users that reside in Skøyen Omsorg+ (Oslo), a care unit for older adults over 65 years old. The different competences in the group were also helpful in data collection, privacy considerations, and participant consent. It was not possible to recruit users with dementia, nor to receive definitive participant diagnoses. We only received the subjective opinion of the health experts on the level of user impairment.

Figure 1.2 presents the project workflow from the technology research point of view. The residents were seen more as co-researchers than users and the purpose of the trial and of the sensor system deployed in the apartments was decided in close collaboration with the participants through interviews and dialogue cafés [137]. This input from the residents would serve as an input for technology research, that would then translate the needs and challenges pointed out by the participants into self-learning functions in a smart home.

The industrial partner in the project installed sensors in the apartments and set up the network for data collection. This thesis focuses on the collection and use of these data for the implementation of smart functions in the future. The installation of sensors can be very challenging in personalised apartments and a feasibility study on the first installation of sensors in one apartment and the lessons learned can be found in [58].

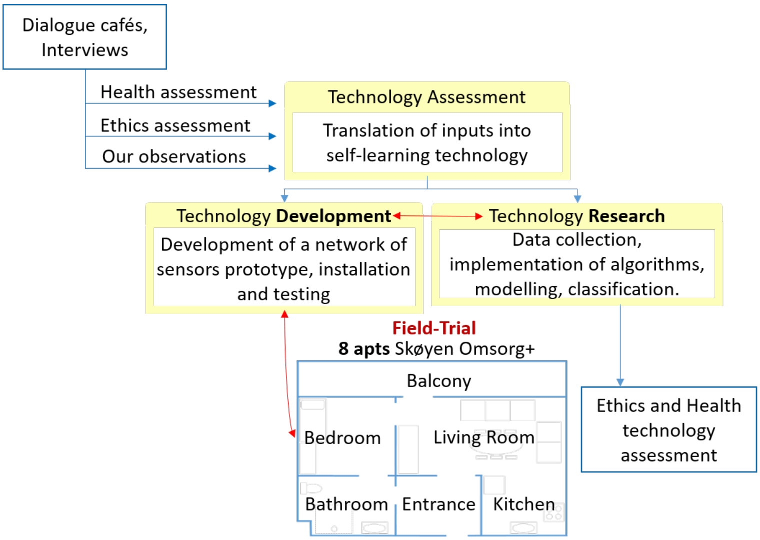



Figure 1.2: Workflow in the ALP, from the perspective of the technology work.

## 1.3 Research Objectives

The main aim of this thesis is to apply, compare, and identify performance, applicability and limitations of state-of-the-art sensor and activity prediction algorithms in smart homes. Our study aims, in particular, at having older adults as users, as ALT can be very helpful in supporting their independent lives at home. These algorithms enable the realisation of several assistive functions such as prompting systems that provide reminders or encouragement to the resident, improved operation of automation functions (e.g. turn the lights on in the bathroom when it is predicted the resident will go into the bathroom at night), prediction and prevention of anomalies, and indication of the onset of diseases.

A number of sequence prediction methods have been tested in controlled environments that simulate smart homes and have shown promising performance. This work aims to investigate the performance of these algorithms when used on data collected from a real world setup. The setup also entails other aspects such as privacy concerns and apartment aesthetics, which in this work resulted in a limited number of non-intrusive binary sensors.

The research question (RQ) of this thesis can, in order to address the above scope, be formulated as follows:

> *What is the performance and what are the limitations and the possible applications of state-of-the-art sequence prediction methods in real homes with older adults, where only a limited number of non-intrusive binary sensors can be used?*

This research question may further be divided into the following sub-research questions (SRQ):

**SRQ 1** What performance can be obtained when predicting the *next sensor event* and its *time of occurrence* in real homes, where a limited number of non-intrusive sensors are installed? What are the limitations and possible applications of each studied state-of-the-art algorithm (e.g. required amount of data for the model to converge, suitable sensor network, execution time)?

**SRQ 2** To what extent can a prediction model be generalised *across* apartments and users? Can the learned model in one apartment be used in other apartments where there is no available training data?

**SRQ 3** Is it possible to derive *human activities* through sensor events? Can activities be predicted with a better accuracy than sensor events?

These questions were assessed using data collected from smart homes in which real users, both male and female older adults, aged 70-95 years, live. This work resulted in six research papers: three published and two accepted (to be published) at peer-reviewed conferences, and one journal publication. Figure 1.3

illustrates how the research papers relate to the research questions and to each other.

Paper I, Paper II, Paper III, and Paper IV address **SRQ 1**. Paper I evaluates the performance of two state-of-the-art *probabilistic* algorithms when applied to the prediction of the *next sensor* to be activated or deactivated in a smart home. Data from *one* real home over a period of *two weeks* were available for use in this study. The algorithms were analysed and compared for factors such as best accuracy, memory length (i.e. number of sensor events required to predict the next one with best accuracy), amount of data required for convergence, and number of sensors in the dataset. Paper II evaluates the performance of *recurrent neural networks* (RNN) with long short-term memory (LSTM) in the prediction of the *next sensor* to be activated in a smart home and compares this with a baseline method. *17 weeks* of data from the same home as in Paper I were used. Factors such as best accuracy, memory length, required amount of data for convergence, and number of sensors in the dataset were analysed in the same way as in Paper I.

Paper III provides an in-depth comparative analysis of the *probabilistic* algorithms and *LSTM networks* in the prediction of the *next sensor* to be activated or deactivated in the home. This study used data collected over *30 weeks* in this apartment. This volume of data was sufficient to analyse the studied algorithms in a definitive manner. We compare optimal memory length, required amount of data for convergence, top accuracy, and training and testing execution time. The LSTM network was the best-performing algorithm. This algorithm was therefore improved in Paper IV by including *time* information. We predicted the next sensor event based on previous sensor events and their time of occurrence, then predicted both the next sensor and time of occurrence information based on these. Several methods for including the time as a feature were investigated in this paper.

Paper V focuses on **SRQ 2**. The performance of the best performing algorithm presented in Paper IV was analysed based on data from *four* other apartments. The paper also presents the advantages and limitations of applying *transfer learning* to our setup. Two methods for transfer learning were investigated. These were training the base network using data from four apartments and fine-tuning with data from the target apartment; and training with data from the apartment that has best accuracy when modelled individually and fine-tuning with data from the target apartment.

Paper VI concludes this thesis by applying the algorithms to datasets for *all* eight of the apartments in the field trial. It was also investigated whether the algorithms present the same behaviour for all users. An analysis similar to that in Paper V was carried out. This paper in this way reinforces the results found in the previous papers that address **SRQ 1** and **SRQ 2**. This paper also addresses **SRQ 3**. Associating sensor events to activities is explored in two different rule-based algorithms. The prediction of *activities* is then evaluated using the best methods found for the prediction of the next sensor.

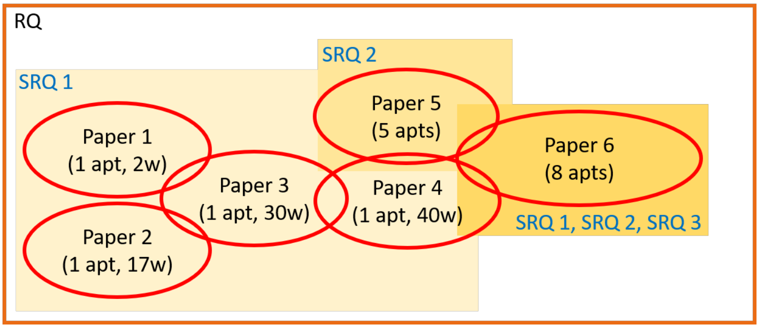



Figure 1.3: Relation between research questions and papers. The number of apartments with data available for the paper, and the number of weeks of data are given in parentheses. No weeks is specified where the collection periods for each apartment differ.

## 1.4 Thesis Outline

This thesis is a collection of six research papers that constitute the research contribution. Chapter 2 presents the background for the thesis, which consists of (i) the ethics approach on which the ALP is based, (ii) the dialogue cafés with users, (iii) a literature survey of ALTs, and (iv) a literature survey of sensor and activity prediction methods in the literature, these providing the basis for the development of ALTs. Chapter 3 introduces our field trial and the prediction algorithms that were applied to the evaluation of the real data collected. Chapter 4 presents an overview of the contributions of the research papers, and individual summaries for each paper. Chapter 5 then discusses the findings of this research and presents conclusions and suggestions for future work. The thesis finally ends with the collection of papers.

The source code for the implemented methods in this thesis are available on my github account [1].

[1] Github account: https://github.com/flaviadcasag/assisted_living_phd_project

Chapter 2

# Background

This chapter presents the background for this research work. The background includes the ethics concerns within the interdisciplinary group, the input acquired from the older adults participating in the project, and the literature survey carried out both on assisted living technologies and sequence prediction algorithms.

## 2.1 Ethics Framework within the Assisted Living Project

The ALP follows the Responsible Research and Innovation (RRI) approach as mentioned in Section 1.2. The RRI term is, in fact, not in much use anymore. There has also never been a conclusive definition of what it really entails. RRI has been explained as being a "strategy of stakeholders to become mutual responsive to each other and anticipate research and innovation outcomes underpinning the "grand challenges" of our time for which they share responsibility" [111]. RRI conveys, in this project, that technology research should be carried out in a responsible way, together with users, from idea to the further development of the technology.

RRI was one of the cornerstones of the interdisciplinary project. The RRI/ethics experts therefore had a strong influence on the technical decisions taken throughout the project. There were many discussions around the type and the number of sensors that should be installed. The view of the ethics experts, and to a certain extent of the health experts, was that many sensors may disturb the residents and be extremely privacy-invasive. It was also expected that too many sensors would make recruitment much more challenging. Using non-intrusive sensors and keeping the number of sensors to the minimum possible were therefore *prerequisites*.

## 2.2 Dialogue Cafés with Users in the Assisted Living Project

The ALP has introduced dialogue cafés as a user involvement method. This approach was inspired by world cafés [40], dialogue conferences [1] and scenario workshops [93]. The idea is that research becomes an interactive process with users through involving different stakeholders in discussions on topics that matter to them, in a safe environment. A number of dialogue cafés were conducted during the ALP at the care-residence Skøyen Omsorg+ (Oslo, Norway), see Figure 2.1. The users (all over 65 years old) were invited to share their opinions and experiences on technology, on its usability and acceptability, challenges in everyday life, and other topics of interest to the project participants and stakeholders.

Figure 2.1: Dialogue café at Skøyen Omsorg+.

One of the cafés focused on needs and challenges in everyday life. This provided important input that directed the course of the technology research. The needs and challenges are described below in the order of importance to the café participants (about 14 residents).

1. Fear of falls. Detection and prevention of falls was extremely important to them. They only, at the beginning of the project, used a necklace with a button that they could push if they fell. This, however, does not always work well (e.g. long time until someone comes to help or the device does not trigger an alarm). They currently also have an ambient depth camera for fall detection. This is provided by the municipality.

2. Difficulties sleeping. Many of the residents expressed problems getting to sleep and waking frequently and/or too early.

3. Orientation at night. Some residents have indicated that lights and switches are not well positioned in the home. This makes it difficult to get around at night.

4. “Button-phobia”. Many expressed having difficulties operating devices with many buttons and being afraid of pushing the wrong button.

5. Everyday self-sufficiency. A few participants indicated that they wanted to be able to take care of themselves and carry out day-to-day activities independently.

6. Mobility and safety outdoors. Some participants mentioned finding getting around difficult and of not being safe when getting out of their home.

One solution to these needs and challenges would be to develop a device/system for each need and challenge. This could potentially work. Many of the items may, however, be correlated. For example, not having a a good night’s sleep (item 2) may affect balance when walking around at home and can lead to falls (item 1). Falls (item 1) can also occur at night, when it is difficult to walk around the apartment (item 3). Most of the items are to some degree related to everyday self-sufficiency (item 5): the residents will need someone to help them if they cannot operate a device with many buttons, to check on them at a certain frequency to make sure they have not fallen, to help them go to the bathroom or get a glass of water during the night, and assist them outdoors. These correlations can provide more information to a unique system that could address all the items at the same time and therefore perform better. A more holistic approach would therefore be beneficial and could potentially be a requirement.

A second solution could be to develop an integrated smart home system that would address all (or most) of these needs and challenges. Such a solution could potentially fully address difficulties with sleep, orientation at night, "button-phobia", and everyday self-sufficiency (items 2-5). It could also partially address falls (item 1). They would, however, still need someone to come and help them if they fell. Mobility and safety outdoors (item 6) could, however, not directly be covered by a home system. Having access to information on activity levels during the day or week may, however, be helpful in advising residents on whether going out is a good idea at that point in time.

A smart home system such as this would have to take into account several automation functions. The system should be able to predict and identify falls and call for help (item 1). Lights and devices such as the TV and microwave should be able to be switched on and off automatically whenever they are needed (item 3 and 4). It should also be possible to provide a personalised activity pattern analysis that could be useful, for example, with item 2, e.g. identify which activities during the day lead to better sleep. All of these also impact self-sufficiency, item 5.

Activity recognition and prediction algorithms must be used in the implementation of these functions. The system must be able to recognise current activities and predict future ones in order to assist the resident and/or take action. For example, the system ought to be able to predict when the resident is likely to turn the TV on and understand if the resident has problems with the TV so that the controller can turn it on for him/her. The correct and robust operation of these algorithms is essential to the implementation of assistive functions in smart homes.

## 2.3 Survey on Assisted Living Technology

This section presents an overview of the current status of Assisted Living Technologies (ALTs). A short summary of commercial technologies is firstly presented, followed by the state-of-the-art for the research field for technologies for older adults and older adults with MCI/D. A more in-depth review that was conducted in this project is found in [37].

### 2.3.1 Commercial Assisted Living Technologies

ALTs have been progressively incorporated into homes to provide automation, to increase comfort, safety and security and to reduce energy consumption and monitor health [1]. Simple smart home devices can be very helpful in improving the comfort of users and promoting an energy efficient home. A number of commercial solutions are also available that control lighting and ambient temperature, some of these including a learning system. Learning systems adapt the system to the users' preferences based on user input over a period of time. Home security can be increased by cameras that can stream the home environment to, for example, smart phones. Smart locks also increase security by allowing users to see and speak with visitors before letting them in, through viewing streaming video on, for example, their smart phone. Health factors can be measured and monitored using a wide variety of the biosensors that are available on the market and by using smart watches.

Most smart devices and sensors are used to improve comfort and increase energy efficiency through using home automation that is designed for the general public. There are, however, far fewer devices and systems for older adults and older adults with MCI/D, these systems generally being focused on increasing safety and providing assistance with daily life activities. The overall maturity of the technology for older adults is also lower. The most common devices and systems on the market for older adults are medication dispensers and fall detection/alarms. There are also a number of calendar/socialising apps, some devices/solutions that address wandering, and some solutions that give reminders based on a calendar (e.g. reminders of appointments and events). There are finally some commercial systems for the elderly, including for those with MCI/D, that claim to carry out some sort of behaviour pattern monitoring. These systems currently provide limited added value.

Smart sensors and devices have evolved significantly from very simple automation systems to smarter systems. A great deal of improvement and development, however, remains to be achieved. The trend today is towards more complete smart home systems that integrate many sensors into a single network. Small scale systems can today be set up by the user, for example by installing a hub for smart speakers and by using compatible devices. A system can also be set up by companies using existing technology. A number of systems are available on the market. The degree of integration, however, varies but is increasing steadily.

### 2.3.2 Research on Assisted Living Technologies

A number of ALTs for older adults have, in recent years, been under research. These include (i) reviews of smart homes, wearable devices, and robotics, projects grouped into Europe, United States, Asia and Australia research projects [22]; (ii) review of smart homes, robotics, virtual reality and gaming, telemedicine,

[1] Please refer to the report [37] to find references of the commercial technology cited in this section.

and social support [85]; reviews that group technologies and projects into smart homes, mobile, and wearable sensors and robotics, and point out algorithms in activity recognition, context modelling, anomaly detection, location and planning, and review applications in health monitoring tools, wandering prevention, and cognitive orthotics [102], and (iv) an extensive review of smart homes, activity recognition activities and all aspects related to this including sensors, features selection, and classification [88]. This subsection presents work in the literature that relates to the application area of the work carried out in this thesis, including research that refers to the needs and challenges pointed out by the residents in our project, as well as smart homes.

#### 2.3.2.1 Stand-alone Devices

A lot of ongoing research is being conducted into fall detection, this indicating that commercial technology has not yet reached maturity. A number of fall detection methods have been proposed e.g. using depth cameras [75, 131], floor-vibration sensors [4], and accelerometers [8]. These use tracking methods, geometrical analysis, and threshold values. The accuracy obtained in these experiments is very good, close to or 100%. They are, however, very dependent on the rules designed by the developers. Machine learning may, however, be more robust for these applications, generalising the application to different environments and people. Other work on fall detection has used machine learning algorithms and data from depth sensors [116], accelerometers [92], or a combination of different sensors [66, 132]. These systems still generate considerable numbers of false alarms, despite the great progress in this area. These are very inconvenient for users. Most of these systems were not tested using data from real environments.

There are also quite a few memory-aid devices in the research literature. The Memory Aiding Prompting System (MAPS) is one example. This is primarily designed for adults with cognitive disabilities, to help them complete activities of daily living [18]. Autominder, another device, was developed for people with memory impairment. It uses AI methods to provide personalised reminders that adapt over time [98]. A further device reported in the literature is a smart reminder system that used two accelerometers on each wrist to recognise activities through artificial neural networks and send reminders when a predicted activity was not being executed [21].

Some projects have conducted research into mobility devices. A smart cane that provides physical support when walking/standing and guidance was tested in an elderly living facility [38]. The cane's functions include right path guidance, obstacle detection and avoidance, monitoring of the user's vital signs, and the identification of the user's intent. Another mobile orientation device with three functions was developed for people with dementia. The three functions are orientation assistance, which consists of alerting and requesting a caretaker when the user leaves home, an appointment management function that reminds the user of upcoming events by voice through the device, and an emergency service that allows two-way voice communication [45, 46]. Both the cane and the orientation device were evaluated by elderly users. The users, however, thought that they were difficult to operate.

Stand-alone devices have potential. Studies have, however, indicated the need for more user-friendly devices [35]. Older adults also prefer ambient sensors rather than wearables [35]. Using ambient sensors instead of wearables or robots is also an important factor, particularly for older adults with MCI/D. Older adults with MCI/D can forget what the devices are used for and not use them properly. As mentioned previously, an integrated system can have advantages, as every activity in a home can be correlated. It therefore may be an advantage to use a system that addresses everything. This may be why smart home systems have been so widely researched in recent years. Smart home systems under ongoing research are described in the following.

#### 2.3.2.2 Smart Homes and Smart Home Systems

A number of smart home projects have been reported in reviews in recent years [3, 22, 109]. The aim of the research is usually to install a number of sensors in a home that can gather data and be used in the implementation of functions that can improve comfort, energy efficiency, safety and/or assist with activities of daily living (ADL). Table 2.1 presents smart homes that are relevant to this thesis (i.e. that provide safety and/or assistance with ADLs). The table lists the aim of the research, the type and number of sensors and where the algorithms were applied/tested.

Table 2.1: Smart homes in the research literature

| Smart home | Aim | Sensors | Dataset |
|---|---|---|---|
| Casattenta [41] | Track the inhabitant and identify dangerous situations (e.g. falls, gas leak)<br>Graphical interface for social interaction (video-conference), alarms, reminders, home surveillance and safety | Wearable sensors and ambient sensors | Labs and industry environments |
| Easy Line+ [12, 15, 97] | Help elderly and disabled people to have more autonomy in the kitchen | Radio frequency identification (RFID) sensors in appliances; gas, fire, smoke and flooding sensors; magnetic sensors for the doors and motion sensors | Labs |
| Sweet-home [19, 20, 122] | Audio interaction in which the user can easily control their home environment<br>Recognize vocal orders, distress sentences and equipment usage | Microphones (8) and ambient sensors (motion sensors, switches, temperature, etc.) | Domus testbed (more than 150 sensors) |
| Smarter Safer Home [133, 134] | To support older people with mild dementia<br>To assess the severity of dementia and detect abnormal behaviours | Motion sensors, accelerometers in mattresses, power sensors, acoustic sensors for water pipes, temperature and humidity sensors, magnetic sensors and pressure sensors | The system was going to be implemented in a trial platform in twenty different homes, but to our knowledge no reports exist of this implementation. |

*Continued on next page*

Table 2.1 – *Continued from previous page*

| **Smart home** | **Aim** | **Sensors** | **Dataset** |
|---|---|---|---|
| COACH – Cognitive Orthosis for Assisting aCtivities in the Home [13, 14, 55, 56, 79, 80] | To support people with dementia with ADLs, focus on handwashing activity | Color camera | Labs and care facility |
| SPHERE – Sensor Platform for HEalthcare in a Residential Environment [135] | Development of sensors<br>Multi-sensor data fusion<br>Detect and management of various health conditions | More than 40 ambient sensors, wearable sensors and color and depth cameras | 100 homes |
| CASAS – Center for Advanced Studies in Adaptive Systems [23, 27, 31, 33, 34, 57, 69, 82, 83, 84, 86, 87, 100, 101, 103, 113, 114, 115, 128] | Several smart home testbeds and public datasets<br>Activity recognition and prediction<br>Prompting systems<br>Early detection of MCI/D | About 50 binary and sampling sensors | CASAS testbeds |
| DOMUS – DOMotics at the Université de Sherbrook [10, 16, 48, 54, 106, 117] | Cognitive assistance, medical monitoring and televigilance for people with cognitive disorders<br>Cognitive orthesis for meal preparation to help people with head trauma<br>Early detection of dementia<br>Autonomous outdoor mobility | Movement detectors, electromagnetic contacts, tactile carpets, microphones, loudspeakers, readers and RFID tags, etc. | DOMUS testbed |

*Continued on next page*

Table 2.1 – *Continued from previous page*

| Smart home | Aim | Sensors | Dataset |
|---|---|---|---|
| NOCTURNAL – Night Optimised Care Technology for UseRs Needing Assisted Lifestyles [6, 76, 89, 90, 126] | Mainly designed for people with early dementia who can often be confused when awakening Orientation during the night time | Motion and bed movement sensors | 8 people with dementia for 3 months |
| Dem@Care – Dementia Ambient Care [62, 63, 68, 77, 108, 110] | Activity recognition, data fusion and mining Knowledge-representation Intelligent decision-making support Early dementia diagnosis | Color and depth cameras, power sensors, RFID sensors in objects, motion sensors, wristwatch (physical activity measurement), microphones and pressure sensors in the bed | 1 person with MCI for 3 months |
| MavHome – Managing an Intelligent Versatile Home [30, 32] | Increase the comfort of residents Reduce the operation costs | About 50 sensors: power sensors, acoustic sensors, temperature sensors, motion sensors, RFID sensors, wearable vital sign sensors | MavHome testbed |

Most of the systems have been developed and/or evaluated using data collected in controlled environments (labs and/or testbeds). Only a few have used real homes and real users (i.e. target group of study, for example, people with dementia). The systems tested in real homes seem, however, to not progress forward. There are, to the best of our knowledge, no reports of further experiments in some of these projects, e.g. NOCTURNAL, Smarter Safer Home, Dem@Care. This might be due to difficulties with finding volunteers, or because the method was not robust enough for application in real homes. Smart home technology may also still be immature because of inadequate algorithms, improper activity recognition methods, and low rates of prediction accuracy [3].

Most of the projects in Table 2.1 used a large number of sensors (50-150) in the smart homes. Some of the smart homes even use colour cameras, which older adults have said are obtrusive and violate the resident's privacy [35]. A smaller number of sensors may be preferable to reduce user surveillance, lower costs, and to reduce the impact on home aesthetics. It has also been shown that too many sensors do not always improve activity recognition results [28].

## 2.4 Survey on Sensor Event, Activity, and Time Prediction

Many of the assistive functions in smart homes require robust sensor/activity prediction to work properly. Well functioning algorithms can lead to, for example, an improved operation of automation functions (e.g. adjust the temperature early enough prior to the person waking up); enable the realisation of prompting systems (e.g. prompt the resident if the predicted activity has not been performed) [57]; or identify changes and anomalies in certain behaviour patterns (e.g. movement, everyday habits, etc.), so indicating the onset or the progress of a condition [105]. Sensor event/activity prediction is comprised of two main tasks: *event* prediction (sensor or activity) and *time* prediction.

A number of algorithms for sequential event prediction have been studied in recent years [130]. These algorithms train a model to predict the next symbol, based on a sequence of symbols. The Active LeZi (ALZ) is a probabilistic method that has been extensively employed in the prediction of sequential data [49]. The algorithm firstly finds patterns in a sequence based on a text compression algorithm. It then builds a tree using these patterns, including frequency of occurrence. Please refer to Figure 3.3 and Section 3.3.1 for a better understanding of this method. An algorithm that computes probabilities based on the tree is then applied to predict the next symbol. ALZ achieved a peak accuracy of 47% when applied to the Mavlab testbed dataset (Figure 2.2, – 50 binary and sampling sensors) [49]. The dataset included events from six persons and their interaction with 50 different devices.

The Sequence Prediction via Enhanced Episode Discovery (SPEED) algorithm builds on an idea that is similar to ALZ [2]. It constructs a tree from the patterns found in sequences. SPEED, however, introduces *episodes* to find patterns, episodes being sequences that start and end with an activation or deactivation of the same sensor, or vice-versa. Patterns are then derived from the episodes.

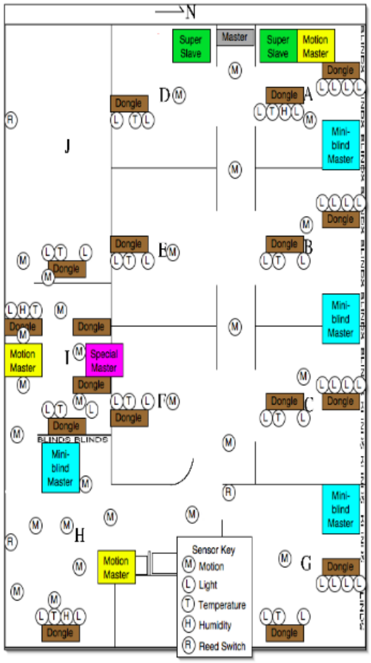



Figure 2.2: Mavlab testbed [123].

Please refer to Figure 3.4 and Section 3.3.2 for a better understanding of this method. SPEED was applied to the same dataset as ALZ (Mavlab — Figure 2.2) and achieved an accuracy of 88.3%. The same dataset was used for both training and testing. A comparison of the prediction accuracy of SPEED and ALZ is presented in Figure 2.3.

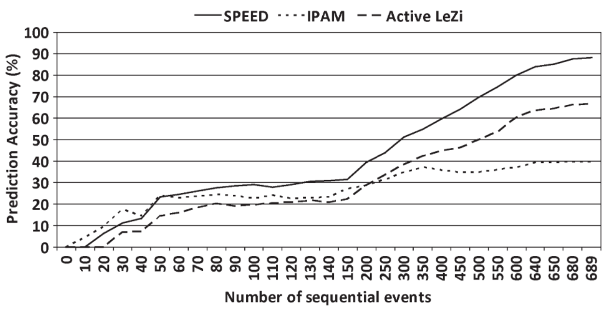



Figure 2.3: SPEED compared to other sequence prediction methods [2].

Neural networks have also been used for sensor event prediction, achieving notable performance. Recurrent neural networks (RNN) are usually used for this application [72, 96, 124]. RNN networks were designed for and often work better on tasks that involve sequential inputs (e.g. speech and language) [70]. RNN networks keep an internal memory in their hidden units and can therefore contain information on the history of past inputs. Three RNN models, Echo State Network (ESN), Back Propagation Through Time (BPTT), and Real Time Recurrent Learning (RTRL), were applied to a ten-day dataset with six binary sensors (four motion and two magnetic, see Figure 2.4) [72]. One person with dementia lived in the apartment. The networks were implemented such that the number of input and output values corresponded to the number of sensors in the dataset. Each sensor assumed a value of "0" or "1", "off" or "on", in a certain time slot. Prediction was computed for the next six hours. The ESN model performed best, the root square mean error (RMSE) being 0.06 [72].

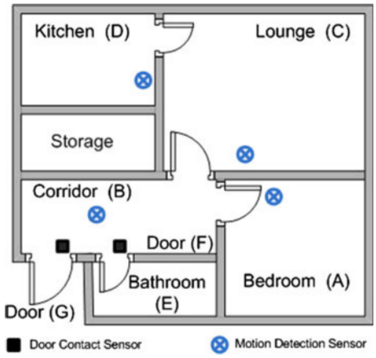



Figure 2.4: Real home with binary sensors for data collected in [72].

A similar study was carried out on a 16-room office environment [124]. The dataset in this case was collected through an app installed on the personal data assistants (PDA) of participating employees, the participants manually registering when they entered or left a room. There were four participants in the study and an Elman network and a multilayer perceptron network were applied to predict the next room a person would enter. The Elman network achieved the best results, which ranged from 70% to 91% accuracy depending on the user. There were 16 rooms in the office. Each room was therefore codified in four bits. The input corresponded to two previous rooms and the output to the predicted next room. This study also applied other methods – Bayesian network, state prediction, and Markov predictor. Comparable results were achieved (Figure 2.5) [96].

A similar study was carried out for a 16-room office environment [124]. The dataset in this case was collected through an app installed on the personal data assistant (PDA) of participating employees that had to register manually whenever they entered/left a certain room. An Elman network and a multilayer perceptron network were applied to predict the next room a person would go to. There were four participants in the study and the Elman network attained the best results, ranging from 70% to 91% accuracy depending on the user. Each room was codified in four bits as there were 16 rooms in total. The input corresponded to two rooms and the output to the predicted next room. This work also applied other methods – Bayesian network, state prediction, and Markov predictor – where comparable results were achieved [96].

| | Elman net | MLP | Bayesian network | State predictor | Markov predictor |
|---|---|---|---|---|---|
| Person A | 91.07% | 87.39% | 85.58% | 88.39% | 90.18% |
| Person B | 78.88% | 75.66% | 86.54% | 80.35% | 78.97% |
| Person C | 69.92% | 68.68% | 86.77% | 75.17% | 75.17% |
| Person D | 78.83% | 74.06% | 69.78% | 76.42% | 78.05% |

Figure 2.5: Comparison of methods for different users in a 16-room office environment for the prediction of the next room [96].

Such algorithms should also be able to predict when the next sensor event or activity in a sequence will occur. Time series methods such as Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) have been extensively applied in the literature to predict the time of the next event or activity [17]. These methods assume the time series to be linear. This is, however, not the case for activities in a home [83]. Rule-based algorithms have been developed for time forecasting [7, 57], the algorithms extracting temporal patterns between related activities to generate predictions. They are quite useful and have been shown to be able to achieve good performance. These algorithms are however not good at predicting very complex activities and activities that do not occur very often.

Non-linear time series models would be more suitable for time prediction in smart homes, e.g. artificial neural networks. A Non-linear Autoregressive Network (NARX) showed promising results in the prediction of the start and the end time of sensor activation [73]. This is a type of RNN that is usually applied to long term time series prediction. Simulated data and real data were both used in this study. The real data were collected over 20 days and contained events from six binary sensors, see Figure 2.4. Each sensor had its own network, where input and output nodes had start and end time of the sensor's activation.. The NARX network performed better, with a RMSE ranging from 0.06 to 0.09 depending on the sensor. An example of time prediction in the bedroom using simulated data is shown in Figure 2.6.

Decision trees have been used to predict the point in time at which a specific activity would take place [83]. This method relies on several features being extracted from sensor event sequences, classified by a regression tree, each leaf node containing multivariate linear models. The method was applied to data from 25 testbeds each with 51 binary and sampling sensors and achieved an average normalised RMSE of 0.01.

Bayesian networks have been applied to predict the next location, time of day, day of the week and (as a consequence) the activity label of what the person is doing [86]. This work is the closest to ours in terms of predicting next sensor/activity and time using a single model, see Figure 2.7. It has achieved an accuracy of 46-60% when predicting the next location, 66-87% for time of day (slots of 3 hours in the day), 89-97% for predicting the day of the week, and

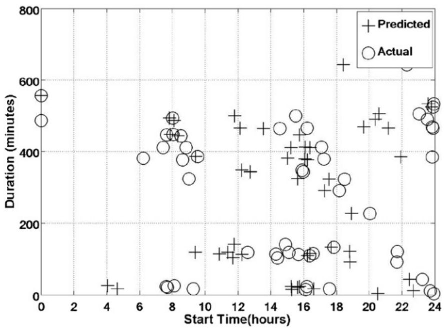



Figure 2.6: Prediction results for the bedroom sensor from a simulated environment [73].

61-64% for activity recognition. This algorithm was employed in two apartments each with around 30 binary sensors.

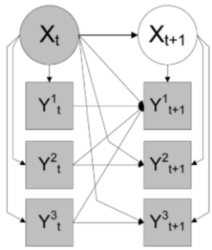



Figure 2.7: Bayesian network structure for predicting next location $Y^1$, time of day $Y^2$, day of the week $Y^3$ and activity $X$ [86].

## Transfer Learning

Each individual has her/his unique habits. Smart homes may also have different layouts and different limitations for the deployment of sensors. It is therefore important that prediction algorithms are able to adapt to each home and to each resident's habits. Transfer learning (TL) is a technique that entails the training and learning of parameters from a source dataset that is different yet related to a target dataset (e.g. different labels and data distributions) [94]. Applying TL can therefore reduce the training time of a model and the data required from the new domain (e.g. apartment). It is also possible to investigate how general a model is, i.e. whether the model can be applied to new apartments without any fine-tuning of the source model.

Transfer learning has been used in a number of fields, e.g. image and language classification, computer networks, automated planning, mathematical problems, and activity recognition [29, 94]. It has not, however, been fully explored for time series data [42]. This might be due to the lack of available general purpose pre-trained models [42]. This may, however, change soon as transfer learning has proved to provide many advantages in other domains.

It has, for example, been shown that models trained on features extracted using a pre-trained RNN perform better or at least as well as RNNs trained for a specific task within electronic health record data prediction [52]. Transfer learning can also dramatically decrease the required amount of data in the target dataset, as proved for mortality prediction algorithm [36] and for activity recognition [59, 99]. It also allows datasets with different feature spaces to transfer knowledge between each other [59, 100]. Transfer learning can be applied to a number of algorithms: RNNs [52], Hidden Markov Models [60], statistical inference [36], support vector machine [59].

A cross-domain activity recognition algorithm combined with transfer learning and a similarity function between activities, was proposed by a study for transfer learning in smart homes [59]. Three different datasets were used in this study. One dataset was collected over 28 days from a real home of a 26- year-old man. Peak accuracy was 65% for seven activities. Other studies have examined the transfer of knowledge of activities from multiple physical source spaces to a different target physical space [99], an algorithm being proposed that automatically maps activities from source to target environment and that classifies activities based on a weighted majority vote method. The data used in this study were collected from six testbeds in which volunteers lived for 2-3 months. The data included 5 to 11 activities, and a peak accuracy of about 80% was achieved. HMM and transfer learning have also been used in combination across three apartments with five recorded activities and achieved a F1-score of 0.65 in the best case [65].

Transfer learning has its limitations. It has been shown that it can either improve or degrade the prediction accuracy of models, depending on the dataset used for transfer learning. This degrading of the prediction accuracy is known as negative learning [94]. It is important, in these cases, to detect which source dataset is the best to use for a certain problem. For example, Dynamic Time Warping may be beneficial for measuring inter-dataset similarities [42].

## 2.5 Summary

This chapter has described the background to this thesis. The background consists of a number of inputs, including results from dialogue cafés with older adults, discussions within the ALP, and literature surveys on both assisted living technologies and activity recognition and prediction methods.

A number of needs and challenges pointed out by the older adults in dialogue cafés in the ALP could be addressed by an integrated smart home. Studies have shown that non-intrusive sensors are preferable, and that ambient sensors are preferred over wearables. The RRI framework chosen for the ALP also meant that a limited number of sensors were used, to accommodate privacy issues. Our smart home contains events from 13-17 binary sensors, i.e. twice as many as used in [72, 73], and less than one third of the number of sensors used in [2, 49, 83]. Our study is comparable to the work reported in [96] (16 rooms). However, the events in that study were inserted by each user rather than being collected by sensors. This implies less noise in the data.

A number of activity recognition and prediction algorithms have been reported in the literature. The ALZ algorithm has been widely used. SPEED, even though it is similar to ALZ, does however give better results. Recurrent neural networks have also shown promising results. These three are therefore the state-of-the-art methods in this field. Nevertheless, most of the work in the literature, especially studies using these methods, use data collected in controlled environments, many being based on scripted activities. It is also quite common for there to be many sensors in an environment (50 or more). This may not be applicable to real homes due to privacy concerns, budget constraints, and home aesthetics. There are, furthermore and to the best of our knowledge, no comprehensive comparative studies that investigate state-of-the-art algorithms applied to data collected in real homes and comparing a number of aspects on the performance. This deficit is addressed in the work presented in this thesis.

# Chapter 3

# Material, Users, and Methods

This chapter describes the field trial including the sensors, the system installed in the homes, and the users. Data preprocessing, the algorithms used for sensor event and activity prediction and the transfer learning technique are also presented in this chapter.

## 3.1 Field Trial

### 3.1.1 Sensors

The industrial partner in the project, Sensio, could supply three types of ambient binary sensors [112]: *motion* sensors, *magnetic* sensors, and *power* sensors (Figure 3.1). *Motion* sensors (Pyroelectric/Passive Infrared – PIR) detect motion through detecting a change in infrared radiation in the sensor's field of view. The sensor generates an event with message "1" each time motion is detected. It otherwise sends no event. *Magnetic* sensors consist of two components, a reed switch and a magnet. They are fitted opposite each other on doors, windows, and drawers to indicate whether they are open or closed. An electric current is created when the two pieces are close to each other, the circuit otherwise being broken. Events with message "1" are generated for open and "0" for closed. *Power* sensors measure the electricity usage of an appliance. They can therefore indicate whether the appliance is turned on or off, events with message "1" being for on and "0" for off. The power sensors themselves do not, however, generate the "1" and "0" messages. Sensio therefore had to manually find each device's power thresholds for on and off, for each apartment and each piece of equipment.

The data from these sensors include timestamp (date and time, precision in seconds), sensor ID, and sensor message (binary). Table 3.1 shows an example of data collected from the sensor network.


(a) Motion sensor (b) Magnetic sensor (c) Power sensor

Figure 3.1: Sensors for smart homes from the project's industrial partner Sensio [107].

Table 3.1: Sample of binary sensor data

| Timestamp | Sensor ID | Sensor message |
|---|---|---|
| 01.09.2017 07:58:05 | 2 | 1 |
| 01.09.2017 08:00:14 | 12 | 1 |
| 01.09.2017 08:01:01 | 4 | 1 |
| 01.09.2017 08:02:56 | 5 | 1 |
| 01.09.2017 08:03:05 | 12 | 0 |

### 3.1.2 Data Collection and Storage

A great deal of preparatory work must be carried out before personal data can be collected and stored. The first step is to apply to the Norwegian Centre for Research Data (NSD – *Norsk Senter for Forskningsdata*) for project approval. The application included information on how participant consent would be obtained, how the data would be collected, stored and processed, and how the security and anonymization of the users' data would be guaranteed.

The approval given by NSD contained the requirement that the data from the binary sensors is pushed to and stored on the secure server at the University of Oslo, Services for Sensitive Data (TSD – *Tjenester for Sensitive Data*) [119]. TSD was furthermore responsible for the anonymity, encryption, and security of the collected data.

Sensio, the industrial partner in the project, was responsible for installing the sensor system in the homes. The sensors were connected wirelessly through Z-Wave and xComfort protocols to a Raspberry Pi 3, which then transferred the data to TSD. The connection to TSD was handled by Sensio, who were responsible for securing the anonymity and encryption TSD requires.

The aim of the research was presented to the residents at a combined Dialogue Café and Recruitment meeting. Interested residents were asked to register their names on a list at the meeting. They were then individually informed (orally and in writing) about the project and about consent by the project's health researchers. All participants were capable of giving their informed consent, this being determined by their health record or from the knowledge of the housekeeper.

### 3.1.3 Field Trial

Our field trial involved eight apartments in a community care facility. All apartment residents were more than 70 years old, and the apartments had similar layouts. All have a bedroom, a living room, an open-plan kitchen area, a bathroom, and an entrance hall (Figure 3.2).

