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.

https://doi.org/10.3384/ecp18153236

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

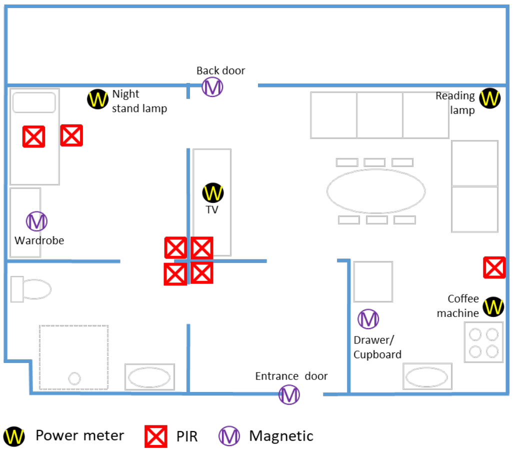

Figure 1. Sensors system in the field trial apartment.

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.

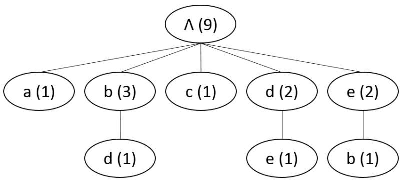

Figure 2. Tree generated by ALZ from sequence "abcdebdeb".

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

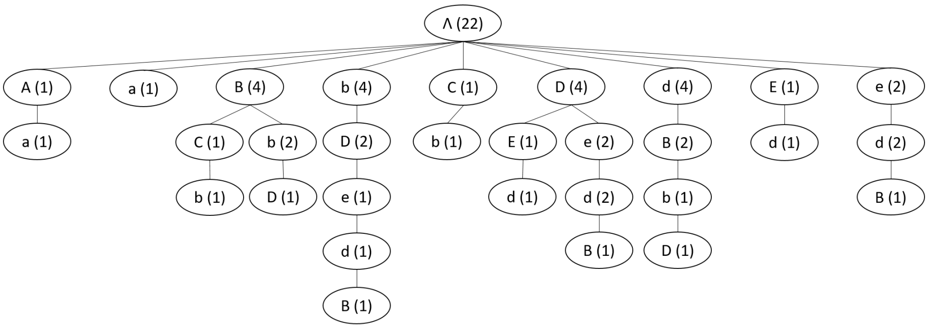

Figure 3. Tree generated by ALZ from sequence "AaBCbDEdBbDedB".

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

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

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

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 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.

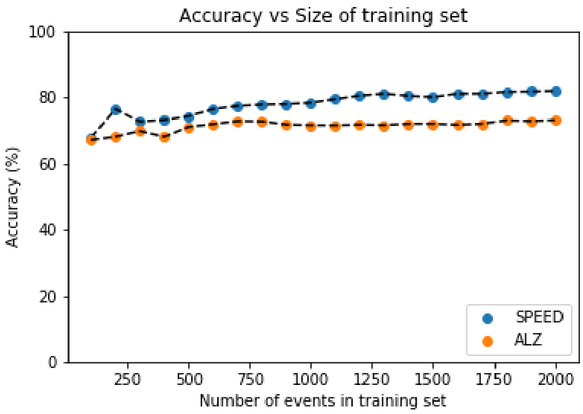

Figure 4. SPEED and ALZ prediction accuracy vs. the size of training set.

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

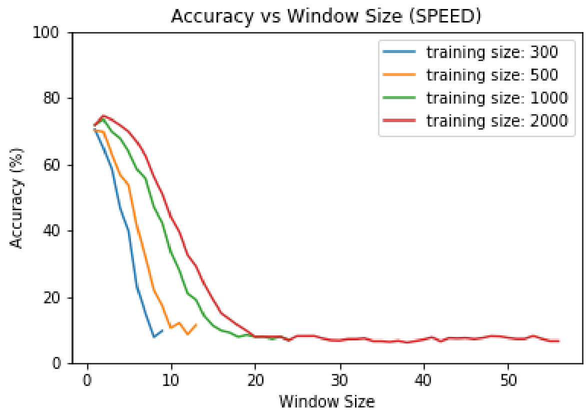

Figure 5. SPEED prediction accuracy vs. the window length, for several training set sizes.

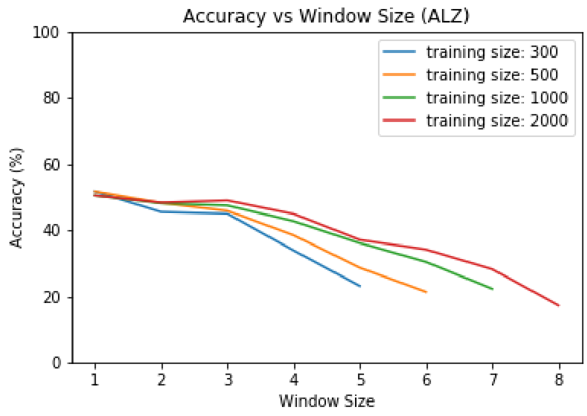

Figure 6. ALZ prediction accuracy vs. the window length, for several training set sizes.

