Human Activity Recognition based on Real Life Scenarios

Fadi Al Machot and Suneth Ranasinghe
Applied Informatics,
Alpen-Adria-Universität Klagenfurt,
Klagenfurt, Austria
{Fadi.AlMachot, Suneth.Ranasinghe}@aau.at

Johanna Plattner
Institute for Applied Research on Ageing,
Carinthia University of Applied Sciences,
Villach, Austria
j.plattner@fh-kaernten.at

Nour Jnoub
Multimedia Information System,
University of Vienna,
Vienna, Austria
nour.jnoub@univie.ac.at

***Abstract*—In Active and Assisted Living (AAL) systems, a major task is to support old people who suffer from diseases such as Dementia or Alzheimer. To provide required support, it is essential to know their Activities of Daily Living (ADL) and support them accordingly. Thus, the accurate recognition of human activities is the foremost task of such an AAL system, especially when non-video/audio sensors are used. It is common that one or more sensors could share or represent a unique activity, and consequently, finding out the most optimal window size among them to represent such an activity is challenging. This paper proposes a Recurrent Neural Networks (RNN) based on a windowing approach for subject-independent human activity recognition. The proposed RNN model is trained based on dynamic systems perspectives on weight initialization process. In order to check the overall performance, this approach was tested using the popular CASAS dataset and the newly collected HBMS dataset. The results show high performance based on different evaluation metrics.**

***Index Terms*—Classification, Recurrent Neural Network (RNN), Activity Recognition, Windowing, Active and Assisted Living environments (AAL), Smart Homes**

I. Introduction

Several countries suffer from demographic trends resulting in a growing demand for products and services to support elderly or disabled people to live independently in their home environments. By using modern technologies, Active and Assisted Living (AAL) systems provide innovative and cost-effective solutions to increase the safety of inhabitants in order to enhance their quality of living.

In the field of AAL, the usage of non-intrusive devices is recommended by considering that non-audio/video sensors are more suitable to minimize privacy issues. The role of sensors is to generate sensor events of the observed environment to reflect the individual behavior and associated intentions. Usually, for such purposes, a sensor setting is used to create a simple activity database which can be further used to classify activities. A simple activity is an activity which is reflected by a specific sensor event. These simple activities are subsequently processed to extract high-level features, about the general status of the individual. Consequently, a supervised machine learning model can be trained to recognize complex human activities.

Human Activity Recognition (AR) is challenging due to (a) the subject independent AR, (b) the uncertainty of sensor measurements, (c) specification of activity window size in real time when new sensor events occur, (d) determining the optimal way to guide and support subjects, (e) multi-class classification, and (f) differentiating human activities among the sensors which might share the same sensors.

This work is established in the frame of the Human Behavior Modeling and Support (HBMS[1]) project. HBMS focuses on supporting people by introducing a fully functional AAL system. One of the major tasks of HBMS is to build human cognitive models, based on Human Cognitive Modeling Language (HCM-L) that can be used to assist inhabitants when they require support for their daily activities [1] [2]. For example, consider the sequence of activities that may be followed by an individual, who suffers from Alzheimer, when preparing a meal. Due to the illness, the person may stop in a while during the process of meal preparing, without knowing what to do next. In such cases, HCM-L offers the possibility to access the "prepare a meal"-model, i.e. the reserved knowledge of the person, which can be applied to guide and support him to successfully complete the intended task [3].

Accordingly, the HBMS system required the continuous observation and recognition of human behavior in order to support them. During this research, we offer an intensive analysis of AR datasets when non-video/audio sensors are used. Moreover, we show the recognition performance using different linear and non-linear classification models compared to a Recurrent Neural Model whose weights are initialized using dynamic systems perspectives. The approach is tested on CASAS project datasets [4] and the HBMS dataset [5]. Both sets were created from real-life scenarios to support residents in smart home environments.

The paper is organized as follows: Section 2 gives a brief overview of the state-of-the-art deep learning AR approaches. Section 3 presents the overall architecture of the proposed windowing and AR system. Section 4 illustrates the HBMS laboratory and dataset. Section 5 discusses the results and the overall performance evaluation. The paper is finalized with a conclusion in Section 6.