A minimal number of binary sensors were installed in the apartments. This was to minimise resident surveillance and due to the project's technical and financial constraints. A set of sensors was selected that can potentially identify daily activities and that enable the realisation of useful functions for older adults, indicated in dialogue cafés with users [137]. The sensors generate events that can indicate occupancy patterns (movement around the apartment), and daily activities such as kitchen activities, dressing, being in bed, and leisure activities such as reading, watching TV and listening to the radio.

The proposed sensor system is shown in Figure 3.2. Physical limitations, however, meant that not all apartments were installed with an identical set of sensors. Examples of physical limitations include a fridge door with a gap that was too wide for a magnetic sensor to be used and different equipment in the apartments, for example the fact that residents have either a coffee machine or a kettle. All the motion sensors were installed in all eight apartments. The power and magnetic sensors installed in the apartments vary from apartment to apartment, as shown in Table 3.2.

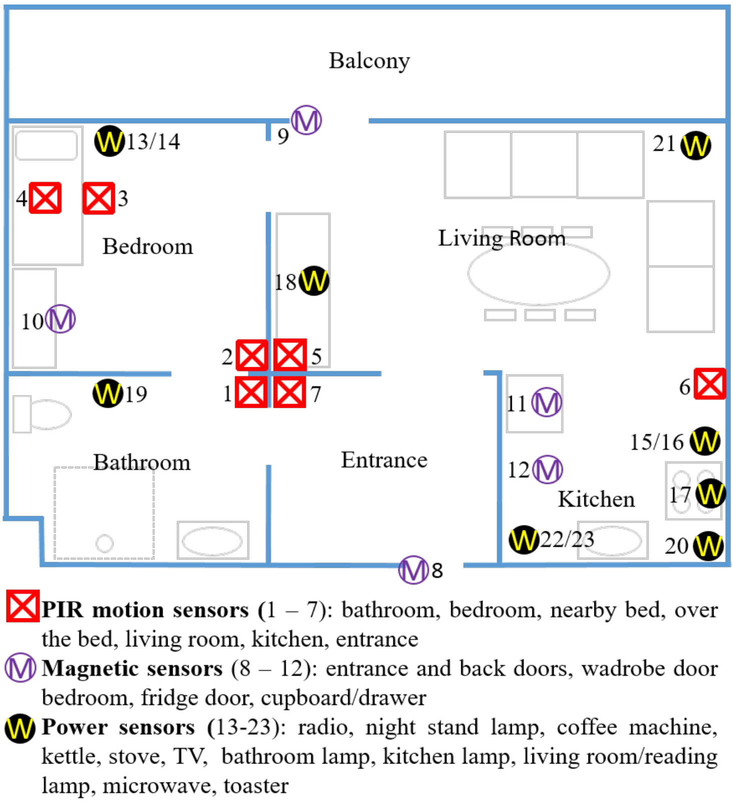



Figure 3.2: Proposed sensor system for field trial apartments.

Table 3.3 presents the age and gender of each apartment resident. Some health and cognitive ability scores are also given, each being scored by the resident themselves. These scores were acquired using RAND-12 [121] and the Cognitive Function Instrument (CFI) [78, 125]. RAND-12 uses a scale from 1 to 5, this being for poor (1), fairly good (2), good (3), very good (4), and

Table 3.2: Sets of sensors in each apartment in addition to the standard set of motion sensors

| Apt. ID | Power Sensors | Magnetic Sensors |
|---|---|---|
| 1 | Night stand lamp, coffee machine, living room/reading lamp, TV | Cupboard/drawer, entrance door |
| 2 | Night stand lamp, coffee machine, living room/reading lamp, microwave, TV | Fridge, entrance door |
| 3 | Kettle, living room/reading lamp, microwave, toaster | Fridge, cupboard/drawer, entrance door |
| 4 | Night stand lamp, coffee machine, living room/reading lamp, TV | Fridge, entrance door |
| 5 | Kettle, TV | Fridge, cupboard/drawer, entrance door |
| 6 | Night stand lamp, coffee machine, kettle, living room/reading lamp, TV, microwave | – |
| 7 | Night stand lamp, coffee machine, kettle, living room/reading lamp, TV | Wardrobe, cupboard/drawer, entrance door |
| 8 | Night stand lamp, TV | Wardrobe, entrance door |

excellent (5) perceived health. CFI is a questionnaire that gives a total score of between 0 and 14, 0 indicating no cognitive impairment and 14 indicating severe cognitive impairment. Two of the project's health experts also provided their subjective opinion on the physical and cognitive abilities of each resident using the RAND-12 scale. They also scored how active the person is in the home, i.e. their movement around the home and level of activity. The scale used for this was from 1 to 4 – passive (1), fairly active (2), active (3), and very active (4). The health experts noted that the residents of apartments 5 and 8 experienced memory deterioration during the study.

## 3.2 Data Preprocessing

The motion sensors only send activation messages ("1"). The corresponding "off" events ("0" message) were therefore inserted so that the data are consistent for every sensor. Motion sensor "off" events were therefore inserted when another motion sensor is activated or another type of sensor is activated in another room. Moreover, data acquired from binary sensors often contain faulty events such as erroneous activation of motion sensors by sunlight and motion sensor switch-off delays [39]. Such noise can significantly affect the performance of the models. The following preprocessing step was therefore introduced. Motion sensors occasionally do not send an activation event when they should. Missing

Table 3.3: Characteristics of residents participating in the field trial

| **Apt. ID** | **Age (Gender)** | **Self-perception** | | **Opinion of health experts** | | |
|---|---|---|---|---|---|---|
| | | *Health* | *Cognition* | *Physical* | *Cognition* | *Activity* |
| 1 | 90 (f) | 1 | 6 | 1/2 | 4/2 | 1 |
| 2 | 84 (m) | 3 | 1 | 3 | 5 | 2 |
| 3 | 71 (m) | 2 | 0 | 4 | 4 | 2 |
| 4 | 83 (f) | 3 | 5 | 4 | 4 | 3 |
| 5 | 80s (f) | 4 | 4.5 | 2/3 | 1 | 4 |
| 6 | 88 (f) | 2 | 6 | 1/2 | 4 | 1 |
| 7 | 95 (f) | 3 | 5.5 | 1/3 | 4 | 2 |
| 8 | 86 (m) | 3 | 1 | 3/4 | 4/2 | 4 |

events were therefore inserted to correct the data. For example, it is not possible to go into the bedroom from the kitchen without passing through the living room. The living room activation event was therefore inserted if it was missing. In cases where two sensor events are possible, such as where there are two paths through the apartment, then events are inserted such that the final percentage distribution of the two options remains as observed in the original data. The time of the inserted event is the mean of the previous and the next event. This does not compromise data accuracy as faulty events are usually associated with relatively fast motions around the apartment. The elapsed time between the events is therefore quite short.

## 3.3 Probabilistic Algorithms

Two probabilistic prediction methods were used in the thesis, *Active LeZi* (ALZ) and *Sequence Prediction via Enhanced Episode Discovery* (SPEED). Both methods convert sensor data into a sequence of *letters* and then identify sequence patterns. The patterns and their frequency of occurrence are used to generate a *tree*. This tree is then used to calculate the most probable next event. The prediction of the next symbol is performed by the Prediction Partial Matching algorithm (PPM) [25, 26]. Please refer to Section 3.3.3 for an explanation of the PPM algorithm.

Table 3.4 presents a possible scenario of actions performed by a resident and the corresponding sensors that are triggered. Each sensor is assigned a letter, as dictated by ALZ and SPEED and as shown in Table 3.5.

### 3.3.1 Active LeZi (ALZ)

ALZ is a sequence prediction algorithm that is based on a text compression algorithm [49]. The input to ALZ is a sequence of lower-case letters, each letter representing a specific sensor event. For example, the sequence for the

Table 3.4: Actions scenario

| **Action performed** | **Sensor event**[a] |
|---|---|
| Wake up | PIR bedroom (on) |
| Go to living room | *PIR bedroom (off)*, PIR living room (on) |
| Turn on TV | Power TV (on) |
| Go to kitchen | *PIR living room (off)*, PIR kitchen (on) |
| Turn on coffee machine | Power coffee machine (on) |
| Go to living room and watch TV while coffee is being made | *PIR kitchen (off)*, PIR living room (on) |
| Go to kitchen | *PIR living room (off)*, PIR kitchen (on) |
| Turn off coffee machine | Power coffee machine (off) |
| Go to living room | *PIR kitchen (off)*, PIR living room (on) |

[a]Events in *italic* have been inserted in the data preprocessing step.

Table 3.5: Assignment of letters to sensors

| **Sensor** | **Letter** |
|---|---|
| PIR bedroom | a/A |
| PIR living room | b/B |
| Power TV | c/C |
| PIR kitchen | d/D |
| Power coffee machine | e/E |

scenario described in Table 3.4 would be “abcdebdb”. Only activation events (i.e. “on” events) are included. ALZ uses the procedure dictated by the LZ78 text compression algorithm to generate patterns that occur in a sequence [136]. One small modification was, however, made to the procedure.

A given sequence $x_1, x_2, \ldots, x_i$ is parsed into $n_i$ sub-sequences $w_1, w_2, \ldots, w_{ni}$ such that for all $j > 0$, the prefix of the sub-sequence $w_j$ is equal to some $w_i$ for $1 < i < j$. For example for the sequence “abcdebdb”, the patterns found by LZ78 would be “a”, “b”, “c”, “d”, “e”, “bd”. Note that the last pattern “bd” was identified as “b” had already been found. ALZ also generates more patterns from the suffixes where this is possible. For example, “bd” would also generate another “d”. This accounts for patterns that were not perceived by the LZ78 algorithm but that are possible in the smart home environment. This increases the convergence rate of the model [49].

The frequency of occurrence of the patterns is counted when the sequence has been completely parsed and the patterns have been derived from this. An order-k-1 Markov *tree* is then constructed based on the *patterns* and their *frequencies*. $k$ corresponds to the longest pattern found in a training sequence. PPM (Section 3.3.3) is then used to calculate the next most probable event. The tree generated for the example scenario sequence of “abcdebdb” is shown in Figure 3.3.

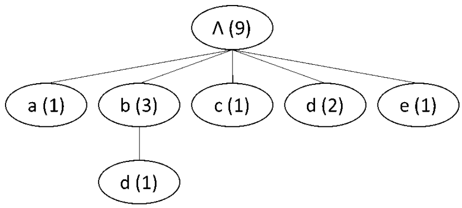



Figure 3.3: Tree generated by the ALZ algorithm for the sequence “abcdebdb”. The numbers within parenthesis represent the frequency of occurrence of the patterns found in the respective sequence.

### 3.3.2 Sequence Prediction via Enhanced Episode Discovery (SPEED)

SPEED is another sequence prediction algorithm. It is based on the occurrence of frequent patterns in home environments [2]. The main difference between SPEED and ALZ is the procedure used to find patterns in the sequence. SPEED defines an *episode* as the sequence between an initial and ending point of an activity. For example, the moment a coffee machine is turned “on” is the initial point of a coffee making episode, the episode lasting until the coffee machine is turned “off”. An “off” event cannot occur unless an “on” event has occurred. “Off” events therefore always occur after an “on” event for an activity (or sensor), and vice-versa.

The data received from the sensors in the smart home are represented using a sequence of letters. Upper-case letters represent a sensor “on” event and lower-case letters represent a sensor “off” event. The sequence for the example scenario presented in Table 3.4 would therefore be “AaBCbDEdBbDedB”.

The main idea behind the SPEED algorithm is to extract *episodes* from a sequence of data and derive *patterns* from these episodes. The patterns are then used to generate a *tree* that keeps track of the learned episodes and their frequencies. The *height* of the tree is the length of the longest episode, which defines *maximum episode length*. The algorithm, for every event in a sequence, searches for the opposite event in following events. If the opposite event can be found, then an episode is found. The first episode sequence found in the example scenario is “Aa”. The patterns generated from this episode are “A”, “a”, and “Aa”. Table 3.6 shows the episodes and patterns derived from the example sequence. We keep track of these and count the frequency of occurrence to generate an order-k-1 Markov model, $k$ being the maximum episode length. A

Table 3.6: SPEED episodes and patterns for the sequence "AaBCbDEdBbDedB"

| Episode | Patterns found |
|---|---|
| Aa | A, a, Aa |
| BCb | B, C, b, BC, Cb, BCb |
| Ded | D, e, d, De, ed, Ded |
| Bb | B, b, Bb |
| dBbD | d, B, b, D, dB, Bb, bD, dBb, BbD, dBbD |
| Ded | D, e, d, De, ed, Ded |
| bDedB | b, D, e, d, B, bD, De, ed, dB, bDe, Ded, edB, bDed, DedB, bDedB |

tree for the example sequence is presented in Figure 3.4. The PPM algorithm is then finally used for prediction.

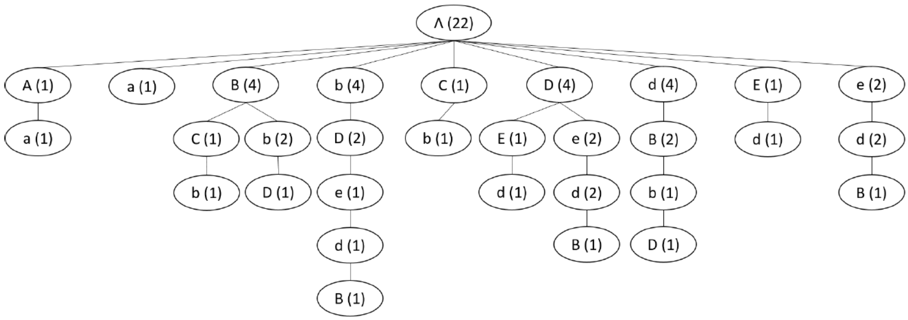



Figure 3.4: Tree generated by the SPEED algorithm for the sequence "AaBCbDEdBbDedB". The numbers within parenthesis represent the frequency of occurrence of the patterns found in the respective sequence.

### 3.3.3 Prediction by Partial Matching (PPM)

The PPM algorithm calculates the probability distribution of each possible event in a given sequence by taking into consideration the different order Markov models and the different weights [25, 26]. The weights are given by the escape probability for the model, for going from a higher order to a lower one. The predicted symbol is the symbol with the highest probability.

ALZ and SPEED use slightly different PPM strategies. ALZ uses the exclusion strategy. This means prediction is performed using the suffixes of the given sequence, but not the sequence itself. Therefore, for the sequence "bd", the patterns used to calculate the probability of each letter being next are "b" and the null context. For example, if we want to calculate the probability of a "c" after "bd" using ALZ, based on the tree in Figure 3.3, then the probability would be given by Equation 3.1. In an order-2 model, the probability of a "c" after a “b” is 0/3 and the probability of escaping to order-1 is 2/3. In order-1, the probability of a “c” after a null context is 1/9.

In SPEED, however, the patterns used for calculating probabilities after a certain sequence would be all the suffixes, including the sequence itself. For the sequence “dB”, the patterns used would therefore be “dB”, “d”, and the null context. The probability of a “b” after this sequence, based on the tree in Figure 3.4, would be as given by Equation 3.2. We start in order-2, where the probability of a “b” after “dB” is 1/2 and the probability of escaping to the lower order is 1/2. In order-1, the probability of a “b” after “d” is 0/4 and the probability of escaping to the lower order is 2/4. Finally in the lowest order, the probability of a “b” after a null context is 4/22.

$$p(c, bd) = \frac{0}{3} + \frac{2}{3}\left(\frac{1}{9}\right) = 0.074 \tag{3.1}$$

$$p(b, dB) = \frac{1}{2} + \frac{1}{2}\left(\frac{0}{4} + \frac{2}{4}\left(\frac{4}{22}\right)\right) = 0.545 \tag{3.2}$$

## 3.4 Long Short-Term Memory Recurrent Neural Networks (LSTM RNN)

The converted data used in ALZ and SPEED were also used as input for the Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). RNN maintains an internal *memory* and has therefore been broadly applied to sequence prediction. It achieves good performance for inputs that are sequential in time and has been applied, for example, to text generation [74], speech recognition [51], and pattern recognition in music [47]. LSTM [53] is an RNN architecture that is designed to be better at storing and accessing information than the standard RNN [50].

We used an LSTM network configured as a text generation network. The number of inputs is a certain number of previous sensor events and is equal to the *memory length* (i.e. number of previous events used to predict the next one). The output is the predicted next event in the sequence (Figure 3.5). The input and output are one-hot encoded. One-hot encoding represents each symbol using a vector of bits that has a length that is equal to the number of symbols in the sequence. All values are zero, except for the value that corresponds to the symbol (Figure 3.5). The number of values in our input vector for the one-hot LSTM encoding is equal to the number of symbols in the sequence. For example, there are 30 values in the input for 15 sensors, each having an “on” and “off” state.

The LSTM network models were implemented in Python 3 using Keras open source library for neural networks [24]. A number of hyperparameters and parameters were first tuned using grid search and then manually tweaked to find the optimal values. Optimal values were as follows: one hidden layer with

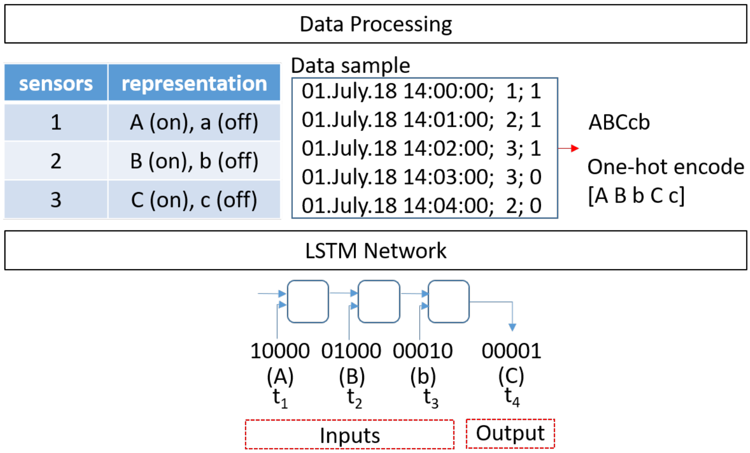



Figure 3.5: LSTM network configuration.

hyperbolic tangent activation and 64 neurons; batch size (i.e. number of samples used for training each iteration of the epoch) of 512; *Adam* as the optimisation function (adaptive learning rate optimisation algorithm [67]) with a learning rate of 0.01 and categorical cross-entropy as loss function; output layer that used a *softmax* activation function. The early stopping method was also applied, allowing a maximum of 200 epochs for each model's training, and a dropout rate of 50%. They were both used to avoid overfitting. We used weights in the training process to balance the number of samples for each sensor. These are computed using the "`compute_class_weight`" function from the Scikit-learn open source library [95]. The weight corresponds to the total number of samples divided by the number of occurrences of the class.

## 3.5 Inclusion of Time to Sequential Data

The time is incorporated by adding a number to the letter to signify a slot of time. The letters assigned to each sensor are kept as before. Generated sensor events are, in all cases, treated as independent events. This means that the number of inputs to the LSTM where there are four possible time slots for each sensor and 15 sensors with "on" and "off" events, is $4 \times 15 \times 2 = 120$.

In the following, the types of time slots that were investigated are described.

### 3.5.1 Sensor Event and Period of Day

Four periods of the day are distinguished: morning (from 7am to noon), afternoon (from noon to 6pm), evening (from 6pm to 10pm), and night (from 10pm to 7am). The period of the day is indicated by adding a number between 0 and 3 to the letter for the event. For example, the symbol "A0" would be generated for a motion sensor event in the bedroom going "on" in the morning.

### 3.5.2 Sensor Event with Class Intervals

A number is added to the sensor's letter to indicate the slot of time that elapses to the next event. We define a set of 4 time-class intervals [$<$ 1min, 1-15min, 15min-1h, $>$ 1h] and a set of 8 time-class intervals [<1min, 1-5min, 5-15min, 15-30min, 30min-1h, 1-2h, 2-5h]. We therefore assign the numbers 0-3 or 0-7 to the event, depending on the intervals used. For example, the symbol "A1" would be generated if the motion sensor in the bedroom (assigned letter a/A) is activated in the morning and 10 minutes later the person went to the bathroom.

### 3.5.3 Sensor Event and Time-Cluster with Hour of the Day and Elapsed Time to the Next Event

The K-means algorithm is an unsupervised learning method that was used for clustering sensor samples. In the K-means algorithm, the dataset samples are classified into K clusters such that the sum of square distances (SSD) within the clusters is minimised [9]. In this study the algorithm was used to cluster the samples of each sensor by the hour of the day in which they occurred and the time elapsed to the following sensor event. Each cluster contains a centroid, which is given by the mean value of each feature of the algorithm. K-means is performed for between 1 and 8 clusters (K), the best K being chosen manually using the elbow method [61]. The elbow method consists of plotting an SSD vs. K graph and choosing the K that resembles an "elbow" (the point of inflection on the curve). This is the best fit for that problem. Figure 3.6 shows an example of the clustering of samples for the motion sensor in the kitchen. This sensor gives four clusters each represented by a different colour and uses the elbow method based on the graph in Figure 3.7. If this sensor is represented by the letter B, has an "on" event at noon (blue cluster), and the next sensor event took place 3 minutes later, then this would generate "B2" (2 representing the blue cluster).

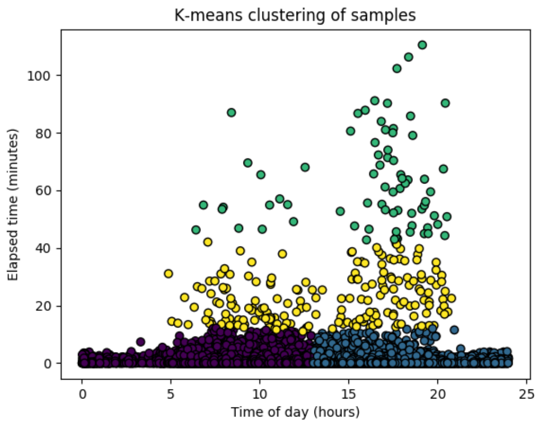



Figure 3.6: K-means clustering of samples of motion sensor events in the kitchen.

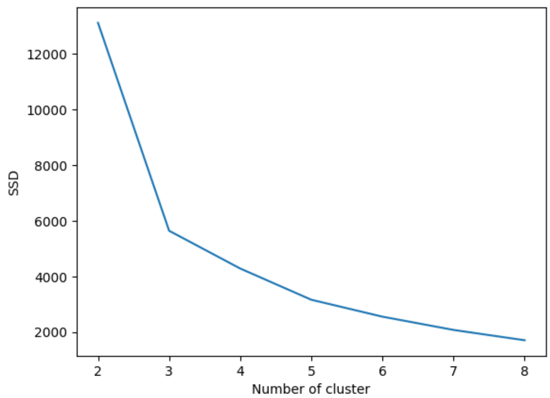



Figure 3.7: SSD vs. number of clusters for motion sensor events in the kitchen.

### 3.5.4 Separate Dataset and Model for each Period of the Day

A separate dataset for each period of the day was used in this approach. The periods were morning (from 7am to noon), afternoon (from noon to 6pm), evening (from 6pm to 10pm), and night (from 10pm to 7am). Each dataset was modelled in separate neural networks.

## 3.6 Activity Recognition and Prediction

Binary sensor events indicate activities of daily living and can therefore be associated with specific activities. The limited number of sensors in our set-up, however, meant that only high-level activities could be registered. The following classes were included in the dataset, after grouping sensor events: watching TV, being in bed, being out, bedroom activities, living room activities, kitchen activities, bathroom activities and transitions between bedroom/bathroom/entrance/living room. In total 11 classes.

Two rule-based algorithms were implemented to derive the activities from binary sensors. These are referred to as *sequential* activities and *concurrent* activities. The set of rules is described in Table 3.7. In *sequential* activities, no more than one activity takes place at the same time, another activity starting as soon as one activity ends. Time information is the elapsed time to the next activity, which in this case is the duration of the activity. In *concurrent* activities, each activity has a start and an end, indicated by a “1” and “0”. Several activities can therefore occur in parallel. For example, the resident can be in the kitchen preparing coffee and still be watching TV. This implies that the duration of concurrent activities can often be longer than sequential activities. Time information is inserted such that the activity *start* contains the *duration* of the activity (time elapsed until the end of the activity) and activity *end* contains

Table 3.7: Rules for deriving activities from sensor events

| **Activities** | **Rules** |
|---|---|
| Kitchen activities | Whenever power and magnetic sensors located in the kitchen are activated or motion sensor in the kitchen is active for more than 1 minute. |
| Living room activities | Whenever power and magnetic sensors located in the living room (except TV) are activated or motion sensor in the living room is active for more than 5 minutes. |
| Watching TV | Whenever the resident is in the living room for more than 5 minutes and the power in the TV is on. |
| Bedroom activities | Whenever power and magnetic sensors located in the bedroom (except sensors around the bed) are activated or motion sensor in the bedroom is active for more than 5 minutes. |
| Being in bed | Whenever motion sensors around the bed are consecutively activated for more than 5 minutes. |
| Bathroom activities | Whenever the motion sensor located in the bathroom is active for more than 1 minute. |
| Being out | Whenever the entrance door “off” and “on” events happen consecutively and for more than 5 minutes; or when the entrance door is the last active motion sensor for more than 10 minutes (for an apartment in which the entrance door was not installed). |
| Transitions | Being in the entrance is always considered as a transition as there are no relevant activities in that area. Other rooms have a subjective transition time chosen based on the distance between rooms and conditions of the residents (e.g. walking speed, use of rollator, etc.). |

the *elapsed time* to the start of the next activity event. Figure 3.8 shows an example of the two sequences, for simplicity without the time.

The method used for the prediction of sensor events using the LSTM network was also applied to the prediction of activities. Each activity is assigned a letter, the time information being included as a number indicating the K-means time-cluster. Only the input data for the LSTM networks uses the transition classes. There are therefore only *seven* output classes. The LSTM network has the same parameters configuration as is used for the prediction of sensor events. We also, because our dataset is imbalanced, use the Synthetic Minority Oversampling Technique (SMOTE). SMOTE is an over-sampling technique that creates synthetic samples for the minority classes [91]. The Imbalanced-Learn

| activities | representation |
|---|---|
| Watching TV | A (on), a (off) |
| Preparing coffee | B (on), b (off) |
| In bed | C (on), c (off) |

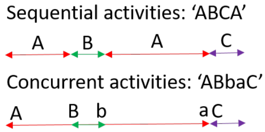



Figure 3.8: Types of activity sequences, sequential and concurrent. The example represents a scenario in which the resident watches TV, goes to the kitchen to prepare coffee while watching TV, and then goes to bed.

library was used to implement this [71].

## 3.7 Transfer Learning

Transfer learning, as mentioned in Section 2.4, involves a model being trained using a *source* dataset, the trained model then being applied to a different and somewhat related *target* dataset [94]. The trained model often needs to be fine-tuned using data from the target dataset. Transfer learning can result in a reduction in the time required to train the model and lower volumes of the target training data. It also provides a solution for unlabelled data. Transfer learning furthermore allows some characteristics in the training and testing datasets to be different, such as labels and data distributions. It is therefore well suited to and very useful in the field of smart homes, as each home usually has a unique layout, a different network of sensors and residents with different habits.

Different sets of power and magnetic sensors were installed in the eight field trial apartments (Table 3.2). Sensors that relate to the same activity were re-labelled in the tests which compare prediction accuracy and apply transfer learning across apartments. The new labels and the sensors assigned to these are given in Table 3.8. Lamp power sensor and wardrobe door magnetic sensor events were removed from the datasets in these tests, as it was not possible to associate them with an activity that was common to most of the apartments.

Table 3.8: Re-labelling of sensors

| New labels | Sensors |
|---|---|
| Kitchen sensor | P[a]: toaster, microwave; M[b]: fridge, cupboard/drawer |
| Beverage sensor | P[a]: coffee machine, kettle |

[a]Power and [b]magnetic sensors.

Chapter 4

# Research Summary

This chapter provides an overview of the thesis papers, their relation to each other and to the research questions defined in the thesis. This is followed by a summary of each paper, the paper's motivations and main contributions.

## 4.1 Overview

This thesis is composed of six research papers that address the research questions introduced in Section 1.3. Please refer to Figure 1.3 for the relations between the papers and the research questions.

The research questions require the application and evaluation of state-of-the-art sequence prediction methods to data from real smart homes in which a limited number of binary sensors were installed. The literature survey has pointed out two probabilistic methods, ALZ and SPEED, and recurrent neural networks. These have been previously applied in this domain and have been shown to achieve good performance. These were therefore the chosen methods for this thesis.

An in-depth comparison of the two probabilistic methods for the prediction of the *next sensor* event was carried out first. A *2-week* dataset from *one* apartment was available at this point in time. Factors such as peak accuracy, memory length, amount of data required for convergence, and number and type of sensors in the dataset were analysed and the performance for each algorithm was compared. This resulted in Paper I. The LSTM network was also compared with a baseline method for these aspects and this task. A *17-week* dataset for this apartment was used. This resulted in Paper II.

Paper III presents the comparison of the ALZ and SPEED probabilistic methods with the LSTM networks for the prediction of the next sensor event. A larger dataset was used, *30 weeks* for the same apartment as used in the previous papers. This provided a more solid and more thorough comparative analysis.

The best-performing algorithm for predicting the next sensor event was the LSTM network with SPEED-text. We investigated whether this algorithm could be improved by incorporating the time component, using four different ways of including slots of time. We explored this in Paper IV. We also carried out predictions of both the next sensor event and its mean time of occurrence within a single model. These methods were applied to *40 weeks* of data from the same apartment as for the previous datasets.

The papers cited so far address **SRQ 1**, which addresses the evaluation of the performance of state-of-the-art methods for the prediction of the next sensor event and time of occurrence information in real homes, for a limited number of sensors.

We subsequently investigated the extent to which the best prediction model can generalise its performance to other apartments and residents. Data from *four more* apartments were available at this point in time. We first modelled a LSTM network for each apartment individually. Then we experimented transfer learning in two ways. This work is presented in Paper V, and addresses **SRQ 2**.

The study is concluded by Paper VI. Data from the eight apartments in the field trial were available at this point in time. This paper confirms the results of the previous papers that were acquired using data from fewer apartments. We also confirmed the behaviour of the algorithms by applying these to data from all apartments. This paper therefore addresses both **SRQ 1** and **SRQ 2** through expanding the analysis to more apartments. We also carried out activity recognition from binary sensor events using a rule-based algorithm. This allowed us to analyse the performance of activity prediction rather than sensor events. This work addresses **SRQ 3**.

## 4.2 Papers

### 4.2.1 Paper I

**Occupancy and Daily Activity Event Modelling in Smart Homes for Older Adults with Mild Cognitive Impairment or Dementia**

This paper presents an in-depth evaluation and comparison of two state-of-the-art *probabilistic* methods in the prediction of the *next sensor* to be activated or deactivated in a smart home. The goal of this work was to investigate the performance of these algorithms in a *real* home scenario. The performance of these methods had previously primarily been evaluated using data acquired from labs and testbeds for about 50 sensors and for scripted activities.

The two algorithms, ALZ and SPEED (Section 3.3), were applied to data collected over a period of *two* weeks from a single real home fitted with 15 sensors, including motion, magnetic, and power sensors.

We first reproduced the results of both algorithms as in the original work, using the same dataset for both training and testing. However, this procedure leads to overfitting. Our dataset therefore uses a separate training and testing set. We also modified the method, to improve accuracy, by introducing a validation step that calculates the optimal number of last events on which to base the prediction, i.e. the window that leads to the maximum overall prediction accuracy, which we refer to as the *optimal window*. We refer to this parameter, in the following papers, as *memory length*.

ALZ and SPEED were compared regarding several factors. The first test analysed the prediction accuracy according to the optimal window for each algorithm, and for different sizes of training set. Smaller window sizes (1-4 events) provide better accuracy, for both algorithms. The accuracy deteriorates very quickly with increasing window size. This behaviour is mainly due to two facts: (i) the smart home has a small number of sensors, so that there are not several patterns for the same actions in the home; and (ii) long sequences of events are not bound to be repeated frequently and are, therefore, more difficult to provide correct predictions.

A second test evaluated peak accuracy and accuracy in relation to the size of the training set. SPEED achieved an accuracy of 75% with an optimal window of two and ALZ an accuracy of 53% with an optimal window of one. SPEED collects a significantly higher number of contexts and frequencies. This may be the reason why SPEED leads to better accuracy (see for example trees generated by ALZ and SPPED for the same sequence of events in Figures 3.3 and 3.4). Maximum accuracy was achieved using SPEED where training sets were larger than around 800 events. ALZ, however, reached a maximum accuracy for a training set of 300 events or more. ALZ therefore converges to its maximum accuracy faster than SPEED, but achieves a much lower prediction accuracy.

A last test was performed to reveal the dependence of the prediction accuracy on the number and type of sensors. Four alternatives were investigated: all sensors (15), only motion sensors (7), only motion and magnetic sensors (11), and only motion and power sensors (11). Accuracy did not, in most of the cases, change significantly with the number of sensors in the dataset. A clear exception was when SPEED is applied to a dataset of only motion sensors. The prediction accuracy was then very poor (50%). This is due to only motion sensors being included, the longest episode being two events long (i.e. "on" and "off" events for the same sensor consecutively). ALZ is therefore better suited where events are not highly intertwined. The prediction accuracy for events that involve magnetic sensors is relatively high, as doors and drawers are often closed right after they are opened. This makes this a relatively easy pattern to predict. On the other hand, power sensor events can occur somewhat randomly, many other events occurring between power events. This makes the prediction of associated events more inaccurate.

*This paper resulted in a comparative analysis of two probabilistic methods that were applied to data from one apartment with 15 sensors, collected over a period of two weeks. These algorithms have been broadly used in the research literature, however with data collected from labs and testbeds, including many binary sensors (about 50). SPEED achieves better performance when all types of sensors are used in the dataset, and both algorithms could achieve convergence with about 2 days of data.*

### 4.2.2 Paper II

**Sensor Event Prediction using Recurrent Neural Network in Smart Homes for Older Adults**

The main goal of this paper was to evaluate the performance of the LSTM network to predict the *next sensor* activation event when applied to data collected from a *real* home. This work is, to the best of our knowledge, the first in the research literature to apply LSTM networks to the prediction of sensors in smart home environments. Data from the apartment referred to in Paper I had, at this point in time, been collected over a period of *17 weeks.* A baseline method was also implemented for comparison purposes.

The literature survey indicates that neural networks perform well in smart home environments. *Recurrent* neural networks are usually used, as they maintain an internal memory that is suitable for inputs that are sequential in time. We chose to apply an LSTM network, as it has been shown to be promising in several applications, including text generation, which is similar to our application. In our setup, the LSTM network used ALZ-text (only "on" events) as input, as described in Paper I for the ALZ method. We compared its performance with a baseline method, which is based on a table in which the probability of each sensor being the next activated depends on the preceding sensor(s). The predicted next sensor event is the one with the highest probability of being activated right after the last sensor(s) in the sequence.

The data were collected over 17 weeks. We therefore analysed the prediction accuracy for 2, 13, and 17-week datasets. We also had datasets that contained data from only motion sensors, and from all sensors in the apartment. A first test consisted of analysing the optimal memory length of each algorithm. A memory length of three events led to the highest accuracy for the baseline method, whilst the LSTM network required six or more events. This shows that LSTM networks are more effective in the learning of patterns and finding temporal relations between features.

The LSTM network obtained a peak prediction accuracy of 69% for the set with all sensors and 75% for the set with motion sensors only. The baseline method achieved a 58% prediction accuracy for all sensors, and 67% for motion sensors. Increasing the size of the data did not have a significant effect on the accuracy of the baseline. Accuracy for the LSTM network, however, increases steadily for larger training dataset sizes.

For the dataset sizes we investigated, the accuracy increased by about 8-10% for the set with only motion sensors in relation to the set with all sensors. When a limited amount of data (2-week dataset) is used the baseline achieves better accuracy than the LSTM. An accuracy of 67% is achieved with two weeks of data for motion sensors only, and of 61% for 13 weeks of data from all sensors. The LSTM network achieves better accuracy than the baseline when the amount of data is sufficient (13 and 17-week datasets). Indeed, as the amount of data increases, the model improves from 60% to 67% mean accuracy for all sensors and from 70% to 74% mean accuracy for only motion sensors. This confirms that two weeks was an insufficient amount of data. The model tends to overfit then, so leading to lower accuracy on the test set.

*The work resulted in an evaluation of LSTM networks for the prediction of the next sensor event on data collected in a real apartment over a period of 17 weeks. The LSTM network performed better than the implemented baseline method when a sufficient amount of data (more than 2 weeks) were available.*

### 4.2.3 Paper III

**Comparison of Probabilistic Models and Neural Networks on Prediction of Home Sensor Events**

In this paper the two probabilistic methods, ALZ and SPEED, which were compared in Paper I, and LSTM networks which was compared with a baseline in Paper II, are compared. ALZ-text and SPEED-text input was used in all methods. To the best of our knowledge, this was the first time such a comprehensive comparison of state-of-the-art methods for the prediction of next sensor event has been conducted using data collected from real apartments. The data were collected over a period of *30 weeks* from *one* home. We analysed the methods using a dataset with all 15 sensors and with only the seven motion sensors.

The work showed that probabilistic methods can achieve a high prediction accuracy (close to their peak accuracy) with a relatively small amount of data (typically 2 days of data). LSTM networks, however, require a larger dataset (about 3 weeks for SPEED-text and 10 weeks for ALZ-text) to reach accuracies close to the peak. Probabilistic methods were also found to base prediction on a relatively small number of previous events. An optimal memory length of four for ALZ and three for SPEED was established. LSTM networks, however, base prediction on a sequence of eight last events or more. This indicates that these networks are better at finding longer-term dependencies and patterns in a sequence of events. The accuracy attained in LSTM networks is also quite stable for memory lengths that are larger than the optimal. Probabilistic methods, however, have an optimum memory length, accuracy decreasing both for shorter and for longer memory lengths than the optimum. For the most accurate models (the ones with SPEED-text as dataset), the LSTM required 1/7 of the time SPEED required for training and testing.

Our best result (83%) for the dataset containing events from the 15 sensors was achieved by the LSTM network with SPEED-text. SPEED achieved an accuracy that was only 1% lower. This, however, required a considerably longer training time. SPEED may therefore be a good choice in applications where modelling using a small amount of data is an advantage and in which execution time is also not too critical, as an accuracy close to the peak can be achieved with little data. Our results have, in general, shown that it is possible to achieve accuracy close to the peak with little data. SPEED and LSTM with SPEED-text achieved better results than ALZ and LSTM with ALZ-text. This is not surprising, as the conversion of data to SPEED-text sequences gives more information (both “on” and “off” events). This can also be confirmed by the trees formed by ALZ and SPEED (Figures 3.3 and 3.4). The best choice for a dataset with no intertwined events, such as for our dataset with only the seven motion sensors, is the LSTM with SPEED-text. SPEED does not work well on this dataset. The tree has a height of two, meaning that only “off” events can be predicted reliably, as also found in Paper I.

Another interesting finding is that more than 10 weeks of data does not significantly improve the results of any of the methods applied. A change in the algorithms and/or in the way the data are input, or the provision of additional information would therefore be required to improve prediction accuracy.

Finally, a larger number of sensors could potentially lead to better prediction accuracy, as this entails more information on which a prediction can be based. A smaller number may, however, be preferable to reduce surveillance, lower costs, and reduce the impact upon home aesthetics. Our work shows that it is possible to achieve acceptable prediction accuracy with few sensors.

*This work resulted in a thorough comparative analysis of the use of probabilistic methods and LSTM networks on the prediction of the next sensor event in a smart home. The algorithms were applied to a dataset with sufficient data (30 weeks), collected from one real apartment. The LSTM network using SPEED-text performed best in our setup. It provides the best peak accuracy, requires the lowest training and testing execution time, shows good performance for different types of datasets (i.e. both with all types of sensors and with only motion sensors in dataset), and converges with a limited amount of data (three weeks).*

### 4.2.4 Paper IV

**Prediction of the Next Sensor Event and its Time of Occurrence in Smart Homes**

In this paper, sequential *sensor* events and the *time* of occurrence information were predicted using data collected from *one* apartment over a period of *40 weeks*. We used the best-performing algorithm from our previous analysis (Paper III), which is the LSTM network with SPEED-text. We included time information in the model. We first predicted the next sensor event based on previous sensor events and time information. We also predicted both next sensor event and time information in a single model.

The time information was incorporated in several ways: period of the day (morning, afternoon, evening, night), 4-class and 8-class time-intervals, and K-means time-cluster (Section 3.5). We also investigated performance for separate networks for the four periods of the day.

