

# Unsupervised Online Activity Discovery Using Temporal Behaviour Assumption

Hristijan Gjoreski, Daniel Roggen

Wearable Technologies Laboratory, University of Sussex, Brighton BN1 9RH, UK  
h.gjoreski@sussex.ac.uk, daniel.roggen@ieee.org

## ABSTRACT

We present a novel unsupervised approach, UnADevs, for discovering activity clusters corresponding to periodic and stationary activities in streaming sensor data. Such activities usually last for some time, which is exploited by our method; it includes mechanisms to regulate sensitivity to brief outliers and can discover multiple clusters overlapping in time to better deal with deviations from nominal behaviour. The method was evaluated on two activity datasets containing large number of activities (14 and 33 respectively) against online agglomerative clustering and DBSCAN. In a multi-criteria evaluation, our approach achieved significantly better performance on majority of the measures, with the advantages that: (i) it does not require to specify the number of clusters beforehand (it is open ended); (ii) it is online and can find clusters in real time; (iii) it has constant time complexity; (iv) and it is memory efficient as it does not keep the data samples in memory. Overall, it has managed to discover 616 of the total 717 activities. Because it discovers clusters of activities in real time, it is ideal to work alongside an active learning system.

## ACM Classification Keywords

H.1.2 Models and Principles: User/Machine Systems; I.5.4 Pattern recognition: Applications; H.5.m Miscellaneous: HCI

## Author Keywords

Activity recognition; Activity discovery; Online temporal clustering; Accelerometer; Segmentation

## INTRODUCTION

Human Activity Recognition (HAR) relies on supervised learning [1], which limits it to recognizing pre-defined "closed" set of activities for which training data is available. In practice, the activities that a person may perform is potentially unbounded. Activity discovery aims to discover potential new activities beyond a predefined set through unsupervised learning [2]. It can be tackled through unsupervised clustering [3] (feature-based approaches), motif discovery techniques [4]–[7], or topic models [8][9] (symbols-based). Activity discovery is meant to be part of a larger system which also attempts to discover the semantic

meaning of the discovered activity. One approach for this is to use active learning (ask the user) [10][11]. This becomes a viable option as recent wearables (e.g. smartwatches, Google Glass) support this through micro-interactions. For example, the Withings smartwatch [12] have enabled users to manually label activities on top of the automatically detected ones. Currently, over 200,000 users collectively logged, equivalent to over 109 years' worth of activity data.

We focus on discovery of activities characterised by sensor signals that are periodic or static. With motion sensors, such activities are: walking, running, gym exercises, activities of daily living (shovelling, scrubbing floor, brushing teeth) as well as still postures such as sitting, standing, lying. Discovering such activities is important because their analysis can provide basis for health applications (e.g. analysing changes in daily routines) or fitness application (e.g., analysing work-out exercises). These activities have two key characteristics: i) last for some time (e.g. tens of seconds brushing teeth, minutes during gym exercises, hours during sleep); and ii) it is unlikely that frequent transition between such activities occur.

In this paper, we propose a feature-based, activity discovery method that exploits these characteristics. It is based on the online clustering of Guedalia et al. [13] and Zhang et al. [14], and improves upon by: i) exploiting the human behaviour characteristics and discovering clusters that are *continuous in time* and last for a *predefined minimum duration*; ii) including a mechanism to regulate *sensitivity to brief outliers*; iii) discovering *multiple clusters overlapping in time* which allows to better model activities comprising deviations from nominal behaviour; iv) being *unbounded* (open-ended) and not requiring specification of the number of clusters to discover; v) being *online* and *memory and time efficient* by not keeping the data samples in memory and having an execution time which does not grow with the number of activities discovered.

We evaluate the method on two datasets: the JSI-ADL dataset comprising 14 activities recorded by 10 subjects [15] and the REALDISP [16] dataset comprising 33 fitness activities recorded by 17 subjects. We compare the method against Density-based spatial clustering of applications with noise (DBSCAN) [17] and online agglomerative clustering [14]. Our findings show significantly better performance compared to competing approaches on majority of the performance measures, with the added benefits of online operation and reduced computational and memory needs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
ISWC '17, September 11–15, 2017, Maui, HI, USA  
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM 978-1-4503-5188-1/17/09...\$15.00  
<https://doi.org/10.1145/3123021.3123044>

## RELATED WORK

The general assumption of activity discovery is that activities that repeat over time (but not necessarily continuously) will lead to identical motifs that repeat (exploited by motif-based approaches) or feature vectors that cluster nearby (exploited by feature-based approaches).

*Motif discovery* approaches analyse the sensors data and try to find frequent, reoccurring motifs (patterns), i.e., repeating activities [4]–[7]. They can detect only activities for which the motif is found (a priori in a supervised way [4] or in unsupervised way [6]). Therefore, for an activity to be detected it has to repeat frequent enough so that a motif is found. This usually results in detection of quite limited set of activities (e.g., 6 in [4][6]). Our approach can detect an activity even if it appears only once (given it is periodic and long enough), but it cannot recognize if the same activity repeats multiple times. This makes the two approaches complementary and potentially used together in future implementations.

