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

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 preliminary results on sensor data prediction in a smart home environment with a limited number of binary sensors. The data has been collected from a real home with one resident over a period of 17 weeks. We apply Recurrent Neural Network with Long Short-Term Memory to a text sequence derived from the sensors' events to predict the next event in a sequence. We compare our system's characteristics and results to a baseline method and to similar work in the area. Our implementation achieved a peak accuracy of 69% for a set with 13 sensors in total - motion, magnetic and power sensors - and 75% for five motion sensors.**

***Keywords*—smart home, sensor data prediction, binary sensors, recurrent neural network**

I. Introduction

The Assisted Living project is an interdisciplinary project with experts in the field of nursing and occupational therapy, ethics, and technology [1]. The aim is to develop assisted living technology (ALT) to support older adults with mild cognitive impairment or dementia (MCI/D) to live a safe and independent life at home. MCI and dementia consist of a cognitive decline that can affect attention, concentration, memory, comprehension, reasoning, and problem solving [2]. In order to support older adults with MCI/D in their everyday life, several functions in smart home environments have been investigated in the past years. This includes assisting functions such as prompting with reminders or encouragement, diagnosis tools, as well as prediction, anticipation and prevention of hazardous situations. The majority of these functions requires reliable activity recognition and prediction algorithms to work properly.

Even though several algorithms have been reported in the literature for activity recognition and prediction, to the extent of our knowledge such prediction algorithms have not yet been proven to be accurate enough to be implemented in real homes. In addition, there is no complete study comparing the different available algorithms, testing different configurations for input of data, or providing guidelines as to which application areas they are best suited for. In a previous work that is currently in press [3], we applied two state-of-the-art probabilistic algorithms on binary sensor data acquired from a real home and carried out sequence prediction. We compared their performance in several configurations and provided guidelines on applications. In this paper, we report results for the same task applying recurrent neural network (RNN) with long short-term memory (LSTM).

II. Related Work

There has been a great deal of research on data prediction algorithms in the past years [4]. Such algorithms can be applied in a large range of domains, including sensor event and activity prediction for several functionalities in smart homes. Examples include improved automation functions (e.g. turn on the heater sufficient time prior to the person arriving at home); prompting systems (e.g. prompt a person to execute a necessary activity in case it was not performed) [5]; or anomaly detection in certain behavior patterns (e.g. movement, everyday habits, etc.) and therefore indicate the onset or progress of a condition [6].

Among others, probabilistic methods have been investigated for sequential data prediction. The Active LeZi (ALZ) was applied on the dataset from Mavlab testbed, which comprises 50 binary sensors, and achieved a peak accuracy of 47% [7]. The SPEED (sequence prediction via enhanced episode discovery) algorithm 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 [8]. Both algorithms translate the data of the binary sensors to a sequence of letters and build a tree based on the past observations. They are based on Markov models, hence the most probable next event can be predicted based on the current state, by using the Prediction by Partial Matching algorithm (PPM) [9].

Besides probabilistic algorithms, neural networks have also been used for sensor event prediction, typically recurrent neural networks. In [10] the models Echo State Network (ESN), Back Propagation Through Time (BPTT), and Real Time Recurrent Learning (RTRL) were compared and the ESN achieved a root square mean error (RMSE) of 0.06 in a fourteen-day dataset from six binary sensors (four motion and two magnetic). In these networks, the number of input and output values corresponds to the amount of sensors in the dataset, and each can assume value '0' or '1' for being "off" or "on" at a certain time slot. The prediction was made for the next six hours. In a later work, a Non-linear Autoregressive Network (NARX) was compared to an Elman network, with both using as input and output the start time and end time of a sensor's activation [11]. In this case, each sensor had its own network that was trained and tested on a twenty-day dataset from the same six binary sensors. The NARX performed better when predicting only the

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

next step, with a RMSE ranging from 0.06 to 0.09, depending on the sensor.

A similar study was conducted for an office environment comprising sixteen rooms. The dataset was collected via an app the employees had in a personal data assistant (PDA), where they registered manually whenever they arrived and left a certain room [12]. Both the Elman network and a multilayer perceptron network to predict the next room a person would go to in the office were applied. There were four participants in the study and the Elman network presented the best results, with a maximum accuracy of 91% for one participant and a minimum of 70% accuracy for another. In this study, each user was modelled by a separate network. The input corresponded to the last two rooms codified in four bits each since there were 16 rooms in total. The output was the next room to be occupied and was codified in the same manner. This work compared these neural networks with 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].