The accuracy improved by 1-1.4% for 4-class and 8-class time-intervals and K-means time-cluster methods when we included time information in the input for the prediction of the next sensor event. Our best performing model for predicting the next sensor event included a 4-class time-interval information. It achieved a peak average accuracy of 84% for a set of 15 sensors, motion, magnetic, and power sensors. This is 1.4% better than without including the time information, and only 0.2% better than when using K-means time-cluster. The time elapsed between events therefore contains information that improves prediction, however, only marginally. The apartments are quite small and there are a limited number of sensors. The information is therefore still very limited. The accuracies for the separate networks are marginally lower than where all the data are in one network. This is as expected, as information is lost when separating into networks, more data therefore being required to compensate for this.

Our best prediction of both the next sensor event and the time of occurrence information was obtained using K-means time-clusters. This implementation attained an accuracy of 80% for a set of 15 sensors (4% higher than using 4 and 8-class intervals), for a dataset with 70000 events, which corresponds to approximately 20 weeks of data. This method also attained an accuracy of 83.4% for a set of seven motion sensors.

Little work has been reported in the literature on the prediction of the time of occurrence together with sequential sensor events in smart homes. There is, to the best of our knowledge, only one work in the literature that predicts the next event and its time of occurrence information in the same model. Bayesian networks are used in this work [86]. Our method attains better overall accuracy when predicting both the sensor event and its time of occurrence information in the same LSTM network.

*This work resulted in an analysis of different input sequences for LSTM network with SPEED-text, the time component being included in the prediction models. The dataset was collected over a period of 40 weeks from one real home. We carried out a prediction of the next sensor event based on preceding sensor events and time information, and a prediction of the next sensor event and time of occurrence information using both these as input. Including the time as a K-means time-cluster of each sensor samples led to the best results.*

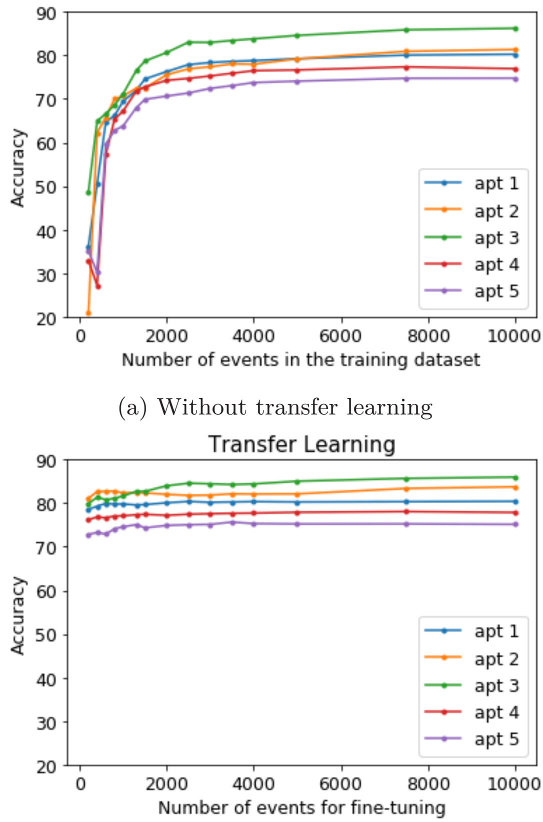
### 4.2.5 Paper V

**Prediction of Next Sensor Event and its Time of Occurrence using Transfer Learning across Homes**

This paper presents the results of sensor event prediction and time of occurrence prediction using *transfer learning* between apartments. The motivation for transfer learning in this work was to evaluate the generalisation of the model to other apartments, reduce the time required for data collection in individual apartments before the system can be operated, and reuse the learning of previously trained models. LSTM networks with SPEED-text were used. We implemented two different ways of transfer learning and compared this with each apartment being modelled individually, i.e. without transfer learning. We, in addition to the usual data preprocessing steps, also re-labelled the sensors that relate to the same activity, so that the sensors are similar for all apartments (Section 3.7). Data from *five* real apartments were used in this study, collected over different periods of time (69-291 days).

We first trained an individual model for each apartment. Then we applied transfer learning as follows: (i) train the LSTM network with data from four apartments and fine-tune with data from the target apartment, and (ii) train with data from one apartment (that had best accuracy when modelled individually) and fine-tune with data from the target apartment. The data from the target apartment, which had not been used in training, were split and used for fine-tuning and testing the network.

Accuracies when predicting the next sensor event for each apartment modelled using its own LSTM network, were in the range 81-87%. Accuracies when applying transfer learning, without fine-tuning, were in the range of 66-77%. These lower accuracies show that fine-tuning the model is indeed required to achieve good prediction accuracy. The mean peak accuracies achieved after fine-tuning were 81-86% for both tests (when the source model was trained




(a) Without transfer learning



(b) With transfer learning

Figure 4.1: Accuracy of prediction of the next sensor vs. number of events in the training dataset using (a) individual LSTM networks and b) transfer learning with fine-tuning.

using four apartments and using the one best performing apartment). These accuracies are marginally lower than when each home is modelled individually. The accuracy when using transfer learning with very little data (less than a day), is much higher than for the apartment modelled using its own network with the same amount of data, see Figure 4.1. The results are similar for predicting both the next sensor event and its K-means time-cluster. Accuracies are 74-81% when modelled individually and 73-81% when using transfer learning.

Transfer learning has been shown, for our set of apartments, to work successfully up to a certain number of events. Where the number of events in the training dataset is low (up to around 4000 events), then transfer learning leads to higher prediction accuracy than each apartment modelled individually. This means that the prediction algorithm can work well straight away where a new apartment is added to the study, and attain a relatively good accuracy

(70-80%) from the first day. The prediction accuracy for larger training datasets is, however, approximately the same. In fact, it is in most cases marginally higher when each apartment is modelled individually.

*This work resulted in an evaluation of the use of transfer learning across five of the apartments that were included in the field trial. Fine-tuning was shown to be required for transfer learning to be successful between the homes. It was also shown that good prediction accuracy could be achieved from the first day (70-80%) where a new apartment acquires the system, where transfer learning is applied.*

### 4.2.6 Paper VI

**Predicting Sensor Events, Activities, and Time of Occurrence Using Binary Sensor Data from Homes with Older Adults**

We, in this work, firstly expand the comparisons and analysis performed in the previous papers by applying the methods to data collected from the *eight* apartments in our field trial. Each apartment has 13-17 binary sensors and different periods of data collection. Our best performing implementation, the LSTM network with SPEED-text, achieved an accuracy of 77-87% for predicting the next sensor event, and an accuracy of 73-83% when predicting both the next sensor event and the mean time elapsed to the next sensor event (K-means time-cluster). One hypothesis for the 10% variability is that there are different sources of noise in the data for each apartment. For example, residents are often visited by family members and healthcare assistants. Another aspect could be that some people are more predictable in their patterns around the apartment than others.

The conclusions in Paper V were confirmed when applying transfer learning across the apartments. One apartment, however, presented a different curve behaviour. The accuracy for this apartment was not good from the start, although it required less data (about 4000 events) than the case without transfer learning (about 5000 events), see Figure 4.2.

The very new contribution of this paper was *activity recognition* and *prediction* in smart homes using the binary sensors. Two rule-based methods for associating sensor events with activities were implemented. We refer to these as *sequential* and *concurrent* activities, as explained in Section 3.6. Each activity is assigned a letter, as for sensor events. The time information represented as K-means time-cluster was selected based on the best performance in previous analysis. The *transition between rooms* classes are only used in the input for the LSTM network. There are therefore only seven output classes (watching TV, being in bed, being out, bedroom activities, living room activities, kitchen activities, bathroom activities).

The accuracy results for the *concurrent* activities dataset were better in all cases. Accuracy was 5.4-14% higher than when predicting only the next activity based on previous activities and time; and 5.1-16.1% higher when predicting both the next activity and time-cluster. There is, however only one resident and a relatively small number of sensors in each apartment. Sensors also do not

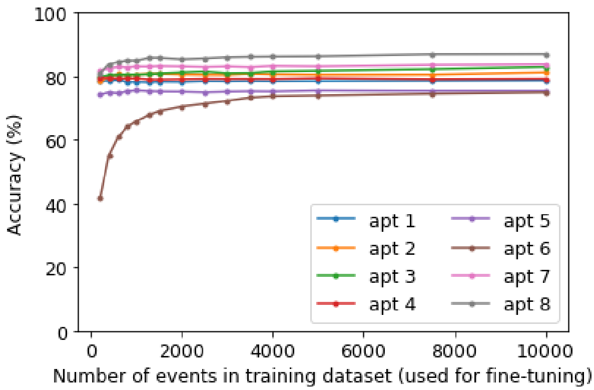



Figure 4.2: Accuracy of prediction of the next sensor event vs. number of events used for fine-tuning. Source model trained with data from seven apartments, fine-tuning with and testing on the target apartment.

relate in a high degree to other sensors. There are therefore, in reality, only few concurrent activities. Most of the "start" activity events in the concurrent dataset are therefore immediately followed by the "end" of the same activity. Most of the "end" of activities are therefore predicted with 100% accuracy. This explains the higher accuracies for this method. This implementation may, nevertheless, be a good option for smart home environments where several activities can occur at the same time, e.g. multi-resident smart homes. This is not the case for our setup. The *sequential* activity dataset is therefore probably a fairer algorithm.

It is possible to investigate in which activities confusions have occurred. They are in fact similar for both types of datasets. *Bedroom activities* are mostly predicted as *in bed* and *kitchen activities*. This is understandable as bedroom activities often take place after having been in bed or in the living room, which has access to the kitchen. *Living room activities* are confused with *watching TV*, and to a lesser extent with kitchen activities, for the same reasons as stated in the previous comment. Finally, *being out* has been often predicted as *kitchen activities*, as the entrance door also has a connection to the living room. An interesting result is that the *watching tv* activity has, in the example apartment, been very well predicted at 86.5%. This could be useful for smart functions involving the TV, e.g. if the resident has difficulties operating the remote control. *Bathroom* and *kitchen activities* have also shown good accuracy (77.9% and 83.2%). This range of accuracy may be useful in the analysis of patterns in the home, potentially for anomaly detection.

Our best model achieved 75-95% accuracies for predicting the next activity for the concurrent activity dataset. The best model achieved an accuracy of 64-85% for predicting the next activity and the mean duration and time of occurrence. The results for the sequential activity dataset are worse. Our best model achieved 62-90% accuracies for predicting the next activity, the best model achieving 50-80% accuracies for predicting the next activity and its duration and time of occurrence.

*This work resulted in a comprehensive study of state-of-the-art algorithms for the prediction of sensor events and activities of daily living in smart homes, and time of occurrence information. The methods used in the previous papers were applied to data from all the apartments in the field trial (eight). The results were confirmed when the evaluation of the methods was expanded to all the apartments in the field trial, and so were the conclusions for transfer learning across apartments. Rule-based methods for associating sensor events with activities were also implemented. Activity and time of occurrence information were then predicted. Activity prediction has, in most cases, not achieved better prediction accuracy than sensor prediction.*

Chapter 5

# Discussion

This chapter discusses the findings of the thesis that address the research questions. Limitations faced throughout the development of the project are also discussed. Conclusions on the accomplishments of the thesis are presented at the end of the chapter, and future work is discussed.

## 5.1 Research Questions

The research question was defined and divided, in Section 1.3, into three sub-research questions (**SRQ**). The findings for each sub-research question are discussed in this section.

### SRQ 1 – Performance of State-of-the-art Prediction Methods on Data from Real Homes

The performance of two probabilistic methods, ALZ and SPEED, and LSTM networks using ALZ-text and SPEED-text, were compared for the prediction of the next sensor event in a sequence. The methods were applied to data from eight apartments, collected over a period of 75-385 days, the period varying with apartment. The comparison took into account factors such as memory length, the required amount of data for convergence, peak accuracy, model training and testing execution time, and number and type of sensors in the dataset.

Both probabilistic methods achieved a high prediction accuracy (close to their peak accuracy) when tested on data from one apartment, using a relatively small number of events in the training set (about 1500-2000 events or 2-3 days, for this apartment). ALZ achieved 69% accuracy and SPEED achieved 82% accuracy. LSTM networks required a larger training dataset (about 4000 events/14 days for SPEED-text and 7500 events/60 days for ALZ-text) to reach an accuracy close to its peak. The accuracies achieved were, however, higher at 84% for LSTM networks with SPEED-text and 72% for ALZ-text. The SPEED-text dataset contains more information (i.e. both "on" and "off" events) and, therefore, more patterns, this contributing to better accuracies. SPEED and LSTM network with SPEED-text were therefore applied to data from the other seven apartments in the field trial. SPEED achieved a 75-82% accuracy, the required number of events for the peak accuracy being between 2000 and 7500 events. The LSTM network with SPEED-text achieved 75-85% accuracy. The LSTM network provided better accuracy than SPEED in all apartments, accuracy being 1.5-5% higher.

Probabilistic methods were found to base the prediction on a relatively small number of previous events (memory length). Four for ALZ and three for SPEED. Accuracy furthermore decreases both for shorter and for longer memory lengths than the optimal. LSTM networks, however, base prediction on a sequence of eight previous events or more. Accuracy is, for this, quite stable for memory lengths that are larger than the optimal. This indicates that such networks are better at finding longer-term dependencies and patterns in a sequence of events.

The prediction accuracy for all the methods improves by about 3% where the number of sensors is reduced from fifteen to seven (only motion sensors). One exception is SPEED, which gives poor accuracy no matter how many events there are in the training dataset. This occurs when there are no intertwined events. Only "off" events can therefore be predicted reliably. LSTM with SPEED-text is, for this case, by far the best method.

The execution time for training and testing each algorithm has been measured. The probabilistic methods require a much longer processing time, which increases as more events are added to the dataset. It was established that SPEED required eight times as long to model as the LSTM with SPEED-text. It should, however, be noted that SPEED reaches a high accuracy with less data.

SPEED may, taking these aspects into account, be a good choice for applications in which time is not too critical and there is a limited amount of data available. This does, however, assume that events are intertwined. LSTM network with SPEED-text has, however, been shown to achieve better prediction accuracy in a much shorter time, and also perform well with a limited dataset (even if it requires more data than SPEED). This comparison was presented in Paper I (probabilistic methods study), Paper II (LSTM network study), and Paper III (probabilistic methods compared to LSTM networks). Only data from one apartment were available to these papers. Paper VI confirmed these results by applying the algorithms to data collected from the eight apartments.

The LSTM network with SPEED-text algorithm was the most suitable for our application and was further developed in the following studies. The time component was included in the dataset to improve prediction, and was included in four different ways: (i) period of day (morning, afternoon, evening, night); (ii) 4-class or 8-class time-intervals (elapsed time to the next sensor event); (iii) K-means time-cluster including information on the mean hour of the day and the mean time elapsed to the next sensor event; and (iv) separate networks for the four periods of the day. This analysis was presented in Paper IV, data being from one apartment, and in a shorter version in Paper VI, data being from eight apartments.

Our best performing model in the prediction of next sensor event included 4-class time-interval information. It attained a peak average accuracy of almost 87%. This is 2% better than without the time information. The time that elapsed between events therefore contains information that improves prediction. This improvement is, however, only marginal, the improvement in other apartments varying from 0.5-4.5%. We also predicted both the next sensor event and the mean time of occurrence using a single model, the best results being obtained by using K-means time-clustering. This implementation attained an accuracy of 83%. Other apartments achieved accuracies in the range of 73-83%.

In summary, state-of-the-art prediction algorithms in the research literature that have been primarily tested using data collected from controlled environments, can achieve relatively good accuracy in the prediction of the next sensor event when using data from real homes with few sensors. Our best accuracy for predicting the next sensor event was 87%, and 83% for predicting both the next sensor event and mean elapsed time to the next sensor event.

The required accuracy for real world environments depends on the application. For example, an accuracy of 95% may be required for the prediction of events for automation or prompting systems. Accuracies lower than this for these functions could be an inconvenience for the resident. The accuracy achieved in this thesis could, however, be sufficient for a monitoring system that is, for example, used by healthcare personnel. Such a system could provide an indication of future anomalies, e.g. an unusual sequence of events is predicted, indicating that the resident therefore needs assistance. Our results have also shown that it is possible to achieve the peak accuracy of algorithms using relatively little data, two days to three weeks for our smart home layout and sensor network.

### SRQ 2 – Generalisation of Model to Other Apartments and Users

**SRQ** 2 was primarily explored in Paper V, using data from five apartments. It was also confirmed in Paper VI using data from all eight apartments. There was for SQ2, as for SQ1, around a 10% variability in prediction accuracy between the apartments for both SPEED and LSTM network with SPEED-text. The coefficient of variation (standard deviation divided by the mean) for the 10% variability of the predictions is around 0.04, which is lower than 1 and therefore considered to be a low variance. We, nonetheless, present some hypotheses for this variability below.

There are a number of factors that can have an impact on prediction accuracy in a home. These include level of cognitive impairment of the resident, level of activity and movement around the apartment and walking speed and use of a rollator or wheelchair. Each apartment can also have its own sources of noise in the collected data. This could be due to a physical aspect that affects sensor events, such as different light intensities or damaged furniture and/or electrical devices. Or it could be that more than one person is in the home, e.g. family member or healthcare personnel visits. The movement/activity patterns of some people around the apartment may also be more predictable than others, e.g. always take the same path from the kitchen to the bathroom. Some of these aspects could not be measured in the setup of this project, e.g. walking speed and noise. A fair and thorough explanation of the variability in the predictions for the apartments could therefore not be arrived at.

We also evaluated the feasibility of transfer learning between the apartments. Two methods of transfer learning were investigated, using the data available from five apartments: (i) train an LSTM model using data from four apartments and fine-tune and test with data from the target apartment, and (ii) train an LSTM model using data from the apartment with the highest accuracy when modelled individually and fine-tune and test using data from the target apartment. The network with data from the target apartment achieved accuracies of 66-79% with method (i) without fine-tuning. Accuracies were 79-86% where the network was fine-tuned. Some apartments achieved good accuracies without fine-tuning. Fine-tuning is, however, much preferable. Transfer learning leads to higher prediction accuracy when the source network is fine-tuned than where each apartment is modelled individually, for a *low number of events* (around 200 events and up to about 4000 events). Most of the apartments achieved around 80% accuracy or more with few events (around 200 events, less than a day). This means that the prediction algorithm can work well straight away when a new apartment is added to the study, and achieve a relatively good accuracy from the first day. Prediction accuracy is, however, approximately the same for *larger training datasets*, for both individual and transfer learning models. Accuracy is in fact marginally higher in most cases when each apartment is modelled individually and there is a considerable amount of data (more than 4000 events, about 16 days).

### SRQ 3 – Activity Recognition and Prediction from Binary Sensors

Our last sub-research question, **SRQ 3**, was addressed in Paper VI. We carried out activity recognition in a rule-based manner from the binary sensors' events and performed activity prediction using the LSTM with SPEED-text algorithm.

Two types of activity datasets were analysed: *sequential* and *concurrent*. *Sequential* is where no more than one activity takes place at the same time. As soon as one activity ends, another starts. *Concurrent* is where each activity has a start and an end, allowing several activities to occur in parallel. For the *concurrent* activity dataset, the models achieved 75-95% accuracy when predicting the next activity only, and accuracies of 64-85% when predicting the next activity and the mean duration and time of occurrence information. For the *sequential* activity dataset, the results are worse. The models achieved 62-90% accuracy when predicting the next activity, and 50-80% when predicting the next activity and its duration and time of occurrence information.

The sequential activity algorithm may be the fairer rule-based algorithm for activity extraction for our setup, as there are relatively few activities that can be derived and that occur concurrently. Predicting the next activity for most of the apartments led to a lower accuracy than predicting the next sensor event. Even though the accuracy of predicting activities is not high, the rule-based algorithms are useful for evaluation of activity levels of each resident. For example, the analysis of activity levels over a period of time can be useful to potentially indicate the onset of diseases and/or indicate correlations between activities and so the need for improvements in daily habits and how this can be achieved.

## 5.2 Limitations

The project had inherent limitations right from the start. The project used a RRI ethics-based approach and a predefined industrial partner. These limitations had a strong impact on the development and results of the thesis. The type of sensors that we could install were, firstly, very limited by two factors: the requirement that sensors were non-intrusive and the availability of sensors from the industrial partner. Binary sensors are one type of non-intrusive sensors that can be installed in a smart home. They provide very limited information and anonymous data. This does address privacy concerns. But it also greatly restricts the amount of information. This led to the number of activities that could be recognised, predicted, and analysed being very limited. The system is also not able to perceive whether there is more than one person in the room, a fact that contributes to noisy data. The industrial partner limited the type of binary sensors that could be deployed in the homes. The only ambient sensors available for data collection were motion, power, and magnetic sensors. We could therefore not gather the information required that would allow us to detect detailed activities. Only high-level activities such as bathroom activities could be detected (i.e. no distinction between showering, toilet, and sink use). The number of sensors was also quite small. This is of course preferable to reduce user surveillance, lower costs, and reduce the impact on home aesthetics. However, having more sensors could have allowed us to study how many sensors would actually be needed and also potentially obtain better prediction rates. A better granularity for activities could also have been achieved.

Another limitation is noisy data. As mentioned previously, data obtained from binary sensors can contain a number of faulty events, e.g. activation of motion sensors by sunlight, missing events (for example caused by very slow movements), and delayed events sending. We furthermore assume that the data is from only one person living in the apartment. It is, however, evident that the residents are visited by family members and health personnel on a regular basis. A lot of noise also originates from power sensors, as the electrical power threshold for on and off events had to be measured manually by the industrial partner for each device. The thresholds were not always reliable. We, however, carried out a data preprocessing step in which we attempted to clean and correct the raw data. We checked that electrical devices had consecutive on and off events, or vice-versa, and missing events were inserted. We believe that a major element of prediction error is still noisy data.

## 5.3 Conclusions

This thesis has contributed to a comprehensive analysis of state-of-the-art methods used in the literature as applied to data from real homes. A careful literature survey of methods that are often used for sequence prediction in smart homes was first conducted. The methods used in the literature were then tested for several aspects for data collected from eight one-bedroom apartments located in a care-dwelling facility for older adults over 65 years old. The ultimate goal is to realise smart home systems that can assist older adults to live a safe and independent life. The objective of this thesis specifically was to evaluate the potential and the limitations of sensor event and activity prediction methods when applied to a read world scenario. This was inspired by most of the studies in the literature using data from controlled environments and usually many sensors. It is evident that every study should start with tests within a controlled environment, e.g. labs and testbeds. However, research at some point should be performed in real world scenarios. There are several factors that will impact the performance of the methods, including noisy data, the unpredictability of activities and residents' habits, different numbers of persons at home at different times, privacy concerns, and home aesthetics. To the best of the authors' knowledge, the papers produced and presented in this thesis are the first research publications of such a complete feasibility and comparison study using data from real homes of older adults and with a limited number of non-intrusive sensors.

The main contributions of the thesis can be summarised as follows:

- A comprehensive comparative analysis of probabilistic methods (ALZ and SPEED), and subsequent comparison of these with LSTM neural networks, in the prediction of the next sensor event for data collected from real homes. This study investigated a number of factors that affect the performance of the algorithms: memory length, required amount of data for convergence, peak accuracy, execution time, and type and number of sensors in the dataset. These findings can be useful in deciding which analysis and prediction methods to use based on project constraints (e.g. the number of available sensors, user privacy, budget limitations, etc.).

- We have introduced a validation step that calculates optimal memory length. This step leads to a considerable improvement in performance. The algorithms originally used the maximum context length found in the training step (tree height) as the length of the sequence of prior events from which the next symbol was predicted. This was, however, shown not to be optimal for prediction that uses probabilistic methods. Prediction accuracy, after a certain length, decays constantly for higher memory lengths.

- Several configurations for inputting the data from binary sensors were investigated. As shown in this thesis, best performance is achieved by using a sequence of previous sensor events in an one-hot input vector arrangement at the input of the LSTM network.

- It was shown that prediction algorithms reach their peak accuracy using little data collected in the homes. Both academic research and the industry have assumed that the availability of big data is a requirement for achieving good Machine Learning technique prediction accuracy. In this work, we have shown that probabilistic methods require 2-5 days and LSTM neural networks require 6-16 days of data to achieve their peak accuracy in a smart home environment with a limited number of binary sensors.

- A number of ways of including the time information in the predictions were explored. We established that using the K-means clustering algorithm to define the mean elapsed time for each sensor's samples was the best solution. This entails clustering each sensor's samples by the hour of the day it occurred and the time elapsed to the next sensor event. Sensors are therefore clustered automatically to one of 3-4 possible time-clusters, each representing a slot of mean elapsed time to the next event.

- Transfer learning has been used across the apartments. It was demonstrated that good accuracy can be achieved with very little data (less than one day) when firstly using data from several apartments to model a network that is later fine-tuned to a new apartment (target apartment). This shows that the sensor event prediction algorithm could work reasonably well straight away after a system is installed in a new apartment.

- Two rule-based algorithms to associate binary sensor events with activities were implemented. Activity prediction was then carried out. It has been shown that the accuracy of predicting activities was, in most cases, lower than for predicting sensor events. Using activities rather than sensor events is, however, useful in the analysis of the activity levels of the residents, which can potentially identify changes in patterns over time.

We believe that the results have proven the current capabilities and limitations of the application of methods widely used in the literature to real world environments. The results also indicate methods for achieving improved performance. The availability of additional sensors in the system could have given better sensor event/activity recognition and prediction. Our set of sensors also proved to be somewhat limited for the task, only high-level activities being identified. A small number of sensors, as in our study, is however more realistic for real applications, as it meets ethics and privacy concerns, budget limitations, and home aesthetics considerations. The thesis shows that it is possible to achieve acceptable prediction accuracy with few sensors and relatively little data (up to three weeks). The findings of our study can also be useful when deciding which prediction methods to use in relation to project constraints (e.g. the number of available sensors, user privacy, etc.).

## 5.4 Future Work

Future work should focus on improving the performance and robustness of activity recognition and prediction methods in real homes. Much of this implies finding a solution to the limitations presented in this chapter. For example, additional sensors in the apartments could have been very beneficial and allowed a more thorough study. Considering other types of sensors would also have been an advantage. For example, pressure and acoustic sensors can provide valuable information on activities. However, the aesthetics of the home, ethical approval, and the residents' consent would still need to be taken into account. It would be an advantage, before installing the sensors, to carry out a study of the optimal positioning of sensors, as was recently carried out [118].

Privacy is always an issue. Binary sensors are therefore very well-suited to these applications as they are non-intrusive. It has, however, been shown that other types of sensors that are relatively intrusive, such as depth cameras, are well accepted by older adults, providing they can improve important aspects of day-to-day life such as safety. The older adults in our project have clearly expressed that they would trade privacy for better safety. Sensors such as depth cameras provide a lot more information than binary sensors and therefore have a greater potential in the implementation of assistive functions. Similar sensors that convey more information and are not too intrusive include radar and thermal cameras. The data are a little more intrusive. They are, however, still much less intrusive than a colour camera, for example. It should also be noted that data collection only applies to the period required for research and development, which can be short as it was shown in this thesis. Once a support function has been developed, implemented and tested, data collection is not required for the operation of such a system. Network security still, however, remains a potential challenge.

It is very important for both prediction and activity recognition algorithms that the data are as clean as possible. Finding and implementing additional noise removal methods would therefore be beneficial and would most likely improve prediction accuracy. This is evident from the difference in prediction performance of the algorithms within the apartments in our field trial compared to datasets from controlled environments.

Finally, intelligent assistive functions could be implemented and tried out in a real home when the achieved accuracy is high. Examples of possible functions that were indicated during the thesis are: a system that is able to turn on the TV when it identifies that the user wants this, but cannot manage to turn it on; a system that can register that the user has fallen asleep and forgotten to turn the stove off, and turns it off automatically. These are functions that hopefully are not far from being implemented, as soon as activity prediction and recognition algorithms achieve better performance.

# Papers

I

Paper I

# Occupancy and Daily Activity Event Modelling in Smart Homes for Older Adults with Mild Cognitive Impairment or Dementia

**Flávia Dias Casagrande, Evi Zouganeli**

Published in Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59).
DOI: 10.3384/ecp18153236

# Occupancy and Daily Activity Event Modelling in Smart Homes for Older Adults with Mild Cognitive Impairment or Dementia

Flávia Dias Casagrande    Evi Zouganeli

Department of Electronic Engineering, OsloMet – Oslo Metropolitan University, Norway.
{flavia.casagrande,evi.zouganeli}@oslomet.no

## Abstract

In this paper we present event anticipation and prediction of sensor data in a smart home environment with a limited number of sensors. Data is collected from a real home with one resident. We apply two state-of-the-art Markov-based prediction algorithms – Active LeZi and SPEED – and analyse their performance with respect to a number of parameters, including the size of the training and testing set, the size of the prediction window, and the number of sensors. The model is built based on a training dataset and subsequently tested on a separate test dataset. An accuracy of 75% is achieved when using SPEED while 53% is achieved when using Active LeZi.

*Keywords: smart home, prediction models, sensor data, occupancy modelling, event modelling*

## 1 Introduction

We present results from the Assisted Living project, an interdisciplinary project that aims to develop assisted living technology (ALT) to support older adults with mild cognitive impairment or dementia (MCI/D) live a safe and independent life at home. The project is carried out by experts in the field of nursing and occupational therapy, ethics, and technology (Zouganeli et al., 2017). MCI and dementia involve cognitive decline, which can affect attention, concentration, memory, comprehension, reasoning, and problem solving. Smart homes can potentially include a number of intelligent functions that can provide valuable support to older adults with MCI/D, such as prompting support e.g. in order to assist or encourage, diagnosis support tools, as well as prediction, anticipation and prevention of hazardous situations. Activity recognition and prediction is a prerequisite and a necessary tool for achieving the majority of these.

We present our first results on prediction of binary sensor data in a smart home environment. Several algorithms have been reported in the literature for this purpose. However, to the extent of our knowledge, such prediction algorithms have not yet been tested in a real home, nor have they been proven to be accurate enough to be implemented in real homes. In addition, there is no comprehensive study comparing the different available algorithms or providing guidelines as to which application areas they are best suited for. In this paper we apply two algorithms on data from a real home, compare their performance, and shed some light regarding their application areas.

## 2 Related Work

Data prediction algorithms have been extensively researched on in the literature (Wu et al., 2017). Event or activity prediction can for example lead to an improved operation of automation functions (e.g. turn on the heater sufficient time prior to the person arriving at home); facilitate useful prompting systems (e.g. prompt the resident in case the predicted next activity is not performed) (Holder and Cook, 2013); or detect changes/ anomalies in certain behaviour patterns (e.g. movement, everyday habits, etc.) and hence assist to indicate the onset or the progress of a condition (Riboni et al., 2016). The Active LeZi (ALZ) algorithm has been extensively applied for prediction on sequential data (Gopalratnam and Cook, 2007). The algorithm was tested on the Mavlab testbed dataset and was shown to achieve a 47% accuracy. Some of the ideas of ALZ have been used in the implementation of a new algorithm, the sequence prediction via enhanced episode discovery (SPEED)(Alam et al., 2012). SPEED was tested on the same dataset as ALZ and achieved an accuracy of 88.3% when the same dataset was used both for training and for testing. These algorithms are based on Markov models, where at any given point in time the next state depends solely on the previous one (Rabiner and Juang, 1986). Hence, the most probable next event can be estimated based on the current state.

Besides probabilistic algorithms, neural networks have also been used for event prediction. A root square mean error (RMSE) of 0.05 using Echo State Network (ESN) and Non-linear Autoregressive Network (NARX) was reported by using a number of input/output configurations (Lotfi et al., 2012; Mahmoud et al., 2013). Other relevant research includes prediction of the time when a certain activity will happen using decision trees (Minor and Cook, 2016) or time series (Moutacalli et al., 2015). Prediction of the next activity as well as the time, location, and day it would occur has also been reported (Nazerfard and Cook, 2015).

In this paper, we use the Active LeZi and SPEED algorithms for the prediction of the next sensor to be activated/ deactivated in an event sequence obtained from a real home with one resident.

# 3 Field Trial

Our field trial involves ten independent one-bedroom apartments within a community care facility for people over 65 years old. Each apartment comprises a bedroom, a living room, open kitchen area, a bathroom, and an entrance hall (Figure 1).

The purpose of the trial and the sensor system to be deployed have been decided upon in close collaboration with the residents (Zouganeli et al., 2017). A minimal number of binary sensors has been deployed in our trial in order to both minimize surveillance of the residents in their private homes, and comply with the technical and economic constraints imposed by the research project this work is a part of. The set of sensors has subsequently been chosen so that it can enable the realization of useful functions for older adults with MCI/D as these were indicated after dialogue cafes with the users (Zouganeli et al., 2017). We chose to include sensors that indicate occupancy patterns (movement around the apartment) and some daily activities like eating/ drinking, dressing, sleeping, and leisure activities (reading, watching TV, listening to radio). Hence, the system comprises motion, magnetic, and power sensors. A motion sensor (Pyroelectric/Passive Infrared – PIR) detects motion through the change of the infrared radiation in its field of view. It sends a message '1' when a motion is detected. Magnetic sensors indicate whether doors/ windows/ drawers are open or closed, by sending messages '1' and '0', respectively. Power sensors measure the electricity usage of a certain appliance, and can therefore indicate whether it is turned on or off, and send messages '1' and '0' respectively. Figure 1 shows a schematic of the apartment. There are 15 sensors installed in total: seven motion sensors (one in each area of the apartment and two over and by the bed to indicate whether the person is in bed); four magnetic sensors (back and entrance doors, wardrobe, and cutlery drawer); and four power sensors on appliances (nightstand lamp, coffee machine, TV, and living room/ reading lamp).

The sensors are connected wirelessly through Z-Wave and xComfort protocols to a Raspberry Pi 3, which receives the data and transfers it for storage in a secure server (TSD). The data comprises timestamp (date and time with precision up to seconds), sensor ID, and sensor message (binary) – see example in Table 1.

**Table 1.** Binary sensors data.

| *Timestamp* | *Sensor ID* | *Sensor message* |
|---|---|---|
| 01.09.2017 17:58:05 | 4 | 1 |
| 01.09.2017 17:58:40 | 6 | 1 |
| 01.09.2017 17:59:02 | 10 | 1 |
| 01.09.2017 17:59:05 | 10 | 0 |

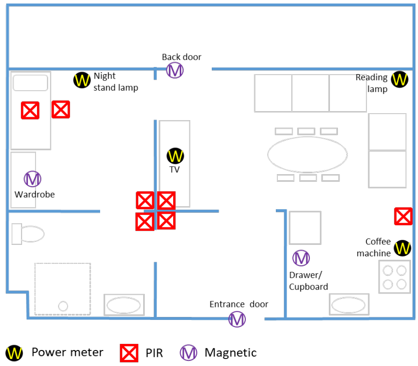



**Figure 1.** Sensors system in the field trial apartment.

# 4 Prediction Algorithms

Both ALZ and SPEED translate the data acquired from the sensors into a sequence of letters and identify patterns that occur frequently, so-called contexts. The contexts and their frequency of occurrence are used to generate a tree, which is then used to calculate the next most probable event to occur. This last step is performed by the Prediction Partial Matching algorithm (PPM) (Cleary and Witten, 1984; Cleary et al., 1997). Table 2 presents a possible scenario in a smart home of performed actions by the resident and the corresponding sensors being triggered. For ALZ and SPEED, each sensor is assigned with a letter, as shown in Table 3.

**Table 2.** Actions scenario.

| *Action performed* | *Activated sensor* |
|---|---|
| Wake up | PIR bedroom (on) |
| Go to living room | PIR living room (on) |
| Turn on TV | Power TV (on) |
| Go to kitchen | PIR kitchen (on) |
| Turn on coffee machine | Power coffee machine (on) |
| Go to living room and watch TV while coffee is being made | PIR living room (on) |
| Go to kitchen | PIR kitchen (on) |
| Turn off coffee machine | Power coffee machine (off) |
| Go to living room | PIR living room (on) |

## 4.1 Active LeZi

ALZ is a sequence prediction algorithm based on a text compression algorithm (Gopalratnam and Cook, 2007). The input in ALZ consists of a sequence of lower case letters, where each letter represents event from one sensor. For example, the sequence corresponding to the scenario described in Table 2 would be "abcdebdeb". ALZ uses the

**Table 3.** Assignment of letters to sensors.

| *Sensor* | *Letter* |
|---|---|
| PIR bedroom | a/A |
| PIR living room | b/B |
| Power TV | c/C |
| PIR kitchen | d/D |
| Power coffee machine | e/E |

idea from the LZ78 text compression algorithm to generate patterns that occur in a sequence and create a tree with these and their frequencies (Ziv and Lempel, 1978).

A given sequence $x_1, x_2, \ldots, x_i$ is parsed into $n_i$ subsequences $w_1, w_2, \ldots, w_{ni}$ such that for all $j > 0$ the prefix of the subsequence $w_j$ is equal to some $w_i$ for $1 < i < j$. For example, if we have the sequence "abcdebdeb", the dictionary would have the following words "a", "b", "c", "d", "e", "bd", "eb". These words correspond to contexts derived from the sequence. ALZ generates more contexts from their suffixes, if possible. For example, "bd" would also generate "d", and "eb" would generate "b". This accounts for contexts that were not perceived by the LZ78 algorithm and that are possibilities in a smart home environment. This increases the convergence rate of the model (Gopalratnam and Cook, 2007).

When the sequence is parsed completely and the contexts are derived from it, their frequency of occurrence is counted. An order-k-1 Markov tree is then constructed based on the contexts and their frequencies, where k corresponds to the longest word found in a training sequence. Then PPM is used to calculate the next most probable event. The generated tree for the example scenario with sequence "abcdebdeb" is shown in Figure 2.

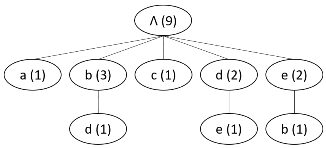



**Figure 2.** Tree generated by ALZ from sequence "abcdebdeb".

## 4.2 SPEED

SPEED is a sequence prediction algorithm that is based on the occurrence of frequent patterns in home environments (Alam et al., 2012). It assumes that human activity is predictable since usually certain patterns are repeated daily. SPEED defines an episode as the sequence between an initial and ending point of an activity. For example, the moment a coffee machine is turned "on" is the initial point of a coffee making episode, which lasts until the coffee machine is turned "off". An "off" event cannot happen unless an "on" event has happened before. Therefore "off" events always happen after an "on" event of the same activity (or sensor), and vice-versa.

The data received from the sensors in the smart home are represented as a sequence of letters, where upper case letters represent a sensor's "on" event and lower case letters represent a sensor's "off" event. For the example scenario presented in Table 1, the sequence would be "AaBCbDEdBbDedB".

The main idea of the SPEED algorithm is to extract episodes from a sequence of data and derive contexts from them. These contexts are used to generate a decision tree that keeps track of the learned episodes and their frequencies. The height of the tree is the length of the longest episode found in the sequence, defined as the maximum episode length. For every event in a sequence, the algorithm searches for its opposite event in the window and if it exists, an episode was found. In the previous sequence, the first episode found is "Aa", the contexts generated from it would be "A", "a" and "Aa". We keep track of these and count their occurrences to generate an order-k-1 Markov model, where k is the maximum episode length. A tree for the example sequence is presented in Figure 3. Finally, the PPM algorithm is used for prediction.

## 4.3 PPM Algorithm

PPM calculates the probability distribution of each possible event based on a given sequence by taking into consideration the different order Markov models with different weights (Cleary and Witten, 1984; Cleary et al., 1997). The weights are given by the escape probability, which allows the model to go from a higher-order to a lower one. The advantage of PPM is that it assigns a greater weight to the probability calculated in higher-order models if the symbol being predicted is actually found in the tree (Gopalratnam and Cook, 2007). The predicted symbol is the one with the highest probability.

ALZ and SPEED use slightly different strategies of PPM. ALZ uses the exclusion strategy, which means the prediction is performed with the suffixes of the given sequence, except the sequence itself. Therefore, in the case of the sequence "eb", the contexts used to calculate the probability of each letter being the next would be "e" and the null context. Suppose we want to calculate the probability of having an "e" after "eb" using ALZ, based on the tree in Figure 2. The probability would be given by Equation 1: in an order-2 model, the probability of having an "e" after an "e" is 0/2 and we escape to the order-1 with 1/2 probability. In order 1, the probability of having an "e" after a null context is 2/9.