[1]https://ae-ainf.aau.at/hbms/

978-1-5386-3227-7/18/$31.00 ©2018 IEEE

II. Related Works

Recently, the deep learning approaches have become popular on account of their high performance in different research fields and applications [6] [7].

In the context of deep learning, [8] proposed a systematic feature learning method using deep convolutional neural networks (CNN) to detect human activities. This method has been used to automatically learn the features from raw inputs. Also, [9] suggests a method to automate the feature extraction process from raw sensor data using a deep convolutional neural network. They have proposed a generic deep framework based on convolutional and long short-term memory (LSTMs) recurrent units. Other works like in [10] proposed a technique based on a deep learning methodology to offer real-time activity recognition to be applied to limited power devices. Their approach considers the invariance against changes in sensor orientation, sensor placement, and sensor acquisition rates. Moreover, a human activity recognition model based on RFID and deep convolutional neural network is introduced by [11]. The major idea is to feed the RFID data into a deep convolutional neural network for activity recognition instead of selecting features.

In [12], they tried to find the best fitting features for K-Nearest Neighbor (KNN) classifiers to improve the classification performance. They extracted 5 different feature representation approaches from an accelerometer dataset collected from two body locations, wrist and thigh. The results show that deep features using CNN produced the best results for kNN classifier. Further approaches for AR can be found in [13] and [7].

III. The proposed Approach

In this paper, we aim at creating a robust AR system by providing Recurrent Neural Network that recognizes human activities with respect to the multi-class classification, incomplete knowledge, and dominant classes problems.

Additionally, it is presented that the positive-definite constraints on weight initialization can be applied successfully to train RNNs based on ReLU (Rectified Linear Unit) nonlinearity [14]. The overall contribution is based on an intensive analysis of human activities and their recognition in the field of AAL.

A. Dynamic Windowing

There are three major challenges which deal with the windowing of streaming data. One major challenge of windowing is choosing an appropriate window size. Proposing, a fixed time windowing approach may offer a quite simple method to learn the activity models during the training phase, but in real-tlife secnarios, some activities may spread over more than one-time slice as discussed in [15] and [16].

Windowing based on an equal number of sensors may be an option as the resulting windows provide varying durations. Nevertheless, there might be situations that may occur with less or more sensor activations as described in [17].

Furthermore, probabilistic dynamic windowing uses a probabilistic approach that maximizes the probability of the most likely window size for a specific activity. The idea is to incorporate the time decay and mutual information using weightings of sensor events within a window [18]. A limitation of such approach is the inefficiency of modeling complex activities when similar sensor sharing situations. Considering the aforementioned challenges, the goal is to analyze the dataset based on different sensor events and define an algorithm that determines the optimal sensors for each activity, In other words, we look for "best fitting sensor set", for each activity. We call this step the "offline phase" [19] [20].

For the given dataset, the overall steps to specify the "best fitting sensors" can be summarized as follows [19]: First, a set of features[2] has to be extracted from the observed labeled dataset for each activity label. Next, the Random Forest [21] approach is applied to determine the importance of sensors based on their best representative feature. In our observation, we considered the best feature that is "the number of activation for each sensor".

Finally, in the online phase, the final window should be created by collecting sensor data until all the best fitting sensors of an activity were activated. This means that the final window may contain not only the best fitting sensors but also other sensors that could be activated in between. Therefore, it is considered to be a dynamic windowing approach. The obtained windows are then used to train the RNN classification model.

B. Classification

To apply an effective classification approach, an intensive evaluation of the CASAS dataset [4] was performed. The goal of the analysis was (see Section V-B) (a) to check the quality of our windowing approach, (b) to determine the suitability of neural models, (c) to find out which classes are the most dominant classes and (d) to see how the data are distributed in the space.

To choose an appropriate classification approach, the data distribution should be considered. For this purpose, Fisher mapping [22] was applied to plot the three major scores out from the samples that have been observed.

Figure 1 shows a scatter plot of the first three Fisher scores. It is clear that the data is highly overlapped. Additionally, Figure 1 shows that the dataset obviously has a class imbalance problem, which was already expected due to the nature of human behavior. The unbalanced dataset is balanced using Synthetic Minority oversampling technique (SMOTE) which attempts to balance the dataset by creating synthetic instance [23].