*Topic models* have been frequently applied on video data, images, environmental sensors and less frequently to wearable sensors [9]. Usually these approaches are parametric and assume a fixed model complexity, thus the performance critically depends on the number of topics specified. Seiter et al. [8] overcame these issues by using semantic temporal priors to discover 5 high-level activities. However, the semantic priors are defined with context words, which are obtained by a hand-crafted classifier. Compared to this work, our approach is more related to the classifier that provides the context words, which suggests potential usage of our approach as input to the topic modelling approach for higher level activities.

*Feature-based approaches* use clustering techniques (DBSCAN and K-Means in [3], and Expectation Maximization in [18]) applied on feature vectors extracted from window-segments of the raw sensor data. This was demonstrated with low number of activities (e.g., 5 in [3]).

Traditional clustering techniques (DBSCAN, K-Means) are not able to take advantage of the temporal characteristics of most stationary and periodic human activities, i.e., that when we engage in these activities, they last for some period of time and it is unlikely that frequent change between them occurs. For these reasons, *temporal clustering* techniques may be more suitable. Often, these techniques belong to the wider family of spatial-temporal clustering techniques [19]. However, in spatio-temporal applications, usually the temporal component is used as a simple filter to partially select the data in some time period (hour, day, year) [20].

Temporal clustering is a process of grouping similar data samples close both in feature space and in time. Recently, temporal clustering of periodic human motions has been applied on data acquired from motion capture system and by video cameras [21]. The main drawbacks of the

approach are: the number of clusters to discover have to be specified beforehand; it operates offline on the whole dataset; and the computational complexity is quadratic with the length of the window. In [22], the authors proposed a temporal clustering approach for activity segmentation using wearable sensors data. The proposed method is based on Linear Dynamical Model and was tested on datasets with limited number of simple and highly distinct activities (straight walking, running, jumping, sideways walking, punching, and body rotation). A limitation of the method is that it relies on an ad-hoc segmentation technique which uses the curvature of the gyroscope data, which appears unsuitable for real life. Also, the computational complexity relies on the number of segments, and the method requires manual selection of the optimal number of clusters.

Most approaches reviewed above rely on offline processing which requires the availability of the entire dataset and often prevents embedded implementations. Thus, as a basis for our method we used an online agglomerative clustering method introduced in [13]. The algorithm is fairly simple: (i) for each new point it moves the closest centroid towards it; (ii) it merges the two closest centroids; (iii) the new point becomes also a centroid. Additionally, the method uses Gaussian kernel-function for distance measures to better handle noise [14]. This method is online because it does not keep all the examples in memory at the same time, but processes them one by one, keeping only aggregates (cluster centre, and cluster size).

Overall, the key limitations of the state of the art approaches is that they have addressed a small number of activities (at most 10 in [21]). Our method addresses these limitations and builds upon [13] and [14] by incorporating the temporal assumption of the human behaviour, i.e., samples closer in time are likely to form a cluster that represents an activity. The proposed method can handle large number of activities (17 and 33 in our experiments).

## METHOD

Let us assume the scenario shown in Figure 1. At the top is the sensor signal (e.g., acceleration), segmented by a sliding window. The user is performing 3 activities (A1, A2 and A3) resulting in 3 *activity segments*. An activity segment is defined as an interval between the start and the end of an activity. A traditional clustering method takes the all of the data windows in bulk (not taking into account the temporal dependence and sequentially of the windows) as an input, performs the clustering, and assigns a cluster number to each data window. An illustration is given in Figure 1, where the clustering method found 3 clusters (C1, C2 and C3), which roughly correspond to the 3 activities. However, the analysis of the data windows on a time scale, shows that these 3 clusters are scattered in time, and there are 9 cluster segments (marked with CS1 to CS9). A *cluster segment* is obtained when a sequence of data windows map to a same cluster. In an unsupervised scenario, where the method is not aware of which cluster corresponds to which activity,

each cluster segment is a potential query to users asking about the activity (active learning). Thus, an ideal clustering method should cluster the sensor signal into cluster segments which corresponds to the activity segment.

In everyday life there are numerous situations in which some activity is interrupted or deviated from for short periods, e.g., walking, interrupted with short stops. A traditional clustering method risks clustering each of these stops as a separate cluster segments and this way fragmenting the walking activity.

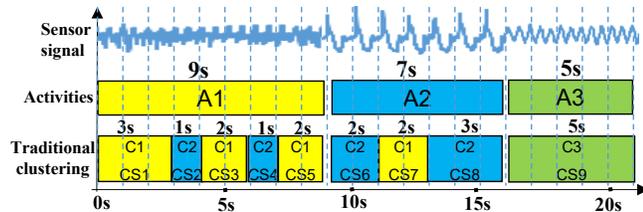


Figure 1. Illustrative clustering of 3 activities.

Another issue with the offline clustering methods is that they require all the data to be present in memory at the same time before performing the clustering. Assuming one day's worth of data, this means clustering could only be performed at the end of the day. This implies both a larger memory requirement than online approaches, but it is also not well suited for inclusion within a broader active learning framework where the user could be prompted in real time to label a potentially discovered activity.