In the case of SPEED, the contexts used for calculating probabilities after a certain sequence would be all the suffixes, including the sequence itself. Suppose we have the sequence "dB". We would use contexts "dB", "d" and the null context. The probability of having a "b" after this sequence based on the tree in Figure 3, would be given by Equation 2: we start in order 2 model, where the probability of having a "b" after "dB" is 1/2 and escape to the lower order with probability 1/2. In order-1, the probability of having a "b" after "d" is 0/4 and we escape to the

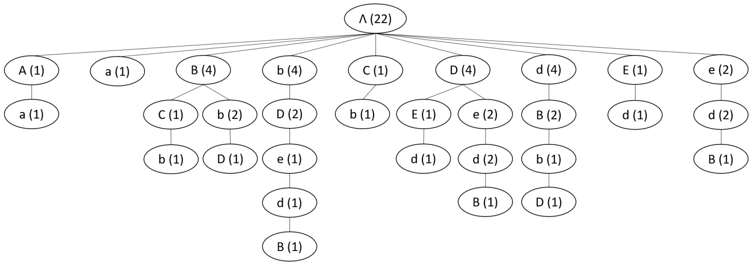



**Figure 3.** Tree generated by ALZ from sequence "AaBCbDEdBbDedB".

lower order with probability 2/4. Finally, in the lowest order, the probability of "b" after a null context is 4/22.

$$p(e, eb) = \frac{0}{2} + \frac{1}{2}\left(\frac{2}{9}\right) = 0.111 \tag{1}$$

$$p(b, dB) = \frac{1}{2} + \frac{1}{2}\left(\frac{0}{4} + \frac{2}{4}\left(\frac{4}{22}\right)\right) = 0.545 \tag{2}$$

# 5 Results and Discussions

Data has been collected from the apartment described in section 3 over a period of two weeks. In total, there are 6182 raw sensor events. The data was translated to the format required by ALZ and SPEED, which resulted in 4629 and 9062 events respectively. In the SPEED sequence, we performed noise removal such that "on" events only come after "off" events of the same sensor, or vice-versa. We ended up with 9044 events. In the SPEED algorithm, the next event is predicted based on the last sequence of size equal to the maximum episode length (Alam et al., 2012). Firstly, we reproduced the results using the same dataset and method as reported in that paper (Alam et al., 2012). Subsequently we modified the testing procedure somewhat by calculating the optimal number of last events to base the prediction on, i.e. the window that leads to the maximum overall prediction accuracy, which we refer to as the optimal window. Window sizes up to the maximum episode length are considered.

In order to be able to compare our results to the performance of the same algorithms when they are used on the data from the Mavlab testbed (Alam et al., 2012), we firstly compute the prediction accuracy that is attained when using the same dataset for both the training and the testing, as performed in their work. Figure 4 presents the results when training and testing using the same sequence of n events, where n = {100, 200, ..., 2000}.

In this test, SPEED had an optimal window of five and ALZ of six events, when the training and testing sets consisted of 2000 events. An accuracy of 82% and 73% is achieved by SPEED and ALZ respectively. Clearly training and testing with the same dataset leads to overfitting. As a result, the apparent accuracy may keep increasing

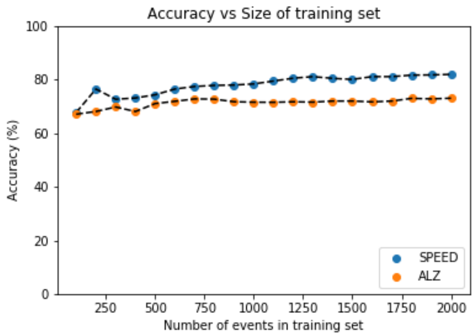



**Figure 4.** SPEED and ALZ prediction accuracy vs. the size of training set.

when we increase the dataset size. For a dataset size equal to 700, as in the Mavlab dataset, SPEED and ALZ attain 77% and 73% prediction accuracy respectively when used on our data.

In order to evaluate the actual prediction accuracy of the algorithms, our data is split into a training set, a validation set, and a testing set. The training set is used to construct the tree, the validation set is used to find the optimal window, and the testing set is used to calculate the prediction accuracy.

We first analyse the importance of choosing the optimal window to predict events from. Figure 5 shows the prediction accuracy for different sizes of the window, and for four different sizes of the training dataset when using the SPEED algorithm. Similarly, Figure 6 shows the effect of the window size in the case of the ALZ algorithm. The validation set comprised 1000 events in all cases.

We notice that smaller window sizes (1-4 events) provide better accuracy, for both algorithms. The accuracy deteriorates very quickly with increasing window size. This behaviour is as expected in particular for a setup with a small number of sensors, since long sequences of events are not bound to be repeated frequently. In the case of SPEED, for example, bathroom activities would be maximum two-events long ("on-off" bathroom motion sensor). These graphs are in addition a manifestation of the fact

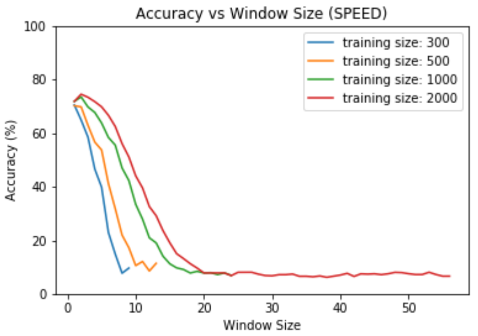



**Figure 5.** SPEED prediction accuracy vs. the window length, for several training set sizes.

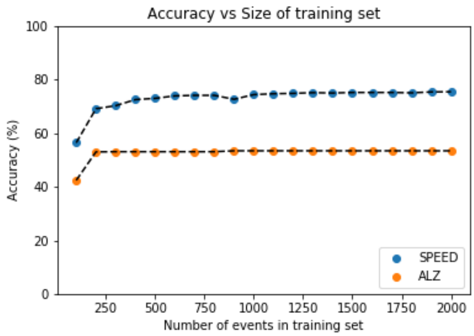



**Figure 7.** SPEED and ALZ prediction accuracy vs. the size of training set.

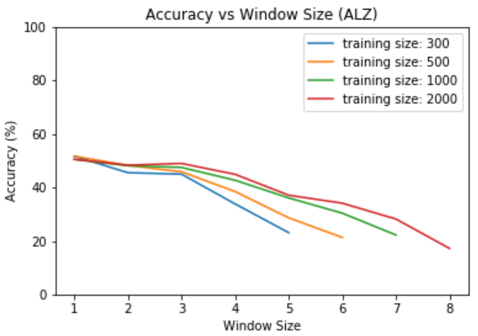



**Figure 6.** ALZ prediction accuracy vs. the window length, for several training set sizes.

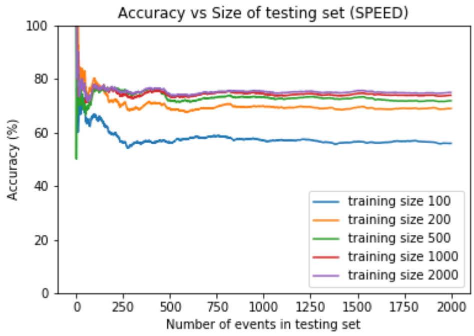



**Figure 8.** SPEED prediction accuracy vs. the size of testing set.

that SPEED creates a tree of much longer height than ALZ does. The tree height corresponds to the longest episode in SPEED, whereas in ALZ it corresponds to the longest context. This is evident from Figures 3 and 2 where the respective trees are shown for the same example scenario.

Once the optimal window was calculated from a validation set of 1000 events, we computed the accuracy for different number of training events. We trained the algorithms with a number of events i = {100, 200, 300, ..., 2000}. The prediction accuracy was computed based on a testing set of 1500 events. Figure 7 shows the results for both SPEED and ALZ.

SPEED achieved an accuracy of 75% and ALZ an accuracy of 53%, with optimal windows of two and one respectively. We observe that this maximum accuracy is achieved with SPEED for training sets larger than about 800 events, while ALZ reaches a maximum accuracy for a training set of 300 events or more. Hence, ALZ converges to its maximum accuracy faster than SPEED, however, it achieves a much poorer prediction accuracy than SPEED. Using a larger number of events for the training does not increase the accuracy significantly for neither of the algorithms.

At this point, we can associate some of these results to the trees generated for both algorithms for the example scenario in sections 4.1 and 4.2. The height of the tree is significantly larger in SPEED for the same performed actions. It can also be noted from Figure 3 that SPEED collects a significantly higher number of contexts and frequencies, which may be the reason why SPEED leads to better accuracy.

In the following we examine the dependence of the prediction accuracy on the size of test dataset. Figure 8 shows the prediction accuracy attained by SPEED as a function of the size of the testing dataset for different sizes of the training dataset. Figure 9 shows the same results for ALZ.

In the case of SPEED, the prediction accuracy is quite variable for a test dataset of up to about 250 events due to the small number of predicted events. The maximum accuracy is achieved for test set sizes larger than about 500 events when the training is performed based on a set with 500, 1000 and 2000 events. This confirms that the algorithm is quite robust. ALZ shows similar behaviour and achieves its maximum prediction accuracy for test datasets larger than about 200 events.

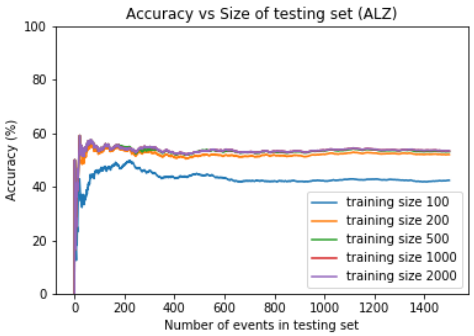



**Figure 9.** ALZ prediction accuracy vs. the size of testing set.

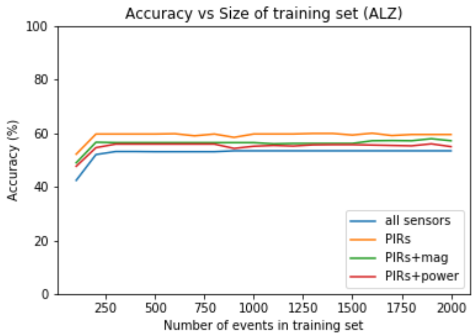



**Figure 11.** ALZ prediction accuracy vs. the size of training set for different sets of sensors.

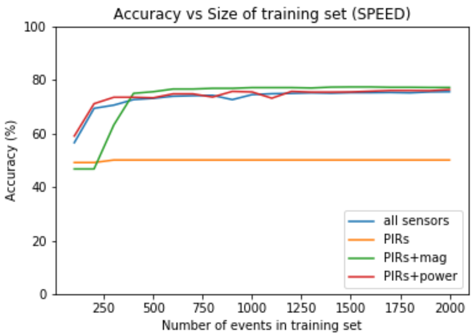



**Figure 10.** SPEED prediction accuracy vs. the size of training set for different sets of sensors.

A last test was performed to reveal the dependence of the prediction accuracy on the number and type of sensors. Four alternatives were investigated based on our current data: all sensors (15), only PIR sensors (7), only PIR and magnetic sensors (11), and only PIR and power sensors (11). The last two sets have the same number of sensors, however, magnetic and power sensors can affect accuracy differently. The results are shown in Figure 10 for SPEED and Figure 11 for ALZ.

Both algorithms show relatively good robustness with respect to the number of sensors. The accuracy is not significantly dependent upon the number of sensors in the dataset, in most of the cases. A clear exception is the case when only PIRs are used for prediction using SPEED. The prediction accuracy is very poor in this case. Note that in this case the longest episode will be two events. For example, if the resident would go from the bedroom to the living room and then to the kitchen, the resulting sequence would be "AaBbCc". There is no context connecting the living room to the bedroom, or the kitchen to the living room. Hence, while the "off" events are easily correctly predicted, the prediction of the next sensor to be activated will often be quite inaccurate in this case. Note that in this case the tree created by SPEED will have a maximum length of two. On the other hand, ALZ is better suited to such cases where events are not highly interweaved. When SPEED is used the remaining sensor sets achieved a prediction accuracy that is similar to that achieved by the full set of sensors. The alternative where the power sensors are not included provides slightly better results indicating that events related to appliances are more difficult to predict.

In the case of ALZ, the best accuracy is achieved when fewer sensors are used. This is a result of the fact that the average probability of occurrence for each event increases when the number of possible events decreases. The prediction accuracy of events that involve magnetic sensors is relatively high as doors and drawers are often closed right after they have been opened, thus making this a relatively easy pattern to predict. On the other hand, power sensors can occur somewhat randomly with many other events happening in between, thus making the prediction of the associated events more inaccurate.

## 6 Conclusions and Future Work

Activity recognition and prediction in a smart home environment with binary sensors has received a lot of attention in recent years. Most of the reported work is carried out in testbeds and lab environments where users are often asked to execute pre-scripted activities. Such smart-home testbeds typically include a quite large number of sensors, e.g. the CASAS testbed utilized around 50 sensors (Gopalratnam and Cook, 2007).

In this paper we have presented preliminary results on event prediction based on data from a real home collected using just 15 binary sensors. We have used two prediction algorithms, ALZ and SPEED, to predict the next sensor event in a sequence. To the extent of our knowledge, this is the first time these algorithms are used on a dataset obtained from a real home. We compare the prediction accuracy of the two models and examine the dependence of their performance on a number of parameters – the size of the training dataset, the size of the testing dataset, and the size of the window used for the prediction. We reached an accuracy of 75% with SPEED and 53% with ALZ when training with a dataset of 2000 events and testing on a separate dataset of 1500 events. Increasing the number of events in either the training or the testing dataset, did not improve the attained accuracy. In addition, we examined the dependence of the prediction accuracy on the number of sensors for both algorithms. Our results show that robust prediction accuracy can be attained by a relatively low number of sensors.

However, a much higher prediction accuracy is required before such algorithms are applicable to real homes. Future work will include the time component in order to improve the accuracy of our models as this has been indicated to lead to a considerable improvement (Marufuzzaman et al., 2015).

# Acknowledgements

The authors wish to thank the residents and the housekeepers at the seniors' care unit Skøyen Omsorg+; Torhild Holthe and Dr. Anne Lund (OsloMet) for recruiting participants for the trial and communicating with the residents throughout; Dejan Krunić and Øyvind Width (Sensio AS) for installations of the sensors, and the rest of the participants of the Assisted Living Project for a fruitful interdisciplinary collaboration; Dr. Muhammad Raisul Alam for clarifications regarding the SPEED algorithm and the way it was applied in his work; Prof. Jim Tørrensen, University of Oslo, for his support. The project is financed by the Norwegian Research Council under the SAMANSVAR programme (247620/O70).

# References

Muhammad Raisul Alam, M. B.I. Reaz, and M. A. Mohd Ali. SPEED: An inhabitant activity prediction algorithm for smart homes. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 42(4):985–990, 2012. ISSN 10834427. doi:10.1109/TSMCA.2011.2173568.

John G. Cleary and Ian H. Witten. Data Compression Using Adaptive Coding and Partial String Matching. *IEEE Transactions on Communications*, 32(4):396–402, 1984. ISSN 00906778. doi:10.1109/TCOM.1984.1096090.

John G. Cleary, W. J. Teahan, and Ian H. Witten. Unbounded length contexts for PPM. In *Proceedings DCC '95 Data Compression Conference*, pages 52–61, 1997. ISBN 0-8186-7012-6. doi:10.1109/DCC.1995.515495.

Karthik Gopalratnam and Diane J. Cook. Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm. *IEEE Intelligent Systems*, 22(1), 2007.

Lawrence B. Holder and Diane J. Cook. Automated activity-aware prompting for activity initiation. *Gerontechnology*, 11(4):534–544, 2013. ISSN 15691101. doi:10.4017/gt.2013.11.4.005.00.

Ahmad Lotfi, Caroline Langensiepen, Sawsan M. Mahmoud, and M. J. 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. ISSN 18685137. doi:10.1007/s12652-010-0043-x.

Sawsan Mahmoud, Ahmad Lotfi, and Caroline Langensiepen. Behavioural pattern identification and prediction in intelligent environments. *Applied Soft Computing Journal*, 13(4):1813–1822, 2013. ISSN 15684946. doi:10.1016/j.asoc.2012.12.012.

M Marufuzzaman, M. B. I. Reaz, M. A. M. Ali, and L. F. Rahman. A Time Series Based Sequence Prediction Algorithm to Detect Activities of Daily Living in Smart Home. pages 262–270, 2015. ISSN 0026-1270. doi:10.3414/ME14-01-0061.

Bryan Minor and Diane J. Cook. Forecasting occurrences of activities. *Pervasive and Mobile Computing*, 2016.

Mohamed Tarik Moutacalli, Abdenour Bouzouane, and Bruno Bouchard. Sensors activation time predictions in smart home. *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, page 37, 2015. doi:10.1145/2769493.2769530.

Ehsan Nazerfard and Diane J. Cook. CRAFFT: An Activity Prediction Model based on Bayesian Networks. 33(4):395–401, 2015. ISSN 1527-5418. doi:10.1038/nbt.3121.ChIP-nexus.

L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16, 1986. ISSN 07407467. doi:10.1109/MASSP.1986.1165342.

Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Rim Helaoui. SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment. *Artificial Intelligence in Medicine*, 67:57–74, 2016. ISSN 18732860. doi:10.1016/j.artmed.2015.12.001.

TSD. Services for sensitive data (TSD). `https://www.uio.no/english/services/it/research/sensitive-data/index.html`. Accessed: 2018-04-30.

Shaoen Wu, Jacob B. Rendall, Matthew J. Smith, Shangyu Zhu, Junhong Xu, Honggang Wang, Qing Yang, and Pinle Qin. Survey on Prediction Algorithms in Smart Homes. *IEEE Internet of Things Journal*, 4(3):636–644, 2017. ISSN 23274662. doi:10.1109/JIOT.2017.2668061.

Jacob Ziv and Abraham Lempel. Compression of Individual Sequences via Variable-Rate Coding. *IEEE Transactions on Information Theory*, 24(5):530–536, sep 1978. ISSN 15579654. doi:10.1109/TIT.1978.1055934.

Evi Zouganeli, Flavia D. Casagrande, Torhild Holthe, Anne Lund, Liv Halvorsrud, Dag Karterud, Adele Flakke-Johannessen, Hilde Lovett, S.K. Mørk, J. Strøm-Gundersen, Erik Thorstensen, R. Norvoll, R.T. Meulen, M.-R. Kennedy, R.J. Owen, M. Ladikas, and Ellen-Marie Forsberg. Responsible development of self-learning assisted living technology for older adults with mild cognitive impairment or dementia. *ICT4AWE 2017 - Proceedings of the 3rd International Conference on Information and Communication Technologies for Ageing Well and e-Health*, (Ict4awe):204–209, 2017. doi:10.5220/0006367702040209.

Paper III

# Comparison of Probabilistic Models and Neural Networks on Prediction of Home Sensor Events

**Flávia Dias Casagrande, Jim Tørresen, Evi Zouganeli**

To be published in International Joint Conference on Neural Networks (IJCNN) 2019.

III

# Comparison of Probabilistic Models and Neural Networks on Prediction of Home Sensor Events

Flávia Dias Casagrande
*Dept. of Mechanical,*
*Electronics, and Chemical Engineering*
*OsloMet – Oslo Metropolitan University*
Oslo, Norway
flacas@oslomet.no

Jim Tørresen
*Dept. of Informatics*
*University of Oslo*
Oslo, Norway
jimtoer@ifi.uio.no

Evi Zouganeli
*Dept. of Mechanical,*
*Electronics, and Chemical Engineering*
*OsloMet – Oslo Metropolitan University*
Oslo, Norway
evizou@oslomet.no

***Abstract*—We present results and comparative analysis on the prediction of sensor events in a smart home environment with a limited number of binary sensors. We apply two probabilistic methods, namely Sequence Prediction via Enhanced Episode Discovery – SPEED, and Active LeZi – ALZ, as well as Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) in order to predict the next sensor event in a sequence. Our dataset has been collected from a real home with one resident over a period of 30 weeks. The binary sensor events are converted to two different text sequences as dictated by SPEED and ALZ, which are also used as inputs for the LSTM networks. We compare the performance of the algorithms regarding the number of preceding sensor events required to predict the next one, the required amount of data for the model to reach peak accuracy and stability, and the execution time. In addition, we analyze these for two different sets of sensors. Our best implementation achieved a peak accuracy of 83% for a set with fifteen sensors including motion, magnetic and power sensors, and 87% for seven motion sensors.**

***Index Terms*—smart home, sensor data prediction, binary sensors, recurrent neural network, probabilistic models**

## I. Introduction

The Assisted Living Project (ALP) is an interdisciplinary project involving health, ethics, and technology experts [1]. The aim is to develop assisted living technology (ALT) to support older adults with Mild Cognitive Impairment (MCI) or Dementia (D) live a safe and independent life at home. MCI/D is a cognitive decline that can affect attention, concentration, memory, comprehension, reasoning, and problem solving [2]. A fair amount of research on smart home functions has aimed at assisting older adults with MCI/D in their everyday life. Examples are functions such as prompting with reminders or encouragement, diagnosis tools, as well as prediction, anticipation and prevention of hazardous situations. These require quite robust and reliable activity recognition and prediction algorithms in order to be deployed in real homes.

Activity recognition and prediction can be performed by various algorithms that have been reported in the literature. Most of this work has used data collected in the lab based on scripted activities. In addition, there is no comparative study investigating different configurations for input of data, the required data size for accurate predictions, or providing guidelines as to the applicability of these. In this work, we apply state-of-the-art sequence prediction algorithms, both probabilistic methods and recurrent neural networks, to binary sensor data acquired from a real home with a relatively small number of sensors over a period of 30 weeks. We compare the performance of these methods for sensor event prediction with regard to the amount of data, the time used for training and testing the models, and the number of preceding events required to predict the next event (memory length). We further analyze the performance of the algorithms for two different sets of sensors – one with events from fifteen sensors (motion, magnetic and power) and one with events from seven motion sensors only.

Section II gives an overview of algorithms used for sensor sequential prediction in the literature. Section III describes our field trial. Section IV presents the methods used in the current work. In section V we present our results and discuss our findings. The paper concludes in Section VI with a short summary and ideas for improvement and future work.

## II. Related Work

Several sequential data prediction algorithms have been investigated in the past years [3]. These have a broad range of application areas, including sensor event and activity prediction – the basis of several functions in smart homes. Such algorithms can for instance lead to an improved operation of automation functions (e.g. turn on the heater a sufficient time prior to the person arriving at home); enable the realization of prompting systems (e.g. prompt the resident if the predicted activity has not been performed) [4]; or identify changes and anomalies in certain behaviour patterns (e.g. movement, everyday habits, etc.) and thus indicate the onset or the progress of a condition [5].

The Active LeZi (ALZ) is a probabilistic method that has been extensively employed for prediction on sequential data [6]. It achieved a peak accuracy of 47% when applied on the Mavlab testbed dataset, that includes 50 binary sensors [6]. Based on the ALZ, the Sequence Prediction via Enhanced Episode Discovery (SPEED) algorithm was implemented [7]. SPEED was applied on the Mavlab dataset and reached an

Financed by the Norwegian Research Council under the SAMANSVAR programme (247620/O70).

accuracy of 88.3% when the same dataset was used both for training and for testing. Both algorithms convert the data of binary sensors to a sequence of letters and build a tree based on the observed patterns and corresponding frequency of occurrence. The tree is Markov model-based, where at any given point in time the next state depends solely on the previous one [8]. Hence, the most probable next event can be estimated based on the current state, by using the Prediction by Partial Matching algorithm (PPM) [9].

Neural networks have also been used for sensor event prediction with notable performance, typically recurrent neural networks (RNN). Three RNN models – Echo State Network (ESN), Back Propagation Through Time (BPTT), and Real Time Recurrent Learning (RTRL) – were applied on a fourteen-day dataset with only six binary sensors (four motion and two magnetic). The ESN performed better with a root square mean error (RMSE) of 0.06 [10]. In these networks, the number of input and output values corresponded to the number of sensors in the dataset, and each assumed value "0" or "1" for being "off" or "on" at a certain time slot. The prediction in this case was computed for the next six hours. In a subsequent work, a Non-linear Autoregressive Network (NARX) was compared to an Elman network. Both used as input and output the start and end time of a sensor's activation [11]. In this study, each sensor had its own network trained and tested on a twenty-day dataset with the same six binary sensors. The NARX performed better when predicting only the next step, with a RMSE ranging from 0.06 to 0.09, depending on the sensor.

A similar study was carried out for a 16-room office environment [12]. The dataset in this case was collected through an app the employees had installed on a personal data assistant (PDA). They would register themselves whenever they entered/left a certain room. An Elman network and a multilayer perceptron network were applied to predict the next room a person would go to. There were four participants in the study and the Elman network attained the best results, ranging from 70% to 91% accuracy depending on the user. Each room was codified in four bits as there were 16 rooms in total. The input corresponded to two rooms and the output to the predicted next room. This work also applied other methods – Bayesian network, state prediction, and Markov predictor – where comparable results were achieved [13].

Other related research includes prediction of the next activity as well as the time, location, and day it would occur using Bayesian networks, which achieved 74% of activity prediction [14]. Prediction of the time when a certain activity will take place has also been investigated using decision trees [15] and time series [16].

Our dataset was collected from a real home, while most datasets from the cited works have been collected through scripted activities primarily in lab environments. In addition, it contains events from fifteen binary sensors, i.e. twice as many as used in [10], [11], and less than one third of the number of sensors used in the Mavlab testbed. The number of sensors is comparable to the work in [12] (16 rooms), however in that study the events were inserted by each user in their PDA rather than being generated automatically.

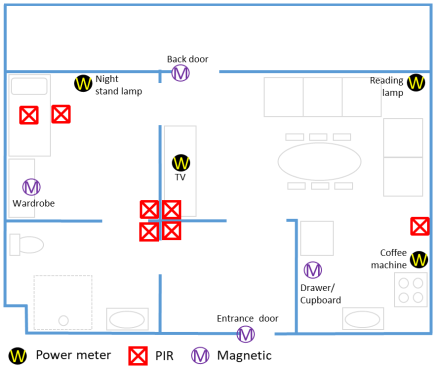



Fig. 1. Sensors system installed in the field trial apartment.

## III. Field Trial

Our field trial includes nine apartments in a community care facility with residents over 65 years old. In this work we use data from one of the apartments where we have collected 30 weeks of data. The apartments comprise a bedroom, a living room, an open kitchen area, a bathroom, and an entrance hall (Fig. 1).

The purpose of the trial and the sensors system deployed in the apartments have been decided in close collaboration with the participants [1]. We installed a minimal number of binary sensors in order to both minimize surveillance of the residents and comply with the technical and economic constraints imposed by the project. The set of sensors has subsequently been chosen so that it can enable the realization of useful functions for older adults with MCI/D as these were indicated at dialogue cafés with the users [1]. Hence, our set of sensors contains motion, magnetic, and power sensors. These enable inference of occupancy patterns (movement around the apartment) and some daily activities – kitchen-related activities, dressing, being in bed –, and leisure activities – reading, watching TV, listening to radio. Motion sensors (Pyroelectric/Passive Infrared – PIR) detect motion through the change of the infrared radiation in its field of view. It sends a message "1" every time a motion is detected, otherwise it sends no other message. In our dataset we had to insert the "off" events ("0" message) so that the data are consistent for all sensors. Magnetic sensors indicate whether doors, windows, and drawers are open or closed, by sending messages "1" and "0", respectively. Power sensors measure the electricity usage of a certain appliance, and can therefore indicate whether it is turned on or off, and send messages "1" and "0" respectively.

Fig. 1 shows the schematic of the apartment we collected 30 weeks of data from, with 15 sensors in total:

- Seven motion sensors: one in each room of the apartment, and two over and by the bed to indicate whether the person is in bed;
- Four magnetic sensors: entrance and back doors, wardrobe, and cutlery drawer;
- Four power sensors on appliances: night stand lamp, coffee machine, TV, and living room/reading lamp.

The sensors are connected wirelessly through Z-Wave and xComfort protocols to a Raspberry Pi 3, which transfers the data for storage in a secure server. The data comprise timestamp (date and time with precision of seconds), sensor ID, and sensor message (binary). Table I presents events generated by the following example scenario: the resident wakes up (PIR bedroom "on"), goes to the living room (PIR living room "on"), turns on the TV (power TV "on"), goes to the kitchen (PIR kitchen "on"), starts the coffee machine (power coffee "on"), goes back to the living room (PIR living room "on") while coffee is prepared, goes back to kitchen (PIR kitchen "on") to get the coffee (power coffee "off") and drink it in the living room (PIR living room "on").

## IV. Sensor Data Prediction Methods

### A. Preprocessing

The preparation of the data includes two steps: data correction and data conversion. The data correction is necessary because the data acquired from binary sensors often contain faulty events e.g. erroneous activation of motion sensors by sunlight, bouncing of contact sensors, or switch-off delays of motion sensors [17]. Such flawed data may substantially affect the performance of the models that will learn erroneous patterns. In our system, we observed that sometimes the motion sensors do not send an activation event, as they should. Missing sensor events have been inserted to correct for this. For example, it is not possible to go to the bedroom directly from the kitchen without passing through the living room. If the living room motion sensor activation event is missing, it is inserted. In the case where there are two possible sensor events (e.g. two possible paths in the apartment), the choice of the inserted sensor event is done such that the distribution of the inserted events corresponds to the percentage distribution of the two options as observed in the data. This process had a significant effect on the obtained accuracy.

TABLE I
Binary sensors data

| *Timestamp* | *Sensor ID* | *Sensor message* |
|---|---|---|
| 01.09.2017 07:58:05 | 2 | 1 |
| 01.09.2017 07:58:40 | 4 | 1 |
| 01.09.2017 07:59:02 | 10 | 1 |
| 01.09.2017 07:59:50 | 5 | 1 |
| 01.09.2017 08:00:14 | 12 | 1 |
| 01.09.2017 08:01:01 | 4 | 1 |
| 01.09.2017 08:02:56 | 5 | 1 |
| 01.09.2017 08:03:05 | 12 | 0 |
| 01.09.2017 08:03:33 | 4 | 1 |

Subsequently, the corrected data is converted to two sequences of letters, as dictated by the ALZ and SPEED algorithms. The resulting sequences are also fed into LSTM networks that are configured as text generation networks.

The conversion assigns a dedicated letter to each of the sensors. In the case of ALZ, only "on" events are taken into account, and hence only lower-case letters are used. SPEED, on the other hand, differentiates "on" and "off" events of the same sensor by using upper- and lower-case letters, respectively. Table II presents the assigned letters corresponding to the example scenario in a smart home described in Table I.

### B. Active LeZi

ALZ [6] is a largely used algorithm for sequence prediction. From the sequence of lower-case letters, ALZ derives several patterns and their frequency of occurrence. This is based on the LZ78 text compression algorithm [18]. Given a certain sequence $x_1, x_2, \ldots, x_i$, the LZ78 will parse it into $n_i$ subsequences $w_1, w_2, \ldots, w_{ni}$ such that for all $j > 0$ the prefix of the subsequence $w_j$ is equal to some $w_i$ for $1 < i < j$.

For example, ALZ would generate the sequence "abcdebdb" for the scenario in Table I. The derived patterns according to LZ78 would be "a", "b", "c", "d", "e", "bd". ALZ generates these and even more patterns from the original ones, if possible. For example, "bd" also generates the pattern "d". This addition accounts for patterns that were not perceived by the LZ78 algorithm and that are still possible in a smart home environment. This modification increases the convergence rate of the model [6]. Besides the patterns, their frequency of occurrence is also counted. An order-k-1 Markov tree is then constructed based on the patterns and their frequencies. Note that k corresponds to the longest pattern found in a training sequence. Fig. 2 shows the generated tree for the example scenario with sequence "abcdebdb".

Subsequently, the PPM algorithm is used for predicting the next event. The PPM algorithm calculates the probability distribution of each possible event based on a given sequence

TABLE II
Assignment of letters to sensors

| *Sensor (ID)* | *Letter* |
|---|---|
| PIR bedroom (2) | a/A |
| PIR living room (4) | b/B |
| Power TV (10) | c/C |
| PIR kitchen (5) | d/D |
| Power coffee machine (12) | e/E |

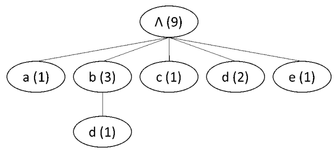



Fig. 2. Tree generated by the ALZ algorithm for the sequence "abcdebdb".

by taking into consideration the different order Markov models in the formed tree with different weights [9].

### C. Sequence Prediction via Enhanced Episode Discovery

SPEED is, like ALZ, a sequence prediction algorithm based on the occurrence of frequent patterns in home environments [7]. SPEED defines an *episode* as the sequence between an "on" and an "off" event of the same sensor, or vice-versa. For example, the events that occur between the TV turned "on" and "off", these included, is an *episode*.

Upper- and lower-case letters represent a sensor's "on" and "off" events. For the example scenario presented in Table I, SPEED would generate the sequence "AaBCbDEdBbDedB".

SPEED extracts episodes from a given sequence and derive patterns from them. In the previous sequence, the first episode that is found is "Aa" and the patterns derived from it would be "A", "a" and "Aa". These are used to generate a decision tree that keeps track of the learned episodes and their frequencies, as performed by ALZ. A tree for the example sequence is presented in Fig. 3. Note that the height of the tree is the length of the longest episode found in the sequence. The PPM algorithm is also used for the prediction of the next event.

### D. Long Short-Term Memory Network

RNN has been broadly applied to sequence prediction due to its property of keeping an internal memory. Hence, it attains a good performance for inputs that are sequential in time. Examples of applications include text generation [19], speech recognition [20] and pattern recognition in music [21]. The LSTM [22] is an RNN architecture designed to be better at storing and accessing information than the standard RNN [23].

In this work the LSTM network is configured as a text generation network. The number of inputs is a certain number of sensor events – equal to the memory length – and the output is the predicted next event in the sequence (Fig. 4). The input and output are one-hot encoded. In the one-hot encoding representation, each letter is represented by a vector of bits of length equal to the number of letters. All values are zero, except for the one corresponding to that letter (see Fig. 4).

A stateless LSTM network model was implemented in Python 3 using Keras open source library for neural networks. A number of parameters were tuned in order to find the optimal values. Memory length (i.e. number of events that are used to predict the next event) was set to 9. The model has one hidden layer with 64 neurons. The number of samples used for training each iteration of the epoch (i.e. batch size) was 512 and learning rate of 0.01. Adam was used as the optimization function, categorical cross-entropy as loss function, and the activation functions in the hidden layer and output layer were set as hyperbolic tangent and softmax, respectively. We used the early stopping method to avoid overfitting and unnecessary computations, allowing a maximum of 200 epochs for each model's training.

## V. Results and Discussion

Data have been collected from one apartment over a period of 30 weeks. Table III shows the number of sensor events for ALZ- and SPEED-text sequences, after data correction and conversion, for the two sets of sensors we analyze (all 15 sensors and only the 7 PIR sensors).

### A. Training and Testing Configuration

In the SPEED algorithm, the next event is predicted based on the last sequence of events with length equal to the maximum episode length [7]. In [7], the authors use the same dataset for both training and testing, which leads to overfitting.

We have modified the testing procedure by calculating the optimal number of last events to base the prediction on, i.e. the number of events that leads to the maximum overall prediction accuracy, which we refer to as the optimal memory length. Memory lengths up to the maximum episode length have been considered. In a previous paper [24], we applied the SPEED method on our data that were obtained from the same home as reported here over a period of two weeks. When using the same procedure as in [7], we achieved an accuracy of 82% – compared to 88% on the Mavlab dataset. When splitting the data into training (60%), validation (20%), and testing (20%), and optimizing the memory length as described above, we achieved an accuracy of 75% on our data obtained from a real home.

Similarly for ALZ we obtained 73% (compared to 47% in [6]) when using the same dataset for training and testing, and 53% when using different datasets for training, validation and testing, and optimizing the memory length as described above. Hence we use this modified method for SPEED and ALZ in the following sections.

In the case of SPEED and ALZ, the training set is used to build the tree, the validation set is used to find the optimal memory length, and the testing set is used to compute the model's accuracy.

We use the same split rates for the sets used in the LSTM network, where the training set is used to train the network, the validation set is used for tuning the parameters and the testing set to calculate the accuracy. We can notice from Table III that the majority of the events are from motion sensors. Therefore, during the training process in the neural networks, we use weights for each sensor to compensate the fewer samples from the magnetic and power sensors. These are computed using the "`compute_class_weight`" function of the Scikit-learn open source library. The weight corresponds to the total number of samples divided by the number of occurrences of the class. In addition, for all the methods the results show the mean accuracy achieved using a 5-fold cross-validation process (using 60% of the data for training, 20% for validation, and 20% for testing).

TABLE III
Number of events in dataset

| *Set of Sensors* | *Number of Events* | |
|---|---|---|
| | *ALZ* | *SPEED* |
| All sensors (15) | 60961 | 121922 |
| PIR sensors (7) | 55302 | 110604 |

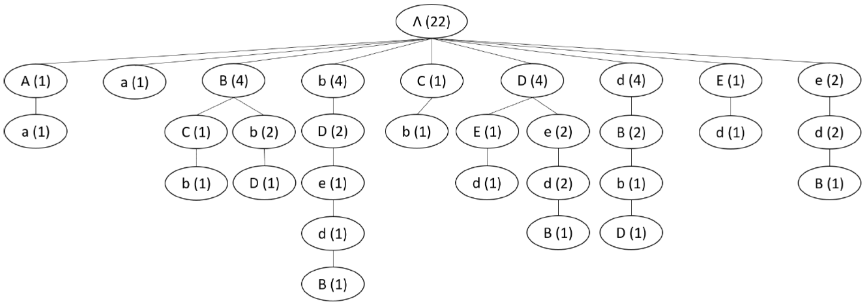



Fig. 3. Tree generated by the SPEED algorithm for the sequence "AaBCbDEdBbDedB".

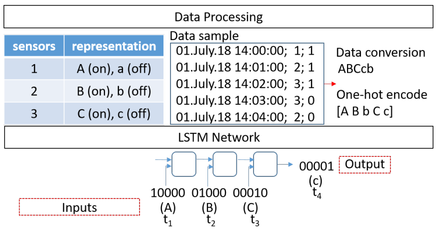



Fig. 4. LSTM network configuration.

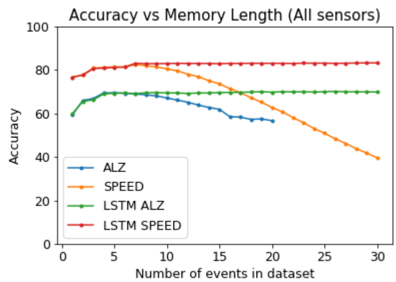



Fig. 5. Accuracy vs memory length for all algorithms on a dataset with all fifteen sensors.

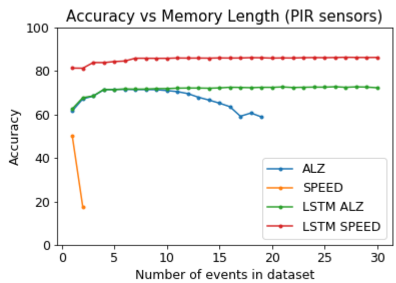



Fig. 6. Accuracy vs memory length for all algorithms on a dataset with seven motion sensors.

In the following, we examine the accuracy attained by the four algorithms (ALZ, SPEED, LSTM with ALZ-text and LSTM with SPEED-text) first against the memory length and then against the size of the dataset given in weeks. Further we compare our results to previous related work and summarize our discussion in this section.

### B. *Optimum memory length*

We examine the accuracy achieved on the validation set for several values of memory length ranging from 1 to 30 events. This is performed first for a dataset containing events from all fifteen sensors (magnetic, power and motion) – Fig. 5 – and then for a dataset containing only the seven motion sensors – Fig. 6.

When using a dataset with fifteen sensors (Fig. 5), ALZ achieved a best accuracy of 69% while SPEED reached 82%. The optimal memory length was 4 events for ALZ and 7 for SPEED. The LSTM networks achieved accuracies of 70% and 83% when using ALZ- and SPEED-text, respectively. In both cases the optimal memory length is equal or larger than 8. The larger optimum memory length for LSTM indicates that probabilistic methods predict the next event based on fewer previous events, in other words the LSTM is more efficient at detecting patterns and correlations over a longer sequence.

It is also interesting to notice how the accuracy is affected by memory lengths larger than the optimal. The accuracy of the probabilistic methods drops substantially as the memory length gets larger. In contrast, the LSTM networks roughly stabilize at the peak accuracy for larger memory length values. A reason for this is that probabilistic methods are based on certain patterns happening quite frequently. Since our dataset has few sensors, short patterns are more likely to happen more often and therefore, they provide better predictions. The LSTM, on the other hand, has the ability to find patterns in long sequences and can therefore predict the next event based on many past events and longer term patterns and dependencies. Increasing the memory length further does not improve the accuracy, however, which can imply that the model has reached its best performance for this configuration.