In a previous work, we applied both SPEED and ALZ to our data from one real smart-home and reached an accuracy of 75% and 53%, respectively [3]. In the current work, we use neural networks, since they have been shown in the literature to achieve good performance for this task. Our dataset contains events from 13 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], however, the sensor testbed in that work comprised primarily motion sensors.

III. 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 (Fig. 1).

The purpose of the trial and the deployed sensor system have been decided in close collaboration with the residents [1]. 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 [1]. We chose to include sensors that indicate 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). 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, 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 an apartment that comprises 15 sensors 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;
- 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). An example scenario is shown in Table I: the resident goes to the living room (sensor ID 4), turns on the TV (sensor ID 23), goes to the kitchen (sensor ID 5) and makes coffee (on and off sensor ID 20) and goes back to the living room.

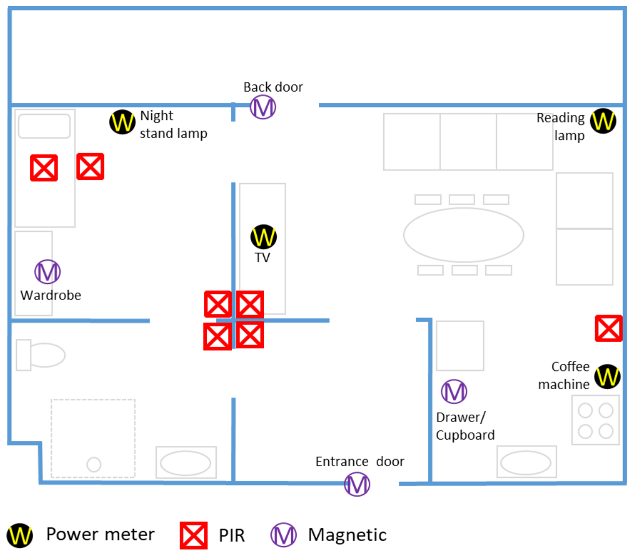

Fig. 1. Sensors system in the field trial apartment

Table I. Binary Sensors Data for an Example Scenario

Timestamp	*Sensor ID*	*Sensor message*
2018-03-25 14:35:55	4	1
2018-03-25 14:37:46	23	1
2018-03-25 14:38:13	5	1
2018-03-25 14:39:00	20	1
2018-03-25 14:41:02	20	0
2018-03-25 14:41:58	4	1

IV. Sensor Data Prediction

We apply LSTM network to predict the next sensor event in a sequence. In addition, we implement a baseline for comparison.

A. *Baseline*

The baseline consists of a table with the probability of each sensor being the next activated sensor depending on the preceding sensor(s). The memory length, i.e. the number of previous events the prediction is based on, can vary. The predicted next event (next activated sensor) is the one with the highest probability of being activated right after the last sensor(s) in the sequence.

An example from our data is shown in Table II. The table considers a memory length of one event, i.e. the probability prediction is based on the previous event only. For instance, from a training dataset we computed that after the bedroom motion sensor was activated, the most probable next sensor event was the bathroom motion sensor in 53% of the samples. Based on that, the event predicted after the bedroom would therefore be the bathroom motion sensor. Similarly, if the last sensor in a sequence were the living-room motion sensor, the next predicted event would be the motion sensor in the kitchen.

B. *Recurrent Neural Network*

RNNs have been extensively applied to sequence prediction tasks because of the property of keeping an internal memory, which is a great advantage for inputs that are sequential in time. Examples of applications can be text generation [17], speech recognition [18] and pattern recognition in music [19]. The LSTM [20] is an RNN architecture designed to be better at storing and accessing information than the standard RNN [21].

1) *LSTM Network Configuration*

The input and output of the LSTM network are the data from the sensors translated to a sequence of letters, as it was performed by the ALZ algorithm [7]. Each sensor is represented by a letter, hence an event sequence becomes a text sequence. For instance, the example scenario data in Table I would be represented as “abcda”, where ‘a’, ‘b, ‘c’ and ‘d’ represent sensors ID 4, 23, 5 and 20.