To tackle these two issues, we propose the *UnADeVS* method (Unbounded Unsupervised Activity Discovery using the Temporal Behaviour Assumption). *UnADeVS* is an online method (it clusters streams of data in real time) and is based on the agglomerative clustering method proposed in [13][14]. *UnADeVS* improves upon [13][14] by introducing a mechanism to exploit the temporal assumption of continuity of human behaviour. It has 3 mechanisms (controlled by 3 *algorithm parameters*) to handle the temporal dependence of the data:

- First, at any point in time, the method keeps a pool of active clusters, so that multiple deviations can be clustered into multiple temporally overlapping clusters. These clusters would effectively be distinct in terms of their class centres but potentially overlapping in time. The *activePool* parameter defines this number of clusters in the pool of active clusters. Note that this is not the total number of clusters (which is open ended).
- Second, each active cluster has some waiting period (*tolerance*) in which it stays active and can be updated. This way it better handles short outliers (such as brief deviations from the nominal behaviour) without creating a separate cluster for each deviation. Basically, the *tolerance* parameter defines the duration a cluster is allowed to remain in the pool of active clusters without being updated. A larger value allows to tolerate more outliers or transients before deciding to "freeze" the

cluster which means deciding that a cluster segment has been discovered. A smaller value would decrease the tolerance to noise and may increase the number of discovered clusters.

- Finally, the method has a minimum duration parameter (*minDur*) which is the minimum duration that a cluster must reach, otherwise the cluster is discarded. In this way, the method discards the potential small cluster segments that are found for each short deviation. Chiefly, if we are interested in discovering activities lasting at least *N* seconds then *minDur* is set to *N*.

The pseudo code is given with Algorithm 1. First, the data is segmented with a sliding window, and a feature vector (*fv*) is computed on the window. Next, the closest cluster from the pool of active clusters is found by calculating the distance between the cluster centre (*cen*) and the *fv*. We are using Gaussian kernel as a distance metrics because it better handles noise and outliers [14]:

$$K(fv, cen) = \exp\left(-\frac{\|fv - cen\|^2}{2\sigma^2}\right) \quad (1)$$

Next, the closest cluster is updated by adding the *fv* in the cluster and updating the cluster's centre, size and end time. Then, the two closest clusters are merged, and new cluster is created with the *fv* (giving it a chance to grow a new cluster if it is a start of a new activity). Finally, if a cluster is complete (not updated for *maxDur* time), it is outputted.

The code and example visualization is publicly available at: <http://www.sussex.ac.uk/strc/research/wearable/research-ll>.

---

#### Algorithm 1: UnADeVS algorithm

---

```

parameters: activePool, tolerance, minDur
input: accStream
output: discovered_clusters
FOR EACH sample IN accStream:
  window.add(sample)
  IF window.size > win_size:
    fv = Preprocess(window) #filter & extract features
    window.clear()
    #PERFORM THE CLUSTERING
    cc = Find_closest_cluster(fv, active_clusters)
    Update(cc) #updates the cluster: centre, size, and time
    IF active_clusters.size > activePool:
      Merge(active_clusters) #merge two closest clusters
      #create a new cluster for the point and add it
      new_cluster = Create_cluster(fv)
      active_clusters.add(new_cluster)
    #if there is old, non-active cluster, output it
    FOR EACH cluster IN active_clusters:
      IF cluster.time - fv.time > tolerance:
        #if the cluster is bigger than the minimum
        IF cluster.size > minDur:
          discovered_clusters.add(cluster)
          OUTPUT: "Cluster is discovered:" + cluster
          active_clusters.remove(cluster)

```

## EXPERIMENTAL SETUP

### Data

We evaluated the method on 2 established HAR datasets. The first dataset is the Jozef Stefan Institute – Activities of Daily Living (JSI-ADL) dataset [15][23]. It was chosen because it contains activities of daily living (walking, lying, sitting, cycling, etc.), and also includes some more complex repetitive activities such as: shovelling, scrubbing the floor,

washing dishes and working on PC. These more complex activities usually consist of multiple atomic activities. For example, shovelling comprises bending to push the shovel in the snow, rotating to throw the snow away, pausing. The dataset contained 14 periodic activities (the in-betweens and the transitions were marked as NULL), which were executed by 10 users. It was recorded by an accelerometer attached on the right thigh, sampled at 50 Hz.

The second dataset is the REALDISP HAR dataset [16]. It contains 34 (33 + NULL) fitness activities performed by 17 subjects. It contains data logged from 9 inertial sensors sampled at 50 Hz, from which we have chosen the left wrist accelerometer because of practical reasons (smartwatch). We chose this dataset because it contains a representative number of subjects (17) and a large number of repetitive fitness activities (33, more than 3 times the number of activities used in the previously largest activity discovery evaluation in [21]). Therefore, this dataset should be challenging enough to show the potential of the presented method. The activities include: walking, jogging, running, jumping (up, sideways, etc.), cycling, rowing, and various stretching exercises (wrist rotations, arms rotations, bending, etc.).