Subsequently, we compare the accuracy results of a dataset with fifteen sensors (Fig. 5) to the accuracy results of a dataset that contains only the seven motion sensors (Fig. 6). The accuracy curves for the LSTM network models show a similar dependency to memory length. The optimal memory length is 9 or larger. The LSTM with SPEED-text achieves 87% while with ALZ-text achieves 73%. The ALZ method also shows similar behaviour, and same optimal memory length of 4, with a higher accuracy of 71%. SPEED presents a very peculiar behaviour. The maximum memory length is 2. This is a consequence of the fact that SPEED builds the tree based on episodes, and the longest episode in this case is two events. For example, if the resident would go from the bedroom to the living room and then to the kitchen, the resulting sequence would be "AaBbCc". There are no intertwined events, since when one motion sensor activates, another deactivates. Hence, the "off" events are easily predicted. When it comes to "on" events, the sensor that is most frequently activated will always be the one predicted to activate next, leading to lower accuracy for "on" events.

### C. *Required amount of data for good accuracy*

In the following, we investigate the behaviour of the accuracy with respect to the size of the dataset used for the complete process of training, validating, and testing the models. The accuracy results are computed within the testing set and using the optimal memory length found in the previous analysis. Fig. 7 and 8 show the results when the algorithms are applied to a dataset with all fifteen sensors and with only seven sensors, respectively.

The best accuracy is achieved for 10 weeks of data or above. There is no significant improvement in the accuracy for larger datasets, we therefore show the plots for dataset sizes up to 10 weeks for better clarity on the lower range of the graph.

We first examine the accuracy in the dataset with all sensors (Fig. 7). A peak accuracy of 83% was achieved by LSTM with SPEED-text, while the SPEED algorithm achieved a peak accuracy of 82%. The accuracy achieved by the LSTM with ALZ-text was considerably lower at 69%. In this case, stability is achieved much later than with the other methods. Finally, the ALZ method reached a top accuracy of 70% with 4 weeks of data. However, this method does not seem to be as stable as the other algorithms.

Note that the probabilistic methods attain a good accuracy (close to the peak accuracy) with only 2 days of data. By comparison, the LSTM networks need approximately 2-3 weeks of data to start approaching their top accuracy. This correlates well with the previously discussed ability of the LSTM to learn longer term patterns and dependencies, and attain better accuracy based on these.

Next we examine the accuracy results for the dataset using only the seven motion sensors (Fig. 8). As expected, the accuracy is higher since there are fewer sensors in this set. Moreover, motion sensor events happen sequentially, without intertwined events. The LSTM with SPEED-text achieved an accuracy of 87%, by far the best among the methods. The peak accuracy was achieved with slighlty less than 2 weeks of data. In addition, stability is reached with less data compared with the case in Fig. 7. The LSTM with ALZ-text and the ALZ achieved very similar accuracies of 73%, and 74% respectively. The SPEED method, however, achieved a poor accuracy in this case. This is due to the short memory length and lack of intertwined events, as discussed when presenting Fig. 6. Also here, it is confirmed that probabilistic methods require a rather small amount of data to achieve a considerable accuracy, close to the peak accuracy that can be reached by these methods. The LSTM with SPEED-text also achieved a good accuracy with only a few days of data. However, the LSTM network with ALZ-text needed considerably larger amounts of data to attain acceptable prediction accuracy.

Most of the models reached a peak accuracy with 10 weeks of data or more. It may appear somewhat surprising that the best accuracy was reached for the same amount of data – 10 weeks – for both sets of sensors. However, as we pointed out earlier, the majority of the events in the dataset is in fact from motion sensors, and therefore, the two datasets are of similar size.

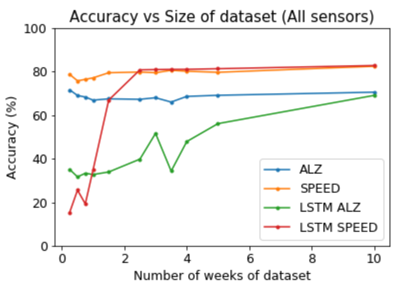



Fig. 7. Accuracy vs size of dataset for all algorithms on a dataset with all sensors.

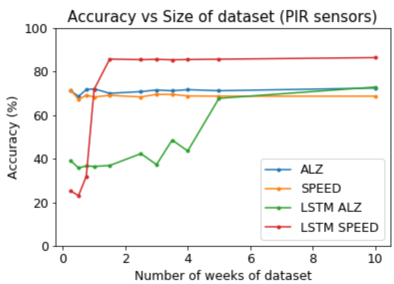



Fig. 8. Accuracy vs size of dataset for all algorithms on a dataset with motion sensors.

### D. Execution time

Lastly, we examine the execution time to train and test the models. Table IV shows the results for the set with all sensors. In general, the probabilistic methods require longer processing time, although the ALZ needs only slightly longer time than the LSTM networks. SPEED requires eight times longer time to model than the LSTM with SPEED-text. Note that when using all the sensors these two models achieve similar prediction accuracy. However, SPEED reaches a high accuracy with much less data.

### E. Discussion

We have applied two probabilistic methods on our data and have achieved comparable results to those obtained from the Mavlab testbed dataset. That testbed includes 50 sensors while our dataset was obtained from a real home with fifteen sensors, i.e. considerably fewer than the Mavlab testbed. In addition, in that work the same dataset was used both for training and for testing, which results in overfitting and overestimating the accuracy of the model. We use separate datasets for training, validation, and testing.

We have compared the performance of these two probabilistic methods with LSTM networks. To our knowledge this is the first time LSTM networks have been applied to this specific task. ESN, that is an RNN like LSTM, has shown good results [10], [11]. It is also the first time that probabilistic methods are compared to LSTM neural networks for sensor event prediction.

In our work, the best accuracy was achieved by the LSTM network with SPEED-text, 83% with all the fifteen sensors and 87% with seven motion sensors. In [12] an Elman network was applied to a dataset with 16 rooms and achieved peak accuracy of 91%, which is higher than our results. However, the dataset in that study was generated by the users themselves rather than being collected by sensors, a fact that is expected to lead to considerably fewer faulty events.

Our work showed that probabilistic methods can achieve a high prediction accuracy (close to their peak accuracy) with a relatively small amount of data (typically 2 days of data). LSTM networks require a larger dataset (about 3 weeks with SPEED-text and 10 weeks with ALZ-text) to reach good accuracy. Also, probabilistic methods are found to base the prediction on a relatively small number of previous events – an optimal memory length of four for ALZ and seven for SPEED was established in this work. On the other hand, LSTM networks base the prediction on a sequence of eight last events or more. This indicates that such networks are better at finding longer-term dependencies and patterns in a sequence of events. In addition, in LSTM the attained accuracy is quite stable for memory lengths that are larger than the optimal. On the other hand, probabilistic methods have an optimum memory length, hence the accuracy decreases both for shorter and for longer memory lengths than the optimum.

TABLE IV
EXECUTION TIME OF ALGORITHMS

| *Algorithm* | *Execution time (min)* |
|---|---|
| ALZ | 2.8 |
| SPEED | 16.5 |
| LSTM with ALZ-text | 2.1 |
| LSTM with SPEED-text | 1.5 |

For the dataset containing events from the fifteen sensors, our best result was achieved by the LSTM network with SPEED-text (83%). SPEED achieved only 1% lower accuracy, however, after considerably longer training time. Hence in applications where it is an advantage to model with a small amount of data where in addition execution time is not too critical, SPEED may be a good choice, since it can achieve an accuracy close to its peak with little data. In general, our results have shown that it is possible to achieve good accuracy with much less data than thought previously. SPEED and LSTM with SPEED-text achieve better results than ALZ and LSTM with ALZ-text. This is not surprising since the conversion of data to SPEED-text sequences contains more information (both "on" and "off" events). This can also be confirmed by the trees formed by ALZ and SPEED (Fig. 2 and 3).

For a dataset with no intertwined events though – the case of our dataset with only the seven motion sensors – the best choice is the LSTM with SPEED-text. SPEED does not work well in this case, since the tree has a height of 2 so that only "off" events can be predicted reliably.

Another interesting finding is that more data than 10 weeks does not improve significantly the results for any of the applied methods. Hence, a change in the algorithms and/or in the way the data are input, or additional information, is required to improve the prediction accuracy.

Finally, regarding the number of sensors. A larger number of sensors can lead to better prediction accuracy to the extent that it entails more information to base the prediction on. A smaller number may, however, be preferable both in terms of reduced surveillance for the user, lower cost, and less nuance for the esthetics of the home. Our work shows that it is possible to achieve acceptable prediction accuracy with much fewer sensors than thought previously.

## VI. Conclusions and Future Work

Activity recognition and prediction algorithms in smart home environments using binary sensors have been indicated to be useful for a number of functions. Most of the work reported in the literature has been carried out using data collected in lab environments and testbeds, with scripted activities. Such smart home testbeds typically include a quite large number of sensors, e.g. the Mavlab testbed deployed around 50 sensors [6].

In this paper we presented results on sensor sequence prediction using state-of-the-art methods: two probabilistic methods (ALZ and SPEED) and LSTM networks with both SPEED- and ALZ-text sequence inputs. Our dataset was obtained from a real home with an older adult ($>$ 65 years old) and with a relatively small number of sensors (15).

We compared all the methods with regard to a number of factors: the required number of preceding events to predict the next event (memory length), the necessary amount of data to achieve good accuracy and stability, the time used for training/testing, and the number of sensors in the dataset. To the extent of our knowledge, this is the first time such a comparison has been carried out. Our best implementation achieved an accuracy of 83% with LSTM with SPEED-text for a set with fifteen sensors in total – motion, magnetic and power sensors – and 87% with LSTM with SPEED-text as input for seven motion sensors. For the most accurate models using the SPEED-text, the LSTM required around 1/7 of the time SPEED required to do the modelling. On the other hand, the LSTM required about 3.5 weeks of data before reaching considerable (close to its peak) accuracy, whereas the probabilistic methods only needed 2 days of data for reaching considerable accuracy. The findings of our study can be useful for deciding which methods to use in accordance with project constraints (e.g. the number of available sensors, user privacy, etc.) and the area of application.

Clearly, a higher prediction accuracy is required before such algorithms can be applicable to real homes. Future work will include the time information as part of the input in order to improve the accuracy of our models. In addition, we will investigate the reproducibility of the best prediction model in other apartments with similar sensors and hence the variability of the predictions. Moreover, we will examine the possibility of using transfer learning methods across the apartments. These will be published in future work.

## Acknowledgment

The authors would like to thank the residents and the housekeepers at the seniors' care unit Skøyen Omsorg+; Torhild Holthe and Dr. Anne Lund (OsloMet) for recruiting participants for the trial and communicating with the residents throughout the trial; Dejan Krunić and Øyvind Width (Sensio AS) for installations of the sensors; and the rest of the participants of the Assisted Living Project for a fruitful interdisciplinary collaboration.

## References

[1] E. Zouganeli, F. Casagrande, T. Holthe, A. Lund, L. Halvorsrud, D. Karterud, A. Flakke-Johannessen, H. Lovett, S. Mørk, J. Strøm-Gundersen, E. Thorstensen, R. Norvoll, R. Meulen, M.-R. Kennedy, R. Owen, M. Ladikas, and E.-M. Forsberg, "Responsible development of self-learning assisted living technology for older adults with mild cognitive impairment or dementia," *ICT4AWE 2017 - Proceedings of the 3rd International Conference on Information and Communication Technologies for Ageing Well and e-Health*, no. Ict4awe, pp. 204–209, 2017.

[2] B. Winblad, K. Palmer, M. Kivipelto, V. Jelic, L. Fratiglioni, L.-O. Wahlund, a. Nordberg, L. Bäckman, M. Albert, O. Almkvist, H. Arai, H. Basun, K. Blennow, M. de Leon, C. DeCarli, T. Erkinjuntti, E. Giacobini, C. Graff, J. Hardy, C. Jack, a. Jorm, K. Ritchie, C. van Duijn, P. Visser, and R. C. Petersen, "Mild cognitive impairment–beyond controversies, towards a consensus: report of the International Working Group on Mild Cognitive Impairment." *Journal of internal medicine*, vol. 256, no. 3, pp. 240–246, 2004.

[3] S. Wu, J. B. Rendall, M. J. Smith, S. Zhu, J. Xu, H. Wang, Q. Yang, and P. Qin, "Survey on Prediction Algorithms in Smart Homes," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 636–644, 2017.

[4] L. B. Holder and D. J. Cook, "Automated activity-aware prompting for activity initiation," *Gerontechnology*, vol. 11, no. 4, pp. 534–544, 2013.

[5] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui, "SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment," *Artificial Intelligence in Medicine*, vol. 67, pp. 57–74, 2016.

[6] K. Gopalratnam and D. J. Cook, "Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm," *IEEE Intelligent Systems*, vol. 22, no. 1, 2007.

[7] M. R. Alam, M. B. Reaz, and M. A. Mohd Ali, "SPEED: An inhabitant activity prediction algorithm for smart homes," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 42, no. 4, pp. 985–990, 2012.

[8] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[9] J. G. Cleary and I. H. Witten, "Data Compression Using Adaptive Coding and Partial String Matching," *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.

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

[11] S. Mahmoud, A. Lotfi, and C. Langensiepen, "Behavioural pattern identification and prediction in intelligent environments," *Applied Soft Computing Journal*, vol. 13, no. 4, pp. 1813–1822, 2013.

[12] L. Vintan, A. Gellert, J. Petzold, and T. Ungerer, "Person movement prediction using neural networks," in *Proceedings of the KI2004 International Workshop on Modeling and Retrieval of Context*, vol. 114, no. 4, 2004, pp. 1–12.

[13] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Next Location Prediction Within a Smart Office Building," *Cognitive Science Research Paper Unversity of Sussex CSRP*, vol. 577, p. 69, 2005.

[14] E. Nazerfard and D. J. Cook, "CRAFFT: An Activity Prediction Model based on Bayesian Networks," vol. 33, no. 4, pp. 395–401, 2015.

[15] B. Minor and D. J. Cook, "Forecasting occurrences of activities," *Pervasive and Mobile Computing*, 2016.

[16] M. T. Moutacalli, A. Bouzouane, and B. Bouchard, "Sensors activation time predictions in smart home," *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, p. 37, 2015.

[17] N. E. Elhady and J. Provost, "A Systematic Survey on Sensor Failure Detection and Fault-Tolerance in Ambient Assisted Living," *Sensors*, vol. 18, no. 1991, p. 19, 2018.

[18] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, 1978.

[19] J. Martens, "Generating Text with Recurrent Neural Networks," *Neural Networks*, vol. 131, no. 1, pp. 1017–1024, 2011.

[20] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech Recognition With Deep Recurrent Neural Networks," *Icassp*, no. 3, pp. 6645–6649, 2013.

[21] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 115–143, 2003.

[22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] A. Graves, "Generating Sequences With Recurrent Neural Networks," *arXiv*, 2014.

[24] F. D. Casagrande and E. Zouganeli, "Occupancy and Daily Activity Event Modelling in Smart Homes for Older Adults with Mild Cognitive Impairment or Dementia," in *Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59)*, 2018, pp. 236–242.

Paper V

# Prediction of Next Sensor Event and its Time of Occurrence using Transfer Learning across Homes

**Flávia Dias Casagrande, Evi Zouganeli**

To be published in the Proceedings of the 16th ACS/IEEE International Conference on Computer Systems and Applications AICCSA 2019.

V

# Prediction of Next Sensor Event and its Time of Occurrence using Transfer Learning across Homes

***Abstract*—We present results on the prediction of sequential sensor events and time of occurrence using transfer learning with Recurrent Neural Network with Long Short-Term Memory, between five apartments. Our dataset has been collected from real homes with one resident each and contains data from a set of 13-17 sensors, depending on the apartment, including motion, magnetic, and power sensors. We compare the prediction accuracy and the required dataset size for the prediction when each apartment is modelled individually, and when transfer learning is used. Transfer learning is used in two configurations – a) training with data from four apartments and fine-tuning and testing on each of the target apartments, and b) training with one apartment and fine-tuning and testing on each of the target apartments. In our best prediction models, a top accuracy of 87% is attained when predicting the next sensor event, and 81% when predicting both the next sensor event and the mean time elapsed to the next sensor event. There is a variability of 10% in the attained prediction accuracy across apartments. For a small number of events in the target dataset, having a network pre-trained with data from four apartments and fine-tuned with the target apartment provides the best prediction models.**

***Index Terms*—smart home, sequence prediction, time prediction, binary sensors, recurrent neural network, transfer learning**

## I. Introduction

Activity recognition and prediction can be performed by a number of algorithms that have been reported in the literature. In the past years the use of transfer learning in the development of such algorithms has grown [1]. Transfer learning refers to training and learning parameters from a source dataset that is different yet related to a target dataset. It can result in reduced time for training the model and a lower amount of data required for the training. In addition, it is a solution for unlabeled data. Transfer learning also allows that some characteristics in the training and testing datasets are different, e.g. labels and data distributions [2]. It is therefore extremely applicable and useful for smart homes, where usually each home has its own layout, a different network of sensors and where, moreover, the residents may have individual habits.

Most of the reported work in this field has used data collected in the lab or in testbeds based on scripted activities. In addition, to the extent of our knowledge, there is no study that uses transfer learning to predict both the next sensor event and the time it will occur in the same algorithm, using long short-term memory (LSTM) recurrent neural network (RNN). This is the aim of the current paper where we use data collected from real homes.

Our work is part of an interdisciplinary project that involves experts in health, technology, and ethics [3]. The aim of the project is to develop assisted living technology (ALT) to support older adults with Mild Cognitive Impairment (MCI) or Dementia (D) live a safe and independent life at home. Functions that are able to assist older adults with MCI/D in their everyday life have been extensively studied. These can be for example functions for prompting the residents with reminders or encouragement, diagnosis tools, as well as prediction, anticipation and prevention of hazardous situations. These rely on quite robust and accurate activity recognition and prediction algorithms to perform well in real homes.

The performance of probabilistic methods and neural networks for the prediction of the next sensor event with data from one apartment has been compared in previous work [4], [5]. The best performing algorithm was shown to be the LSTM network with binary sensor events "on" and "off" converted to a text sequence. This algorithm was further developed to include the time information and predict both the next sensor event and its time of occurrence using data from the same apartment [6]. In the present paper we apply transfer learning with the best performing algorithms from that work to build prediction models for five apartments in total. The motivation for transfer learning is to reduce the time required for data collection in individual apartments before the system can be operated, and rather reuse the learning in already trained models.

The paper is organized as follows. Section II gives an overview of algorithms implemented in the literature for sensor event and time prediction, and for transfer learning. Section III gives an overview of our field trial and the sensor system in the apartments. Section IV describes the data preprocessing, used as input for the prediction method described in Section V. Section VI presents the results and discussion. Finally, in Section VII we conclude the paper and discuss future work.

## II. Related Work

### A. Prediction of Sensor Events and Time of Occurrence

Activity prediction includes mainly two tasks: sequence prediction and time prediction. A number of algorithms for sequence prediction have been studied in the past years [7]. These algorithms usually train a model based on a sequence of symbols to predict the next symbol. Active LeZi (ALZ) uses Markov Models to predict the next symbol in a sequence [1]. Inspired by ALZ, the Sequence Prediction via Enhanced Episode Discovery (SPEED) algorithm predicts the next sensor event based on the frequency of observed patterns [8]. Neural networks have also been used to predict the next sensor in a sequence with notable performance, typically recurrent neural networks (RNN) [9]–[11]. Other reported methods are Bayesian network, state prediction, and Markov predictor [12].

In addition to the next sensor event or activity in a sequence, such algorithms should also be able to predict when the event will occur. The time of occurrence is important to enable a number of smart home functions, for example improved operation of automation features; prompting systems [13]; or anomaly detection in certain behaviour patterns [14]. Time series methods such as Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) have been extensively applied in the literature [15]. However, they assume the time series to be linear, which is not applicable to activities in a home [16]. Rule-based algorithms have been developed for time forecasting [13], [17]. They are quite useful, however they do not account for more complex activities. Non-linear time series models would be more suitable for time prediction in smart homes, e.g. artificial neural networks. A Non-linear Autoregressive Network (NARX) showed promising results to predict the start and the end time of sensor activation [18]. Decision trees have been used to predict the time a certain activity would take place [16]. Poisson process has also been applied to predict the time an observed activity would occur [19]. Bayesian networks have been applied to predict the next location, time of day, day of the week and as a consequence the activity label of what the person is doing [20]. This work is the closest to ours and has reached an accuracy of 46-60% when predicting the next location, 66-87% for time of day (slots of 3 hours along the day), 89-97% predicting the day of the week, and 61-64% for activity recognition.

Our dataset was collected from a real home, while most datasets from the cited works have been collected through scripted activities primarily in lab environments. It contains events from 13-17 binary sensors, i.e. twice as many as used in [9], [18], and less than one third of the number of sensors used in [1], [8], [16]. Even though it is comparable to the work in [12] (16 rooms), in that study the events were inserted by each user rather than being generated using sensors. In addition, we predict both the next sensor event and the mean elapsed time to the next event in the same model.

### B. Transfer Learning

Transfer learning has been used in several fields, e.g. image and language classification, computer networks, automated planning, mathematical problems, and activity recognition [2], [21]. It has not been fully explored yet for time series data [22]. This might be because of the lack of available general purpose pre-trained models [22], as there are for image classification. However, this may change soon as transfer learning has proved to provide many advantages. For instance, it has been shown that models trained on features extracted using a pre-trained recurrent neural network (RNN) perform better or at least as well as RNNs trained for a specific task for electronic health records data prediction [23]. In addition, transfer learning can dramatically decrease the required amount of data in the target dataset, as proved for a mortality prediction algorithm [24] and for activity recognition [25], [26]. It also allows that datasets with different feature spaces can transfer the knowledge between each other [26], [27]. Besides, it can be applied to several algorithms: RNNs [23], Hidden Markov Models [28], statistical inference [24], support vector machine [25]. When it comes to transfer learning in smart homes, a cross-domain activity recognition algorithm combined with transfer learning and a similarity function between different activities was proposed [25]. In this work, three different datasets are used, where one is collected over 28 days from a real home of a 26-year-old man, and a peak accuracy of 65% is achieved with seven activities. Other work transfers the knowledge activities from multiple physical source spaces to a different target physical space [26]. The authors propose an algorithm that maps automatically activities from source to target environment and classifies the activity based on a weighted majority vote method. They use data collected from six testbeds where volunteers lived for 2-3 months, containing 5 to 11 activities, and achieve a peak accuracy of about 80%. HMM and transfer learning have also been combined and used across three apartments with five recorded activities and achieve 0.65 F1-score in the best case [29].

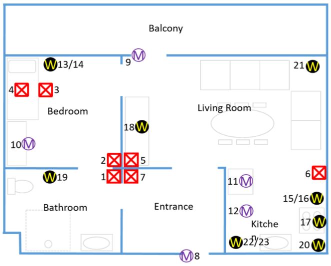
Transfer learning has its limitations. It has been shown that it can either improve or degrade the prediction accuracy of models depending on the dataset used for transfer, which is known as negative learning [2]. In these cases, it is important to detect which is the best source dataset to a problem, for example using Dynamic Time Warping to measure inter-dataset similarities [22].

To the extent of our knowledge LSTM networks have not been previously used for the prediction of sequential sensor events using transfer learning. In addition to predicting the next sensor event, we predict the mean elapsed time of occurrence in the same model.

## III. Field Trial

Nine residents over 80 years old participate in our field trial. In this work we use data from five apartments. All apartments are part of a community care facility and have similar layouts – comprising a bedroom, a living room, an open kitchen area, a bathroom, and an entrance hall (Fig. 1). A minimal number of binary sensors has been installed in the apartments to minimize surveillance of the residents. The set of sensors has been chosen so that it can enable the realization of useful functions for older adults with MCI/D as these were indicated at dialogue cafes with the users [3]. Hence, our set of sensors contains motion (passive infrared sensor – PIR), magnetic, and power sensors. These enable inference of occupancy patterns (movement around the apartment) and some daily activities – kitchen related activities, dressing, being in bed –, and leisure activities - reading, watching TV, listening to radio.

Unfortunately, not all apartments could have the exact same set of sensors due to physical limitations (e.g. fridge door with a too big gap to enable the use of magnetic sensor) and/ or different equipment (e.g. residents either have a coffee machine or a kettle). However, all the participants had the same initial proposal of set of sensors, as shown in Fig. 1. The five apartments that provided data to this work have all the motion sensors and the power sensor in the TV, while the rest of the sensors are different, as shown in Table I.




**PIR motion sensors**
1 – 7: bathroom, bedroom, nearby bed, over the bed, living room, kitchen, entrance
**Magnetic sensors**
8 - 12: entrance and back doors, wadrobe door bedroom, fridge door, cupboard/drawer
**Power sensors**
13-23: radio, night stand lamp, coffee machine, kettle, stove, TV, bathroom lamp, kitchen lamp, living room/reading lamp, microwave, toaster

Fig. 1. Proposed sensors system for field trial apartments.

TABLE I
SENSORS IN EACH APARTMENT

| Apt. ID | Sensors |
|---|---|
| 1 | P[a]: night stand lamp, coffee machine, living room/reading lamp; M[b]: cupboard/drawer |
| 2 | P[a]: night stand lamp, coffee machine, living room/reading lamp, microwave; M[b]: fridge |
| 3 | P[a]: kettle, living room/reading lamp, microwave, toaster; M[b]: fridge, cupboard/drawer |
| 4 | P[a]: night stand lamp, coffee machine, living room/reading lamp; M[b]: fridge |
| 5 | P[a]: kettle; M[b]: fridge, cupboard/drawer |

[a]Power and [b]magnetic sensors.

The sensors are connected wirelessly through Z-Wave and xComfort protocols to a Raspberry Pi 3, which transfers the data for storage in a secure server. The data comprise timestamp (date and time with precision of seconds), sensor ID, and sensor message (binary). Table II shows a data sample.

TABLE II
BINARY SENSORS DATA

| Timestamp | Sensor ID | Sensor message |
|---|---|---|
| 01.09.2017 07:58:40 | 4 | 1 |
| 01.09.2017 08:00:14 | 12 | 1 |
| 01.09.2017 08:03:05 | 12 | 0 |

## IV. DATA PREPROCESSING

### A. Data Correction

Data acquired from binary sensors often contain faulty events e.g. erroneous activation of motion sensors by sunlight and switch-off delays of motion sensors [30]. Such noise can significantly affect the performance of the models. Hence, we have carried out a data correction preprocessing. Occasionally the motion sensors do not send an activation event when they should. We insert missing events so the data can be more accurate. For example, it is not possible to go to the bedroom directly from the kitchen without passing through the living room. When the living room activation event is missing, it is inserted. If there are two possible sensor events (e.g. two possible paths in the apartment), the choice of the inserted sensor event is done randomly and in a way that the final percentage distribution of the two options remains as observed in the original data. This is done in all data, i.e. both the training and the testing sets. The time of the inserted event is the mean between the previous and next samples. This does not compromise the data accuracy because the faulty events are usually between relatively fast motions around the apartment, which means the elapsed time between the samples is not long.

### B. Data Conversion

The corrected data are subsequently converted to a text sequence. The sensor's ID and message are assigned a letter. This is inspired by the SPEED algorithm [8]. SPEED is a sequence prediction algorithm based on the occurrence of frequent patterns in home environments. SPEED uses upper- and lower- case letters to represent a sensor's "on" and "off" events. For the sample data in Table II, SPEED would generate the sequence "ABb", where sensors 4 and 12 are assigned the letters a/A and b/B, respectively. With each sensor being assigned a letter, we now include the time information. This was performed in two ways, depending on what we are predicting, using the best performing algorithms according to previous work [6].

*1) Sensor Event with Time Elapsed to the Next Event:* When predicting the next sensor event only, we use together with the sensor's letter a number that indicates the time elapsed to the next event. We define a set of 8 time intervals: [$<$ 1min, 1-5min, 5-15min, 15-30min, 30min-1h, 1-2h, 2-5h, $>$ 5h]. Hence, we assign numbers 0-7 to the event. For example, if the motion sensor in the bedroom (assigned letter a/A) were activated in the morning and 10 min later the person went to the bathroom, the generated symbol would be "A2".

*2) Sensor and Cluster with Hour of the Day and Elapsed Time to the Next Event:* When predicting both the next sensor and time elapsed to the next sensor event, we add a number after the sensor letter to indicate a time-related cluster as follows. We apply the K-means algorithm to cluster each sensor event according to the hour of the day occurrence and the time elapsed to the following sensor event. The samples of each sensor are classified into K clusters such that the sum of square distances (SSD) within the clusters is minimized [31]. Each cluster contains a centroid, given by the mean values of each feature of the algorithm. We perform K-means for a maximum number of clusters (K) equal to 8 and choose the best K manually according to the elbow method [32]. This method consists of plotting an SSD vs K graph and choosing the K that resembles an “elbow” (the point of inflection on the curve), which is the best fit for that problem. Fig. 2 shows an example of clustering the samples of the motion sensor in the kitchen. This sensor results in four clusters (represented by the different colors). Suppose this sensor is represented by letter B, has had an “on” event at noon (blue cluster), and the next sensor event took place 3 minutes later. This would generate “B2” (2 representing the blue cluster).

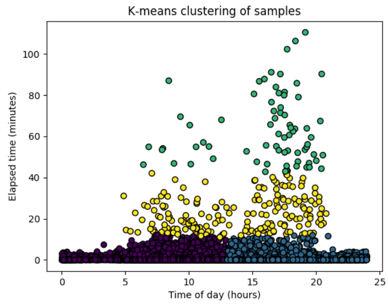



Fig. 2. K-means clustering of motion sensor events in the kitchen.

## V. Prediction Method

### A. Sensors Mapping

As described in Section III, some power and magnetic sensors differ within the five apartments (Table I). In order to transfer the learning across the apartments, we re-label the sensors that refer to the same activity. The new labels and the sensors assigned to these are shown in Table III. Lamp power sensor events were removed from the datasets since we did not manage to assign them to an activity that was common for all lamps and apartments.

TABLE III
RE-LABELING OF SENSORS

| New labels | Sensors |
|---|---|
| Kitchen sensor | P[a]: toaster, microwave; M[b]: fridge, cupboard/drawer |
| Beverage sensor | P[a]: coffee machine, kettle |

[a]Power and [b]magnetic sensors.

### B. Long Short-Term Memory

RNN [33] is a neural network that has the property of keeping an internal memory, and has therefore been widely applied to inputs that are sequential in time [34], [35]. The LSTM [36] is a type of RNN designed to be better at storing and accessing information than the standard RNN.

We employ an LSTM network configured as a text generation network as this was the best performing algorithm for our data and sensor system’s configuration [4], [5]. The number of inputs is a certain number of sensor events – equal to the memory length – and the output is the predicted next event in the sequence (Fig. 3). The input and output are one-hot encoded. In the one-hot encoding representation, each symbol is represented by a vector of bits of length equal to the number of symbols in a sequence. All values are zero, except for the one corresponding to that symbol (Fig. 3).

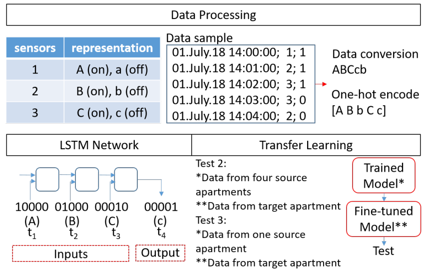



Fig. 3. LSTM network configuration and transfer learning process.

We first train an individual model for each apartment, which we refer to test 1. When using transfer learning, we first train the LSTM network with data from four apartments (test 2) or from one apartment (test 3). In this case, the data from the target apartment – that have not been used in the training – are split to be used in the fine-tuning of the network, keeping the weights of the best-fit model, and in the testing (Fig. 3).

A stateless LSTM network model was implemented in Python 3 using Keras open source library for neural networks. A number of parameters were tuned in order to find the optimal values. The memory length (i.e. number of symbols that are used to predict the next symbol) had value 10. The model has 1 hidden layer with hyperbolic tangent activation and 64 neurons. Our batch size (i.e. number of samples used for training each iteration of the epoch) was 512. We used Adam as the optimization function with learning rate of 0.01 and categorical cross-entropy as loss function. The output layer was a softmax activation function. We used the early stopping method and dropout rate of 50% to avoid overfitting, allowing a maximum of 200 epochs for each model’s training.

## VI. Results and Discussion

Table IV shows the number of sensor events in the dataset of each apartment and the number of days it has been collected. In all cases, the LSTM network was trained based on a certain number of events and tested on a test set containing 3000 events. This process is repeated three times and the accuracy values in the graph correspond to the mean of the best test accuracy of each training. We show graphs up to 10000 training events for better clarity at the lower range in the graph, for comparison purposes.

We have three tests in each subsection, both when predicting the next sensor event only, and when predicting the next sensor event as well as the time-related cluster. In test 1, we model an individual LSTM network for each apartment. In test 2, transfer learning is used where the LSTM network is trained with data from four apartments and fine-tuned with data from the fifth apartment. In test 3, transfer learning is used where the model is trained with data from the apartment that obtained the best accuracy in the first test and fine-tuned for each of the test apartments.

TABLE IV
NUMBER OF EVENTS PER APARTMENT

| Apt. ID | Number of Events | Number of Days |
|---|---|---|
| 1 | 169082 | 264 |
| 2 | 94209 | 217 |
| 3 | 17130 | 69 |
| 4 | 115744 | 291 |
| 5 | 190796 | 258 |

### A. *Prediction of the Next Sensor*

We investigate the accuracy of predicting the next sensor event only using as input the sensor event and the elapsed time to the next event (see Section IV-B1). Fig. 4 shows the prediction accuracy of the first test performed, with individual LSTM network models for each apartment. Note that with a small number of events the accuracies are pretty low, and increases sharply with dataset size. After 4000 events in the training dataset, the accuracy increases very slowly. The top accuracies achieved and the number of events required are presented in Table V. The best accuracy of 86.96% is achieved for apartment 2. Somewhat lower but comparable accuracies are achieved in apartments 1 and 3, while the lowest accuracy is obtained in apartments 4 and 5. It is interesting to compare the number of events required to reach the top accuracy for each apartment. Twice as many events were required in apartment 1 as in apartment 2. On the other hand, in apartment 3, 1/8 of the number of events was required compared with apartment 2 to achieve approximately the same accuracy.

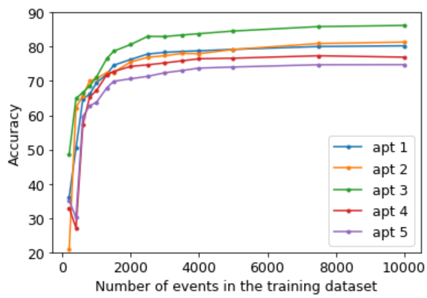



Fig. 4. Accuracy of prediction of the next sensor vs. number of events in the training dataset, using as input both sensor events and elapsed time to the next sensor event. Individual LSTM model per apartment (test 1).

Fig. 5 shows the accuracy curves when we perform test 2 – train the network with data from four apartments and fine-tune with the fifth apartment that is the test apartment. A very low number of events is required for the fine-tuning to achieve quite high accuracy straight away. The accuracy increases slowly as more events are added for the fine-tuning. In Table V we see the attained accuracies and the required number of events for test 2. For all apartments, the accuracies are marginally lower ($< 1\%$) than when using individual models for each apartment (test 1). The number of events required in the training dataset is the same. We have also computed the accuracy without fine-tuning prior to testing, i.e. when we test the network trained with data from four apartments directly on the test dataset of the fifth apartment. This led to top accuracies of 76.87%, 79.13%, 66.20%, 75.70%, and 71.1%, in apartments 1 to 5 respectively. This shows that the fine-tuning of the model is indeed required to achieve good prediction accuracy when using transfer learning across apartments.

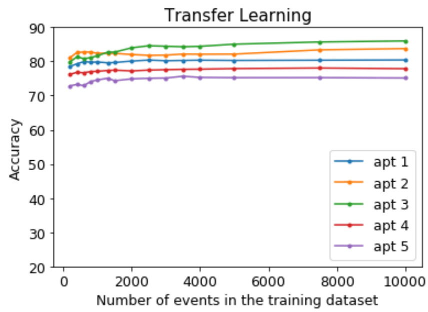



Fig. 5. Accuracy of prediction of the next sensor vs. number of events in the training dataset, using as input both sensor event and elapsed time to the next sensor event. Transfer learning, test 2 – training the model with data from four apartments, fine-tuning with and testing on the target apartment.

Lastly, we investigate the prediction accuracy when training the model with the apartment that achieved best accuracy in test 1 (apartment 2) and fine-tuning with each test apartment individually (Fig. 6). In this case the accuracies obtained for a low number of events in the training set are lower than in the previous transfer learning (test 2). This is as expected as we now have much less data to train the network with. However, the prediction accuracy for the low range of the graphs is still quite higher than in test 1. From Table V we observe that the top accuracies are very similar as for test 2 and the amount of data required is similar. However, in test 3, the time to train the network is also considerably lower since data from only one apartment are used.

TABLE V
PREDICTION ACCURACY OF THE NEXT SENSOR EVENT

| Apt. ID | Top Mean Accuracy (Number of Events Required) | | |
|---|---|---|---|
| | *Test 1* | *Test 2* | *Test 3* |
| 1 | 84.90% (160000) | 84.81% (160000) | 84.64% (160000) |
| 2 | 86.96% (80000) | 86.20% (70000) | – |
| 3 | 86.23% (10000) | 86.00% (10000) | 86.09% (10000) |
| 4 | 78.91% (100000) | 78.90% (100000) | 78.90% (100000) |
| 5 | 80.78% (180000) | 80.53% (180000) | 80.61% (180000) |

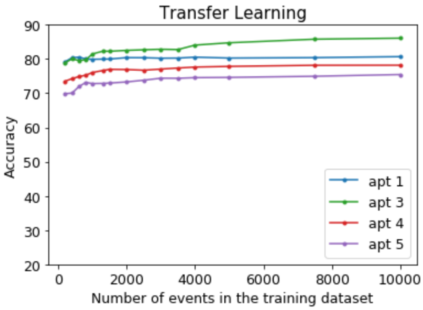



Fig. 6. Accuracy of prediction of the next sensor vs. number of events in the training dataset, using as input both sensor event and elapsed time to the next sensor event. Transfer learning, test 3 – training the model with data from apartment 2, and fine-tuning with and testing on the target apartment.

### B. Prediction of the Next Sensor Event and Mean Elapsed Time to the Next Event

In the following we predict both the next sensor event and the time-related cluster containing information for the mean elapsed time to the next event (see Section IV-B2). Fig. 7 shows the prediction accuracy when each apartment is modelled individually. In all cases, the accuracy increases sharply with increasing number of training events, and stabilizes from about 7000 events. Thereafter, the accuracy increases very slowly as more data are added. Table VI shows the top accuracies attained. The accuracies range from 74-81%. Once again the highest accuracy is attained in apartments 1, 2 and 3, and the lowest in apartments 4 and 5.

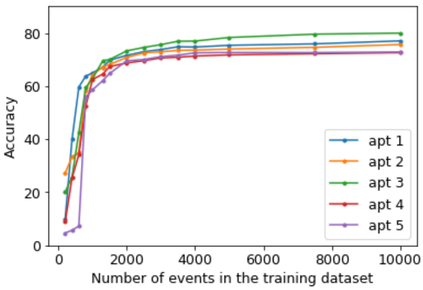



Fig. 7. Accuracy of prediction of the next sensor and time cluster vs. number of events in the training dataset, using as input both sensor event and time cluster. Individual LSTM model per apartment (test 1).

Fig. 8 and 9 present the accuracy results when transfer learning is carried out as above, tests 2 and 3. The overall results are similar to these obtained when predicting the next sensor event only. The prediction accuracy attained for a small number of events is much higher when using transfer learning as compared with each apartment being modelled individually. The top accuracies are very similar, and so is the number of required events to attain these. When comparing transfer learning from four apartments to transfer learning from one apartment (Table VI), it turns out that the top prediction accuracies are only marginally different.

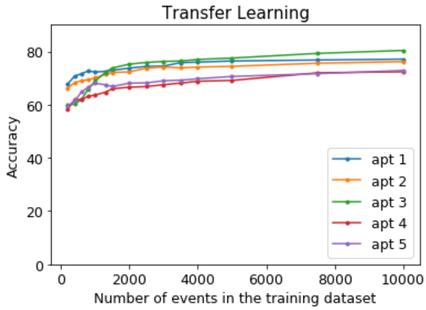



Fig. 8. Accuracy of prediction of the next sensor and time cluster vs. number of events in the training dataset, using as input both sensor event and time cluster. Transfer learning, test 2 – training the model with data from four apartments, fine-tuning with and testing on the target apartment.