Generally, it is known that neural networks with sufficient hidden layers (with nonlinear activation functions) are capable of parameterizing arbitrary complex nonlinear functions, and it is trained in a way to preserve the local structure of the

[2]Number of activations of each sensor, activation duration of each sensor, number of activated sensors for each activity, and the location of the sensor

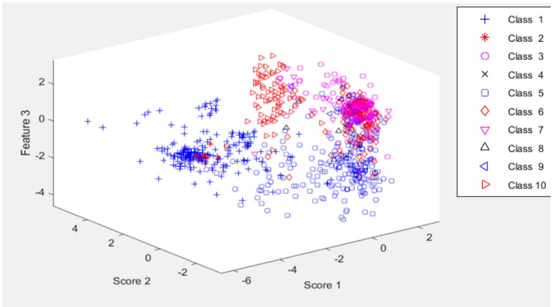

Fig. 1. Scatter plot of the first three Fisher scores

data in the latent space. Herein, the cost function which is minimized in the training of the network that is adapted from Mean Square Error (MSE).

A RNN [24] is proposed to process a sequence of arbitrary length by recursively applying a transfer function to its internal hidden state vector h_t of the input sequence. The activation of the hidden state h_t at time-step t is computed as a function f of the current input symbol x_t and the previous hidden state h_{t-1}.

At each time step, this network computes the subsequent hidden state from the previous state and the input of the current state. Thus, if h_{t-1} is the previous hidden state and x_t is the input of the current state, the subsequent hidden state is given by equations 1 and 2, where W is the weight matrix and b is the bias (is set to zero). We use a softmax layer on the last layer to extract probabilities for classification of all the observed activities, where W_o refers to weights matrix at the output layer and where W_h refers to weights matrix at the hidden layer.

$$h(t) = f(W_h h(t-1) + W_x x(t) + b_h) \tag{1}$$

$$y(t) = softmax(W_o h(t) + b_o) \tag{2}$$

$$softmax(z_c) = \frac{e^{z_c}}{\sum_{d=1}^{c} e^{z_d}} \tag{3}$$

In our learning approach, we have used RNN because human activities do generate sequences which contain sensor records that are dependent. The idea behind this approach is that our prediction depends on our previous states where the RNN has a memory about all the information that has been calculated so far and uses this knowledge to predict the next activity.

Regarding the initialization of W_h, in [14], it is recommended that W_h should be positive definite with respect to ReLU nonlinearity. As a result, the normalized positive-definite weight matrix tries to reduce the sensitivity of the hidden nodes to input perturbations by changing the dynamics to be in a one-dimensional manifold.

$$A = \frac{1}{N}\langle M^T, M\rangle \tag{4}$$

$$e = max(\lambda(A + I)) \tag{5}$$

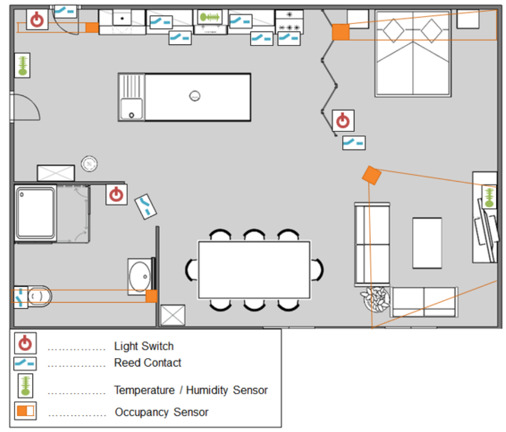

Fig. 2. Laboratory flat equipped with used sensor setting

$$W_h = \frac{A + I}{e} \tag{6}$$

where, $\langle\rangle$ indicates the dot product, N is the number of hidden nodes, I is the identity matrix, M is a standard normal matrix with values generated i.i.d. from a Gaussian distribution with mean zero and unit variance; and $\lambda(A)$ is the list of all eigenvalues of A. Equation 6 guarantees that W_h is a positive definite matrix with the highest eigenvalue of unity and all the remainder eigenvalues are less than 1.

IV. HBMS Laboratory Dataset

The sensors included in the setting were chosen by considering important factors which have to be deliberated when implementing them in real environments, i.e. the sensors had to be small and concise to attract as little attention as necessary. Moreover, the maintainability of the energy supply had to be considered, too. To avoid cables and installation overhead, wireless sensors were selected and another important fact was using non-intrusive sensors which do not record any private data of the residents.

