

Representing Temporal Dependencies in Smart Home Activity Recognition for Health Monitoring

Glenn Forbes*, Stewart Massie*, Susan Craw*, Lucy Fraser†, Graeme Hamilton†

*School of Computing Science & Digital Media

Robert Gordon University, Aberdeen, UK

{g.r.forbes, s.massie, s.craw}@rgu.ac.uk

www.rgu.ac.uk/dmstaff/lastname-firstname

†Albyn Housing Society Ltd, Invergordon, UK

Abstract—Long term health conditions, such as fall risk, are traditionally diagnosed through testing performed in hospital environments. Smart Homes offer the opportunity to perform continuous, long-term behavioural and vitals monitoring of residents, which may be employed to aid diagnosis and management of chronic conditions without placing additional strain on health services. A profile of the resident’s behaviour can be produced from sensor data, and then compared over time. Activity Recognition is a primary challenge for profile generation, however many of the approaches adopted fail to take full advantage of the inherent temporal dependencies that exist in the activities taking place. Long Short Term Memory (LSTM) is a form of recurrent neural network that uses previously learned examples to inform classification decisions. In this paper we present a variety of approaches to human activity recognition using LSTMs which consider the temporal dependencies present in the sensor data in order to produce richer representations and improved classification accuracy. The LSTM approaches are compared to the performance of a selection of baseline classification algorithms on several real world datasets. In general, it was found that accuracy in LSTMs improved as additional temporal information was presented to the classifier.

Keywords—Human Activity Recognition, Temporal Dependency, Smart Homes, Sensors, Ambient Assisted Living.

I. INTRODUCTION

Smart Home technology is becoming increasingly popular but the focus to date has largely been on security and automation. However, there is real potential to employ smart home technology for health monitoring and management. Research has shown that there is a strong relationship between the activities and behaviours that a person can undertake in their day-to-day lives and their future health and predicted lifespan [1]. The opportunity is to capture information on Smart Home residents by using sensors to monitor a their activities, e.g. room transitions, and behaviours, e.g. food preparation. A key advantage is that data is collected in the persons natural environment rather than in a more artificial laboratory setting. Daily or weekly profiles of a resident’s activities and behaviours can be captured, allowing trends in the data to be identified as a resident’s profile changes over time; and a comparison of profiles to benchmark examples that are known to indicate potential health concerns.

There are numerous challenges in building a Smart Home health monitoring system, including the type of sensors that

should be employed; the measurement of similarity between profiles and benchmarks; and the setting of thresholds to initiate interventions. However, the key challenge addressed in this work is how to effectively identify or classify a resident’s activity given the data from the sensors. In contrast to many Human Activity Recognition (HAR) tasks, in which the data is polled at regular time-spans (such as accelerometer data from wearables), Smart House data often contains both polled data (e.g. electricity usage) and more irregular event activation data, from movement sensors, etc. Algorithms and representations that effectively support the different feature types are required. Two broad approaches are employed: Rule-based systems and Machine Learning (ML) approaches. With rule-based systems event sequences are manually mapped to activities without the need for labelled examples. Whereas with ML, examples are annotated with the activity label and used to train a classification model. Both approaches have advantages and limitations but in this work we explore ML solutions that offer greater flexibility to transfer to new activities and home designs.

Sequences are important in activity recognition. An issue in making use of sequential sensor activation data is to develop effective approaches for harnessing temporal dependencies. The aim of this work is to identify and investigate the importance of different types of temporal dependencies and propose specific representations and algorithms which can take advantage of these relationships.

In this paper, we explore the use of everyday, low-cost ambient sensors installed in Smart Homes for the creation of resident profiles which can then be used for assessing similarity to previous profiles and to benchmark cases. Specifically, we identify and discuss the main challenges in building effective representations and applying ML and Deep Learning algorithms to classify a resident’s activity that can then be assembled into their profiles. A key focus is on how to effectively take advantage of different forms of temporal relationships within the case representation. Several alternative approaches to classifying activities from low level, raw data inputs are investigated and effective solutions which leverage temporal relationships are identified. The main contribution of this work is the development of a novel temporal

dependency-aware ML approach for activity recognition from event sequence sensor data.