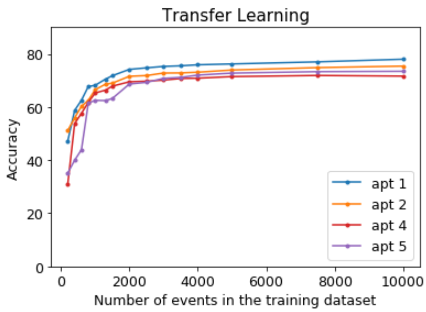



Fig. 9. Accuracy of prediction of the next sensor and time cluster vs. number of events in the training dataset, using as input both sensor event and time cluster. Transfer learning, test 3 – training the model with data from one apartment (apartment 3) and fine-tuning with and testing on the target apartment.

TABLE VI
PREDICTION ACCURACY OF NEXT SENSOR EVENT AND TIME CLUSTER

| Apt. ID | Top Mean Accuracy in Stability (Number of Events) | | |
|---|---|---|---|
| | *Test 1* | *Test 2* | *Test 3* |
| 1 | 80.92% (160000) | 81.26% (160000) | 80.02% (160000) |
| 2 | 80.99% (80000) | 80.39% (80000) | 80.67% (80000) |
| 3 | 79.94% (10000) | 80.37% (10000) | – |
| 4 | 73.97% (100000) | 73.78% (100000) | 73.17% (40000) |
| 5 | 77.78% (180000) | 77.52% (180000) | 77.43% (180000) |

### C. Summary and Discussion

For our set of apartments, transfer learning has been shown to work successfully up to a certain number of events. For a low number of events in the training dataset, up to about 4000 events, transfer learning leads to higher prediction accuracy than when each apartment is modelled individually. This means that when a new apartment is added to the study, the prediction algorithm can work well straight away, and attain a relatively good accuracy (70-80%) from the first day. However, for larger training datasets, the prediction accuracy is approximately the same. In fact, in most cases is it marginally higher when each apartment is modelled individually.

An interesting observation is that different prediction accuracies are attained in the five apartments. This could be due to the amount of data available in each apartment. In Table IV we notice the average number of events per day is roughly 640, 434, 248, 397, 740 for apartments 1 to 5, respectively. This indicates that the degree of activity varies significantly and/or that some of the residents are more active when in the apartment (e.g. apartments 1 and 5) than others. Alternatively, the residents with the low number of events per day might be absent from the apartments for longer periods, however, this does not seem to be the case when inspecting the data. Nonetheless, the average number of events per day does not seem to have a direct influence on the achieved prediction accuracy. For instance, relatively high prediction accuracy (86% and 80%) has been achieved for apartment 3 that only has 248 events per day, whereas much lower prediction accuracies (81% and 78%) are achieved in apartment 5 that has the highest number of events per day (740). Also, comparable accuracies are attained in apartment 1 (85% and 81%) as to apartment 3, although there are on average more than twice as many events per day in the former than in the latter. Hence, there is no correlation between accuracy and the amount of data here. One hypothesis for the variability across apartments can be that the accuracy correlates inversely with the resident's degree of impairment. Indeed, the resident in the apartment that attains the lowest prediction accuracy, apartment 5, was the one with a noticeable degree of impairment according to non-expert observations. However, we are not in the position to quantify the correlation for the remaining residents at the point of writing.

## VII. Conclusion and Future Work

Sequential sensor events and time of occurrence prediction algorithms can enable activity recognition and prediction in smart home environments and be the basis for a number of support functions in the home. Most of the research work in the literature has been carried out using data collected in lab environments and testbeds, typically including a quite large number of binary sensors (e.g. 50 sensors [1]). We collected data from five apartments in a community care facility, with one resident each. Data were collected from about 15 sensors per apartment over a period of time ranging from 69 to 291 days, depending on the apartment.

We have presented results on sensor event sequence prediction and time of occurrence prediction using transfer learning between apartments. We use LSTM networks with text-sequences that indicate the sensor events as inputs. We use two different ways for transfer learning and compare to the case when each apartment is modelled individually, i.e. without transfer learning. In general, the top prediction accuracies are achieved when each apartment is trained individually. However, for a small number of events in the target dataset, pretraining the network with data from four apartments and fine-tuning with data from the target apartment resulted in the best accuracy. This confirms the usefulness of transfer learning when a limited amount of data is available for the target model.

We attain a top accuracy of 87% when predicting the next sensor event, which is higher than what has been achieved previously (80% [26]). In addition, we predict both the next sensor event and the mean time elapsed to the next event with an accuracy of 81%. There is a 10% variability in the attained accuracy across apartments. The correlation of the variability between apartments with possible cognitive impairment, as well as other causes, will be further investigated in future work when the remaining apartments will also be included. Better accuracy is required for the algorithms to be applicable in real homes. In future work we will aim at carrying out activity recognition using the binary sensor events. Our hypothesis is that clustering sensors in activities and predicting these may improve the prediction accuracy.

## References


[1] K. Gopalratnam and D. J. Cook, "Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm," *IEEE Intelligent Systems*, vol. 22, no. 1, 2007.

[2] S. J. Pan and Q. Y. Fellow, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, pp. 279–286, 2009.

[3] *Anonymous. Details omitted for double-blind reviewing*, 2017.

[4] *Anonymous. Details omitted for double-blind reviewing* , 2019.

[5] *Anonymous. Details omitted for double-blind reviewing*, 2018.

[6] *Anonymous. Details omitted for double-blind reviewing*, 2019.

[7] S. Wu, J. B. Rendall, M. J. Smith, S. Zhu, J. Xu, H. Wang, Q. Yang, and P. Qin, "Survey on Prediction Algorithms in Smart Homes," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 636–644, 2017.

[8] M. R. Alam, M. B. Reaz, and M. A. Mohd Ali, "SPEED: An inhabitant activity prediction algorithm for smart homes," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 42, no. 4, pp. 985–990, 2012.

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

[10] *Anonymous. Details omitted for double-blind reviewing*, 2018.

[11] L. Vintan, A. Gellert, J. Petzold, and T. Ungerer, "Person Movement Prediction Using Neural Networks," *Computer*, no. April, 2004.

[12] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Next Location Prediction Within a Smart Office Building," *Cognitive Science Research Paper Unversity of Sussex CSRP*, vol. 577, p. 69, 2005.

[13] L. B. Holder and D. J. Cook, "Automated activity-aware prompting for activity initiation," *Gerontechnology*, vol. 11, no. 4, pp. 534–544, 2013.

[14] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui, "SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment," *Artificial Intelligence in Medicine*, vol. 67, pp. 57–74, 2016.

[15] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. San Francisco, CA, USA: Holden-Day, Inc., 1990.

[16] B. Minor and D. J. Cook, "Forecasting occurrences of activities," *Pervasive and Mobile Computing*, 2016.

[17] A. Aztiria, J. C. Augusto, A. Izaguirre, and D. Cook, "Learning accurate temporal relations from user actions in intelligent environments," in *Advances in Soft Computing*, vol. 51, 2009, pp. 274–283.

[18] S. Mahmoud, A. Lotfi, and C. Langensiepen, "Behavioural pattern identification and prediction in intelligent environments," *Applied Soft Computing Journal*, vol. 13, no. 4, pp. 1813–1822, 2013.

[19] T. Mahmud, M. Hasan, A. Chakraborty, and A. K. Roy-Chowdhury, "A poisson process model for activity forecasting," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 3339–3343.
[20] E. Nazerfard and D. J. Cook, "CRAFFT: An Activity Prediction Model based on Bayesian Networks," vol. 33, no. 4, pp. 395–401, 2015.
[21] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and Information Systems*, vol. 36, no. 3, pp. 537–556, 2013.
[22] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Transfer learning for time series classification," 2018.
[23] P. Gupta, P. Malhotra, L. Vig, and G. Shroff, "Transfer Learning for Clinical Time Series Analysis using Recurrent Neural Networks," pp. 4–7, 2018.
[24] T. Desautels, J. Calvert, J. Hoffman, Q. Mao, M. Jay, G. Fletcher, C. Barton, U. Chettipally, . Yaniv Kerem5, and R. Das1, "Using Transfer Learning for Improved Mortality Prediction in a Data-Scarce Hospital Setting," *African Journal of Neurological Sciences*, vol. 34, no. 1, 2015.
[25] D. H. Hu, V. W. Zheng, and Q. Yang, "Cross-domain activity recognition via transfer learning," *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 344–358, 2010.
[26] P. Rashidi and D. J. Cook, "Activity knowledge transfer in smart environments," *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 331–343, 2011.
[27] D. H. Hu and Q. Yang, "Transfer learning for activity recognition via sensor mapping," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 1962–1967, 2011.
[28] P. Jaini, Z. Chen, P. Carbajal, E. Law, L. Middleton, K. Regan, M. Schaekermann, G. Trimponias, J. Tung, P. Poupart, and M. Engineering, "Online Bayesian Transfer Learning for Sequential Data Modeling," pp. 1–20, 2017.
[29] T. L. M. V. Kasteren, G. Englebienne, and B. J. A. Kr, "Transferring Knowledge of Activity Recognition accross Sensor Networks," *Pervasive*, pp. 283–300, 2010.
[30] N. E. Elhady and J. Provost, "A Systematic Survey on Sensor Failure Detection and Fault-Tolerance in Ambient Assisted Living," *Sensors*, vol. 18, no. 1991, p. 19, 2018.
[31] K. M. Bataineh, M. Najia, and M. Saqera, "A comparison study between various fuzzy clustering algorithms," 2011.
[32] K. D. Joshi and P. S. Nalwade, "Modified k-means for better initial cluster centres," 2013.
[33] A. Graves, "Generating Sequences With Recurrent Neural Networks," *arXiv*, 2014.
[34] J. Martens, "Generating Text with Recurrent Neural Networks," *Neural Networks*, vol. 131, no. 1, pp. 1017–1024, 2011.
[35] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech Recognition With Deep Recurrent Neural Networks," *Icassp*, no. 3, pp. 6645–6649, 2013.
[36] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

Paper VI

# Predicting Sensor Events, Activities, and Time of Occurrence Using Binary Sensor Data from Homes with Older Adults

**Flávia Dias Casagrande, Jim Tørresen, Evi Zouganeli**

Published in: IEEE Access Journal.
DOI: 10.1109/ACCESS.2019.2933994

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2017.DOI*

# Predicting Sensor Events, Activities, and Time of Occurrence Using Binary Sensor Data from Homes with Older Adults

**FLÁVIA D. CASAGRANDE[1], JIM TØRRESEN[2], AND EVI ZOUGANELI.[1]**
[1]Department of Mechanical, Electronics, and Chemical Engineering, OsloMet – Oslo Metropolitan University, Pilestredet 35, 0166 Oslo, Norway (e-mail: {flacas,evizou}@oslomet.no)
[2]Department of Informatics, University of Oslo, Gaustadalléen 23B, 0373 Oslo, Norway (e-mail: jimtoer@ifi.uio.no)

Corresponding author: Flávia D. Casagrande (e-mail: flacas@oslomet.no).

This work was supported by the Norwegian Research Council under the SAMANSVAR programme (247620/O70).

**ABSTRACT**
We present a comprehensive study of state-of-the-art algorithms for the prediction of sensor events and activities of daily living in smart homes. Data have been collected from eight smart homes with real users and 13-17 binary sensors each – including motion, magnetic, and power sensors. We apply two probabilistic methods, namely Sequence Prediction via Enhanced Episode Discovery and Active LeZi, as well as Long Short-Term Memory Recurrent Neural Network, in order to predict the next sensor event in a sequence. We compare these with respect to the required number of preceding sensor events to predict the next, the necessary amount of data to achieve good accuracy and convergence, as well as varying the number of sensors in the dataset. The best-performing method is further improved by including information on the time of occurrence to predict the next sensor event only, and in addition to predict both the next sensor event and the mean time of occurrence in the same model. Subsequently, we apply transfer learning across apartments to investigate its applicability, advantages, and limitations for this setup. Our best implementation achieved an accuracy of 77-87% for predicting the next sensor event, and an accuracy of 73-83% when predicting both the next sensor event and the mean time elapsed to the next sensor event. Finally, we investigate the performance of predicting daily living activities derived from the sensor events. We can predict activities with an accuracy of 61-90%, depending on the apartment.

**INDEX TERMS** Binary sensor, probabilistic method, recurrent neural network, sequence and time prediction, transfer learning.

## I. INTRODUCTION

ACTIVITY recognition and prediction are a prerequisite for the realisation of intelligent support functions in smart homes, including functions that support older adults with mild cognitive impairment or dementia (MCI/D) live a safe and independent life at home. MCI/D is a cognitive decline that can affect attention, concentration, memory, comprehension, reasoning, and problem solving [1]. A fair amount of research on smart home functions has aimed at assisting older adults with MCI/D in their everyday life [2]. Examples are prompting with reminders or encouragement [3], [4], diagnosis tools [5], [6], as well as prediction, anticipation, and prevention of hazardous situations [7], [8].

A number of algorithms for activity recognition and prediction have been reported in the literature. However, most of the work in the literature uses data collected in the lab or in testbeds based on scripted activities. In addition, there is no comparative study investigating state-of-the-art algorithms applied to data collected from real homes, different configurations for input of data, limitations, and suitable applications. This is the focus of this work, where we use data collected from real homes, analyze, and compare the performance of state-of-the-art prediction algorithms. The work has been carried out in an interdisciplinary project, the Assisted Living Project (ALP), that involves experts in health, technology, and ethics [9]. The aim of the project is to develop assisted living technology (ALT) to support older adults with MCI/D live a safe and independent life at home.

In this paper, we start our analysis by comparing the performance of state-of-the-art prediction algorithms – probabilistic methods and neural networks – for the prediction of the next sensor event based on previous sensor events. Their performance is assessed with regard to a number of factors: the required number of preceding events to predict the next event from (which we refer to as "memory length"), the necessary amount of data to achieve good accuracy and convergence, and the number of sensors in the dataset. The best-performing algorithm is further improved by including the time of occurrence information in several ways. Part of this work has been previously published [10]–[13], however, using data from one apartment only. We have also examined the prediction accuracy across some of the apartments and the performance when using transfer learning [14]. In the current paper, we expand the analysis to include all eight apartments in the field trial in order to analyze the variability of the prediction accuracy across residents. In addition, we analyze the feasibility of extracting daily living activities from the sensor events and predicting the next activity rather than the next sensor, as well as its time of occurrence.

The paper is organized as follows. Section II gives an overview of algorithms used for sequential sensor event and activity prediction in the literature, of work related to prediction of the time of occurrence, and of transfer learning. Section III presents our field trial, the sensor system in the apartments, and the format of the collected data. Section IV describes the prediction methods, followed by the description of data preprocessing in Section V. Sections VI, VII, and VIII present the results and discussion for the prediction of the next sensor event and its mean time of occurrence, transfer learning, and activity prediction, respectively. These are illustrated in Fig. 1, for better understanding. Finally, in Section IX, we discuss our findings and conclude the paper.

## II. RELATED WORK

Activity prediction includes mainly two tasks: sequence prediction and time prediction. Such algorithms can for instance lead to an improved operation of automation functions (e.g. adjust the temperature sufficient time prior to the person waking up); enable the realization of prompting systems (e.g. prompt the resident if the predicted activity has not been performed) [15]; or identify changes and anomalies in certain behaviour patterns (e.g. movement, everyday habits, etc.) and thus indicate the onset or the progress of a condition [16].

A number of algorithms for sequence prediction have been studied in the past years [17]. These algorithms usually train a model based on a sequence of symbols to predict the next symbol. The Active LeZi (ALZ) is a probabilistic method that has been extensively employed for prediction of sequential data [18]. It achieved a peak accuracy of 47% when applied on the Mavlab testbed dataset, that includes 50 binary sensors [18]. The Sequence Prediction via Enhanced Episode Discovery (SPEED) algorithm has been implemented based on ALZ [19]. SPEED was applied on the Mavlab dataset and reached an accuracy of 88.3% when the same dataset was used both for training and for testing. Both algorithms convert the data of binary sensors to a sequence of letters and build a tree based on the observed patterns and corresponding frequency of occurrence. Neural networks have also been used for sensor event prediction with notable performance, typically recurrent neural networks (RNN) [11], [20]–[22]. Three RNN models – Echo State Network (ESN), Back Propagation Through Time (BPTT), and Real Time Recurrent Learning (RTRL) – were applied on a fourteen-day dataset with only six binary sensors (four motion and two magnetic). The ESN performed better with a root square mean error (RMSE) of 0.06 [20]. In these networks, the number of input and output values corresponded to the number of sensors in the dataset, and each assumed value "0" or "1" for being "off" or "on" at a certain time slot. The prediction in this case was computed for the next six hours. A similar study was carried out for a 16-room office environment [21]. The dataset in this case was collected through an app installed on the personal data assistant (PDA) of participating employees that had to register manually whenever they entered/left a certain room. An Elman network and a multilayer perceptron network were applied to predict the next room a person would go to. There were four participants in the study and the Elman network attained the best results, ranging from 70% to 91% accuracy, depending on the user. Each room was codified in four bits, as there were 16 rooms in total. The input corresponded to two rooms and the output to the predicted next room. This work also applied other methods – Bayesian network, state prediction, and Markov predictor – where comparable results were achieved [22].

In addition to sequence prediction, these algorithms should also be able to predict when the next symbol (representing either a sensor or an activity) will occur. The time series methods Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) have been extensively applied in the literature [23]. Nevertheless, they assume the time series to be linear, which is not applicable to activities in a home [24]. Rule-based algorithms have been developed for time forecasting as well [15], [25]. They are quite useful, however they do not account for more complex activities. Non-linear time series models would be more suitable to time prediction in smart homes, e.g. artificial neural networks. A Non-linear Autoregressive Network (NARX) was compared to an Elman network to predict a sensor activation's start and end time [26]. In this study, each sensor had its own network trained and tested on a twenty-day dataset with six binary sensors. The NARX performed better, with a RMSE ranging from 0.06 to 0.09, depending on the sensor. Decision trees have been used to predict the time a certain activity would happen [24]. This method relies on several features extracted from sensor events sequences. It was applied on a dataset with 51 both binary and sampling sensors and achieved an average normalized RMSE of 0.01. Bayesian networks have been used to predict the next location, time of day, and day of the week a person would execute an activity [27]. This algorithm was employed in two apartments with about 30 binary sensors each, where the next location was predicted with 47% and 61%. Poisson process

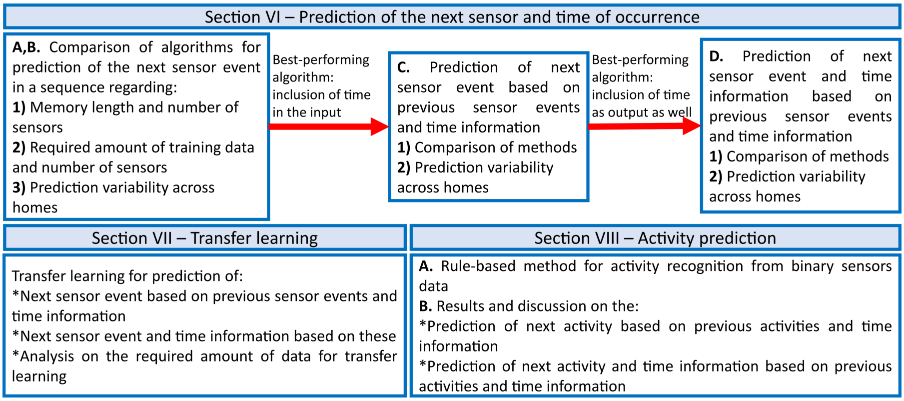



**FIGURE 1.** Content of results and discussion sections in the paper.

has also been applied to predict the time an observed activity would occur [28]. An RMSE of 3.9431 seconds was achieved in this work.

Taking into account that each individual has their unique habits, and smart homes may have different layouts and limitations for deployment of sensors, it is important that the prediction algorithm is able to adapt to each home and resident. Transfer learning can reveal whether the algorithm can adapt. This technique consists of training and learning parameters from a source dataset that is different yet related to a target dataset (e.g. different labels and data distributions [29]). Transfer learning has been used in several fields, e.g. image and language classification, computer networks, automated planning, mathematical problems, and activity recognition [29], [30]. This method has proved to provide many advantages. For instance, it allows that datasets with different feature spaces can transfer the knowledge between each other [31], [32]. In addition, transfer learning can dramatically decrease the required amount of data in the target dataset, as proved for a mortality prediction algorithm [33] and for activity recognition [32], [34]. Besides, it can be applied in combination with several algorithms: RNNs [35], Hidden Markov Models [36], statistical inference [33], support vector machine [34]. In smart homes, a cross-domain activity recognition algorithm combined with transfer learning and a similarity function between different activities was proposed [34]. In that work, three different datasets were used, where one was collected over 28 days from a real home of a 26-year-old man. A peak accuracy of 65% was achieved with seven activities. Another work transferred the knowledge of activities from multiple physical source spaces to a different target physical space [32]. The authors propose an algorithm that maps automatically activities from source to target environment and classifies the activity based on a weighted majority vote method. The data contained 5 to 11 activities, and were collected from six testbeds where volunteers lived for 2-3 months. A peak accuracy of about 80% was reported. Hidden Markov Models and transfer learning have also been combined and used across three apartments with five recorded activities and achieved a F1-score of 0.65 in the best case [37]. Transfer learning has its limitations. It has been shown that it can either improve or degrade the prediction accuracy of models depending on the dataset used for transfer, which is known as negative learning [29]. In these cases, it is important to detect which is the best source dataset to a problem, for example using Dynamic Time Warping to measure inter-dataset similarities [38].

Most datasets in the cited works were collected through scripted activities primarily in lab environments, whereas our dataset has been collected in real homes. It contains events from 13-17 binary sensors, i.e. twice as many as used in [20], [26], and less than one third of the number of sensors used in the Mavlab testbed [18]. The number of sensors is comparable to the work in [22] (16 rooms), however in that study the events were inserted by each user in their PDA rather than being generated using sensors, which may lead to a dataset with less artifacts. To our knowledge, no previous work has carried out a comparison of the performance of state-of-the-art sequence prediction algorithms, moreover applied to real data, nor have LSTM networks been previously used for the prediction of sequential sensor events, including the use of transfer learning. In addition, we predict both the next sensor and the mean elapsed time of occurrence within the same model. From the works cited above, [27] is the closest to ours in the sense that it predicts both the next event and its time information in the same model. That work predicts the next location, time of day (slots of 3 hours through the day), and day of the week using a Bayesian network with reported accuracy of 46-60%, 66-87% and 89-97%. Subsequently, the activity is predicted with an accuracy of 61-64% based on a combination of these features. The authors use data from testbeds collected over 6 and 4 months, and take into account 10 locations and 11 activities. Our work predicts the next sensor event and the time of occurrence for a set with about

15 sensors with better overall accuracy. In addition, activities are predicted with considerably higher accuracy.

## III. FIELD TRIAL

Our field trial includes eight residents over 70 years old in a community care facility. The apartments have similar layouts – comprising a bedroom, a living room, an open kitchen area, a bathroom, and an entrance hall (Fig. 2). The purpose of the trial and the sensor system to be deployed have been decided upon in close collaboration with the residents [9]. A minimal number of binary sensors was installed in the apartments to minimize surveillance of the residents and comply with the technical and economical constraints imposed by the project. The set of sensors has been chosen so that it can potentially identify daily activities and possibly enable the realization of useful functions for older adults with MCI/D as these were indicated at dialogue cafés with the users [9]. Hence, our set of sensors contains motion, magnetic, and power sensors. These generate events that are able to indicate occupancy patterns (movement around the apartment), daily activities – kitchen related activities, dressing, being in bed –, and leisure activities – reading, watching TV, listening to radio. Motion sensors (Pyroelectric/Passive Infrared – PIR) detect motion through the change of the infrared radiation in its field of view. It generates an event with message "1" every time a motion is detected, otherwise it sends no event. In our dataset, we had to insert the "off" events ("0" message) so that the data are consistent for all sensors. Magnetic sensors indicate whether doors, windows, and drawers are open or closed, by generating events with messages "1" and "0", respectively. Power sensors measure the electricity usage of a certain appliance, and can therefore indicate whether it is turned on or off, and generate events with messages "1" and "0", respectively.

Not all apartments could have the exact same set of sensors due to physical limitations (e.g. fridge door with too big gap to enable the use of a magnetic sensor) and/or different equipment (e.g. some residents have a coffee machine, others have a kettle). However, all the participants had the same initial proposal of set of sensors, as shown in Fig. 2. The eight apartments that provided data to this work have installed all the motion sensors, while the rest of the sensors vary between apartments, as summarized in Table 1.

The sensors are connected wirelessly through Z-Wave and xComfort protocols to a Raspberry Pi 3, which transfers the data for storage in a secure server. The data comprise timestamp (date and time with precision of seconds), sensor ID, and sensor message (binary). Table 2 shows a sample of the data collected.

## IV. PREDICTION METHODS

This section describes the prediction methods applied in this work, probabilistic methods – Active LeZi (ALZ) and Sequence Prediction via Enhanced Episode Discovery (SPEED) – and recurrent neural network (RNN) with long short-term memory (LSTM). The probabilistic methods convert the data acquired from the sensors into a sequence of letters and identify sequence patterns. The patterns and their frequency of occurrence are used to generate a tree, which is then used to calculate the next most probable event to occur. This last step is performed by the Prediction Partial Matching algorithm (PPM) [39], [40]. The same converted data is used

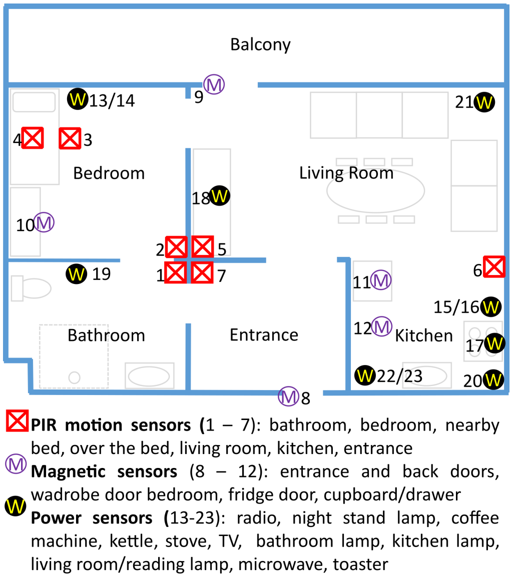



FIGURE 2. Proposed sensors system for field trial apartments.

TABLE 1. Set of Sensors in each Apartment (complementing the standard set of motion sensors)

| Apt. ID | Sensors |
|---|---|
| 1 | P[a]: night stand lamp, coffee machine, living room/reading lamp, TV; M[b]: cupboard/drawer, entrance door |
| 2 | P[a]: night stand lamp, coffee machine, living room/reading lamp, microwave, TV; M[b]: fridge, entrance door |
| 3 | P[a]: kettle, living room/reading lamp, microwave, toaster; M[b]: fridge, cupboard/drawer, entrance door |
| 4 | P[a]: night stand lamp, coffee machine, living room/reading lamp, TV; M[b]: fridge, entrance door |
| 5 | P[a]: kettle, TV; M[b]: fridge, cupboard/drawer, entrance door |
| 6 | P[a]: night stand lamp, coffee machine, kettle, living room/reading lamp, TV, microwave |
| 7 | P[a]: night stand lamp, coffee machine, kettle, living room/reading lamp, TV; M[b]: wardrobe, cupboard/drawer, entrance door |
| 8 | P[a]: night stand lamp, TV; M[b]: wardrobe, entrance door |

[a]Power and [b]magnetic sensors.

TABLE 2. Sample of Binary Sensors Data

| Timestamp | Sensor ID | Sensor message |
|---|---|---|
| 01.09.2017 07:58:05 | 2 | 1 |
| 01.09.2017 08:00:14 | 12 | 1 |
| 01.09.2017 08:01:01 | 4 | 1 |
| 01.09.2017 08:02:56 | 5 | 1 |
| 01.09.2017 08:03:05 | 12 | 0 |

**TABLE 3.** Actions scenario

| Action performed | Activated sensor |
|---|---|
| Wake up | PIR bedroom (on) |
| Go to living room | PIR living room (on) |
| Turn on TV | Power TV (on) |
| Go to kitchen | PIR kitchen (on) |
| Turn on coffee machine | Power coffee machine (on) |
| Go to living room and watch TV while coffee is being made | PIR living room (on) |
| Go to kitchen | PIR kitchen (on) |
| Turn off coffee machine | Power coffee machine (off) |
| Go to living room | PIR living room (on) |

**TABLE 4.** Assignment of letters to sensors

| Sensor | Letter |
|---|---|
| PIR bedroom | a/A |
| PIR living room | b/B |
| Power TV | c/C |
| PIR kitchen | d/D |
| Power coffee machine | e/E |

as input for the LSTM networks that are configured as text generation networks in this case.

Table 3 presents a possible scenario in our smart home with actions performed by the resident and the corresponding sensors triggered. As dictated by ALZ and SPEED, each sensor is assigned with a letter, as shown in Table 4.

### A. ACTIVE LEZI

ALZ is a sequence prediction algorithm based on a text compression algorithm [18]. The input in ALZ consists of a sequence of lower-case letters, where each letter represents event from one sensor. For example, the sequence corresponding to the scenario described in Table 3 would be "abcdebdb". ALZ uses the procedure dictated by the LZ78 text compression algorithm to generate patterns that occur in a sequence and create a tree with these and their frequencies [41].

A given sequence $x_1, x_2, \ldots, x_i$ is parsed into $n_i$ subsequences $w_1, w_2, \ldots, w_{ni}$ such that for all $j > 0$ the prefix of the subsequence $w_j$ is equal to some $w_i$ for $1 < i < j$. For example, if we have the sequence "abcdebdb", the patterns found by LZ78 would be "a", "b", "c", "d", "e", "bd". In addition, ALZ generates more patterns from their suffixes, if possible. For example, "bd" would also generate "d". This accounts for patterns that were not perceived by the LZ78 algorithm and that are possibilities in a smart home environment. This increases the convergence rate of the model [18].

When the sequence is parsed completely and the patterns are derived from it, their frequency of occurrence is counted. An order-k-1 Markov tree is then constructed based on the patterns and their frequencies, where k corresponds to the longest pattern found in a training sequence. Then PPM is used to calculate the next most probable event. The generated tree for the example scenario with sequence "abcdebdb" is shown in Fig. 3.

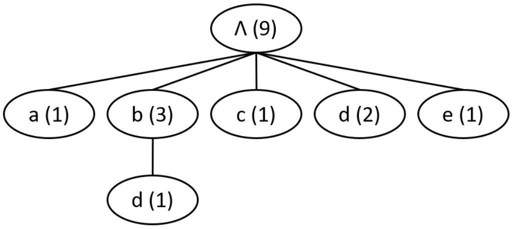



**FIGURE 3.** Tree generated by the ALZ algorithm for the sequence "abcdebdb".

### B. SEQUENCE PREDICTION VIA ENHANCED EPISODE DISCOVERY

SPEED is also a sequence prediction algorithm that is based on the occurrence of frequent patterns in home environments [19]. SPEED builds on the same procedure of ALZ, however, it introduces a different method for finding patterns in the sequence. SPEED defines an *episode* as the sequence between an initial and ending point of an activity. For example, the moment a coffee machine is turned "on" is the initial point of a coffee making episode, which lasts until the coffee machine is turned "off". An "off" event cannot happen unless an "on" event has preceded it. Therefore "off" events always happen after an "on" event of the same activity (or sensor), and vice-versa.

The data received from the sensors in the smart home are represented as a sequence of letters, where upper-case letters represent a sensor's "on" event and lower-case letters represent a sensor's "off" event. The sequence representing the example scenario presented in Table 3 would be "AaBCbDEdBbDedB".

The main idea of the SPEED algorithm is to extract episodes from a sequence of data and derive patterns from them. These patterns are used to generate a decision tree that keeps track of the learned episodes and their frequencies. The height of the tree is the length of the longest episode found in the sequence, defined as the maximum episode length. For every event in a sequence, the algorithm searches for its opposite event in the window and if it exists, an episode was found. In the previous sequence, the first episode found is "Aa", the patterns generated from it would be "A", "a" and "Aa". We keep track of these and count their occurrences to generate an order-k-1 Markov model, where k is the maximum episode length. A tree for the example sequence is presented in Fig. 4. Finally, the PPM algorithm is used for prediction.

### C. PREDICTION PARTIAL MATCHING ALGORITHM

The PPM algorithm calculates the probability distribution of each possible event based on a given sequence by taking into consideration the different order Markov models with different weights [39], [40]. The weights are given by the escape probability, which allows the model to go from a higher-order to a lower one. The advantage of PPM is that

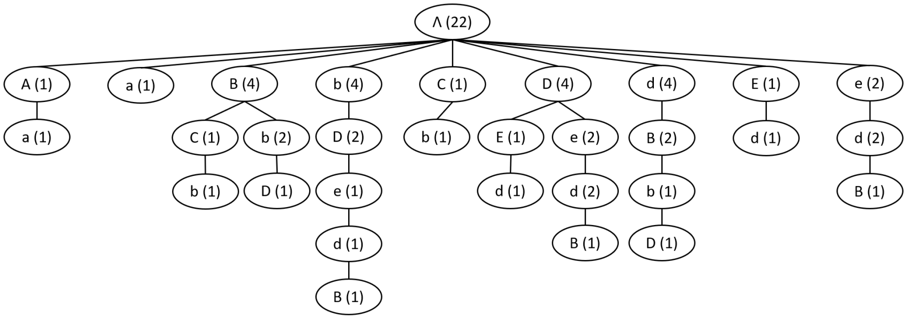



**FIGURE 4.** Tree generated by the SPEED algorithm for the sequence "AaBCbDEdBbDedB".

it assigns a greater weight to the probability calculated in higher-order models if the symbol being predicted is actually found in the tree [18]. The predicted symbol is the one with the highest probability.

ALZ and SPEED use slightly different strategies of PPM. ALZ uses the exclusion strategy, which means the prediction is performed with the suffixes of the given sequence, except the sequence itself. Therefore, in the case of the sequence "bd", the patterns used to calculate the probability of each letter being the next would be "b" and the null context. Suppose we want to calculate the probability of having a "c" after "bd" using ALZ, based on the tree in Fig. 3. The probability would be given by (1): in an order-2 model, the probability of having a "c" after a "b" is 0/3 and we escape to the order-1 with 2/3 probability. In order 1, the probability of having a "c" after a null context is 1/9.

In the case of SPEED, the patterns used for calculating probabilities after a certain sequence would be all the suffixes, including the sequence itself. Suppose we have the sequence "dB". We would use patterns "dB", "d" and the null context. The probability of having a "b" after this sequence based on the tree in Fig. 4, would be given by (2): we start in order 2 model, where the probability of having a "b" after "dB" is 1/2 and escape to the lower order with probability 1/2. In order-1, the probability of having a "b" after "d" is 0/4 and we escape to the lower order with probability 2/4. Finally, in the lowest order, the probability of "b" after a null context is 4/22.

$$p(c, bd) = \frac{0}{3} + \frac{2}{3}\left(\frac{1}{9}\right) = 0.074 \tag{1}$$

$$p(b, dB) = \frac{1}{2} + \frac{1}{2}\left(\frac{0}{4} + \frac{2}{4}\left(\frac{4}{22}\right)\right) = 0.545 \tag{2}$$

### D. LONG SHORT-TERM MEMORY NETWORK

RNN [42] is a neural network that has the property of keeping an internal memory, and has therefore been widely applied to inputs that are sequential in time [43], [44]. The LSTM [45] is a type of RNN designed to be better at storing and accessing information than the standard RNN.

We employ an LSTM network configured as a text generation network. The number of inputs is a certain number of sensor events – equal to the memory length – and the output is the predicted next event in the sequence (Fig. 5). The input and output are one-hot encoded. In the one-hot encoding representation, each symbol is represented by a vector of bits of length equal to the number of symbols in a sequence. All values are zero, except for the one corresponding to that symbol (Fig. 5).

A stateless LSTM network model was implemented in Python 3 using Keras open source library for neural networks. A number of parameters were tuned in order to find the optimal values. The model has one hidden layer with hyperbolic tangent activation and 64 neurons. Our batch size (i.e. number of samples used for training each iteration of the epoch) was 512. We used *Adam* as the optimization function with learning rate of 0.01 and categorical cross-entropy as loss function. The output layer was a softmax activation function. We used the early stopping method and dropout rate of 50% to avoid overfitting, allowing a maximum of 200 epochs for each model's training. In addition, during the training process we use weights for each sensor to balance the number of samples for each sensor. These are computed using the "`compute_class_weight`" function of the Scikit-learn open source library. The weight corresponds to the total number of samples divided by the number of occurrences of the class.

## V. DATA PREPROCESSING

### A. SENSORS MAPPING

As described in Section III, some power and magnetic sensors differ within the eight apartments (Table 1). In the tests where we compare the prediction accuracy and transfer the learning across the apartments, we re-label the sensors that refer to the same activity. The new labels and the sensors assigned to these are shown in Table 5. Lamp power sensors and wardrobe door magnetic sensors' events were removed from the datasets since we did not manage to assign them to

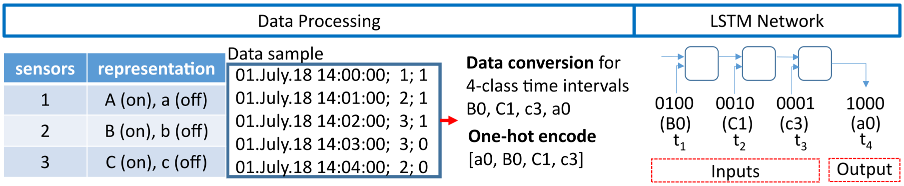



**FIGURE 5.** LSTM network configuration.

**TABLE 5.** Re-labeling of Sensors

| New labels | Sensors |
|---|---|
| Kitchen sensor | P[a]: toaster, microwave; M[b]: fridge, cupboard/drawer |
| Beverage sensor | P[a]: coffee machine, kettle |

[a]Power and [b]magnetic sensors.

an activity that was common for most of the apartments.

### B. DATA CORRECTION

Data acquired from binary sensors often contain faulty events e.g. erroneous activation of motion sensors by sunlight and switch-off delays of motion sensors [46]. Such noise can significantly affect the performance of the models. Hence, we have carried out a data correction preprocessing as follows. Occasionally the motion sensors do not send an activation event when they should. We therefore insert missing events to correct the data. For example, it is not possible to go to the bedroom directly from the kitchen without passing through the living room. When the living room activation event is missing, it is inserted. If there are two possible sensor events (e.g. two possible paths in the apartment), the choice of the inserted sensor event is done such that the final percentage distribution of the two options remains as observed in the original data. The time of the inserted event is the mean between that of the previous and of the next event. This does not compromise the data accuracy because the faulty events usually take place between relatively fast motions around the apartment, which means that the elapsed time between the events is quite short.

### C. DATA CONVERSION

The corrected data are subsequently converted to both ALZ- and SPEED-text sequences, as explained in Section IV. The time inclusion was performed as follows. In all cases the generated sensor events are treated as independent events. In the case of the one-hot encoding for the LSTM, our input vector has as many values as the number of symbols in the sequence. For 15 sensors, we have 30 inputs to represent the "on" and "off" states of each of these.

#### 1) Sensor Event and Period of Day.

In this case, we distinguish between four periods of the day: morning (from 7am to noon), afternoon (from noon to 6pm), evening (from 6pm to 10pm), and night (from 10pm to 7am). This is indicated by a number between 0 and 3 that is added to the letter that represents the event. For instance, an event of the motion sensor in the bedroom going "on" in the morning would generate the symbol "A0". E.g. when the time of day is taken into account, the number of inputs to the LSTM is multiplied by 4 (120 inputs in total) and similarly in the other cases. These are treated as independent events.

#### 2) Sensor Event with Time Elapsed to the Next Event

When predicting the next sensor event only, we use together with the sensor's letter a number that indicates the time elapsed to the next event. We define a set of 4-class time intervals: [$<$ 1min, 1-15min, 15min-1h, $>$ 1h]. Hence, we assign numbers 0-3 to the event. For example, if the motion in the bedroom (assigned letter a/A) were activated in the morning and 10 minutes later the person went to the bathroom, the generated symbol would be "A1".