On average, the JS-ADL dataset contains much longer activity segments (359s) compared to the REALDISP (26s). Table 1 summarises the dataset characteristics.

	#Activities	Amount of data [h]	#Feature vectors	#Subjects	Average activity duration (std) [s]
JSI-ADL	14	18.8	67 798	10	359 (259)
REALDISP	33	9	32 210	17	26 (27)

Table 1. Statistics about the chosen datasets.

### Preprocessing

The proposed method operates on features extracted from a window sliding on the sensor data and outputs cluster segments. The choice of the window size is related to the periodicity of the activities in the dataset and must be large enough so that the stationarity assumption is respected. Here we used 2s windows with 1s overlap for both datasets. We explored longer windows with similar results. On each data window we extract 8 statistical features for each of the 3 acceleration axis and for the magnitude of the acceleration vector (32 features in total). The features are: mean, median, standard deviation, energy, integral, skewness, kurtosis, and root mean square. We chose these features because they are generic: they are commonly used in the literature and they are not highly tuned to the specific activities in the dataset. This makes the evaluation more realistic for an "open-ended" HAR scenario, where the activities that may arise, and henceforth the "ideal" features for that activity cannot be predicted, and therefore the best that one can do is use "generic" features. We removed redundant features using the Univariate feature selection method, implemented as *SelectKBest* in the *scikit-learn* library. We varied the K parameter from 1 to 32 and evaluated the HAR classification accuracy before and after

selecting the top K features. We decided to choose the top 14 and 20 features (REALDISP and JSI-ADL respectively), which was a trade-off between accuracy and sufficient number of features. The features were standardized to zero mean and unit standard deviation. The number of signal windows on which feature vectors are computed is given in Table 1.

### Evaluation Metrics

We performed two types of evaluation: unsupervised (activity discovery) and supervised (activity identification). For each type we explain the rationale and the metrics used.

The *unsupervised evaluation* relates to the potential usage of the method where the method discovers clusters and potentially query the user about the activity label. For this we calculated *unsupervised cluster-activity* confusion matrix and then computed the accuracy and the F1-score. The unsupervised cluster-activity matrix computed for the example given in Figure 1 is shown in Table 2 (left). The rows are the activity segments and the columns are the discovered cluster segments. Each entry in the confusion matrix represents the duration of the overlap between the cluster segment and the activity segment. Next, we use the Hungarian (Kuhn-Munkres) algorithm [21][24] to find the optimum correspondence between discovered clusters and the activity (e.g., choosing only CS1 for A1 in Figure 1, which is also marked in shade in Table 2), and compute the *accuracy*. The accuracy for this example is calculated by choosing CS1 for A1, CS2 for A2 and CS9 for A3 as the most optimal which gives  $accuracy = (3+3+5)/(9+7+5) = 0.52$ . In a similar way, we calculate the macro *F-1 score*, which is the harmonic mean between the recall and the precision averaged over all of the activities [25].

	CS1	CS2	CS3	CS4	CS5	CS6	CS7	CS8	CS9		C1	C2	C3
A1	3	1	2	1	2	0	0	0	0	A1	7	2	0
A2	0	0	0	0	0	2	2	3	0	A2	2	5	0
A3	0	0	0	0	0	0	0	0	5	A3	0	0	5

Table 2. Cluster-Activity confusion matrices corresponding to example shown in Figure 1 (unsupervised left, and supervised right). The cells with dark colour are chosen as the optimal.

The accuracy and the F1-score do not explicitly take into account the number of cluster segments. This may be a disadvantage in our application because each discovered cluster segment could lead to a potential user query in a broader active learning system, thus we would like to penalize discovering too many cluster segments. Therefore, we additionally calculated the *average fragmentation*. The fragmentation of an activity is the number of discovered cluster segments for that activity. In Figure 1, the average fragmentation is 3 because the fragmentation of A1 is 5, of A2 is 3 and of A3 is 1, and thus average fragmentation is  $(5+3+1)/3$ . To keep the values between 0 and 1, in the future reference we will use the transformation  $1/(\text{average fragmentation})$ , which means values closer to 1 are better.

Also, we calculated the *activity detection ratio*, which is the number of detected activity segments divided by the total

number of activity segments (it is 1 in Figure 1 because there is at least 1 cluster segment corresponding to each activity segment).

To evaluate the method in a scenario where it is given limited number of labelled activities (e.g. self-annotated by the user in an active learning system), we also performed a *supervised evaluation*. Note, that this evaluation is different to the standard supervised HAR, since we are not learning a classification model. Instead, we assign a label (colour) to each cluster segment. The label of the cluster segments corresponds to the label of the activity whose centre is the closest in the feature space. In Figure 1, there are 3 cluster labels (colours) that correspond to the activities: yellow (C1), blue (C2) and red (C3). According to the cluster labels, the cluster-activity confusion matrix is computed as shown in Table 2 right, which we use to calculate the accuracy and F-1 score [25]. The difference compared to the unsupervised evaluation is that in this case we combine multiple cluster segments with the same colour during an activity segment (e.g., combining CS1, CS3 and CS5 to identify A1 in Figure 1). Similar to the unsupervised metrics, we also calculate the average fragmentation =  $(3+2+1)/3$  and the activity detection ratio = 1. The difference is that we count only the cluster segments with the same colour as the activity.