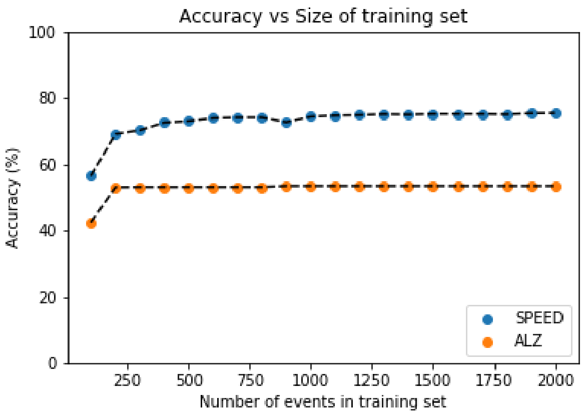

Figure 7. SPEED and ALZ prediction accuracy vs. the size of training set.

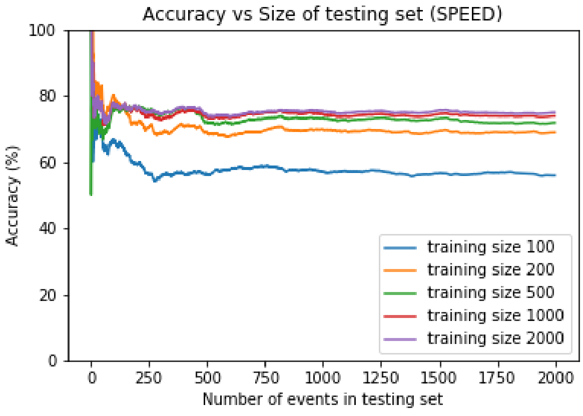

Figure 8. SPEED prediction accuracy vs. the size of testing set.

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.

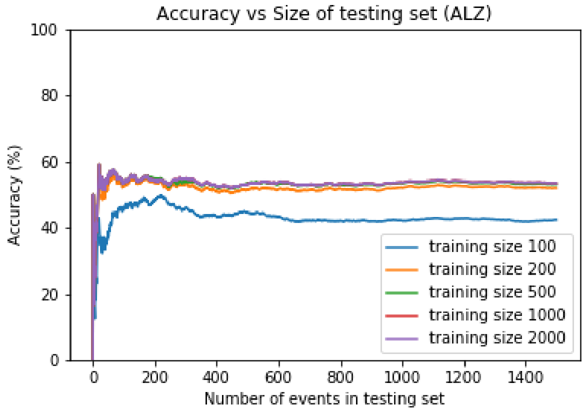

Figure 9. ALZ prediction accuracy vs. the size of testing set.

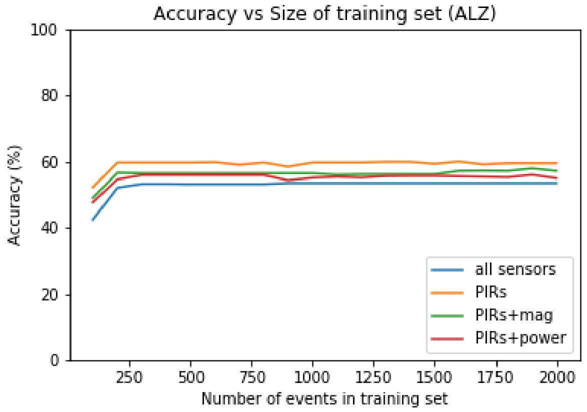

Figure 11. ALZ prediction accuracy vs. the size of training set for different sets of sensors.

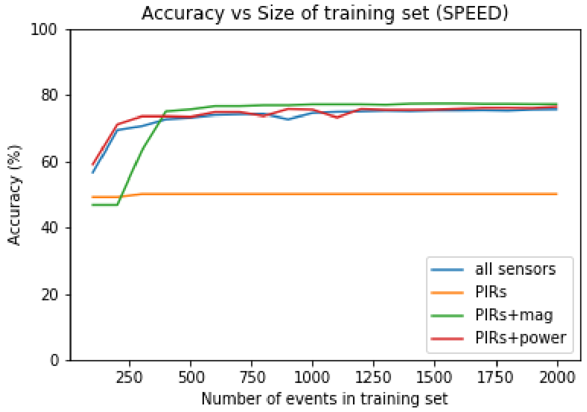

Figure 10. SPEED prediction accuracy vs. the size of training set for different sets of sensors.

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.