Table II. Prediction Table for set of 5 motion sensors for the Baseline Method

Last Sensor	**Next Predicted Sensor**				
	Bathroom	***Bedroom***	***Living room***	***Kitchen***	***Entrance***
Bathroom	0	0.926	0.051	0.007	0.017
Living room	0.018	0.398	0	0.435	0.149
Bedroom	0.527	0	0.454	0.008	0.011
Kitchen	0.005	0.062	0.885	0	0.048
Entré	0.054	0.210	0.630	0.107	0

The LSTM network is configured as a text generation network. The input is a certain number of sensor events – equal to the memory length – and the output is the predicted next event in the sequence (Fig. 2).

2) *Implementation*

A stateless LSTM network model was implemented in Python 3 using Keras open source library for neural networks. A number of parameters were tuned based on the model learning:

- Memory length (number of events) of the network, i.e. number of events to predict the next one from;
- Number of hidden layers and number of neurons in each;
- Batch size: number of samples used for training each iteration of the epoch;
- Learning rate: parameter used in the optimization, which had 0.01 as the optimal for all our trained models;
- Optimization function, loss function and activation functions in the hidden layer and output layer: Adam, categorical cross-entropy, hyperbolic tangent and softmax, respectively.

We have calculated results for two different sets of sensors: 13 sensors and only 5 motion sensors (PIRs). Two motion sensors from the original set – motion by and over the bed – were not used in the modeling due to unreliable data. In addition, we have obtained results for three data collection durations: two, thirteen, and seventeen weeks.

The values of each parameter for our best prediction accuracy are shown in Table III.

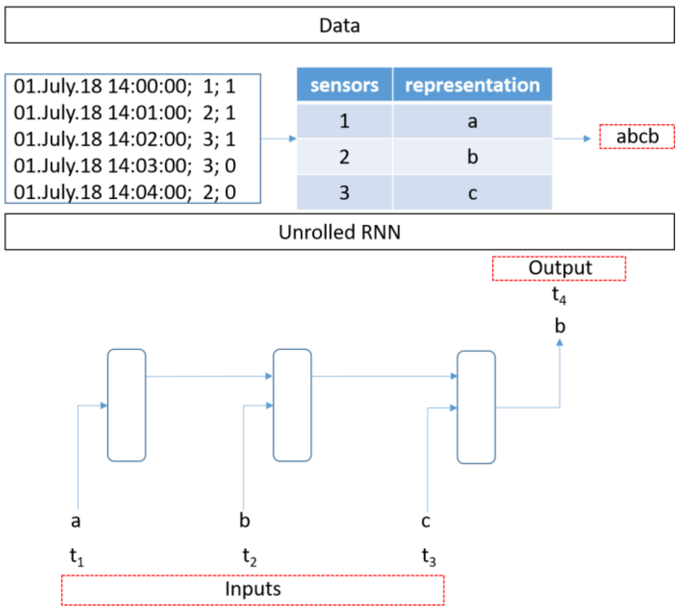

Fig. 2. Configuration of input and output in our LSTM network implementation with ALZ data

TABLE III. PARAMETER VALUES IN OUR LSMT NETWORK IMPLEMENTATION

Parameter	Values in our best model
Memory length	6
Number of hidden layers	1
Number of neuron in hidden layer	64
Batch size	128
Learning rate	0.01
Epochs	500

V. RESULTS AND DISCUSSIONS

For the results presented in this section, the data are split into training (80%) and testing (20%) sets for both the baseline and the LSTM network model. In addition, the results of the LSTM network are acquired by running the model three times with different compositions of training/testing sets from the same dataset.

Table IV shows the number of events for each investigated configuration. The sensors system corresponds to one of the apartments in our field trial, as described in section III – except the motion sensors by and over the bed. We can notice that motion sensors generate a much larger number of events compared to power and magnetic sensors. Therefore, we have balanced our dataset using weights for each sensor. These are computed using the "compute_class_weight" function of the Scikit-learn open source library. The weight is the total number of samples divided by the number of occurrences of the class. Hence, it is used for penalizing mistakes in samples - the higher the weight, the larger the penalty to the error of the corresponding sensor.

Table V presents the accuracy obtained with the baseline for each combination of set of sensors and number of weeks, for a range of memory lengths (number of preceding events).

The baseline achieves a best accuracy of 67% for the set with only motion sensors (5 sensors). This is intuitively understandable since here there are few options of next event to predict and events are not intertwined, meaning that one sensor needs to go off before the next can be activated.

We notice that with the baseline the optimal number of sensor events to predict from is three. Intuitively, one could think that this behavior is not surprising since the number of sensors is rather small and the occurrence of each sensor may thus be associated to the occurrence of very few other sensors. Furthermore, increasing the size of the data does not have a significant effect on the accuracy of the baseline, except when the memory length is relatively long. This may be due to the fact that the longer patterns of five-six events are less frequent and require larger data sizes in order to manifest themselves and affect the overall accuracy. Nevertheless, a memory length of three events still leads to the highest accuracy.