#### 3) Sensor and K-means Time-Cluster with Hour of the Day and Elapsed Time to the Next Event

We apply an unsupervised learning method to cluster the sensor samples, where the K-means algorithm clusters each sensor event according to the hour of the day it has occurred and the time elapsed to the following sensor event. In the K-means algorithm, the samples of each sensor are classified into K clusters such that the sum of square distances (SSD) within the clusters is minimized [47]. Each cluster contains a centroid, given by the mean value of each feature of the algorithm. We perform K-means for a number of clusters (K) between 1 and 8 and choose the best K manually according to the elbow method [48]. This method consists of plotting an SSD vs. K graph and choosing the K that resembles an "elbow" (the point of inflection on the curve), which is the best fit for that problem. Fig. 6 shows an example of clustering the samples of the motion sensor in the kitchen. This sensor results in four clusters (represented by the different colors) – chosen by the elbow method based on Fig. 7. Suppose this sensor is represented by letter B, has had an "on" event at noon (blue cluster), and the next sensor event took place 3 minutes later. This would generate "B2" (where 2 represents the blue cluster).

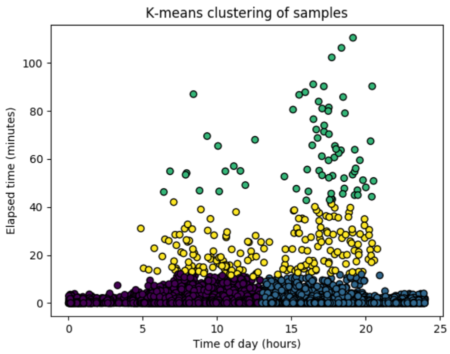



**FIGURE 6.** K-means clustering of samples of motion sensor events in the kitchen.

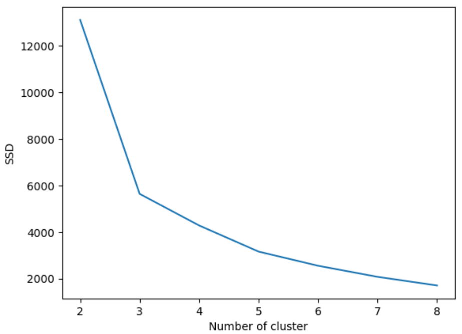



**FIGURE 7.** SSD vs. number of clusters for motion sensor events in the kitchen.

## VI. PREDICTION OF THE NEXT SENSOR AND TIME OF OCCURRENCE

Table 6 shows the number of sensor events in the dataset of each apartment and the number of days it has been collected. This section is organized as follows. We first explain the training and testing procedure for all methods. Subsequently, we perform tests for (i) predicting the next sensor event based on past sensor events, (ii) predicting the next sensor event based on past sensor events and time of occurrence information, and (iii) predicting both sensor event and time of occurrence information based on input including these. In (i) the performance of the four algorithms is tested against a number of factors: the memory length, the amount of data required for good accuracy, and the number of sensors in the dataset. The best-performing algorithm is then further developed for tests (ii) and (iii), where we compare the methods and analyze the accuracy variability across apartments.

**TABLE 6.** Number of Events per Apartment

| Apt. ID | Number of Events | Number of Days |
|---|---|---|
| 1 | 219921 | 358 |
| 2 | 137396 | 311 |
| 3 | 37108 | 163 |
| 4 | 147618 | 385 |
| 5 | 189468 | 260 |
| 6 | 19766 | 96 |
| 7 | 28129 | 75 |
| 8 | 21949 | 75 |

### A. TRAINING AND TESTING CONFIGURATION

In the SPEED algorithm, the next event is predicted based on the last sequence of events with length equal to the maximum episode length [19]. In the work in [19], the authors use the same dataset for both training and testing, which may lead to overfitting and, in addition, may not lead to a generalized model that can be used on other datasets.

We have modified the testing procedure for both ALZ and SPEED by calculating the optimal number of last events to base the prediction on, i.e. the number of events that leads to the maximum overall prediction accuracy, which we refer to as the optimal memory length. Memory lengths up to the maximum pattern found have been considered. In a previous paper [12], we applied the SPEED method on our data that were obtained from one of the apartments reported over a period of two weeks. When using the same procedure as in [19], we achieved an accuracy of 82% – compared to 88% on the Mavlab dataset. When splitting the data into training (60%), validation (20%), and testing (20%), and optimizing the memory length as described above, we achieved an accuracy of 75% on our data obtained from a real home over two weeks. Similarly for ALZ we obtained 73% (compared to 47% in [18]) when using the same dataset for training and testing, and 53% when using separate datasets for training, validation and testing, and optimizing the memory length as described above. Hence, we use this modified method for SPEED and ALZ in the following sections.

In the case of SPEED and ALZ, the training set is used to build the tree, the validation set is used to find the optimal memory length, and the testing set is used to compute the model's accuracy. In the LSTM networks, the training set is used to train the network, the validation set is used for tuning the parameters and the testing set to calculate the accuracy. All models were trained based on a certain number of events, validated on 3000 random events, and tested on 3000 random events. This process is repeated three times and the accuracy values in the graph correspond to the mean accuracy. The fact that the testing set is always random produces some instability in the accuracy when the model is trained with little data, which is evidenced by the instability shown in the lower range in some of the graphs.

### B. PREDICTION OF THE NEXT SENSOR EVENT BASED ON PAST EVENTS

### 1) Choice of Memory Length

We examine the accuracy achieved on the validation set for values of memory length ranging from 1 to 30 events. This is performed first for the dataset of apartment 1 that contains events from fifteen sensors (including magnetic, power and motion sensors) – Fig. 8 – and then for the dataset containing only the seven motion sensors – Fig. 9.

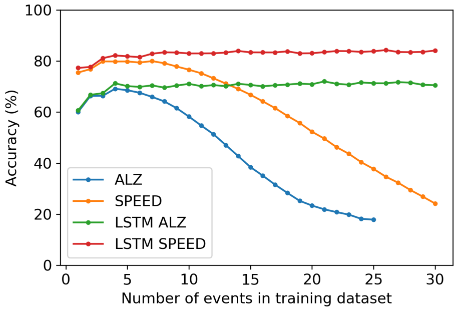



**FIGURE 8.** Accuracy vs. memory length for all algorithms on a dataset with all fifteen sensors.

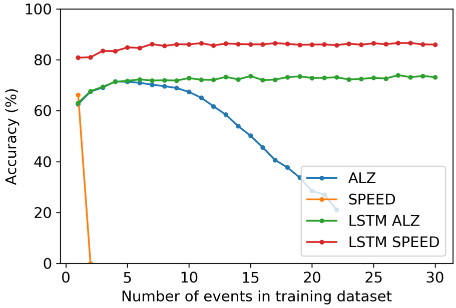



**FIGURE 9.** Accuracy vs. memory length for all algorithms on a dataset with seven motion sensors.

When using the dataset with fifteen sensors (Fig. 9), ALZ achieved a best accuracy of 69.15% while SPEED reached 79.87%. The optimal memory length was four events for ALZ and three for SPEED. The LSTM networks achieved accuracies of 72.02% and 84.12% when using ALZ- and SPEED-text, respectively. In both cases the optimal memory length is equal or larger than eight. The larger optimal memory length for LSTM indicates that these are very efficient at detecting patterns and correlations over a longer sequence, in opposition to probabilistic methods.

It is also interesting to notice how the accuracy is affected by memory lengths larger than the optimal. The accuracy of the probabilistic methods drops substantially as the memory length gets larger. In contrast, the LSTM networks roughly stabilize at the peak accuracy for larger memory length values than the optimal. A reason for this is that probabilistic methods are based on certain patterns happening quite frequently. Since our dataset has few sensors, short patterns are more likely to happen more often, and therefore they provide better predictions. The LSTM, on the other hand, has the ability to find patterns in long sequences and can therefore predict the next event based on many past events and longer term patterns and dependencies. Increasing the memory length further does not improve the accuracy, however, which can imply that the model has reached its best performance for this configuration.

Subsequently, we compare the accuracy results of the dataset with fifteen sensors (Fig. 8) to the accuracy results for the dataset that contains only the seven motion sensors (Fig. 9). The accuracy curves for the LSTM network models show a similar dependency to memory length. The optimal memory length is eight or larger. The LSTM with SPEED-text achieves 86.64% while with ALZ-text achieves 74.00%. The ALZ method also shows similar behaviour, and the same optimal memory length of four, with a peak accuracy of 71.45%. SPEED presents a very peculiar behaviour. The maximum memory length is two. This is a consequence of the fact that SPEED builds the tree based on episodes, and the longest episode in this case is two events. For example, if the resident would go from the bedroom to the living room and then to the kitchen, the resulting sequence would be "AaBbCc". There are no intertwined events, since when one motion sensor activates, another deactivates. Hence, the "off" events are easily predicted. When it comes to "on" events, the sensor that is most frequently activated will always be the one predicted to activate next, leading to lower accuracy for the "on" events.

### 2) Accuracy per Training Set Size

In the following, we investigate the behavior of the accuracy with respect to the size of the training dataset. The accuracy results are computed using the optimal memory length found in the previous analysis. Fig. 10 and 11 show the results when the algorithms are applied to the dataset with all fifteen sensors and with seven sensors, respectively. Since there is no significant improvement in the accuracy for larger datasets, we show the plots for training dataset sizes up to 30000 events for better clarity on the low range of the graph.

We first examine the accuracy in the dataset with all sensors (Fig. 10). A peak accuracy of 83.26% was achieved by LSTM with SPEED-text, while the SPEED algorithm achieved a peak accuracy of 80.65%. The accuracy achieved by the LSTM with ALZ-text was considerably lower at 70.43%. In this case, stability is achieved much later than with the other methods. Finally, the ALZ method reached a peak accuracy of 68.00%. Note that the probabilistic methods attain a good accuracy (close to their peak accuracy) with only 1000 events in the training set. By comparison, the LSTM with ALZ- and SPEED-text require 7500 and 4000, respectively.

Next we examine the accuracy results for the dataset using

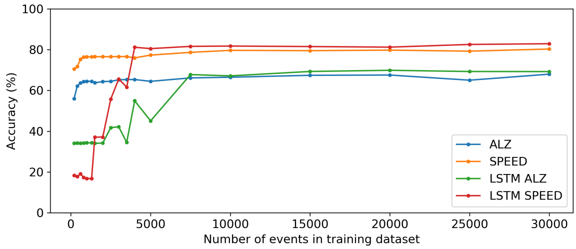



**FIGURE 10.** Accuracy vs. size of training set for all algorithms on the dataset of apartment 1 with all sensors (15).

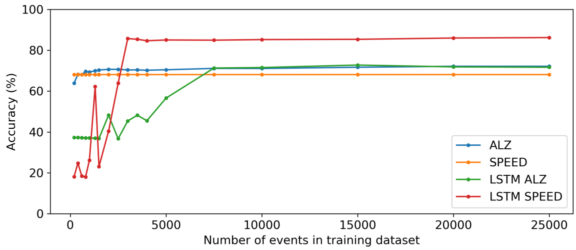



**FIGURE 11.** Accuracy vs. size of training set for all algorithms on the dataset of apartment 1 with seven motion sensors.

only the seven motion sensors (Fig. 11). As seen in the previous analysis, the top accuracy is higher since there are fewer sensors in this set. Moreover, motion sensor events happen sequentially, without intertwined events. Hence "off" events can be predicted more easily. The LSTM with SPEED-text achieved an accuracy of 87.21%, by far the best among the methods. The stability was achieved with about 4000 events. Stability is reached with a similar amount of data compared with the case in Fig. 10. The LSTM with ALZ-text and the ALZ achieved very similar accuracies of 73.24%, and 72.32% respectively. The SPEED method, however, achieved a poor accuracy in this case. This is due to the short memory length and lack of intertwined events, as discussed when presenting Fig. 9. Also here, it is confirmed that probabilistic methods require a rather small amount of data to achieve a considerable accuracy, close to the peak accuracy that can be reached by these methods.

### 3) Prediction Variability across Apartments

In the following, we apply the two prediction methods with higher accuracy – LSTM with SPEED-text and SPEED – on the dataset of each apartment of our field trial. In this case, we perform the mapping (Section V-A) so that the comparison is fair. Table 7 presents the obtained results. SPEED achieved accuracies in the range 74-82% and LSTM with SPEED-text in the range 75-85%. In all cases, the LSTM had an accuracy 1.5-5% higher than SPEED, with one exception (apartment 4), where the accuracies are about the same. On the other hand, in most of the apartments SPEED required less events for a good accuracy and convergence of the model. We noticed that apartments 6, 7, and 8 have have not achieved stability completely yet as the curves keep rising, indicating that higher accuracy can be achieved. They are indeed the apartments with less collected data (Table 6).

It is interesting to notice that SPEED presents less variability across the apartments. This may be due to the fact that SPEED builds a tree where the predictions will be based on the patterns that happen more often, and these are in fact similar to all the apartments since they have similar layouts. The LSTM network, however, is better able to adapt to the resident in this case, taking into account also patterns that do not happen often.

**TABLE 7.** Prediction Accuracy of the Next Sensor Event

| Apt. ID | Top Mean Accuracy (Number of Events for Convergence) | |
|---|---|---|
| | SPEED | LSTM with SPEED-text |
| 1 | 82.24% (2000) | 83.66% (5000) |
| 2 | 77.67% (2000) | 79.13% (5000) |
| 3 | 82.21% (3000) | 84.54% (3000) |
| 4 | 75.01% (2000) | 75.42% (4000) |
| 5 | 76.64% (3000) | 79.17% (4000) |
| 6 | 74.60% (5000) | 76.13% (4000) |
| 7 | 81.78% (7500) | 82.20% (3500) |
| 8 | 79.95% (7500) | 84.50% (3000) |

#### 4) Summary and Discussion

We have compared the performance of two probabilistic methods – ALZ and SPEED – with LSTM networks using ALZ-text and SPEED-text in apartment 1. The best accuracy was achieved by the LSTM network with SPEED-text, 83% with all the fifteen sensors and 87% with seven motion sensors.

The probabilistic methods achieved a high prediction accuracy (close to their peak accuracy) with a relatively small amount of training dataset (about 1000 events). LSTM networks required a larger training dataset (about 4000 event with SPEED-text and 7500 events with ALZ-text) to reach an accuracy close to the peak. Also, probabilistic methods are found to base the prediction on a relatively small number of previous events – an optimal memory length of four for ALZ and three for SPEED was established. On the other hand, LSTM networks base the prediction on a sequence of eight previous events or more. This indicates that such networks are better at finding longer-term dependencies and patterns in a sequence of events. In addition, in the LSTM the attained accuracy is quite stable for memory lengths that are larger than the optimal. On the other hand, probabilistic methods have an optimum memory length, hence the accuracy decreases both for shorter and for longer memory lengths than the optimal.

For the dataset containing events from the fifteen sensors, our best result was achieved by the LSTM network with SPEED-text (83%). SPEED achieved only 2% lower accuracy, however, after considerably longer training time [10]. Hence, in applications where it is an advantage to model with a small amount of data where in addition execution time is not too critical, SPEED may be a good choice, since it can achieve an accuracy close to its peak with little data. In general, our results have shown that it is possible to achieve good accuracy with much less data than thought previously. SPEED and LSTM with SPEED-text achieve better results than ALZ and LSTM with ALZ-text. This is not surprising since the conversion of data to SPEED-text sequences contains more information (both "on" and "off" events). This can also be confirmed by the trees formed by ALZ and SPEED (Fig. 3 and 4).

For a dataset with no intertwined events though – the case of our dataset with only the seven motion sensors – the best choice is the LSTM with SPEED-text. SPEED does not work well in this case, since the tree has a height of two so that only "off" events can be predicted reliably.

Another interesting finding is that when applying these algorithms in different apartments, LSTM with SPEED-text has shown a larger range of accuracies. This indicates that the LSTM can adapt better to the different patterns in the home of each resident than SPEED does. This fact, in addition to the higher accuracy and the shorter execution time, have shown that the LSTM network with SPEED-text is the best model for our smart homes setup. We therefore further develop only this method in the following analysis.

### C. PREDICTION OF THE NEXT SENSOR EVENT BASED ON PAST EVENTS AND TIME INFORMATION

It is important to observe in the results of the previous section that having more than 10000 events in the training set did not improve significantly the results for any of the applied methods. Hence, a change in the algorithms and/or in the way the data are input, or additional information, is required to improve the prediction accuracy. In this section, we include the time of occurrence information in the input of the LSTM network with SPEED-text to investigate whether this leads to an improvement of the prediction accuracy.

#### 1) Comparison of Methods

We predict the next sensor event based on the three proposed input sequences with time information (Section V-C). Fig. 12 shows the performance of the prediction according to the amount of data in the training set in apartment 1. We include the accuracy when using only the previous sensors for comparison purposes. We achieved an accuracy of 83.26% when predicting the next sensor event using previous sensor events as input (i.e. no time information). When we include the period of the day, the class time intervals, and the K-means time-cluster in the input, the models achieve 84.07%, 85.05%, and 84.01%, respectively. The small improvement of 0.8-2% was initially somewhat surprising, as we had expected that the time information would increase accuracy significantly. However, on second thoughts, the apartments are quite small – limiting the number of possible patterns – and there is a limited number of sensors, and hence a lot of information (including for example time information about movements and actions in the home) is not "visible" for the model. The standard deviation of the LSTM models is about 0.02-0.06%, hence the model is quite stable. A significant improvement, however, is in the convergence of the model that occurs with training set sizes of 2500 events, almost half of the events needed for when no time information is included.

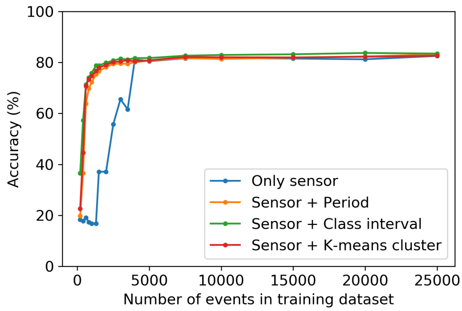



FIGURE 12. Accuracy of prediction of next sensor event vs. the number of events in the dataset.

**TABLE 8.** Prediction Accuracy of the Next Sensor Event based on Past Sensor Events and 4-class Time Intervals with LSTM with SPEED-text

| Apt. ID | Top Mean Accuracy (Number of Events for Stability) |
|---|---|
| 1 | 86.58% (4000) |
| 2 | 83.76% (5000) |
| 3 | 86.38% (5000) |
| 4 | 79.29% (10000) |
| 5 | 80.91% (10000) |
| 6 | 76.63% (5000) |
| 7 | 82.91% (5000) |
| 8 | 86.39% (5000) |

**TABLE 9.** Number of Events per Day and Time Spent Outside the Apartment for each Resident

| Apt. ID | Avg Number of Events per Day | Avg Time Out of Apt (h) |
|---|---|---|
| 1 | 614 | 1.5 |
| 2 | 442 | 1.0 |
| 3 | 227 | 0.6 |
| 4 | 383 | 3.6 |
| 5 | 729 | 1.1 |
| 6 | 206 | 1.75 |
| 7 | 375 | 1.85 |
| 8 | 293 | 7.3 |

2) Prediction Variability across Apartments

The input using the 4-class time intervals has shown marginally better results than the other two methods, and we therefore apply this method on the other apartments. The results are shown in Table 8. Including the time has led to improved accuracy in all the apartments, in a range of 0.5-4%.

The 10% variability between apartments for the prediction of the next sensor could be due to the amount of data available in each apartment. In Table 9, we present the average number of events per day and the average time spent out of the apartment per day. This indicates that the degree of activity varies significantly and/or that some of the residents are more active when in the apartment (e.g. apartments 1 and 5) than others. Nonetheless, the average number of events per day does not seem to have a direct influence on the achieved prediction accuracy. For instance, relatively high prediction accuracy (86%) has been achieved for apartment 3 that only has 227 events per day, whereas much lower prediction accuracy (81%) is achieved in apartment 5 that has the highest number of events per day (729). Also, comparable accuracy is attained in apartment 1 (87%) as to apartment 3, although there are on average more than twice as many events per day in the former than in the latter. Hence, there is no correlation between attained accuracy and the average number of events per day here. Another hypothesis for the prediction accuracy variability is the noise originated by different sources in the data for the apartments. For example, the resident in apartment 6 has often family members visiting. This noise cannot be measured in our setup at this moment.

Furthermore, the coefficient of variation (standard deviation divided by the mean) in this case is about 0.04, which is lower than 1 and therefore, a low variance. The different predictions in this case may simply indicate some people are more predictable in their patterns around the apartment than others.

### D. PREDICTION OF THE NEXT SENSOR EVENT AND ITS MEAN TIME OF OCCURRENCE

1) Comparison of Methods

In the following we examine the accuracy of predicting both the next sensor event and the time of occurrence information. In this case, only the input sequences of class time intervals and K-means time-cluster are considered. Lower accuracy is attained (Fig. 13) than when predicting only the next sensor event, as expected, since now more information is being predicted within the same model. The best accuracy is achieved by the K-means time-cluster (82.00%), 6% better than the class time-intervals (76.63%). For both methods, convergence is achieved with about 2000 events in the training set (Fig. 13). Our hypothesis is that the K-means algorithm clusters the samples in a more balanced way than the 4-class intervals, and this leads to a better prediction accuracy.

**TABLE 10.** Prediction Accuracy of the Next Sensor Event and Time-cluster based on Past Sensor Events and Time-cluster with LSTM with SPEED-text

| Apt. ID | Top Mean Accuracy (Number of Events for Convergence) |
|---|---|
| 1 | 82.97% (5000) |
| 2 | 79.36% (10000) |
| 3 | 80.59% (10000) |
| 4 | 74.87% (10000) |
| 5 | 77.67% (10000) |
| 6 | 72.73% (10000) |
| 7 | 78.09% (5000) |
| 8 | 79.21% (5000) |

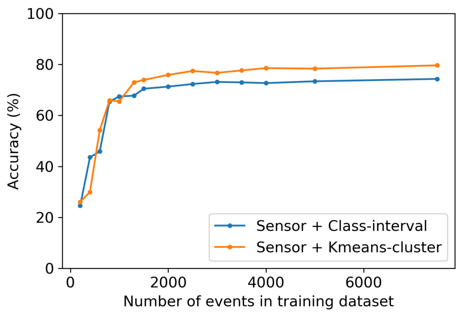



**FIGURE 13.** Accuracy of prediction of next sensor event and time information vs. the number of events in the training dataset.

2) Prediction Variability across Apartments

The K-means time-cluster method attained the highest accuracy when predicting both the next sensor event and time information, and therefore we apply this on the dataset from all apartments. The obtained results are shown in Table 10. The attained accuracy is 3-6% lower than when predicting the next sensor only (Table 8), as expected.

## VII. TRANSFER LEARNING ACROSS APARTMENTS

In this section, we investigate whether the transfer learning technique is feasible and beneficial across the apartments in our field trial. We use transfer learning as follows. We first train an LSTM network with data from seven source apartments and fine-tune and test with one target apartment. In this case, the data from the target apartment – that have not been used in the training – are split to be used in the fine-tuning of the network (keeping the weights of the best-fit model), and in the testing (3000 events). We compute the accuracy of predicting the next sensor event only based on input about previous sensor events, as well as time information using 4-class time of occurrence intervals (Fig. 14). In Fig. 15, we present the obtained accuracy when predicting both the next sensor event and the time-cluster based on inputs about both these.

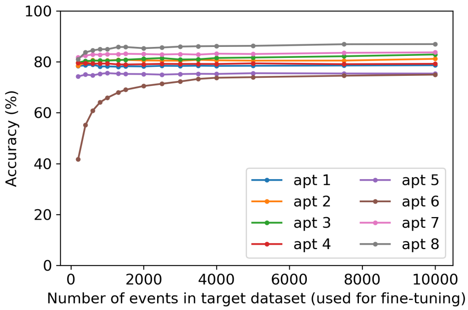



**FIGURE 14.** Accuracy of prediction of the next sensor vs. number of events used for fine-tuning, using as input both sensor event and 4-class time interval. Transfer learning – training the model with data from seven apartments, fine-tuning with and testing on the target apartment.

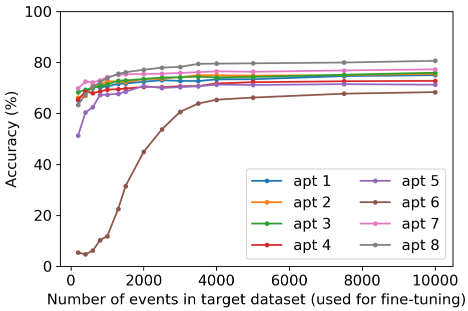



**FIGURE 15.** Accuracy of prediction of the next sensor and time-cluster vs. number of events used for fine-tuning, using as input both sensor event and time-cluster. Transfer learning – training the model with data from seven apartments, fine-tuning with and testing on the target apartment.

Accuracies from about 80% can be achieved straight away with very little data from the target apartment. There is one exception, apartment 6, which takes much longer time to achieve good accuracy. However, also this apartment required less data (about 4000 events) compared to the case without transfer learning (about 5000 events). For larger training datasets, the prediction accuracy is approximately the same as when each apartment is modelled with its own data. In fact, in most cases is it marginally higher when each apartment is modelled individually, except for apartments 4 and 8.

**TABLE 11.** Prediction Accuracy with Transfer Learning without and with Fine-tuning (FT)

| Apt. ID | Top Mean Prediction Accuracy | | | |
|---|---|---|---|---|
| | Next Sensor | | Next Sensor and Time | |
| | No FT | FT | No FT | FT |
| 1 | 75.17% | 82.52% | 60.40% | 78.96% |
| 2 | 77.46% | 84.38% | 58.17% | 79.37% |
| 3 | 68.67% | 85.79% | 52.27% | 79.14% |
| 4 | 79.03% | 80.44% | 60.20% | 75.29% |
| 5 | 71.87% | 79.16% | 49.87% | 75.39% |
| 6 | 2.93% | 75.44% | 0.00% | 69.16% |
| 7 | 80.57% | 84.11% | 59.00% | 78.11% |
| 8 | 78.90% | 87.72% | 53.43% | 81.68% |

Note also that when predicting the time-cluster in addition to the next sensor, a larger amount of data is required to transfer the learning as effectively as when not predicting the time. This is due to the fact that when predicting only the next sensor, the layout of the apartment is what is mostly taken into account given that the apartments are very small and all have the same layout (section III). When predicting the time-cluster, we account in addition for the individual habits of each resident, and hence, additional data are required to fine-tune the network.

Table 11 presents the top accuracy obtained for each apartment. We have also computed the accuracy without fine-tuning prior to testing when applying transfer learning, i.e. when the network has been trained with data from seven apartments and subsequently tested directly on the target dataset. For the prediction of the next sensor only, the accuracy is 4-8% lower than when using fine-tuning. When predicting both the next sensor and the time information, the accuracy is 15-30% lower without as compared to with fine-tuning. This is in accordance with what has been mentioned earlier in this section, i.e. that predicting the time takes into account individual patterns, and therefore needs additional data to fine-tune the network to each resident. In either prediction case, the fine-tuning of the model is indeed required to achieve good prediction accuracy when using transfer learning across apartments.

Subsequently, we investigate how much data are required to transfer learning from the base model (with data from several apartments) such that the target apartment will obtain a good accuracy with very little data. We chose apartment 8 to be the target apartment in this case, since it has shown to have higher accuracy with transfer learning rather than when being modeled with its own data. We use 100 events from the target apartment to fine-tune the network and test on 3000 random events. When predicting only the next sensor, about 40000 events from seven different apartments are required so that in apartment 8 80% prediction accuracy can be achieved with only 100 events, as shown in Fig. 14. For predicting the next sensor and the time-cluster much more data are needed, about 500000 from the seven different apartments. In this case, the accuracy with only 100 events in the target apartment is about 60%. As discussed earlier and presented in Fig. 15, when predicting the time, more data are required from the target apartment to achieve the peak accuracy.

## VIII. ACTIVITY PREDICTION

### A. METHOD

Ultimately, the binary sensor events indicate activities of daily living. In this section, we associate the binary sensor events with activities and predict these. We are only able to register high-level activities as the number of sensors in our set-up is quite limited. Our dataset comprises the following classes: watching TV, being in bed, being out, bedroom activities, living room activities, kitchen activities, bathroom activities, transitions in bedroom/bathroom/entrance/living room – 11 in total.

We implemented two rule-based algorithms for deriving activities from binary sensors that we refer to as *sequential* activities and *concurrent* activities. We decided for a set of rules as described in Table 12. In the case of *sequential* activities, we assume that no more than one activity takes place at the same time, so that as soon as one activity ends, another starts. The time information is the elapsed time to the next activity, which in this case is the duration of the activity. In the case of *concurrent* activities, each activity has a start and an end – indicated by a “1” and a “0”, respectively – , allowing several activities to be happening in parallel. For example, the resident can be in the kitchen preparing coffee and still be watching TV. This implies that, in many cases, the duration of the activities will be longer compared to the sequential activities. The time information is inserted such that activity start contains the duration of the activity (time elapsed until the end of the activity) and activity end contains the elapsed time to the start of the next activity event. Fig. 16 shows an example of the two sequences without including the time, for simplicity.

| activities | representation |
|---|---|
| Watching TV | A (on), a (off) |
| Preparing coffee | B (on), b (off) |
| In bed | C (on), c (off) |

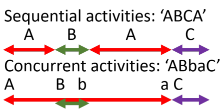



**FIGURE 16.** Types of activities sequences implemented – sequential and concurrent. The example corresponds to a scenario where the resident watches TV, goes to the kitchen to prepare a coffee while watching TV, and then goes to bed.

As we do for the sensor events, each activity is assigned a letter, and the time information with K-means time-cluster is selected due to its best performance. The transition classes are only used in the input of the LSTM network, thus the output classes are in fact only 7. The LSTM network has the same configuration parameters as the one used for the prediction of sensor events. In addition, we use the Synthetic Minority Oversampling Technique (SMOTE) since our data is imbalanced. SMOTE is an over-sampling technique that creates synthetic samples for the minority classes [49]. The library Imbalanced-Learn was used for this implementation [50].

**TABLE 12.** Rules for Deriving Activities from Sensor Events

| Activities | Rules |
|---|---|
| Kitchen activities | Whenever power and magnetic sensors located in the kitchen are activated or motion sensor in the kitchen is active for more than 1 minute. |
| Living room activities | Whenever power and magnetic sensors located in the living room (except TV) are activated or motion sensor in the living room is active for more than 5 minutes. |
| Watching TV | Whenever the resident is in the living room for more than 5 minutes and the power in the TV is on. |
| Bedroom activities | Whenever power and magnetic sensors located in the bedroom (except sensors around the bed) are activated or motion sensor in the bedroom is active for more than 5 minutes. |
| Being in bed | Whenever motion sensors around the bed are consecutively activated for more than 5 minutes. |
| Bathroom activities | Whenever the motion sensor located in the bathroom is active for more than 1 minute. |
| Being out | Whenever the entrance door “off” and “on” events happen consecutively and for more than 5 minutes; or when the entrance door is the last active motion sensor for more than 10 minutes (in one of the apartments where the entrance door was not installed). |
| Transitions | Being in the entrance is always considered as a transition as there are no relevant activities in that area. Other rooms have a subjective transition time chosen based on the distance between rooms and conditions of the residents (e.g. walking speed, use of rollator, etc.). |

**TABLE 13.** Number of Activities per Apartment

| Apt. ID | Number of Activities | |
|---|---|---|
| | Sequential | Concurrent |
| 1 | 70931 | 66136 |
| 2 | 26460 | 52920 |
| 3 | 19344 | 38700 |
| 4 | 53984 | 61222 |
| 5 | 49665 | 99330 |
| 6 | 10577 | 11382 |
| 7 | 10031 | 20090 |
| 8 | 5446 | 8162 |

### B. RESULTS AND DISCUSSION

Table 13 shows the number of activity events in each dataset for each apartment. The LSTM network was trained based on a certain number of events and tested on either 3000 random events or 10% of the total number of events (for the apartments with very few activity events, e.g. 6-8). This process is repeated three times, and the accuracy values in the graphs correspond to the mean of the best test accuracy of each training.

Firstly, we predict the next activity based on previous activities (Fig. 17), and subsequently when including the K-means time-cluster (Fig. 18), for both types of activity sequences. Table 14 presents the prediction accuracy for these. For the sequential activities dataset, the prediction accuracy varies between 58-90% without using the time information in the input, and between 61-90% when including the time information. Including the time in the input resulted in 0 (apartment 7) to 3% improvement. In the case of the concurrent activities dataset, the prediction accuracy varies between 75-92% without using the time information in the input, and between 75-95% when including the time information. Thus the accuracy has improved from 0.3-3.1% across the apartments when the time information is included in the input. Fig. 19 and Table 15 present the accuracy results when predicting both the next activity and its duration/time elapsed to the next activity. The obtained accuracy varies between 64-85% for the concurrent activities, i.e. it is 4.5-11.8% lower compared to above. Similarly, for the sequential activities, an accuracy of 50-80% is achieved, i.e. 9.4-16.2% lower than above. This is expected since now the model has many more classes to predict from and it is in addition predicting more information.

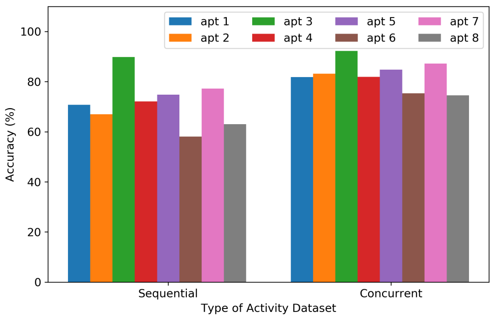



**FIGURE 17.** Prediction accuracy of next activity based on previous activities per apartment and type of activity dataset.

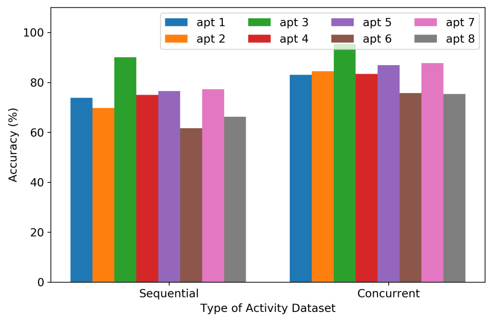



**FIGURE 18.** Prediction accuracy of the next activity based on previous activities and K-means time-cluster per apartment and type of activity dataset.

**TABLE 14.** Prediction of the Next Activity

| Apt. ID | Without time | | With time | |
|---|---|---|---|---|
| | Sequential | Concurrent | Sequential | Concurrent |
| 1 | 70.80% | 81.82% | 73.78% | 83.04% |
| 2 | 66.94% | 83.18% | 69.72% | 84.51% |
| 3 | 89.79% | 92.28% | 90.02% | 95.38% |
| 4 | 72.13% | 81.89% | 75.02% | 83.40% |
| 5 | 74.77% | 84.81% | 76.51% | 86.94% |
| 6 | 58.07% | 75.34% | 61.67% | 75.67% |
| 7 | 77.23% | 87.18% | 77.23% | 87.72% |
| 8 | 63.00% | 74.56% | 66.20% | 75.34% |

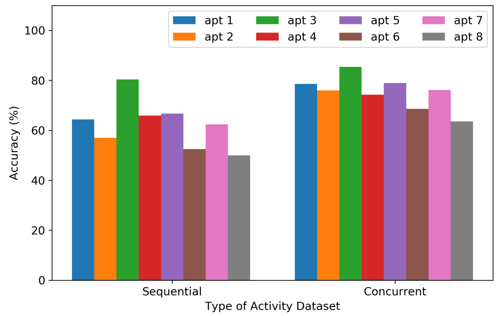



**FIGURE 19.** Prediction accuracy of both next activity and K-means time-cluster per apartment and type of activity dataset.

We can notice that apartment 3 has achieved the best accuracy in all tests. One observation is that this resident did not have the power sensor in the TV, so that this model has one less class to predict (*watching tv*). In addition, it is a class that usually presents much confusion with others, especially with *living room activity*. Apartments 6 and 8 have shown similar and poor accuracies in the tests, however, they do not have enough data for conclusive results (see number of activity events in Table 13). The other apartments – 1, 2, 4, 5, and 7 – present comparable results.

The accuracy results for the concurrent activities dataset were better in all cases – 5.4-14% improvement when predicting only the next activity based on previous activities and time; and 5.1-16.1% when predicting the next activity and K-means time-cluster. However, since there is only one resident and a relatively small number of sensors in each apartment, that moreover do not relate to other sensors, there

**TABLE 15.** Prediction of the Next Activity and K-means Time-Cluster

| Apt. ID | Sequential | Concurrent |
|---|---|---|
| 1 | 64.34% | 78.54% |
| 2 | 56.98% | 75.93% |
| 3 | 80.33% | 85.44% |
| 4 | 65.91% | 74.28% |
| 5 | 66.75% | 78.98% |
| 6 | 52.50% | 68.59% |
| 7 | 62.43% | 76.12% |
| 8 | 50.00% | 63.58% |

are in reality only few concurrent activities. Hence, most of the "start" activity events in the concurrent dataset are immediately followed by the "end" of the same activity. Therefore, most of the "end" of activities is predicted with 100% accuracy, which explains the higher accuracies of this method. This can be confirmed by the confusion matrix obtained with the prediction accuracy results in apartment 1 – Fig. 20. Nevertheless, this may be a good implementation in smart home environments where several activities can happen at the same time, e.g. multi-resident smart homes. This is not the case of our setup, hence the sequential activity dataset is probably a fairer algorithm. An example confusion matrix for this dataset (apartment 1), is shown in Fig. 21. The confusions within classes are similar for both types of datasets. *Bedroom activities* are mostly predicted as *in bed* and *kitchen activities*. This is understandable since bedroom activities happen often after having been in bed or in the living room, which has access to the kitchen. As mentioned before, *living room activities* are confused with *watching TV*, and a little with *kitchen activities*, as the previous comment. And finally, *being out* has been predicted most of the times as *kitchen activities*, as the entrance door also has a connection to the living room. An interesting result is that in this apartment the *watching tv* activity has been very well predicted – 86.5%. This could be useful for smart functions involving the TV, e.g. if the resident has difficulties operating the remote control. *Bathroom* and *kitchen activities* have also shown a considerably good accuracy (77.9% and 83.2%). The range of accuracy may be useful for analyzing patterns in the home and potentially for anomaly detection.

## IX. CONCLUSION

Sequential sensor events, time prediction, and activity recognition and prediction algorithms can enable the development of a number of support functions in smart home environments. Most of the research work in the literature has been carried out using data collected in lab environments and testbeds, typically including a quite large number of binary sensors (e.g. 50 sensors [18]). We collected data from eight apartments in a community care facility, with one resident each (over 70 years old). Data were collected from 13-17 sensors per apartment, over a period of time ranging from 75 to 385 days, depending on the apartment.

To our knowledge, there is no comparative study investigating state-of-the-art sequence prediction algorithms applied to sensor data acquired in homes of real users, as we do in this paper. We compare the performance of these methods regarding factors such as memory length and the required amount of data for good accuracy. When applying two probabilistic methods (ALZ and SPEED) and LSTM networks with both SPEED- and ALZ-text sequence inputs for prediction of the next sensor in a sequence, LSTM with SPEED-text has achieved the highest accuracy of 85%. SPEED achieved 3% lower accuracy and required much longer time to execute. On the other hand, the LSTM required about 4000 events in the training set to reach an accuracy close to its peak, whilst the probabilistic methods only needed about 2000 events. Hence, for datasets with little data SPEED may be beneficial. If there is a considerable amount of data (5000 events in this work), LSTM with SPEED-text is more suitable – it provides a higher accuracy and in much faster execution time than probabilistic methods. When tested in all the apartments, LSTM with SPEED-text achieves results in the range 76-85%.

There is quite limited work in the literature on the prediction of the time of occurrence in addition to the sensor events in smart homes. We study the possibility of improving the best performing algorithm (LSTM with SPEED-text) by including the time component in three different ways: period of the day (morning, afternoon, evening, night), 4-class time interval (elapsed time) to the next sensor event, and K-means time-cluster including information about the mean hour of the day and the mean time elapsed to the next sensor event. Our best performing model for predicting the next sensor event included the 4-class time interval input and attained a peak average accuracy of almost 87%. This is 2% better than without including the time information. Hence, the time elapsed between events contains some information that improves prediction, however, only marginally. In other apartments the improvement varied from 0.5-4.5%. We also predict both the next sensor event and the time of occurrence information, obtaining best results by using K-means time-cluster input. This implementation attained an accuracy of 83%. Other apartments had accuracies in the range 73-83%. Furthermore, we evaluate the variability of the prediction accuracy across the apartments and investigate the feasibility of transfer learning between these. Transfer learning has been shown to work successfully up to a certain number of events. For a low number of events in the training dataset, up to about 4000 events, transfer learning leads to higher prediction accuracy than when each apartment is modelled individually. This means that when a new apartment is added to the study, the prediction algorithm can work well straight away, and attain a relatively good accuracy (70-80%) from the first day in most cases. However, for larger training datasets, the prediction accuracy is approximately the same. In fact, in most cases it is marginally higher when each apartment is modelled individually.