II. RELATED WORK

Assessments of deteriorating health condition, such as mobility, are traditionally performed under laboratory conditions [2], [3]. For example, Vestergaard found a link between performance in the 400-Meter Walking Test and remaining life span. However, these methods are slightly artificial in that the assessments are outwith the normal environment and can be treated as a one-off performance rather than a measure of day-to-day capabilities. In an alternative approach, Stone found long term monitoring in the home environments can lead to effective diagnosis while being less intrusive and potentially more accessible than hospital testing [4]. IoT sensor installations using small ubiquitous sensors, such as infrared passive sensors [5] or binary magnetic switches, can capture behavioural information expressed by residents in Smart Home environments. The HAR challenge here is to infer behaviour effectively based on either a set or sequence of sensor activations.

Rule-based approaches to classifying a resident’s behaviours are well established and suited to basic sensor networks containing few binary sensors [5]. Regular behaviours are easily observable in shallow sensor networks, allowing them to be used for basic health monitoring. While rules are time consuming to generate manually, they work well in static configurations that rarely change and have only a limited number of behaviours that need to be identified.

Specifically engineering a rule-based classifier for a unique environment can be costly and time-consuming [6], and raises several challenges. Theoretically a rule-based system can cover each possible combination of events in a sensor network, however it is often infeasible to write accurate rules that achieve full coverage. Multiple contradictory rules may be activated and priorities need to be established. In addition maintenance can be difficult as small changes can have knock on effect to other rules. While these systems are technically extensible, the time investment required to maintain larger scale systems becomes unmanageable. Consideration of all stages of the network is required when adjusting the system as changes may ripple out through seemingly unrelated functionality.

There has been a recent focus on deep learning in HAR. A key challenge for ML approaches is the acquisition of labelled data; it is expensive to manually annotate sensor data with activity labels, and variations in the labelling decisions of ambiguous sensor sequences may affect the overall quality of the dataset [7]. Smart home data presents its own challenges, as the data is often in the form of an irregular sequence of events rather than a time sequence, more typically of polled sensors with wearables. Only a few public datasets for activity recognition from Smart Home sensor data have been made available by MIT, CASAS and Van Kasteren [8], [9], [10].

As a form of Recurrent Neural Network (RNN), LSTMs offer the potential to capture the temporal relations encoded

in the sequence of features and samples provided to the classifier [11]. In this paper the performance of a range of LSTM designs are investigated with various temporally-aware representations of sensor activation sequences.

The presence and importance of sequential events in the data from Smart Homes gives rise to opportunities to improve activity recognition by capturing the sequences as part of the representation. This may be explicitly by employing a feature engineering approach, however, manually chosen features using heuristic processes may be outperformed by modern deep learning research [11]. Ordóñez found that identifying the temporal dependencies that exist within human activity expressions was key to improving the performance of a HAR classifier. The experiments performed in their work made use of wearable sensor data, though similar relationships with no derived heuristics may exist between binary sensor activations [8]. In Bourobou’s [12] work, HAR classification was performed on a simulated binary sensor dataset factoring in the temporal dependency between subsequent, previous and overlapping activities. These temporal dependencies were identified separately and used to simulate data produced in real home environments. The relationships between temporally relevant activities were analysed using the calculated “importance degree” of each activity. While this approach yielded noteworthy results, additional research is still necessary to assess the effectiveness of alternative approaches to enriching representations by manually engineering the influence of temporal dependence.

III. FITSENSE

The motivation for this work is to improve the flexibility and transferability of the rule-based solution developed for FITsense. FitHomes is a project lead by Albyn Housing Society Ltd aimed at encouraging continued independent living for people with mobility-related health conditions. 16 purpose-built homes have been constructed at Dalmore near Invergordon in the UK; and an additional development of a further 10 homes is currently having sensors installed. Discussion with residents highlighted that they wanted an unobtrusive sensor system, specifically excluding video and wearables as too invasive. Therefore, the houses have been fitted with ambient infrared motion sensors for health data collection.

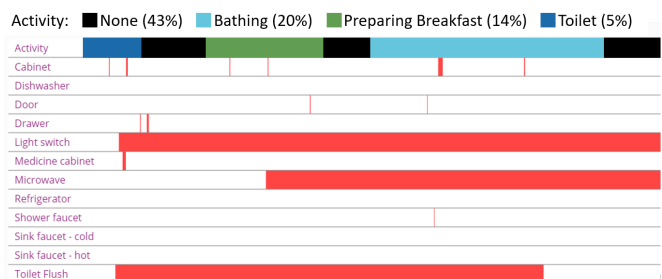


Fig. 1. Timeline of sensor activations and assigned activity labels.