Sensors, namely magnet contacts, switches, temperature and humidity sensors and occupancy sensors, were used in the lab environment. Additionally, to receive the radio signals of the sensors, the EnOcean[3] Pi, a plug-in module for the Raspberry Pi, was used. The module was attached to the FHEM interface, which forwards the collected sensor values directly to the Nimbits database. The used laboratory flat to implement the sensor setting is part of the Living Lab Carinthia, hosted by the Carinthia University of Applied Sciences.

The laboratory consists of an entrance area, a kitchen, a dining area, a living room and a bedroom. Figure 2 shows the final sensor setting within the laboratory flat.

A. Study Design

In order to test the reliability of the implemented sensor network and to create an activity database, a study with 15 participants (adults) was conducted in an equipped flat. Despite

[3]https://www.enocean.com/en/

the fact that the final system should be used in the context of elderly people, the study participants were selected randomly regardless of their background and age. During a single study run, a certain number of tasks had to be performed by the participants. The tasks were chosen to simulate a whole daily schedule, comprised of an overall duration of 45 minutes. The tasks included a sleeping situation, preparing a hot and a cold meal, toileting and activities in the living room such as watching TV and reading on the sofa. Each run was started by entering the flat and finalized by leaving the flat.

B. Activity Annotation

During the study runs, the performed activities were annotated by using an annotation-app on a tablet. For each activity of interest, a separate data channel was created in the database. Each time a predefined activity was started, the corresponding channel was set to 1, and after finishing the activity, the channel was set back to 0. This ensured a precise annotation of the performed tasks with their start and end time, which was important for further data analysis and activity classification.

C. Data Processing

For the preparation of the gathered data for further use, the collected sensor data and the related annotation data for each study run were processed separately. The desired basis format was defined as a file where each row represents one second of the study run including the sensor states and annotation values at this point in time. Using this base-file, we have created dynamic windows for each activity. If the state of a sensor (or of an annotation) is "1" within this window, it will also be "1" in the aggregated row. This approach allows for a dynamic definition of the size of the needed dataset to enable its usage in various fields of application.

V. Results and Discussion

Based on the description of the dataset, the sensors in the smart homes are non-intrusive for human activity recognition of single and multiple residents.

A. Datasets

Three smart home datasets were selected for the evaluation, namely the Aruba and Towor data from CASAS dataset [4] and the HBMS dataset which is newly prepared for the experiment.

Towor dataset represents ADL time series information collected from sensor recordings of two residents, R1 and R2 during summer 2009. Herein, 51 motion sensors, five temperature sensors, fifteen door sensors, a burner sensor, hot and cold water sensor and an electric sensor were used for the recording. It consists of the following annotated activities; *C1: Cleaning*, *C2: Meal Preparation*, *C3: Bed to Toilet*, *C4: Sleeping*, *C5: Working*, *C6: Personal-Hygiene*, *C7: Watching TV* as referred in Table I.

Aruba dataset collected from a house which consists of a single bedroom, a kitchen, a bathroom, a dining room, and an office. The home Aruba included 34 sensors to collect environmental information such as door closure, motion, and temperature of the house setting.All activities are collected from a single inhabitant within the period of 2010-11-04 to 2011-06-11. The following activities are annotated in the dataset as referred in Table II: *C1: Sleeping*, *C2: Go to Bed*, *C3: Meal Preparation*, *C4: Relaxing*, *C5: House Keeping*, *C6: Eating*, *7: Washing Dishes*, *C8: Leave Home*, *C9: Enter Home*, and *C10: Work*.

HBMS laboratory dataset consists of the following classes as referred in Table III : *C1: Go to Bed*, *C2: Preparing Cold Meal*,*C3: Preparing Hot Meal*, *C4: preparing a Drink*, *C5: Sleeping*, *C6: Get Up From Bed*, *C7: Watch TV*, *C8: Reading*, *C9: Sleeping On Sofa*, *C10: Enter Home*, and *C11: Leave Home*.