It is evident from Table II that there is noise in the data. For instance, although it is not possible to go from the entrance hall directly to the kitchen, some of the data represent this (1%). Binary sensors can show faulty activation e.g. erroneous activation of motion sensors by sunlight, bouncing of contact sensors, or switch-off delays of motion sensors [22]. This is not such a big a problem for the baseline method because of the relatively small amount of the noisy data compared to the number of correct events. Therefore, the baseline table of probabilities is hardly affected by the noisy data, as the event with the highest probability, and hence the predicted event, corresponds to possible rather than erroneous occurrences. For the LSTM network on the other hand, such erroneous events in the training set may affect the performance a great deal as the network learns erroneous patterns. We have therefore carried out a data cleaning process where we deleted samples that contain events that are not possible from our dataset. This cleaning process had a significant effect on the attained accuracy.

The LSTM network prediction accuracy and RMSE values vs. the size of the training dataset are shown in Fig. 3 and Fig. 4, respectively. These results were obtained with a network with one hidden-layer, comprising 64 neurons, and a memory length of 6 events, as described in section IV.

We can notice that the accuracy increases steadily for larger training dataset. The peak accuracy is 69% when about 5000 events are used for training, and the mean accuracy is 67%. All models are tested with the same test set, which contains 2000 events. With that amount of data, we cannot see that the network stabilizes. More data is evidently needed for that, and the accuracy may then also improve.

Fig. 5 and 6 shows the prediction accuracy of the LSTM network for several memory lengths. The maximum average accuracy is achieved with six events. The accuracy increases steadily up to this point, when it appears to decrease slowly. Though marginal, this shows that the LSTM network is better able to find the relation between the input features than the baseline is. Also note that the baseline achieved its best performance for a memory length of three events.

TABLE IV. NUMBER OF EVENTS PER SIZE OF DATA AND PER SET OF SENSORS

Set of sensors (number of sensors)	*Number of weeks*	*Number of events*
All (13)	2	2712
	13	12520
	17	16050
PIRs (5)	2	2084
	13	9820
	17	12631

TABLE V. ACCURACY OF BASELINE METHOD

Set of sensors	Number of weeks	Accuracy per memory length (# events)					
		1	*2*	*3*	*4*	*5*	*6*
All	2	0.490	0.576	0.517	0.450	0.376	0.329
	13	0.521	0.565	**0.614**	0.600	0.552	0.501
	17	0.504	0.546	0.583	0.551	0.512	0.470
PIRs	2	0.608	0.644	**0.670**	0.616	0.612	0.577
	13	0.607	0.634	0.626	0.627	0.618	0.592
	17	0.538	0.623	0.624	0.620	0.617	0.600