FITsense is a complementary project aimed at identifying potential risk of falling from FitHome sensor data. A rule-

based HAR approach is currently employed that uses data from sensors to classify the activity being undertaken by the resident at a given point in time. For example, in Figure 1 the sequential activation of sensors is used alongside their locations to classify sequences under 4 activity labels. Each smart house contains sensor equipment which can be used to capture key factors which may be indicative of an increased risk of falling. Selected factors include frequency of toilet use, total sleeptime, number of sleeptime disturbances and others decided to have relevance to fall risk. These factors were chosen through examination of current research and discussion with healthcare professionals.

As new FitHome developments are completed, challenges arise in adapting the FITsense system to new floor plan layouts. Homes in the Dalmore site all follow the same reference layout which allows for the same rules to be used in each house, with some minor refinements to meet personal circumstances. New housing developments have different layouts, and in addition, it is also planned to retrofit FITsense into existing homes with varied layouts. In order to expand the FITsense system for use across all house layouts, the current rule-based solution must be changed. It is not practical to develop and maintain an effective rule-base for all layouts and a more flexible approach is required.

The limitations of rule-based systems mostly stem from the time investment required to develop, update and maintain them. Conversely, the restriction on the development of an ML system is the availability of labelled training data. Labelled data can be difficult to acquire, with some public smart home datasets employing an observer to manually annotate sensor activation sequences [7]. However, a simple and ubiquitous solution is for residents to digitally label sensor data sequences over a relatively short period of time to produce higher quality training data. This is the approach we adopt.

A. Acquisition of Labelled Data

In order to obtain labelled data, a timeline document was circulated to prearranged residents. This timeline broke the day down into 15 minute intervals, with an option to select which room(s) were occupied during the period and which of the tracked ADLs the resident may have been engaged in. Additional space was also left to allow residents to add information they feel may be useful to activity and pattern recognition. The given timeline spanned 7am to 1am with the assumption that residents would be asleep for the remainder of the 24 hour period. The form was designed to ensure that the data collected would not be entirely curated around several rule-based assumptions that had been formed during development of the FITsense system. For instance, residents can select that they performed an ADL spanning across two rooms which the current system does not consider.

Of the evaluation timelines completed by residents, two were selected due to the high quality of data outputted by those houses at the time of evaluation. The key observation was that the event-state system used for FITsense labelling performed well. Resident locations and ADL expressions

matched up well with the established event-states in FITsense data. In most cases where a mismatch occurred a sensor functionality issue existed, or the resident documented unexpected behaviour.

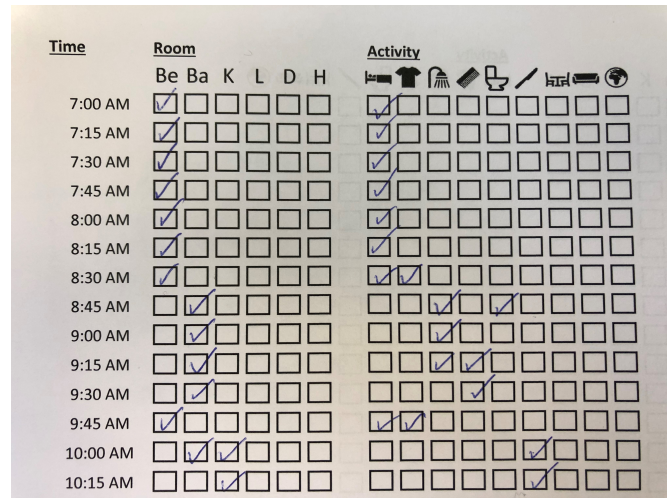


Fig. 2. Example of Evaluation Timeline document.

While the labelled data provided in the evaluation timelines has merit, there are too few examples for use in deep learning. Instead of collecting additional data, an additional 13 surrounding days of unlabelled data was retrieved from the homes. This data was then split using the FITsense windowing technique. By viewing the sensor activation sequence associated with each split, the ADL could be identified considering the residents’ documented behaviour as seen in their returned timeline. For example, in Figure 2, it can be observed that the resident moves from the bedroom into the bathroom after waking up. During this visible period, the resident has transitioned from sleeping to dressing, and then to toilet, shower and grooming. In unlabelled sensor data, this can be viewed as activations of the motion sensors in the bedroom (with sustained movement for a short period) and the bathroom. This sequence then becomes useful for manual annotation, as it allows for long term prediction of regular behaviours. The recurring activity patterns observed in Figure 2 can be used to assist manually annotating the data with ADL labels, producing the fitsense 1 and fitsense 2 datasets.