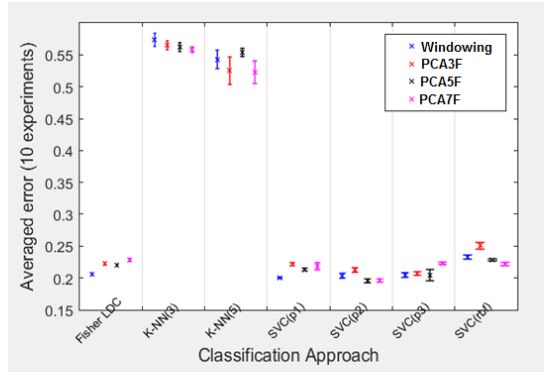

Fig. 3. The classification error and the standard deviation using linear classifiers with kernels where Fisher LDC, K-NN(3) K-Nearest Neighbor Classifier for 3 nearest neighbors without kernel, K-NN(5) K-Nearest Neighbor Classifier for 5 nearest neighbors, SVC(p1) Support Vector Machine without Kernel, SVC(p2) Support Vector Machine using 2nd order polynomial Kernel, SVC(p3) Support Vector Machine using 3rd polynomial Kernel, SVC(rbf) Support Vector Machine using polynomial Kernel radial basis kernel.

B. Data Analysis

We have used CASAS dataset to check the performance of the classification using the complete window based on the proposed windowing approach compared to the following, namely PCA3F (principal components analysis to select the best 3 features), PCA5F (principal components analysis to select the best 5 features) and PCA7F (principal components analysis to select the best 7 features).

Thus, first, a principal component mapping [25] is applied to check how far the principal components already explain the variance. Secondly, a classification error for using different linear and nonlinear classifiers supported by kernels are applied to check how far such models are able to compete with neural models for human AR.

The box plot in Figure 3 shows the classification error and the standard deviation after 10 tests of each classifier. The box plot in Figure 4 shows the classification error and the standard deviation after 10 tests of each classifier.

The back-propagation trained feed-forward neural net classifiers with 2 hidden layers and 5 neurons in each hidden layer as well as using Fisher's linear discriminant classifier based on our windowing approach show the lowest average

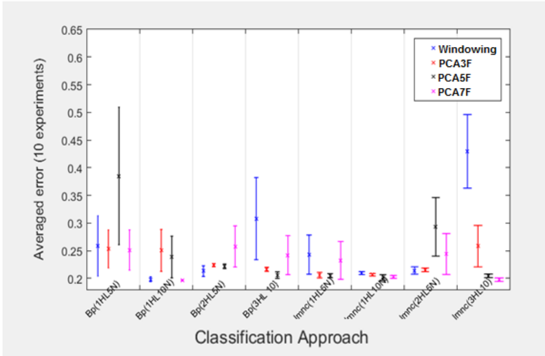

Fig. 4. The classification error and the standard deviation using neural networks where Bp(1HL5N) multi layer perceptron neural network with one hidden layer and five neurons, Bp(1HL10N) multi layer perceptron neural network with one hidden layer and ten neurons, Bp(2HL5N) multi layer perceptron neural network with two hidden layers and five neurons, Bp(3HL10) multi layer perceptron neural network with three hidden layers and ten neurons, lmnc(1HL5N) Levenberg-Marquardt trained feed-forward neural net classifier with one hidden layer and five neurons, lmnc(1HL10N) Levenberg-Marquardt trained feed-forward neural net classifier with one hidden layer and ten neurons,lmnc(2HL5N) Levenberg-Marquardt trained feed-forward neural net classifier with two hidden layers and five neurons, lmnc(3HL10) Levenberg-Marquardt trained feed-forward neural net classifier with three hidden layers and ten neurons

classification errors compared to the performance of reducing the dimensionality of the window to 3, 5 and 7 sensors, respectively. All the other classifiers perform far behind.

Hence, the result is that the proposed windowing works well and choosing neural models are highly recommended because they show best results (see Figures 3 and 4), even better than Fisher's linear discriminant classifier. Based on that, we chose RNN because the generated activities are sequences which contain sensor records that are dependent on time, and the memory of RNN made it an appropriate model.

C. RNN Classification Results

Table I, II and III show the confusion matrix for the corresponding activity classes of each dataset (Aruba, Towor and HBMS, respectively). The number of neurons is 8 and the RNN model is trained using back-propagation technique. Additionally, for multiclass classification, the proposed RNN model is used based on a $10-fold$ cross-validation approach. MATLAB [26], PRtools [27], and Theano [28] have been used for the evaluation tasks and classification.