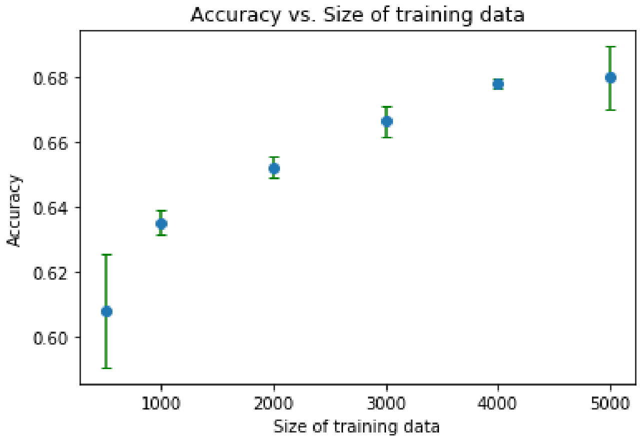

Fig. 3. Accuracy vs. number of events for the LSTM network with text sequence

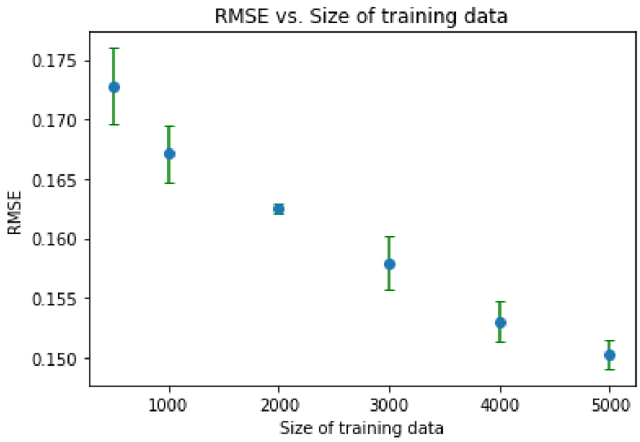

Fig. 4. RMSE vs. number of events for the LSTM network with text sequence

Finally, we analyse the results in more detail in Table VI with both accuracy and RMSE values for training and testing sets, and peak accuracy from the testing set. It is evident that the accuracy can vary dramatically depending on the set of sensors. Having half of the sensors, only motion sensors that cannot have intertwined events, leads to the highest prediction accuracy with a mean of 74%, while with the 13 sensors the mean accuracy is 67%. For the investigated sizes of data, the accuracy increases by about 8-10% when only motion sensors are used as compared to the set with all the sensors.

When the size of the training dataset is insufficient, the model tends to over-fit, and therefore achieves lower test accuracy than training accuracy. This is the case for two weeks of data in both sets of sensors. By increasing the dataset to 13 weeks in the case of PIR sensors only, the difference between training and test accuracy decreases, and with 17 weeks of data the training and testing accuracy are almost equal. This indicates that the network has had sufficient training data to achieve maximum accuracy. In the case of 13 sensors, the difference between training and test accuracy does decrease with increasing amount of data. However, with 17 weeks of data the training accuracy is still considerably better than the test accuracy indicating that more data is required to improve accuracy.

In summary, for a limited amount of data the baseline achieves better accuracy than the LSTM. An accuracy of 67% is achieved with 2 weeks of data from the PIRs only, and an accuracy of 61% is achieved for 13 weeks of data from all sensors. However, for a sufficient amount of data the LSTM achieves better accuracy than the baseline. 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 PIRs. In addition, the LSTM may improve further with additional data while this is not the case of the baseline.

In our previous paper in press [3] we have presented results for sequence prediction using probabilistic methods. The ALZ algorithm reached its maximum accuracy of 53% with just 250 events, from the 2 weeks of data, for all sensors. Training the algorithm with more events did not improve the accuracy. The SPEED algorithm reached 75% accuracy with 2 weeks of data and all sensors. Note that the input in this case includes the 'off' events as well. Since our system does not have many sensors that can have their 'on' and 'off' events intertwined, the 'off' events are easier to predict since they often happen right after the 'on' events. This is confirmed by our results [3].