The FITsense system performs well in real scenarios where houses contained a full complement of functional, sensors. Cogent sequences of ADL expressions can be observed, with few noticeable issues in the event-state transitions. When observing differences between the rule-based labels on the same period of FITsense data, it is notable that the “grooming” ADL was rarely identified. To a manual labeller, “grooming” ADLs can be easily observed after a “toileting” instance, however it is notable that the temporal dependence exhibited here is the key identifier. Through sensor activations alone, it could not be expected that a classifier would be able to correctly identify the dependent ADL. In order to improve the performance of FITsense ADL classification, additional

sensors would need to be added to the configuration in order to allow more ADLs to be correctly telegraphed. Alternatively, the temporal dependencies which exist in the data could be encoded as part of an activity representation for a ML classifier.

IV. IDENTIFYING ACTIVITIES FROM SENSOR DATA

Activities of Daily Living (ADLs) are regular behaviours which a person performs in day to day life. This can include getting out of bed, making a meal or grooming oneself. The specificity of an ADL can vary, with the classification of more complex ADLs (e.g. chopping vegetables) usually requiring additional sensor data. There is a trade-off between the number and cost of the sensor network and the specificity of ADLs that can be identified. On a wider scale, HAR refers to the general task of identifying a person's activity expression from data, whereas ADLs usually refer to regular behaviours in a context around the home. HAR are typically discerned on low level activities, such as standing up, sitting down or running; while ADLs refer to a more general form of behaviour. However, similar approaches can be applied to exploit patterns exhibited in data for both HAR and ADL classification.

The activity recognition task is typically broken down into two separate parts: windowing and ADL classification. Windowing involves separating the continuous sequence of sensor activations into smaller sequences likely to contain a single ADL expressions, while ADL classification is the process of assigning activity labels to each windowed sequences of sensor activations. We currently employ a rule-based approach for windowing and classification. As a first step the plan is to keep the windowing approach but change to a more flexible ML approach for classifying.

A. Rule-based Windowing

A popular approach to split data into windows is to use a sliding window [13]. This has been shown to be effective in real sensor datasets, however there is the potential for windows to overlap and not correctly represent realistic activity behaviours. Our approach to the windowing of long sequences of sensor activations stems from the event-state system used in FITsense, which tracks the resident's movements and activity through the home. This is an additional filter layer between binary sensor activations and the classification task. Location labels associated with event-states can be used to split sequences of sensor activations as the resident transitions from one room to another. Due to the event-state system requiring states to depend on previous locations and behaviour, this rule-based classifier effectively makes use of the implicit temporal dependency in ADLs that occur sequentially. While this provides the benefit of reducing window overlap, the specificity of ADLs which can be captured is reduced.

B. Rule-Based classification

ADL classification in the initial FITsense system lacks granularity but can be effective in real scenarios. The system

could theoretically be expanded to capture more complex ADLs but the work required to write and maintain robust rules which consider the full state of the system is substantial and unmanageable as the system scales. Each room in the home is assigned a set of ADLs which could potentially occur. Within these ADLs are specifications which qualify a sequence as an expression of this ADL. These specifications usually confirm the existence of specific event-state transitions and the total length of an ADL. After larger sequences have been windowed using event-state room transitions, each sequence is checked against the available specifications. The key assumptions behind FITsense ADLs are: ADLs are limited to one per window, and ADLs can only be performed in one room at a time. The master list of ADLs is ordered by importance, to ensure important ADLs are misclassified less often than unimportant ones. In the event more than one ADL specification is matched to a sequence, the specification with the highest importance rating is assigned. The system functions well on the single site with identical home layouts. However, the activity recognition rules rely on manual observation of the specific environment.

The approach has limitations and in particular will not transfer easily to future developments with different sensor configurations or home layouts. This highlights the main issue with rule-based classifiers: the rules must be designed specifically for the environments they operate in. The rules must be tested to ensure edge cases and unexpected sensor behaviours are correctly handled, which is a time consuming process. Rules can also become dependent on 100% sensor uptime. While ambient binary sensors can be cheap, they require upkeep to ensure the general quality of the data being produced is maintained. Additionally, the produced rules are not flexible to sensor configuration changes and other unforeseen events. If a reference house design is not available, rule generation is required for each unique house layout.