As shown in table I, there are misclassifications between C3 (Bed to Toilet) and C6 (Personal-Hygiene).The reason is due to the fact that these two activities are mostly predicted based on similar sensor event activations. Additionally, in Table III, the class C6 "Get Up From Bed" leads to misclassifications due to the same reason. Moreover, it should be mentioned that the size of training data for some classes is still not large enough. However, it is still a difficult task to collect or to find such annotated data neither on the web nor in our lab for a high number of activities. Table IV shows the Precision and Recall values for the considered dataset with the respective class type.

From the obtained results, we can conclude two major points. (a) Similarly shared sensor activations may lead to misclassification which due to the topology of RNN, might be overcome if the size of training data is large enough.(b) The correct initialization of the weights matrix for RNN hidden layers does improve the performance of the classification model for human AR.

Furthermore, AAL support systems should consider the privacy and the sensitivity of old or disabled people. According to [29], the acceptance does not only rely on people's capabilities and limitations, it also depends on the personal, socioeconomic and cultural contexts. To avoid such hinders, non-intrusive sensors are still the best choice.

VI. Conclusion

The paper proposed a human AR approach based on dynamic windowing and RNNs whose weights can be initialized in a proper way to avoid the undesired dynamic behavior of the hidden neuron. An intensive analysis of the datasets has been presented and different linear and non-linear classification models were applied. The results helped to understand the performance of previously mentioned classification and windowing techniques in the frame of human activity recognition using non-intrusive sensors. Moreover, the paper discussed several state-of-the-art approaches considering audio/video and non-intrusive sensors based on deep learning models.

TABLE I
The classification performance using Towor CASAS dataset

	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**CO***
C1	5	0	0	0	0	1	0	6
C2	5	207	0	0	3	0	5	220
C3	0	3	90	0	2	47	3	145
C4	0	0	2	27	5	0	0	34
C5	0	1	0	0	22	0	0	23
C6	2	1	59	2	4	383	2	453
C7	9	1	0	0	12	0	217	239
TO*	21	213	151	29	48	431	227	1120

CO* = Classification Overall, **TO*** = Truth Overall

Acknowledgment

HBMS project is funded by the Klaus Tschira Stiftung GmbH, Heidelberg, Germany.

References

[1] F. Al Machot, H. C. Mayr, and J. Michael, "Behavior modeling and reasoning for ambient support: Hcm-l modeler," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2014, pp. 388–397.

[2] J. Michael, F. Al Machot, and H. C. Mayr, "Adoxx based tool support for a behavior centered modeling approach," in *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2015, p. 41.

[3] H. C. Mayr, F. Al Machot, J. Michael, G. Morak, S. Ranasinghe, V. Shekhovtsov, and C. Steinberger, "Hcm-l: Domain-specific modeling for active and assisted living," in *Domain-Specific Conceptual Modeling*. Springer, 2016, pp. 527–552.

TABLE II
THE CLASSIFICATION PERFORMANCE USING ARUBA CASAS DATASET

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	CO*
C1	76	3	0	0	7	1	0	0	0	0	87
C2	0	24	0	0	0	0	0	0	0	0	24
C3	0	0	151	1	4	4	2	0	1	2	165
C4	0	0	0	13	0	0	0	0	0	1	14
C5	7	0	1	0	64	6	2	2	2	0	84
C6	0	0	3	0	10	40	2	0	0	0	55
C7	0	0	2	0	5	12	17	0	0	0	36
C8	0	0	0	0	0	0	0	78	0	0	78
C9	0	0	0	0	0	0	0	0	82	0	82
C10	0	0	0	0	0	1	1	0	0	15	17
TO*	83	27	157	14	90	64	24	80	85	18	642