### Comparison methods

To compare the performance of the presented method we used the original online agglomerative clustering method [14] (referred as Baseline) and DBSCAN [17]. The Baseline is an online method that requires to specify the maximum number of clusters to discover a-priori. DBSCAN is an offline clustering method that does not need to pre-define the number of clusters to discover and thus is more similar to our approach. Because we are interested in discovering activity segments, we adapted the evaluation for the two methods. First, we ran the Baseline and DBSCAN methods to identify the clusters. Then, we merged consecutive windows that belong to the same cluster which resulted in cluster segments. Then we removed the cluster segments that are shorter than *minDur*. This attempts to make the evaluation of Baseline and DBSCAN as similar as possible to our proposed method.

We introduced two variations of Baseline and DBSCAN referred to as Baseline-T and DBSCAN-T. In particular, we evaluated them by ignoring the cluster labels (colours) and just used the cluster segments (marked with CS in Figure 1). These variations make the 2 approaches more similar to our method, which also assumes that each cluster segment is a separate cluster. The drawback of this variation is that the Baseline-T is not online, and the DBSCAN cannot recognize repeating activities.

### Parameter Optimization

To find the optimal values of the parameters for each of the methods (ours and the 4 competing ones), we performed a

grid search through the algorithm parameter values in order to maximise the unsupervised accuracy and the average fragmentation. The optimization was performed on the data of a single subject (Subject 1 in both datasets), so that we avoid overfitting. The result of the search was a Pareto front from which we have chosen the parameter values that represent trade-off between the accuracy and the average fragmentation. For our method the parameter values were: REALDISP [*activePool* = 3, *tolerance* = 6, *minDur* = 12]; JSI-ADL [*activePool* = 3, *tolerance* = 22, *minDur* = 16].

## EXPERIMENTAL RESULTS

### JSI-ADL Dataset

Figure 2 illustrates the activity discovery results for Subject 1. It shows at the top the 14 activity segments annotated in the dataset, each marked with its class number (the smaller clusters are not marked for space reasons, and some activities repeat such as A0). Underneath, the figure shows the cluster segments identified by the 5 methods each marked with a number and a colour. The cluster segment numbers are always incrementing for our method, Baseline-T and DBSCAN-T, and for the Baseline and DBSCAN the same cluster can repeat in multiple segments. The colour of the clusters corresponds to the colour of the activity whose centre is the closest in the feature space. The empty space between the activities are the NULL segments.

Visually, the traditional clustering method (DBSCAN) and the online method (Baseline), perform the worst as they appear to highly fragment the activity segments. We speculate that this is because they do not use the temporal information about the data windows (points close in time should also be close in the feature space). DBSCAN-T and Baseline-T, perform better for some activities, but the fragmentation is high for the activities that are more "complex" and consist of multiple atomic activities, such as A8 (washing dishes which comprises standing, short walks), A14 (scrubbing floor which comprises on all fours, moving, resting), A7 (shovelling which comprises standing, moving, bending), and A1 (lying exercising: moving legs, resting). For these activities, our method appears to find better cluster segments. The reason for this is two-fold. First, it grows multiple clusters (*activePool*), so that each of the comprising atomic activities can be clustered in a separate cluster (the *minDur* parameter removes the small parallel clusters). Second, each active cluster has some waiting period (*tolerance*) in which it stays active and can be updated. This way it better handles the short transitions between the atomic activities and does not create a separate cluster for each atomic activity, which may be what happens with the traditional methods.

Figure 3 and Figure 4 present the overall performance of the methods averaged over all of the 10 subjects. Additionally, we performed statistical tests (paired T-Test) to confirm the statistical significance of the comparison between the methods with  $p < 0.05$ .

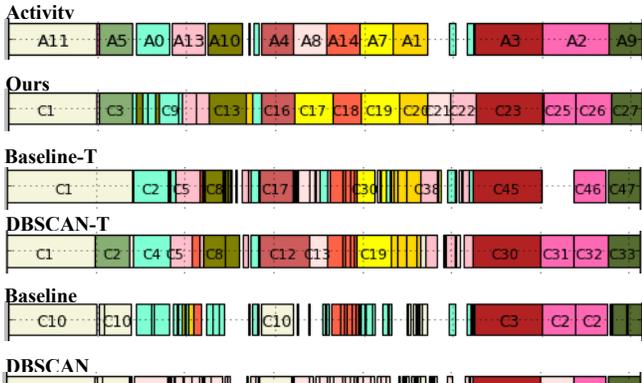


Figure 2. JSI-ADL activity identification results for Subject 1. Activity segments are at the top, identified clusters by each method are below.