C. ML

ML is the standard approach employed for HAR. Advances in deep learning, recurrent and convolutional neural networks have directed the field for classification of complex sensor data, such as accelerometers and gyroscopes [14]. However, more basic sensor configurations, such as the binary sensor networks seen in smart home environments, have been well-served by Naive Bayes, Decision Trees and other established classification algorithms [8], [12].

While these traditionally established ML methods can perform well in ADL classification tasks, they do not make use of the implicit temporal dependencies found in sensor data. The temporal relationships which exist in sensor data have relevance in ADL classification, as they offer an opportunity to extract additional useful knowledge from sensor activation data. Recurrent Neural Networking methods, specifically Long Short Term Memory (LSTMs), can make use of the temporal knowledge encoded in the sequences of sensor activations and ADLs which occur in training data.

We propose using a hybrid method to improve the temporal awareness of an ML-based ADL classifier. First, we plan to enrich sequential sensor data representations by adding relative timestamps between subsequent activations to representations for training data. Then by using LSTMs to learn intrabatch relationships between instances of ADLs.

V. TEMPORAL DEPENDENCIES

In Activity Recognition tasks, it can become observed that some activities can act as natural precursors to others [15]. An example of this is the understanding that after using the restroom it is more likely that the subject will wash their hands. This sequential behaviour is widely present in the FITsense data and may support activity recognition tasks based on a relationship to behavioural routines which occur in daily life. Some classifiers already take advantage of these sequences to a basic degree. We propose that these regular behaviours can be learned more effectively through the representation of temporal dependencies to a ML classifier.

In FITsense data, we have observed two significant forms of temporal dependencies likely to be of use in activity classification:

Implicit Dependencies

- the order and sequence in which ADLs take place; and
- the order and sequence of sensor activations within an ADL.

Explicit Dependencies

- the duration of an ADL;
- the duration of sensor activations within an ADL; and
- the time of day at which events occur.

Implicit dependencies refers to the sequential information incidentally encoded in the order and timing of sensor sequences in the training data. In addition, the classification of previous learned examples can influence the learning and prediction of future sequences. In activity recognition data this allows the sequence of ADLs, as well as the sequence of sensor activations, to be encoded as additional knowledge. Some classifiers, e.g. bi-directional LSTMs, can inherently take advantage of these sequential dependencies.

Explicit dependencies in this context refers to the extension of the feature set to include additional temporal features in the representation. The additional features could potentially capture any additional temporal information, such as the total length of a sensor sequence, the time of day at which it occurred, etc. For instance, by splitting the day into quadrants, a coarse timestamp identifying the quadrant of the day at which a sensor activation occurred could be an explicit representation. In initial experimentation, this specific feature was found to have little impact on the performance of our classifiers. Similar manually engineered features have had varying impact across multiple sensor sequence representations. In order to encourage the discovery of temporal dependencies, we implemented cumulative and relative timestamps into the representation. Manually engineering the representation by selecting specific features relies on observations and assumptions of temporal importance

in timestamp data, whereas fine timestamps allow for the algorithmic identification of key dependencies.

Basic motion sensor data representations typically comprise a set of encoded sensor activations which have occurred within the time window, and fail to capture the implicit or explicit forms of temporal dependency e.g. event sequences and timings. As implicit temporal dependencies incidentally exist in this basic form of sensor data representation, it is the choice of classification algorithm that will determine how effectively these dependencies are represented; Whereas explicit temporal dependencies are represented by manually engineered features their influence will be dependent on how well the manual representations present the additional temporal information for learning.

In this paper a variety of representations and classification algorithms are investigated to understand how different data representations affect the influence of explicit temporal dependencies and different classifiers learn implicit temporal dependencies. It is expected that a temporally aware LSTM classifier supplemented with a richer representation containing explicit temporal information may provide better overall performance over the baseline.

VI. EXPERIMENTS

The aim of these experiments is to evaluate the performance of baseline classifiers on binary sensor datasets and compare their performance with LSTM implementations which have implicit and explicit temporal knowledge made available. A selection of popular classifiers were used to establish the baseline performance of traditional classifiers on this problem. These classifiers do not make use of the implicit temporal knowledge provided through the sequences in the data.

LSTMs can use previous sequential learned examples to inform their decisions [16]. We hypothesise that the performance of an ADL classifier can be improved by forming long term knowledge based on existing temporal dependencies. LSTMs were selected due to their demonstrated strength in time-series classification [17]. Additionally, LSTMs can make use of both the implicit and explicit temporal information from our data representations. Four LSTM configurations were compared to evaluate the performance impact of implicit and explicit temporal knowledge in ADL classification.