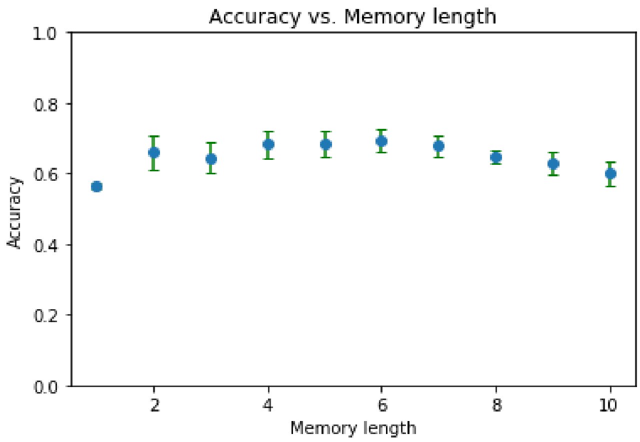

Fig. 5. Accuracy vs. memory length using LSTM network for 17 weeks of data

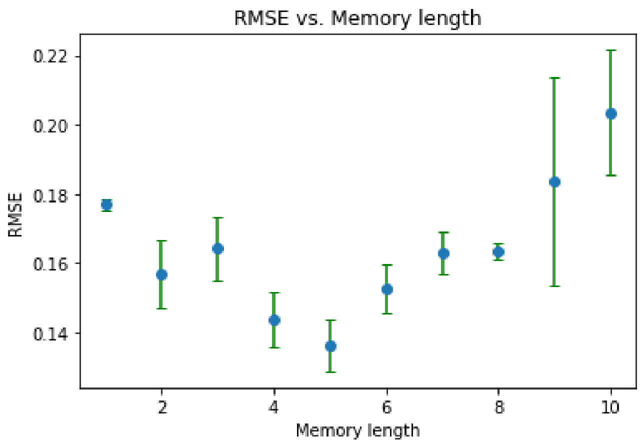

Fig. 6. RMSE vs. memory length using LSTM network for 17 weeks of data

TABLE VI. RESULTS LSTM NETWORK

Set of sensors	Number of weeks	Accuracy			RMSE	
		Train	*Test*	*Test peak*	*Train*	*Test*
All	2	0.652 ± 0.007	0.598 ± 0.027	0.628	0.161 ± 0.007	0.171 ± 0.002
	13	0.691 ± 0.006	0.661 ± 0.014	0.679	0.148 ± 0.005	0.157 ± 0.001
	17	0.699 ± 0.006	0.673 ± 0.016	0.693	0.144 ± 0.006	0.154 ± 0.002
PIRs	2	0.743 ± 0.005	0.699 ± 0.009	0.711	0.192 ± 0.005	0.204 ± 0.003
	13	0.741 ± 0.000	0.735 ± 0.008	0.743	0.195 ± 0.000	0.197 ± 0.003
	17	0.741 ± 0.004	0.742 ± 0.012	0.751	0.197 ± 0.004	0.196 ± 0.004

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 functionalities. 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 [7].

In this paper, we have presented results on sensor event prediction based on data from a real home collected using only 15 binary sensors, over a period of seventeen weeks. We have applied LSTM network and obtained a peak prediction accuracy of 69% and 75% for sets of 13 and 5 sensors respectively. To the extent of our knowledge, this is the first time this type of neural network has been used in the prediction of the next sensors on a dataset obtained from a real home. We have compared its prediction accuracy with a baseline, and investigated both methods with respect to a number of parameters such as memory length, size of the dataset, and the number of sensors. The accuracy achieved by the baseline was 58% for all the sensors over the 17 weeks, with a memory length of three events. The LSTM network achieved the best mean accuracy of 68% over the same period with a memory length of six events. This shows that such networks are more effective in learning patterns and finding temporal relations among features.

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 [10], [23].

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 Krunic 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. D. Casagrande, T. Holthe, A. Lund, L. Halvorsrud, D. Karterud, A. Flakke-Johannessen, H. Lovett, S. K. Mørk, J. Strøm-Gundersen, E. Thorstensen, R. Norvoll, R. T. Meulen, M.-R. Kennedy, R. J. 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 - Proc. 3rd Int. Conf. Inf. Commun. Technol. Ageing Well 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.," J. Intern. Med., vol. 256, no. 3, pp. 240–246, 2004.

[3] 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 press, to be published in Linköping Electronic Conference Proceedings of the 59th International Conference of Scandinavian Simulation Society, SIMS 2018.

[4] 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 Things J., vol. 4, no. 3, pp. 636–644, 2017.

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

[6] 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," Artif. Intell. Med., vol. 67, pp. 57–74, 2016.

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

[8] M. R. Alam, M. B. I. Reaz, and M. A. Mohd Ali, "SPEED: An inhabitant activity prediction algorithm for smart homes," IEEE Trans. Syst. Man, Cybern. Part ASystems Humans, 2012.

[9] J. G. Cleary and I. H. Witten, "Data Compression Using Adaptive Coding and Partial String Matching," IEEE Trans. Commun., 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," J. Ambient Intell. Humaniz. Comput., vol. 3, no. 3, pp. 205–218, 2012.

[11] S. Mahmoud, A. Lotfi, and C. Langensiepen, "Behavioural pattern identification and prediction in intelligent environments," Appl. Soft Comput. J., 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, 2004, vol. 114, no. 4, pp. 1–12.

[13] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Next Location Prediction Within a Smart Office Building," Cogn. Sci. Res. Pap. Unversity 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 Mob. Comput., 2016.

[16] M. T. Moutacalli, A. Bouzouane, and B. Bouchard, “Sensors activation time predictions in smart home,” Proc. 8th ACM Int. Conf. PErvasive Technol. Relat. to Assist. Environ., p. 37, 2015.

[17] J. Martens, “Generating Text with Recurrent Neural Networks,” Neural Networks, vol. 131, no. 1, pp. 1017–1024, 2011.

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

[19] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, “Learning precise timing with LSTM recurrent networks,” J. Mach. Learn. Res., vol. 3, no. 1, pp. 115–143, 2003.

[20] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[21] A. Graves, “Generating Sequences With Recurrent Neural Networks,” arXiv, 2014.

[22] 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.

[23] 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,” pp. 262–270, 2015.