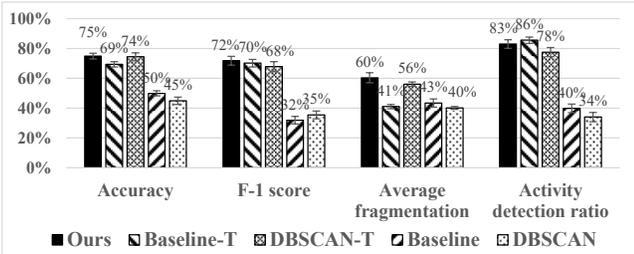


Figure 3. JSI-ADL unsupervised activity discovery comparison of the 5 methods using the 4 evaluation metrics.

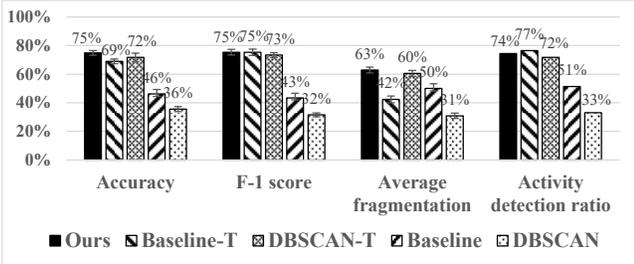


Figure 4. JSI-ADL supervised activity identification comparison of the 5 methods using the 4 evaluation metrics.

Figure 3 shows the results for the unsupervised evaluation, which confirm the observations made in Figure 2: our method performs significantly better in all of the metrics compared to the Baseline and DBSCAN. The accuracy, the F-1 score and the activity detection ratio are similar to the Baseline-T and DBSCAN-T, however our method achieves a significantly better fragmentation, while maintaining discovery of 83% of the 180 activity segments.

Figure 4 shows the results for the supervised evaluation of activity identification. The results show similar accuracy and F-1 score of the three methods: ours, Baseline-T, and DBSCAN-T, which is significantly better compared to the Baseline and the DBSCAN. The comparison of Figure 3 and Figure 4 shows that in general, the supervised performance is similar to the unsupervised for all of the metrics, except for the activity detection ratio which is slightly decreased in the supervised scenario. This suggests that for some activities the discovered clusters do not correspond to that particular activity (wrong identification).

While one could expect that no cluster segments should be discovered during NULL, the results in Figure 2, show that our method discovered clusters C21 and C22. After analysing these segments we found that during this NULL segment the subject was repeating the following activities: walking (5s), sitting down, sitting (5s), standing up, and standing (5s). In a way this is a repetitive complex activity which was correctly detected by our method.

### REALDISP Dataset

Figure 5 presents the detailed results for Subject 1. The top row shows the 33 activities (two of which repeat, resulting in 35 activity segments). Similar to the results of the previous dataset, the Baseline and DBSCAN methods appear to highly fragment the activity segments. We again speculate that this is because they do not use the temporal information about the data windows. DBSCAN-T and Baseline-T perform better, but both fail to identify significant number of activities: they identified only 15 and 20 out of the 35 activity segments, respectively, while our proposed method identified 31. In general, our method better identifies (matches) the activities, both in the matching colour and the number of activities found. This is also confirmed by the detailed results shown in Figure 6 and Figure 7 averaged over all 17 users.

Figure 6 shows the unsupervised evaluation (ignoring the cluster colours). Our method has similar accuracy compared to Baseline-T, and slightly worse fragmentation, but performs significantly better on all the other metrics. The activity detection ratio of our method is significantly higher than the Baseline-T. Our method was able to discover 87% of the activities, i.e., it found 467 out of the total 537 activity segments over all 17 users.

Figure 7 shows the results for the activity identification. The results show that our method has significantly better performance in all of the metrics except for the fragmentation, in which it is slightly worse than the Baseline-T and DBSCAN-T. In general, the supervised accuracy is lower compared to the unsupervised, except for the accuracy of our method, where it increased by 6 percentage points. This suggests combining multiple cluster segments with the same colour, e.g., C2 and C3 for the A1 for our method in Figure 5.

Similarly to most datasets designed for HAR, this dataset comprises periods that are not annotated (NULL class). These are periods where the user does not perform activities that were deemed relevant for the creators of the dataset. However, this does not preclude the user from potentially engaging in some periodic or static behaviours during these NULL segments. While one could expect that no cluster segments should be discovered during NULL, the results in Figure 5, show that, all methods often tend to identify a yellow cluster segment during a NULL activity segment, in-between A32 and A33 for our method in Figure 5). After checking the activity labels we discovered that the yellow

cluster segment corresponds to the walking activity, which suggests that the person was walking around in this non-annotated part of the dataset while waiting to perform the following activity.