Iterations on our LSTM models were used to identify how performance could be improved by supporting implicit temporal dependencies and adding explicit temporal information. The previous learned examples can influence the learning and prediction of future sequences in a traditional (or unidirectional) LSTM. Stateful LSTMs can make use of additional implicit knowledge, specifically the order in which ADLs are performed. Stateful LSTMs retain the hidden states of neurons between batches during training, allowing intrabatch dependencies to be inferred. These implementations are marked in the results with prefix “State”.

Relative timestamps were selected as the preferred form of explicit temporal dependency representation. They were found to be most effective through initial experiments. “Exp”

marks experiments performed using representations containing explicit relative timestamps.

TABLE I
OVERVIEW OF THE DATASETS USED.

Dataset	Classes	Attributes	Instances
adlnormal	5	39	120
kasteren	7	14	242
tapia1	22	76	295
tapia2	24	70	208
fitsense1	7	13	744
fitsense2	7	13	990

A. Labelled Datasets

These experiments make use of three publicly available datasets, alongside two datasets from the FITsense project. The datasets used in these experiments document instances of ADLs and their accompanying binary sensor activations which have been captured using a variety of windowing methods. Details of the six datasets are shown in Table I.

CASAS¹ (adlnormal) This dataset contains the fewest classes, with 5 total activities observed by 39 independent sensors. While there is a small number of instances at 120, the activities tracked in the dataset are diverse enough to present little challenge for most baseline classification methods. As only 5 ADLs are tracked, large timegaps between activities can exist which may impact the performance of stateful LSTMs. This small dataset is non-contiguous.

Van Kasteren² (kasteren) The kasteren dataset follows a structure most similar to that of fitsense1/2, with similar tracked activities and sensors. The ADLs expressed in this dataset have relevance to health monitoring and add a layer of complexity which may present a challenge in classification. Prepare_Breakfast and Prepare_Dinner are observably similar as activation sequences, however the time at which they are performed is important. This small dataset is non-contiguous.

MIT³ (tapia1/2) The most complex of the datasets used in these experiments due to the large number of sensors and classes. Several ADLs could be considered beyond the scope of capability for the sensor network (e.g. Going out shopping vs Going out for entertainment), however the complexity presents a useful challenge for classification. This large dataset is non-contiguous.

FITsense/FitHomes⁴ (fitsense1/2) These datasets contain the largest number of instances while using the least sensors of the datasets used in these experiments. The tracked activities were selected for health monitoring applications such as “sleeping”, “grooming”, and “foodprep”. These large datasets contain contiguous streams of ADLs.

¹<http://casas.wsu.edu/datasets/adlnormal.zip>

²<https://sites.google.com/site/tim0306/kasterenDataset.zip>

³<https://courses.media.mit.edu/2004fall/mas622j>

⁴<https://www.rgu.ac.uk/fitsense>

B. Experimental Design

WEKA was used to run a set of baseline classifiers on the datasets. Each dataset was converted from its original format to a zero-padded sequence of discrete values representing binary sensor activations in the ARFF format. Baseline algorithms were selected to evaluate the performance of a representative variety of classification techniques. LibSVM, J48, Bayes and k-NN were selected due to their established significance in the field. Each of these classifiers were run with default configurations as supplied by WEKA.

Temporally aware LSTM implementations were configured using Keras, using the Tensorflow backend. The LSTMs are trained using the “categorical_crossentropy” loss function and “adam” optimizer. Different batch size, units and epochs values were used for each dataset due to variation in sequence length, sensor makeup and activity composition. Standard LSTMs were used with a batch size of 256, 128 units and run for 100 epochs. Stateful LSTMs were implemented using the “stateful” option in Keras with a batch size of 256, 128 units and run for 100 epochs. ExpLSTMs were implemented with a batch size of 256, 512 units and run for 50 epochs. While LSTMs may offer better classification performance in some scenarios, they require a long training time in comparison to baseline classifiers such as k-NN. Deep learning backends such as Tensorflow offer distributions which may use GPUs for training instead of CPUs, which can significantly improve the length of time taken to train.