A last analysis carried out activity recognition in a rule-based manner from the binary sensors events and performed activity prediction with the LSTM with SPEED-text algorithm. Two types of activity datasets were analyzed: sequential and concurrent. For the concurrent activity dataset, when predicting the next activity only, our best model achieved 95% accuracy, whilst when predicting the next activity and the mean duration and time of occurrence information, the best model achieved an accuracy of 85%. For the sequential activity dataset, the results are worse. When predicting the next activity, our best model achieved 90% accuracy, whilst when predicting the next activity and its duration and time of occurrence information, the best model achieved 80%. However, we indicate that this latter method may be fairer

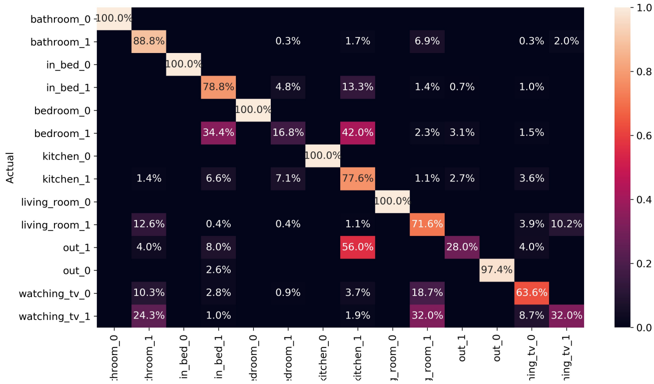



**FIGURE 20.** Confusion matrix of prediction of the next activity based on previous activities and K-means time-cluster for apartment 1, using the concurrent activity dataset.

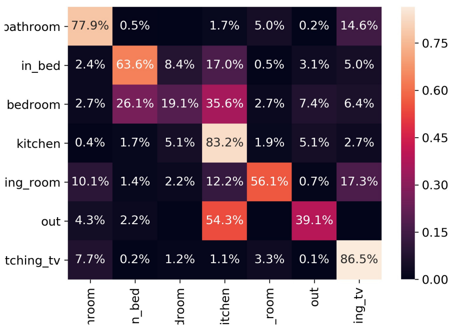



**FIGURE 21.** Confusion matrix of prediction of the next activity based on previous activities and K-means time-cluster for apartment 1, using the sequential activity dataset.

for our dataset where there are relatively few activities happening concurrently. Additional sensors could have been an advantage for better activity recognition and prediction. Our set of sensors proved to be somewhat limited for the task since it can only imply high-level activities. A small number of sensors like ours may, however, be preferable both in terms of reduced surveillance for the user, lower cost, and less nuance for the aesthetics of the home. Our work shows that it is possible to achieve acceptable prediction accuracy with few sensors. In addition, the findings of our study can be useful for deciding which analysis and prediction methods to use in accordance with project constraints (e.g. the number of available sensors, user privacy, etc.) and the area of application.

## ACKNOWLEDGMENT

The authors would like to thank the residents and the housekeepers at the seniors' care unit Skøyen Omsorg+ for their invaluable contribution to this work; Torhild Holthe and Dr. Anne Lund (OsloMet) for recruiting participants for the trial and communicating with the residents throughout the trial; Dejan Krunić and Øyvind Width (Sensio AS) for installations of the sensors; and the rest of the participants of the Assisted Living Project for a fruitful interdisciplinary collaboration.

## REFERENCES

[1] B. Winblad, K. Palmer, M. Kivipelto, V. Jelic, L. Fratiglioni, L.-O. Wahlund, a. Nordberg, L. Bäckman, M. Albert, O. Almkvist, H. Arai, H. Basun, K. Blennow, M. de Leon, C. DeCarli, T. Erkinjuntti, E. Giacobini, C. Graff, J. Hardy, C. Jack, a. Jorm, K. Ritchie, C. van Duijn, P. Visser, and R. C. Petersen, "Mild cognitive impairment–beyond controversies, towards a consensus: report of the International Working Group on Mild Cognitive Impairment." Journal of internal medicine, vol. 256, no. 3, pp. 240–246, 2004.

[2] S. Blackman, C. Matlo, C. Bobrovitskiy, A. Waldoch, M. L. Fang, P. Jackson, A. Mihailidis, L. Nygård, A. Astell, and A. Sixsmith, "Ambient Assisted Living Technologies for Aging Well: A Scoping Review," Journal of Intelligent Systems, vol. 25, no. 1, pp. 55–69, 2016.

[3] A. Mihailidis, J. Boger, M. Canido, and J. Hoey, "The use of an intelligent prompting system for people with dementia," Interactions, vol. 14, no. 4, p. 34, 2007.

[4] J. Hoey, A. V. Bertoldi, P. Poupart, and A. Mihailidis, "Assisting Persons with Dementia during Handwashing Using a Partially Observable Markov Decision Process," Icvs, no. Icvs, 2007.

[5] A. Ashraf and B. Taati, "Automated Video Analysis of Handwashing Behavior as a Potential Marker of Cognitive Health in Older Adults," IEEE Journal of Biomedical and Health Informatics, vol. 20, no. 2, pp. 682–690, 2016.

[6] S. Karaman, J. Benois-Pineau, R. M??gret, J. Pinquier, Y. Gaestel, and J. F. Dartigues, "Activities of daily living indexing by hierarchical HMM for dementia diagnostics," in Proceedings - International Workshop on Content-Based Multimedia Indexing, 2011, pp. 79–84.

[7] P. J. McCullagh, W. Carswell, M. D. Mulvenna, J. C. J. C. Augusto, H. Zheng, P. Jeffers, and W. P. Jeffers, "Nocturnal Sensing and Intervention for Assisted Living of People with Dementia," Healthcare Sensor Networks - Challenges Towards Practical Application, no. September 2011, pp. 283–303, 2012.

[8] M. A. Feki, J. Biswas, and A. Tolstikov, "Model and algorithmic framework for detection and correction of cognitive errors," Technology and Health Care, vol. 17, no. 3, pp. 203–219, 2009.
[9] E. Zouganeli, F. Casagrande, T. Holthe, A. Lund, L. Halvorsrud, D. Karterud, A. Flakke-Johannessen, H. Lovett, S. Mørk, J. Strøm-Gundersen, E. Thorstensen, R. Norvoll, R. Meulen, M.-R. Kennedy, R. Owen, M. Ladikas, and E.-M. Forsberg, "Responsible development of self-learning assisted living technology for older adults with mild cognitive impairment or dementia," ICT4AWE 2017 - Proceedings of the 3rd International Conference on Information and Communication Technologies for Ageing Well and e-Health, no. Ict4awe, pp. 204–209, 2017.
[10] F. D. Casagrande, J. Tørresen, and E. Zouganeli, "Comparison of Probabilistic Models and Neural Networks on Prediction of Home Sensor Events," in Submitted to International Joint Conference on Neural Networks, 2019.
[11] ——, "Sensor Event Prediction using Recurrent Neural Network in Smart Homes for Older Adults," in In Press. International Conference on Intelligent Systems, 2018.
[12] F. D. Casagrande and E. Zouganeli, "Occupancy and Daily Activity Event Modelling in Smart Homes for Older Adults with Mild Cognitive Impairment or Dementia," in Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59), 2018, pp. 236–242.
[13] F. D. Casagrande, J. Tørresen, and E. Zouganeli, "Prediction of the Next Sensor Event and its Time of Occurrence in Smart Homes," in Submitted to Smart Systems: bringing together computer vision, sensor networks and machine learning, 2019.
[14] F. D. Casagrande and E. Zouganeli, "Prediction of Next Sensor Event and its Time of Occurrence using Transfer Learning across Homes," in Submitted to ACS/IEEE International Conference on Computer Systems and Applications AICCSA 2019, 2019.
[15] L. B. Holder and D. J. Cook, "Automated activity-aware prompting for activity initiation," Gerontechnology, vol. 11, no. 4, pp. 534–544, 2013.
[16] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui, "Smart-FABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment," Artificial Intelligence in Medicine, vol. 67, pp. 57–74, 2016.
[17] S. Wu, J. B. Rendall, M. J. Smith, S. Zhu, J. Xu, H. Wang, Q. Yang, and P. Qin, "Survey on Prediction Algorithms in Smart Homes," IEEE Internet of Things Journal, vol. 4, no. 3, pp. 636–644, 2017.
[18] K. Gopalratnam and D. J. Cook, "Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm," IEEE Intelligent Systems, vol. 22, no. 1, 2007.
[19] M. R. Alam, M. B. Reaz, and M. A. Mohd Ali, "SPEED: An inhabitant activity prediction algorithm for smart homes," IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans, vol. 42, no. 4, pp. 985–990, 2012.
[20] A. Lotfi, C. Langensiepen, S. M. Mahmoud, and M. J. Akhlaghinia, "Smart homes for the elderly dementia sufferers: Identification and prediction of abnormal behaviour," Journal of Ambient Intelligence and Humanized Computing, vol. 3, no. 3, pp. 205–218, 2012.
[21] L. Vintan, A. Gellert, J. Petzold, and T. Ungerer, "Person movement prediction using neural networks," in Proceedings of the KI2004 International Workshop on Modeling and Retrieval of Context, vol. 114, no. 4, 2004, pp. 1–12.
[22] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Next Location Prediction Within a Smart Office Building," Cognitive Science Research Paper Unversity of Sussex CSRP, vol. 577, p. 69, 2005.
[23] G. E. P. Box and G. Jenkins, Time Series Analysis, Forecasting and Control. San Francisco, CA, USA: Holden-Day, Inc., 1990.
[24] B. Minor and D. J. Cook, "Forecasting occurrences of activities," Pervasive and Mobile Computing, 2016.
[25] A. Aztiria, J. C. Augusto, A. Izaguirre, and D. Cook, "Learning accurate temporal relations from user actions in intelligent environments," in Advances in Soft Computing, vol. 51, 2009, pp. 274–283.
[26] S. Mahmoud, A. Lotfi, and C. Langensiepen, "Behavioural pattern identification and prediction in intelligent environments," Applied Soft Computing Journal, vol. 13, no. 4, pp. 1813–1822, 2013.
[27] E. Nazerfard and D. J. Cook, "CRAFFT: An Activity Prediction Model based on Bayesian Networks," vol. 33, no. 4, pp. 395–401, 2015.
[28] T. Mahmud, M. Hasan, A. Chakraborty, and A. K. Roy-Chowdhury, "A poisson process model for activity forecasting," in 2016 IEEE International Conference on Image Processing (ICIP), Sep. 2016, pp. 3339–3343.
[29] S. J. Pan and Q. Y. Fellow, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, pp. 279–286, 2009.
[30] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," Knowledge and Information Systems, vol. 36, no. 3, pp. 537–556, 2013.
[31] D. H. Hu and Q. Yang, "Transfer learning for activity recognition via sensor mapping," IJCAI International Joint Conference on Artificial Intelligence, pp. 1962–1967, 2011.
[32] P. Rashidi and D. J. Cook, "Activity knowledge transfer in smart environments," Pervasive and Mobile Computing, vol. 7, no. 3, pp. 331–343, 2011.
[33] T. Desautels, J. Calvert, J. Hoffman, Q. Mao, M. Jay, G. Fletcher, C. Barton, U. Chettipally, . Yaniv Kerem5, and R. Das1, "Using Transfer Learning for Improved Mortality Prediction in a Data-Scarce Hospital Setting," African Journal of Neurological Sciences, vol. 34, no. 1, 2015.
[34] D. H. Hu, V. W. Zheng, and Q. Yang, "Cross-domain activity recognition via transfer learning," Pervasive and Mobile Computing, vol. 7, no. 3, pp. 344–358, 2010.
[35] P. Gupta, P. Malhotra, L. Vig, and G. Shroff, "Transfer Learning for Clinical Time Series Analysis using Recurrent Neural Networks," pp. 4–7, 2018.
[36] P. Jaini, Z. Chen, P. Carbajal, E. Law, L. Middleton, K. Regan, M. Schaekermann, G. Trimponias, J. Tung, P. Poupart, and M. Engineering, "Online Bayesian Transfer Learning for Sequential Data Modeling," pp. 1–20, 2017.
[37] T. L. M. V. Kasteren, G. Englebienne, and B. J. A. Kr, "Transferring Knowledge of Activity Recognition accross Sensor Networks," Pervasive, pp. 283–300, 2010.
[38] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Transfer learning for time series classification," 2018.
[39] J. G. Cleary and I. H. Witten, "Data Compression Using Adaptive Coding and Partial String Matching," IEEE Transactions on Communications, vol. 32, no. 4, pp. 396–402, 1984.
[40] J. Cleary, W. Teahan, and I. Witten, "Unbounded length contexts for PPM," in Proceedings DCC '95 Data Compression Conference, pp. 52–61. [Online]. Available: https://www.cs.waikato.ac.nz/ml/publications/1995/Cleary-Teahan-Witten-ppm.pdf http://ieeexplore.ieee.org/document/515495/
[41] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," IEEE Transactions on Information Theory, vol. 24, no. 5, pp. 530–536, sep 1978. [Online]. Available: http://ieeexplore.ieee.org/document/1055934/
[42] A. Graves, "Generating Sequences With Recurrent Neural Networks," arXiv, 2014.
[43] J. Martens, "Generating Text with Recurrent Neural Networks," Neural Networks, vol. 131, no. 1, pp. 1017–1024, 2011.
[44] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech Recognition With Deep Recurrent Neural Networks," Icassp, no. 3, pp. 6645–6649, 2013.
[45] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[46] N. E. Elhady and J. Provost, "A Systematic Survey on Sensor Failure Detection and Fault-Tolerance in Ambient Assisted Living," Sensors, vol. 18, no. 1991, p. 19, 2018.
[47] K. M. Bataineh, M. Najia, and M. Saqera, "A comparison study between various fuzzy clustering algorithms," 2011.
[48] K. D. Joshi and P. S. Nalwade, "Modified k-means for better initial cluster centres," 2013.
[49] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
[50] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," Journal of Machine Learning Research, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

# Bibliography


[1] Aarhaug, J., Ørving, T., and Kristensen, N. B. *Samfunnstrender og ny teknologi Perspektiver for fremtidens transportsystem (Future ageing and new technology)*. 2009.

[2] Alam, M. R., Reaz, M. B., and Mohd Ali, M. A. "SPEED: An inhabitant activity prediction algorithm for smart homes". In: *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans* vol. 42, no. 4 (2012), pp. 985–990.

[3] Alam, M. R., Reaz, M. B. I., and Ali, M. A. M. "A review of smart homes - Past, present, and future". In: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* vol. 42, no. 6 (2012), pp. 1190–1203. arXiv: TSMCC.2012.2189204 [10.1109].

[4] Alwan, M. et al. "A Smart and Passive Floor-Vibration Based Fall Detector for Elderly". In: *International Conference on Information & Communication Technologies* vol. 1, no. November 2016 (2006), pp. 3–7.

[5] Ashraf, A. and Taati, B. "Automated Video Analysis of Handwashing Behavior as a Potential Marker of Cognitive Health in Older Adults". In: *IEEE Journal of Biomedical and Health Informatics* vol. 20, no. 2 (2016), pp. 682–690.

[6] Augusto, J. C. et al. "Nocturnal Ambient Assisted Living". In: *Second International Joint Conference on AmI*. November 2011. 2011, pp. 350–354. arXiv: 9780201398298.

[7] Aztiria, A. et al. "Learning accurate temporal relations from user actions in intelligent environments". In: *Advances in Soft Computing*. Vol. 51. 2009, pp. 274–283.

[8] Bagalà, F. et al. "Evaluation of Accelerometer-Based Fall Detection Algorithms on Real-World Falls". In: *PLoS ONE* vol. 7, no. 5 (2012), e37062.

[9] Bataineh, K. M., Naji, M., and Saqer, M. "A comparison study between various fuzzy clustering algorithms". In: *Jordan Journal of Mechanical and Industrial Engineering* vol. 5, no. 4 (2011), pp. 335–343.

[10] Bergeron, F. et al. "Indoor Positioning System for Smart Homes Based on Decision Trees and Passive RFID". In: *Proceedings of the 20th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2016*. 2016, pp. 42–53. arXiv: arXiv:1011.1669v3.

[11] Blackman, S. et al. "Ambient Assisted Living Technologies for Aging Well: A Scoping Review". In: *Journal of Intelligent Systems* vol. 25, no. 1 (2016), pp. 55–69.

[12] Blasco, R. et al. "A smart kitchen for ambient assisted living." In: *Sensors (Basel, Switzerland)* vol. 14, no. 1 (2013), pp. 1629–1653.

[13] Boger, J. et al. "A decision-theoretic approach to task assistance for persons with dementia". In: *IJCAI International Joint Conference on Artificial Intelligence*. 2005, pp. 1293–1299.

[14] Boger, J. et al. "A planning system based on Markov decision processes to guide people with dementia through activities of daily living." In: *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society* vol. 10, no. 2 (2006), pp. 323–333.

[15] Bono-Nuez, A. et al. "Ambient intelligence for quality of life assessment". In: *Journal of Ambient Intelligence and Smart Environments* vol. 6, no. 1 (2014), pp. 57–70.

[16] Bouchard, B., Giroux, S., and Bouzouane, A. "A Keyhole Plan Recognition Model for Alzheimer'S Patients: First Results". In: *Applied Artificial Intelligence* vol. 21, no. 7 (2007), pp. 623–658.

[17] Box, George Edward Pelham Jenkins, G. *Time Series Analysis, Forecasting and Control*. San Francisco, CA, USA: Holden-Day, Inc., 1990.

[18] Carmien, S. "End user programming and context responsiveness in handheld prompting systems for persons with cognitive disabilities and caregivers". In: *Conference on Human Factors in Computing Systems - Proceedings* (2005), pp. 1252–1255.

[19] Chahuara, P., Portet, F., and Vacher, M. "Context-aware decision making under uncertainty for voice-based control of smart home". In: *Expert Systems With Applications* vol. 75 (2017), pp. 63–79.

[20] Chahuara, P. et al. "On-line Human Activity Recognition from Audio and Home Automation Sensors". In: *Journal of Ambient Intelligence and Smart Environments (JAISE)* vol. 8, no. 4 (2016), pp. 399–422.

[21] Chaminda, H. T., Klyuev, V., and Naruse, K. "A smart reminder system for complex human activities". In: *International Conference on Advanced Communication Technology, ICACT* (2012), pp. 235–240.

[22] Chan, M. et al. "A review of smart homes-Present state and future challenges". In: *Computer Methods and Programs in Biomedicine* vol. 91, no. 1 (2008), pp. 55–81.

[23] Chen, C., Das, B., and Cook, D. J. "A Data Mining Framework for Activity Recognition in Smart Environments". In: *2010 Sixth International Conference on Intelligent Environments (IE)* (2010), pp. 80–83.

[24] Chollet, F. and others. *Keras*. 2015.

[25] Cleary, J., Teahan, W., and Witten, I. "Unbounded length contexts for PPM". In: *Proceedings DCC '95 Data Compression Conference*, pp. 52–61.

[26] Cleary, J. G. and Witten, I. H. "Data Compression Using Adaptive Coding and Partial String Matching". In: *IEEE Transactions on Communications* vol. 32, no. 4 (1984), pp. 396–402.

[27] Cook, D. J., Krishnan, N. C., and Rashidi, P. "Activity Discovery and Activity Recognition: A New Partnership". In: *Cybernetics, IEEE Transactions on* vol. 43, no. 3 (2013), pp. 820–828.

[28] Cook, D. J. and Holder, L. B. "Sensor selection to support practical use of health-monitoring smart environments". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* vol. 1, no. 4 (2011), pp. 339–351.

[29] Cook, D., Feuz, K. D., and Krishnan, N. C. "Transfer learning for activity recognition: A survey". In: *Knowledge and Information Systems* vol. 36, no. 3 (2013), pp. 537–556. arXiv: NIHMS150003.

[30] Cook, D. et al. "MavHome: an agent-based smart home". In: *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).* (2003), pp. 521–524.

[31] Das, B. et al. "PUCK: An automated prompting system for smart environments: Toward achieving automated prompting-challenges involved". In: *Personal and Ubiquitous Computing* vol. 16, no. 7 (2012), pp. 859–873.

[32] Das, S. and Cook, D. "Health monitoring in an agent-based smart home by activity prediction". In: *Toward a human-friendly assistive environment: ICOST'2004, 2nd International Conference on Smart Homes and Health Telematics*, no. July 2015 (2004), pp. 3–14.

[33] Dawadi, P. N., Cook, D. J., and Schmitter-Edgecombe, M. "Modeling patterns of activities using activity curves". In: *Pervasive and Mobile Computing* vol. 28 (2016), pp. 51–68.

[34] Dawadi, P., Schmitter-Edgecombe, M., and Cook, D. J. "Smart home-based longitudinal functional assessment". In: *ACM UbiComp Workshop on Smart Health Systems and Applications* (2014), pp. 1217–1224.

[35] Demiris, G. et al. "Older adults attitudes towards and perceptions of 'smart home technologies: A pilot study". In: *Informatics for Health and Social Care* vol. 29, no. 2 (June 2004), pp. 87–94.

[36] Desautels, T. et al. "Using Transfer Learning for Improved Mortality Prediction in a Data-Scarce Hospital Setting". In: *African Journal of Neurological Sciences* vol. 34, no. 1 (2015).

[37] Dias Casagrande, F. *Literature Review on Assisted Living Technologies*. Tech. rep. 2017, p. 73.

[38] Dubowsky, S., Genot, F., and Godding, S. "PAMM-a robotic aid to the elderly for mobility assistance and monitoring: a "helping-hand" for the elderly". In: *International Conference on Robotics and Automation*. April. 2000, pp. 1–7.

[39] Elhady, N. E. and Provost, J. “A Systematic Survey on Sensor Failure Detection and Fault-Tolerance in Ambient Assisted Living”. In: *Sensors* vol. 18, no. 1991 (2018), p. 19.

[40] Engage2020 Consortium. *Engage2020 - Tools and instruments for a better societal engagement in "Horizon 2020"*. Tech. rep. 2014.

[41] Farella, E., Falavigna, M., and Ricc, B. “Aware and smart environments: The Casattenta project”. In: *Microelectronics Journal* vol. 41, no. 11 (2010), pp. 697–702.

[42] Fawaz, H. I. et al. “Transfer learning for time series classification”. In: (2018). arXiv: 1811.01533.

[43] Feki, M. A., Biswas, J., and Tolstikov, A. “Model and algorithmic framework for detection and correction of cognitive errors”. In: *Technology and Health Care* vol. 17, no. 3 (2009), pp. 203–219.

[44] Forsberg, E. M. et al. “Assessment of science and technologies: Advising for and with responsibility”. In: *Technology in Society* vol. 42 (2015), pp. 21–27.

[45] Fudickar, S. J. F., Faerber, S., and Schnor, B. “KopAL Appointment User-interface: An Evaluation with Elderly”. In: *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, no. 42 (2011), pp. 1–6.

[46] Fudickar, S. and Schnor, B. “KopAL - A mobile orientation system for dementia patients”. In: *Communications in Computer and Information Science* vol. 53 CCIS (2009), pp. 109–118.

[47] Gers, F. A., Schraudolph, N. N., and Schmidhuber, J. “Learning precise timing with LSTM recurrent networks”. In: *Journal of Machine Learning Research* vol. 3, no. 1 (2003), pp. 115–143.

[48] Giroux, S. et al. “Cognitive Assistance to Meal Preparation: Design, Implementation, and Assessment in a Living Lab”. In: *2015 AAAI Spring . . .* (2015), pp. 01–25.

[49] Gopalratnam, K. and Cook, D. J. “Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm”. In: *IEEE Intelligent Systems* vol. 22, no. 1 (2007).

[50] Graves, A. “Generating Sequences With Recurrent Neural Networks”. In: *arXiv* (2014). arXiv: arXiv:1308.0850v5.

[51] Graves, A., Mohamed, A.-R., and Hinton, G. “Speech Recognition With Deep Recurrent Neural Networks”. In: *Icassp*, no. 3 (2013), pp. 6645–6649. arXiv: arXiv:1303.5778v1.

[52] Gupta, P. et al. “Transfer Learning for Clinical Time Series Analysis using Recurrent Neural Networks”. In: (2018), pp. 4–7. arXiv: 1807.01705.

[53] Hochreiter, S. and Schmidhuber, J. “Long Short-Term Memory”. In: *Neural Computation* vol. 9, no. 8 (1997), pp. 1735–1780. arXiv: 1206.2944.

[54] Hoey, J., Schroeder, T., and Alhothali, A. "Affect control processes: Intelligent affective interaction using a partially observable Markov decision process". In: *Artificial Intelligence* vol. 230 (2016), pp. 134–172.

[55] Hoey, J. et al. "Assisting Persons with Dementia during Handwashing Using a Partially Observable Markov Decision Process". In: *Icvs*, no. Icvs (2007).

[56] Hoey, J. et al. "Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process". In: *Computer Vision and Image Understanding* vol. 114, no. 5 (2010), pp. 503–519.

[57] Holder, L. B. and Cook, D. J. "Automated activity-aware prompting for activity initiation". In: *Gerontechnology* vol. 11, no. 4 (2013), pp. 534–544.

[58] Holthe, T. et al. "The assisted living project: a process evaluation of implementation of sensor technology in community assisted living. A feasibility study". In: *Disability and Rehabilitation: Assistive Technology* vol. 0, no. 0 (2018), pp. 1–8.

[59] Hu, D. H., Zheng, V. W., and Yang, Q. "Cross-domain activity recognition via transfer learning". In: *Pervasive and Mobile Computing* vol. 7, no. 3 (2010), pp. 344–358.

[60] Jaini, P. et al. "Online Bayesian Transfer Learning for Sequential Data Modeling". In: (2017), pp. 1–20.

[61] Joshi, K. D. and Nalwade, P. S. "Modified K-Means for Better Initial Cluster Centres". In: *International Journal of Computer Science and Mobile Computing* vol. 2, no. July (2013), pp. 219–223.

[62] Karakostas, A. et al. "Sensor-based in-home monitoring of people with dementia using remote web technologies". In: *Proceedings of 2015 International Conference on Interactive Mobile Communication Technologies and Learning, IMCL 2015*, no. November (2015), pp. 353–357.

[63] Karakostas, A. et al. "Supporting cognitive skills of people suffering from dementia through a sensor-based system". In: *Proceedings - IEEE 15th International Conference on Advanced Learning Technologies: Advanced Technologies for Supporting Open Access to Formal and Informal Learning, ICALT 2015* (2015), pp. 460–461.

[64] Karaman, S. et al. "Activities of daily living indexing by hierarchical HMM for dementia diagnostics". In: *Proceedings - International Workshop on Content-Based Multimedia Indexing*. 2011, pp. 79–84. arXiv: 1106.4451.

[65] Kasteren, T. L. M. V., Englebienne, G., and Kr, B. J. A. "Transferring Knowledge of Activity Recognition accross Sensor Networks". In: *Pervasive* (2010), pp. 283–300.

[66] Kepski, M. and Kwolek, B. “Fall detection using ceiling-mounted 3d depth camera”. In: *Int. Conf. VISAPP, pages II* (2014), pp. 640–647.

[67] Kingma, D. P. and Ba, J. L. “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION”. In: *3rd International Conference for Learning Representations*. 2014.

[68] König, A. et al. “Ecological assessment of autonomy in instrumental activities of daily living in dementia patients by the means of an automatic video monitoring system”. In: *Frontiers in Aging Neuroscience* vol. 7, no. JUN (2015), pp. 1–11.

[69] Krishnan, N. C. and Cook, D. J. “Activity recognition on streaming sensor data”. In: *Pervasive and Mobile Computing* vol. 10, no. PART B (2014), pp. 138–154.

[70] LeCun, Y., Bengio, Y., and Hinton, G. “Deep learning”. In: *Nature* vol. 521, no. 7553 (2015), pp. 436–444. arXiv: arXiv:1312.6184v5.

[71] Lema{{\^i}}tre, G., Nogueira, F., and Aridas, C. K. “Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning”. In: *Journal of Machine Learning Research* vol. 18, no. 17 (2017), pp. 1–5.

[72] Lotfi, A. et al. “Smart homes for the elderly dementia sufferers: Identification and prediction of abnormal behaviour”. In: *Journal of Ambient Intelligence and Humanized Computing* vol. 3, no. 3 (2012), pp. 205–218.

[73] Mahmoud, S., Lotfi, A., and Langensiepen, C. “Behavioural pattern identification and prediction in intelligent environments”. In: *Applied Soft Computing Journal* vol. 13, no. 4 (2013), pp. 1813–1822.

[74] Martens, J. “Generating Text with Recurrent Neural Networks”. In: *Neural Networks* vol. 131, no. 1 (2011), pp. 1017–1024. arXiv: 9809069v1 [arXiv:gr-qc].

[75] Mastorakis, G. and Makris, D. “Fall detection system using Kinect’s infrared sensor”. In: *Journal of Real-Time Image Processing* vol. 9, no. 4 (2012), pp. 635–646.

[76] McCullagh, P. J. et al. “Nocturnal Sensing and Intervention for Assisted Living of People with Dementia”. In: *Healthcare Sensor Networks - Challenges Towards Practical Application*, no. September 2011 (2012), pp. 283–303.

[77] Meditskos, G. and Kompatsiaris, I. “Dem @ Care : Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support”. In: *Mobihealth Satellite Event*. 2014.

[78] Michelet, M. et al. “The validity of the Norwegian version of the cognitive function instrument”. In: *Dementia and Geriatric Cognitive Disorders* vol. 46, no. 3-4 (2018), pp. 217–228.

[79] Mihailidis, A., Carmichael, B., and Boger, J. “The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home”. In: *IEEE Transactions on Information Technology in Biomedicine* vol. 8, no. 3 (2004), pp. 238–247.

[80] Mihailidis, A., R. Fernie, G., and C. Barbenel, J. “The use of artificial intelligence in the design of an intelligent cognitive orthosis for people with dementia”. In: vol. 13 (2001), pp. 23–39.

[81] Mihailidis, A. et al. “The use of an intelligent prompting system for people with dementia”. In: *Interactions* vol. 14, no. 4 (2007), p. 34.

[82] Minor, B. and Cook, D. J. “Regression tree classification for activity prediction in smart homes”. In: *UbiComp 2014 - Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2014), pp. 441–450.

[83] Minor, B. and Cook, D. J. “Forecasting occurrences of activities”. In: *Pervasive and Mobile Computing* (2016).

[84] Minor, B., Doppa, J. R., and Cook, D. J. “Data-driven activity prediction: algorithms, evaluation methodology, and applications”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), pp. 805–814.

[85] Morris, M. et al. “Smart technologies for older people: a systematic literature review of smart technologies that promote health and wellbeing of older people living at home”. In: *Institute for a Broadband-Enabled Society*, no. 978 0 7340 4781 6 (2012), p. 76.

[86] Nazerfard, E. and Cook, D. J. “CRAFFT: An Activity Prediction Model based on Bayesian Networks”. In: vol. 33, no. 4 (2015), pp. 395–401. arXiv: 15334406.

[87] Nazerfard, E. and Cook, D. J. “Using Bayesian Networks for Daily Activity Prediction”. In: *Proceedings of the 13th AAAI Conference on Plan, Activity, and Intent Recognition.* 2013, pp. 32–38.

[88] Ni, Q., Hernando, A., and Cruz, I. de la. *The elderly’s independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development.* Vol. 15. 5. 2015, pp. 11312–11362.

[89] Nikamalfard, H. et al. “A sleep pattern analysis and visualization system to support people with early dementia”. In: *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops* (2011), pp. 510–513.

[90] Nikamalfard, H. et al. “Knowledge discovery from activity monitoring to support independent living of people with early dementia”. In: *Proceedings - IEEE-EMBS International Conference on Biomedical and Health Informatics: Global Grand Challenge of Health Informatics, BHI 2012* vol. 25, no. Bhi (2012), pp. 910–913.

[91] Nitesh V. Chawla, Bowyer, K. W., and Hall, L. O. “SMOTE: Synthetic Minority Over-sampling Technique Nitesh”. In: *Journal of Artificial Intelligence Research* vol. 2009, no. Sept. 28 (2006), pp. 321–357. arXiv: 1106.1813.

[92] Özdemir, A. T. and Barshan, B. “Detecting Falls with Wearable Sensors Using Machine Learning Techniques”. In: *Sensors* vol. 14, no. 6 (2014), pp. 10691–10708.

[93] Pålshaugen, Ø. “Dialogues in innovation: Interactive learning and interactive research as means for a context sensitive regional innovation policy”. In: *International Journal of Action Research* vol. 10, no. 2 (2014), pp. 213–234.

[94] Pan, S. J. and Fellow, Q. Y. “A Survey on Transfer Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* (2009), pp. 279–286. arXiv: 1808.01974.

[95] Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* vol. 12 (2011), pp. 2825–2830.

[96] Petzold, J. et al. “Next Location Prediction Within a Smart Office Building”. In: *International Workshop on Exploiting Context Histories in Smart Environments* vol. 577 (2005), p. 69.

[97] Picking, R. et al. “E-servant: an intelligent, programmable system to support and integrate assisted living technologies”. In: vol. 1, no. 2 (2014), pp. 64–67.

[98] Pollack, M. E. et al. “Autominder: An intelligent cognitive orthotic system for people with memory impairment”. In: *Robotics and Autonomous Systems* vol. 44, no. 3-4 (2003), pp. 273–282.

[99] Rashidi, P. and Cook, D. J. “Activity knowledge transfer in smart environments”. In: *Pervasive and Mobile Computing* vol. 7, no. 3 (2011), pp. 331–343.

[100] Rashidi, P. and Cook, D. J. “Mining and monitoring patterns of daily routines for assisted living in real world settings”. In: *Proceedings of the ACM international conference on Health informatics - IHI ’10* (2010), p. 336.

[101] Rashidi, P. and Cook, D. J. “Mining Sensor Streams for Discovering Human Activity Patterns Over Time”. In: (2010).

[102] Rashidi, P. and Mihailidis, A. “A survey on ambient-assisted living tools for older adults”. In: *IEEE Journal of Biomedical and Health Informatics* vol. 17, no. 3 (2013), pp. 579–590.

[103] Rashidi, P. et al. “Discovering activities to recognize and track in a smart environment”. In: *IEEE Transactions on Knowledge and Data Engineering* vol. 23, no. 4 (2011), pp. 527–539.

[104] Reaz, M. B. I. "Artificial Intelligence Techniques for Advanced Smart Home Implementation". In: *Acta Technica Corvininesis - Bulletin of Engineering* vol. 6, no. 2 (2013), pp. 51–57.

[105] Riboni, D. et al. "SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment". In: *Artificial Intelligence in Medicine* vol. 67 (2016), pp. 57–74. arXiv: 1501.05581.

[106] Roy, P. C. et al. "A Possibilistic Approach for Activity Recognition in Smart Homes for Cognitive Assistance to Alzheimer's Patients". In: *Activity Recognition in Pervasive Intelligent Environments* (2011), pp. 33–58.

[107] S.A., F. G. *Fibaro*.

[108] Sacco, G. et al. "Detection of activities of daily living impairment in Alzheimer's disease and mild cognitive impairment using information and communication technology." In: *Clinical interventions in aging* vol. 7 (2012), pp. 539–549.

[109] Sanchez, V. G. "A Review of Smart House Analysis Methods for Assisting Older People Living Alone". In: *Journal of Sensor and Actuator Networks* vol. 6, no. 3 (2017), p. 11.

[110] Satt, A. et al. "Evaluation of speech-based protocol for detection of early-stage dementia". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, no. August (2013), pp. 1692–1696.

[111] Schomberg, R. V. "A Vision of Responsible Research and Innovation". In: *Responsible Innovation: Managing the Responsible Emergence of Science and Innovation in Society*. 2013, pp. 51–74.

[112] *Sensio*.

[113] Sprint, G. and Cook, D. J. "Unsupervised detection and analysis of changes in everyday physical activity data". In: *Journal of Biomedical Informatics* vol. 63 (2016), pp. 54–65.

[114] Sprint, G. et al. "Detecting Health and Behavior Change by Analyzing Smart Home Sensor Data". In: *2016 IEEE International Conference on Smart Computing, SMARTCOMP 2016* (2016).

[115] Sprint, G. et al. "Using smart homes to detect and analyze health events". In: *IEEE Computer* (2016), pp. 1–11.

[116] Stone, E. and Skubic, M. "Fall Detection in Homes of Older Adults Using the Microsoft Kinect." In: *IEEE journal of biomedical and health informatics* vol. 19, no. c (2014), pp. 290–301.

[117] Teipel, S. et al. *Information and communication technology solutions for outdoor navigation in dementia*. 2016.

[118] Thomas, B. L., Crandall, A. S., and Cook, D. J. "A Genetic Algorithm Approach to Motion Sensor Placement in Smart Environments." In: *Journal of reliable intelligent environments* vol. 2, no. 1 (2016), pp. 3–16.

[119] TSD. *Services for sensitive data.*

[120] United Nations. *World Population Ageing 2017 - Highlights (ST/ESA/SER.A/397).* Tech. rep.

[121] Usman Iqbal, S. et al. *The veterans RAND 12 item health survey (VR-12): what is it and how it is used.* Tech. rep. Boston University, 2015.

[122] Vacher, M. et al. "The Sweet-Home project: Audio processing and decision making in smart home to improve well-being and reliance". In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (2013), pp. 7298–7301.

[123] Vikramaditya Jakkula, D. J. C. "Mining Sensor Data in Smart Environment for Temporal Activity Prediction". In: *The 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2007).

[124] Vintan, L. et al. "Person movement prediction using neural networks". In: *Proceedings of the KI2004 International Workshop on Modeling and Retrieval of Context.* Vol. 114. 4. 2004, pp. 1–12.

[125] Walsh, S. P. et al. "ADCS Prevention Instrument Project: The Mail-In Cognitive Function Screening Instrument (MCFSI)". In: *Baseline* vol. 20, no. December (2006), pp. 179–190.

[126] Wang, H. et al. "Monitoring and analysis of sleep pattern for people with early dementia". In: *2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops, BIBMW 2010*, no. 2 (2010), pp. 405–410.

[127] WHO (World Health Organization). *Global action plan on the public health response to dementia.* Tech. rep. 2017.

[128] Williams, J. A. and Cook, D. J. "Forecasting Behavior in Smart Homes Based on Sleep and Wake Patterns". In: *Technology and Health Care* (2016), pp. 1–22.

[129] Winblad, B. et al. "Mild cognitive impairment–beyond controversies, towards a consensus: report of the International Working Group on Mild Cognitive Impairment." In: *Journal of internal medicine* vol. 256, no. 3 (2004), pp. 240–246.

[130] Wu, S. et al. "Survey on Prediction Algorithms in Smart Homes". In: *IEEE Internet of Things Journal* vol. 4, no. 3 (2017), pp. 636–644.

[131] Yang, L., Ren, Y., and Zhang, W. "3D Depth Image Analysis for Indoor Fall Detection of Elderly People". In: *Digital Communications and Networks* vol. 2, no. 1 (2016), pp. 24–34.

[132] Zhang, L. J., Jia, H. M., and Jiang, D. P. "Sliding mode prediction control for 3D path following of an underactuated AUV". In: *IFAC Proceedings Volumes.* Vol. 19. 2014, pp. 8799–8804. arXiv: 1204.3968.

[133] Zhang, Q., Su, Y., and Yu, P. "Assisting an Elderly with Early Dementia Using Wireless". In: *IFIP Advances in Information and Communication Technology.* Vol. 426. 2014, pp. 398–404.

[134] Zhang, Q. et al. "Determination of Activities of Daily Living of independent living older people using environmentally placed sensors". In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2013), pp. 7044–7047.

[135] Zhu, N. et al. "Bridging e-Health and the Internet of Things: The SPHERE Project". In: *IEEE Intelligent Systems* vol. 30, no. 4 (July 2015), pp. 39–46.

[136] Ziv, J. and Lempel, A. "Compression of Individual Sequences via Variable-Rate Coding". In: *IEEE Transactions on Information Theory* vol. 24, no. 5 (Sept. 1978), pp. 530–536.

[137] Zouganeli, E. et al. "Responsible development of self-learning assisted living technology for older adults with mild cognitive impairment or dementia". In: *ICT4AWE 2017 - Proceedings of the 3rd International Conference on Information and Communication Technologies for Ageing Well and e-Health*, no. Ict4awe (2017), pp. 204–209.