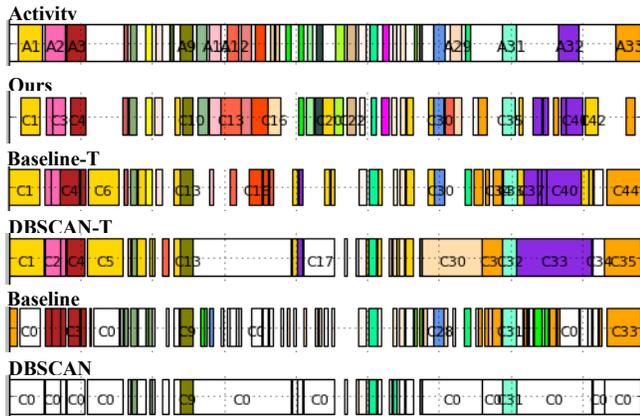


Figure 5. REALDISP activity identification results for Subject 1. Activity segments are at the top, clusters by each method are below.

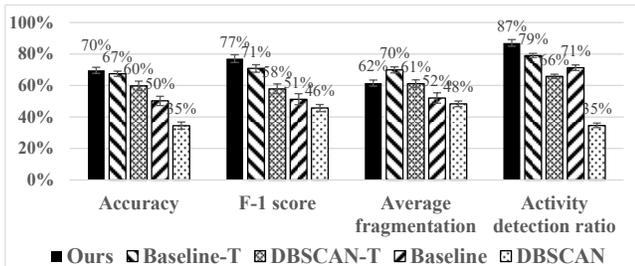


Figure 6. REALDISP activity discovery comparison of the 5 methods using 4 evaluation metrics.

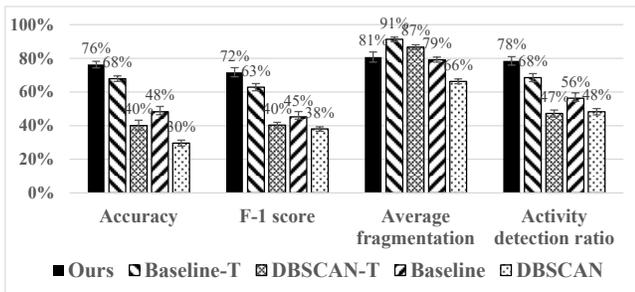


Figure 7. REALDISP Activity identification comparison of the 5 methods using 3 evaluation metrics.

## DISCUSSION AND CONCLUSION

Our work contributes to the body of research in activity discovery. It belongs to the family of methods operating on features extracted from a sliding window. It is suitable for activities whose sensor signal are periodic or stationary, which typically correspond to repetitive movements or stationary postures. Example repetitive movements are: gait (recovery after surgery, Parkinson's), fitness activities, cooking activities (cutting, stirring, etc.). Example stationary postures include: being seated, lying down (in different poses). Therefore identifying such periodic activities and stationary postures can provide important insights to everyday life.

Human activities are also made of sporadic gestures (e.g. drinking from a cup, opening a door). The approach we propose is not suited for such activities. Instead the discovery of sporadic gestures may be more suited to motif discovery techniques operating on the raw sensor signals instead of features [4]–[6].

Motif discovery techniques can also discover periodic activities, such as walking [6]. However, a motif-based method would potentially detect each of the steps individually, thus it is better suited to identify individual steps but less for identification of the whole walk segment.

To the best of our knowledge, this is the first approach to activity discovery in an online setting, and that has been evaluated on two HAR datasets containing large number of activities (14 and 33 respectively). The evaluation showed that it achieved better and sometimes worse but similar performance compared to the competing approaches, and has managed to discover 616 of the total 717 activities.

The *UnADeVS* method has 3 intuitive parameters, which can be controlled by the user in real-life. The user may control the frequency of the clusters discovered (*tolerance*) or the duration of the clusters discovered (*minDur*). In particular, if the method discovers small clusters, the user can increase the *minDur*. Similarly, if the user is asked too often, he can increase the *tolerance* parameter. We have found that the *activePool* parameter should be kept low, either 3 or 4, because with bigger numbers the method grows too many clusters in parallel and this way it finds too many time-overlapping clusters. The experimental evaluation on the two datasets confirmed that for shorter activities (mean activity duration was 26s in REALDISP) the *tolerance* and *minDur* parameters had smaller optimal values (6, 12 respectively) compared to their optimal values (22, 16 respectively) for the longer activities in the JSI-ADL dataset (mean activity duration was 359s). Thorough parameter characterization is considered for future work.

*UnADeVS* is memory efficient because it is online and does not keep all the samples in memory, but instead processes them one by one, keeping only aggregates in memory (cluster centre, cluster duration, cluster size). It is time efficient because the execution depends on the *activePool* parameter, which we found in our experiments on the two datasets to be 3. Basically, that means only 3 calculations are needed for each new data window in order to find the closest cluster centre (constant complexity). This makes *UnADeVS* suitable for embedded and real-time implementations on microcontrollers.

Because the approach discovers clusters of activities in real time, it is especially useful for real-time analysis of discovered clusters. A system could send discovered clusters to the cloud to perform aggregate analysis with other data in a crowdsourced scenario. Alternatively it could be used in a broader active learning scenario to prompt the user for feedback shortly after the activity ends

to label it. This can be combined with query budget allocation strategies to minimise user burden [10].

The approach is general and can be applied to any sensors data suitable for sliding window segmentation. For example, it can be used as a general segmentation method for time-series. Moreover, because it segments streams of data, it may allow experts to use it as a labelling tool and label each of the discovered segments.