CO* = Classification Overall, **TO*** = Truth Overall

TABLE III
THE CLASSIFICATION PERFORMANCE USING HBMS DATASET

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	CO*
C1	0	0	0	0	0	0	0	0	0	0	20	20
C2	2	93	1	40	0	0	1	2	0	0	0	139
C3	0	0	50	0	0	0	1	6	0	0	2	59
C4	1	2	1	56	0	0	0	0	1	60	0	121
C5	1	0	0	0	35	0	0	0	0	0	0	36
C6	20	0	0	0	0	60	0	0	0	0	0	80
C7	0	0	0	0	0	0	8	0	0	0	0	8
C8	0	1	3	0	2	0	1	82	16	0	0	105
C9	0	0	1	0	0	0	0	14	10	0	0	25
C10	0	0	0	0	0	0	0	0	0	30	0	30
C11	0	0	0	0	0	0	0	0	0	0	44	44
TO*	24	96	56	96	37	60	11	104	27	90	66	667

CO* = Classification Overall, **TO*** = Truth Overall

TABLE IV
PERFORMANCE EVALUATION FOR ALL THE DATASETS

	Towor		**Aruba**		**HBMS**	
Class	Precision	Recall	Precision	Recall	Precision	Recall
C1	83.33%	23.81%	87.36%	91.57%	0%	0%
C2	94.09%	97.18%	100%	88.89%	66.90%	96.88%
C3	62.07%	59.60%	91.52%	96.18%	84.75%	89.27%
C4	79.41%	93.10%	92.86%	92.86%	46.28%	58.33%
C5	95.65%	45.83%	76.19%	71.11%	97.22%	94.60%
C6	84.55%	88.86%	72.73%	62.50%	75%	100%
C7	90.80%	95.60%	47.22%	70.83%	100%	72.73%
C8			100%	97.5%	78.10%	78.85%
C9			100%	96.47%	40%	37.04%
C10			88.24%	83.33%	100%	33.33%
C11					100%	66.67%

[4] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 7, 2013.

[5] J. Michael, A. Grießer, T. Strobl, and H. C. Mayr, "Cognitive modeling and support for ambient assistance," in *Information Systems: Methods, Models, and Applications*. Springer, 2012, pp. 96–107.

[6] S. Ranasinghe, F. Al Machot, and H. C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation," *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, p. 1550147716665520, 2016.

[7] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and Vision Computing*, vol. 60, pp. 4–21, 2017.

[8] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[9] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[10] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on*. IEEE, 2016, pp. 71–76.

[11] X. Li, Y. Zhang, I. Marsic, A. Sarcevic, and R. S. Burd, "Deep learning for rfid-based activity recognition," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 164–175.

[12] S. Sani, N. Wiratunga, and S. Massie, "Learning deep features for knn-based human activity recognition." 2017.

[13] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in Bioinformatics*, p. bbx044, 2017.

[14] S. S. Talathi and A. Vartak, "Improving performance of recurrent neural network with relu nonlinearity," *arXiv preprint arXiv:1511.03771*, 2015.

[15] T. Van Kasteren, G. Englebienne, and B. J. Kröse, "An activity monitoring system for elderly care using generative and discriminative models," *Personal and ubiquitous computing*, vol. 14, no. 6, pp. 489–498, 2010.

[16] N. C. Krishnan and S. Panchanathan, "Analysis of low resolution accelerometer data for continuous human activity recognition," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3337–3340.

[17] D. J. Cook and N. C. Krishnan, *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*. John Wiley & Sons, 2015.

[18] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and mobile computing*, vol. 10, pp. 138–154, 2014.

[19] F. Al Machot, H. C. Mayr, and S. Ranasinghe, "A windowing approach for activity recognition in sensor data streams," in *Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on*. IEEE, 2016, pp. 951–953.

[20] F. Al Machot, A. H. Mosa, M. Ali, and K. Kyamakya, "Activity recognition in sensor data streams for active and assisted living environments," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[21] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[22] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 2013.

[23] R. Pears, J. Finlay, and A. M. Connor, "Synthetic minority oversampling technique (smote) for predicting software build outcomes," *arXiv preprint arXiv:1407.2330*, 2014.

[24] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[25] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[26] MATLAB, "version r2015a," Natick, Massachusetts, 2015.

[27] R. Duin, "Prtools version 3.0: A matlab toolbox for pattern recognition," in *Proc. of SPIE*. Citeseer, 2000.

[28] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010.

[29] Q. Ni, A. B. García Hernando, and I. P. de la Cruz, "The elderlys independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development," *Sensors*, vol. 15, no. 5, pp. 11 312–11 362, 2015.