Both WEKA and Keras implementations were run using Leave One Out cross validation, with each dataset being split by day. This ensures fold contain contiguous sequences ensuring realistic meta-sequences are represented, with each fold starting and ending with “sleeping” ADLs. However in the data for baseline classifiers metasequences of sensor activations could be broken up as the implicit temporal dependency between samples is not considered. Each session of training was also repeated three times with fixed seeds to ensure repeatability.

C. Results

TABLE II
BASELINE CLASSIFIER RESULTS (MACRO F1 SCORES).

Dataset	LibSVM	J48	BayesNet	k-NN
adlnormal	0.898	0.934	0.983	0.910
kasteren	0.901	0.891	0.871	0.892
tapia1	0.162	0.303	0.246	0.248
tapia2	0.129	0.314	0.070	0.219
fitsense1	0.281	0.613	0.600	0.667
fitsense2	0.464	0.620	0.530	0.560

The results for the baseline classifiers in shown in Table II. Each algorithm delivers the highest result on at least one dataset, however the overall winner is narrowly J48. This is

due to its performance on more complex datasets such as *tapia1/2* and *fitsense2*. Bayes and LibSVM demonstrate impressive performance on the *adlnormal* and *kasteren* datasets respectively. Performance of the baseline classifiers on the *tapia* datasets is relatively poor in comparison to the strong performance on others. Performance on the *fitsense* datasets is relatively good in comparison to *tapia*, however it still falls short of that seen on *adlnormal* and *kasteren*.

TABLE III
LSTM RESULTS (MACRO F1 SCORES).

Dataset	LSTM	StateLSTM	ExpLSTM	ExpStateLSTM
adlnormal	0.932	0.951	0.975	0.918
kasteren	0.874	0.831	0.867	0.856
tapia1	0.212	0.202	0.331	0.287
tapia2	0.133	0.240	0.359	0.256
fitsense1	0.853	0.833	0.740	0.864
fitsense2	0.676	0.728	0.586	0.752

The LSTMs display a more balanced performance across all datasets. Slightly poorer performance can be observed between the top performing baseline classifiers and ExpLSTM implementations on *adlnormal* and *kasteren* datasets. On the more complex *tapia* datasets, improved performance over J48 can be observed in the ExpLSTM implementations. This improved performance on more complex datasets can also be seen in *fitsense* datasets, with ExpStateLSTM being the clear winner over baseline classifiers. The overall winner in the LSTM implementations is ExpLSTM, with most results showing a leaning towards implementations utilising additional temporal knowledge.

On the *fitsense* datasets, ExpStateLSTM achieves impressive F1 scores (0.864 and 0.752 on *fitsense1* and 2 respectively) and outperforms all the other baseline and LSTM approaches. *fitsense1/2* are the only contiguous datasets in the experiments, and so are expected to benefit most from the implicit temporal dependencies captured by maintaining the sequence of ADLs. The performance of ExpStateLSTM supports this intuition.

D. Discussion

Of the baseline classifiers, J48 shows the most balanced performance across all datasets. Decision trees display strong performance in general activity recognition tasks. A potential future area of interest may be to investigate the methods by which temporal knowledge could be effectively represented in the training of decision trees for ADL classification.

The baseline classifiers did not have access to any temporal relationships, however on the simpler datasets (*adlnormal* and *kasteren*) they all achieved impressive results that outperformed LSTMs with some algorithms. This suggests that for simple classification tasks temporal relationship information is not required. However, on more complex tasks, including

the FITsense data, the baseline algorithms’ performance was poor highlighting the need to harness temporal relationships.

All LSTM implementations displayed more balanced performance than the baseline classifiers, with variants making use of additional temporal information giving better performance on the more complex classification tasks. Stateful LSTMs performed better than temporally unaware LSTMs across several datasets, however they did not win on any overall. While our temporally unaware LSTM narrowly won on the simpler *kasteren* dataset, ExpLSTM gave the best overall performance on the 4 publicly available datasets. This highlights the importance of capturing specific event timings as part of the representation for more complex tasks.

The key motivation for this work is to achieve good performance on FITsense data. On *fitsense 1/2*, ExpStateLSTM was a clear winner, highlighting the importance of the activity sequences for this data. This is because the FITsense datasets are different to the others in that they are formed from a continuous time-stream and have contiguous windows. The inclusion of null or “none” ADL states ensures the complete sequence of activities is retained. As a result, the implicit meta-sequences of ADLs which occur in the data can be effectively used as an additional source of temporal knowledge.

In conclusion, the inclusion of relative time-stamps as explicit temporal information improved performance in most scenarios. This approach to temporal knowledge representation appears to have been successful in encouraging the discovery of temporal relations. The combination of implicit and explicit temporal representation performs best on the *fitsense* datasets, which are completely contiguous.

VII. CONCLUSION

A key focus of the work is to develop case-based representations from simple sensor network inputs that can effectively capture temporal relationships in order to support improved ADL classifications. Specifically, we have presented LSTM solutions for providing additional implicit and explicit temporal knowledge to an ADL classifier, and compared their performance to established baseline algorithms. The proposed methods were evaluated on publicly available datasets, alongside our own FITsense datasets labelled using a hybrid of rule-based windowing and manual sequence annotation.

The 3 key insights found in this work are:

- Additional temporal information has a positive impact on the performance of ADL classifiers, evidenced by ESLSTMs demonstrating the highest performance of any classifier used in our experiments.
- The method by which data is collected has a strong impact on the performance of ADL classifiers. Temporally aware classifiers perform best on contiguous datasets which capture uninterrupted sequences of activities, such as *fitsense1/2*.
- Allowing ADL classifiers to infer temporal dependencies results in better performance rather than manually engineering temporal features based on assumptions.

Future work could potentially investigate the production of lower level representations for deep learning to make use of additional unidentified temporal dependencies, and using other deep learning classifiers such as Convolutional Neural Networks.

ACKNOWLEDGEMENT

This work was part funded by The Scottish Funding Council via The Data Lab innovation centre.

REFERENCES

- [1] N. K. Suryadevara and S. C. Mukhopadhyay. Wireless sensor network based home monitoring system for wellness determination of elderly. *IEEE Sensors Journal*, 12(6):1965–1972, June 2012.
- [2] Sonja Vestergaard, Kushang V. Patel, Stefania Bandinelli, Luigi Ferrucci, and Jack M. Guralnik. Characteristics of 400-Meter Walk Test Performance and Subsequent Mortality in Older Adults. *Rejuvenation Research*, 12(3):177–184, 2009.
- [3] Takehiro Michikawa, Yuji Nishiwaki, Toru Takebayashi, and Yoshiaki Toyama. One-leg standing test for elderly populations. *Journal of Orthopaedic Science*, 14(5):675–685, 2009.
- [4] Erik E. Stone and Marjorie Skubic. Unobtrusive, continuous, in-home gait measurement using the microsoft kinect. *IEEE Transactions on Biomedical Engineering*, 60(10):2925–2932, 2013.
- [5] M. Ogawa, R. Suzuki, S. Otake, T. Izutsu, T. Iwaya, and T. Togawa. Long term remote behavioral monitoring of elderly by using sensors installed in ordinary houses. *2nd Annual International IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology - Proceedings*, pages 322–325, 2002.
- [6] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
- [7] Randy Joseph Rockinson. Activity Recognition with End-User Sensor Installation in the Home. Master’s thesis, Massachusetts Institute of Technology, 2008.
- [8] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, pages 158–175, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [9] Diane Cook, M Schmitter-Edgecombe, Aaron Crandall, Chad Sanders, and Brian Thomas. Collecting and disseminating smart home sensor data in the casas project. In *Proceedings of the CHI workshop on developing shared home behavior datasets to advance HCI and ubiquitous computing research*, pages 1–7, 2009.
- [10] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM, 2008.
- [11] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors (Switzerland)*, 16(1), 2016.
- [12] Serge Thomas Mickala Bourobou and Younghwan Yoo. User activity recognition in smart homes using pattern clustering applied to temporal ANN algorithm. *Sensors (Switzerland)*, 15(5):11953–11971, 2015.
- [13] Jessamyn Dahmen, Brian L. Thomas, Diane J. Cook, and Xiaobo Wang. Activity learning as a foundation for security monitoring in smart homes. *Sensors (Switzerland)*, 17(4):1–17, 2017.
- [14] Ming Zeng, Le T. Nguyen, Bo Yu, Ole J. Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, 6, 2014.
- [15] Susan McKeever, Juan Ye, Lorcan Coyle, Chris Bleakley, and Simon Dobson. Activity recognition using temporal evidence theory. *Journal of Ambient Intelligence and Smart Environments*, 2(3):253–269, 2010.
- [16] Sepp Hochreiter and J Urgen Schmidhuber. Long Shortterm Memory. *Neural Computation*, 9(8):17351780, 1997.
- [17] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. Springer, 2002.