One limitation of the proposed is that reoccurring activities lead each time to a new cluster segment. We envision a further extension which compares the newly found cluster segment to the list of all previously identified cluster segments and finds and regroups the most similar clusters.

An assumption of the proposed approach is that the signal has stationary statistical properties in the sliding window. This generally can be achieved with a window larger than the periodicity of the target activities. In our experiments we observed that 2 seconds window with 1 second overlap satisfies this condition. Thorough window size analysis is considered for future work.

For future work we plan to investigate the active learning paradigm, when, how and how often to ask the user for a label to the discovered cluster [10]. Additionally, we plan to combine *UnADeVS* with topic modelling [8] in order to discover high-level activities, e.g., to use the output of our method as input to the topic modelling method.

## ACKNOWLEDGMENTS

This study was financed by the EPSRC project "Lifelearn: Unbounded activity and context awareness" EP/N007816/1.

## REFERENCES

- [1] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 1, no. June, pp. 1–33, 2014.
- [2] D. J. Cook, N. C. Krishnan, and P. Rashidi, "Activity Discovery and Activity Recognition: A New Partnership," *Cybern. IEEE Trans.*, vol. 43, no. 3, pp. 820–828, 2013.
- [3] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6067–6074, 2014.
- [4] E. Berlin and K. Van Laerhoven, "Detecting leisure activities with dense motif discovery," *Proc. 2012 ACM Conf. Ubiquitous Comput. - UbiComp '12*, p. 250, 2012.
- [5] N. Begum and E. Keogh, "Rare Time Series Motif Discovery from Unbounded Streams," *Vldb*, vol. 8, no. 2, pp. 149–160, 2014.
- [6] D. Minnen, T. Starner, M. Essa, and C. Isbell, "Discovering characteristic actions from on-body sensor data," *Proc. - Int. Symp. Wearable Comput. ISWC*, pp. 11–20, 2007.
- [7] T. Maekawa, D. Nakai, K. Ohara, and Y. Namioka, "Toward practical factory activity recognition: Unsupervised understanding of repetitive assembly work in a factory," *UbiComp 2016*, pp. 1–12, 2016.
- [8] J. Seiter, W. C. Chiu, M. Fritz, O. Amft, and G. Tröster, "Joint segmentation and activity discovery using semantic and temporal priors," *2015 IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2015*, pp. 71–78, 2015.
- [9] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," *Proc. 10th Int. Conf. Ubiquitous Comput. (UbiComp '08)*, pp. 10–19, 2008.
- [10] T. Miu, D. Roggen, P. Missier, and T. Plötz, "On strategies for budget-based online annotation in human activity recognition," *UbiComp 2014 - Adjunct Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, pp. 767–776, 2014.
- [11] M. Stikic, K. Van Laerhoven, and B. Schiele, "Exploring semi-supervised and active learning for activity recognition," *Wearable Comput. 2008. ISWC 2008. 12th IEEE Int. Symp.*, pp. 81–88, 2008.
- [12] "Withings Activity Recognition Smartwatch." [Online]. <https://health.nokia.com/blog/2016/12/21/introducing-our-new-activity-recognition-feature>. [Accessed: 10-Apr-2017].
- [13] I. D. Guedalia, M. London, and M. Werman, "An On-line agglomerative clustering method for nonstationary data.," *Neural Comput.*, vol. 11, no. 2, pp. 521–540, 1999.
- [14] D. Zhang, S. Chen, and K. Tan, "Improving the robustness of 'online agglomerative clustering method' based on kernel-Induce distance measures," *Neural Process. Lett.*, vol. 21, no. 1, pp. 45–51, 2005.
- [15] H. Gjoreski, B. Kaluža, M. Gams, R. Milić, and M. Luštrek, "Context-based ensemble method for human energy expenditure estimation," *Appl. Soft Comput.*, vol. 37, pp. 960–970, 2015.
- [16] M. Damas, O. Amft, A. M. Toth, O. Baños, H. Pomares, and M. A. Tóth, "A benchmark dataset to evaluate sensor displacement in activity recognition," *ACM Conf. Ubiquitous Comput. - UbiComp '12*, vol. 16, p. 1026, 2012.
- [17] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proc. 2nd Int. Conf. Knowl. Discov. Data Min.*, pp. 226–231, 1996.
- [18] P. Siirtola, P. Laurinen, E. Haapalainen, J. Röning, and H. Kinnunen, "Clustering-based activity classification with a wrist-worn accelerometer using basic features," *IEEE Symp. Comput. Intell. Data Min.*, pp. 95–100, 2009.
- [19] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo, "Spatio-temporal clustering," *Data Min. Knowl. Discov. Handb.*, pp. 855–874, 2010.
- [20] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, 2007.
- [21] F. Zhou, S. Member, F. De Torre, and J. K. Hodgins, "Hierarchical Aligned Cluster Analysis for Temporal Clustering of Human Motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582–596, 2013.
- [22] J. Gong, P. Asare, Y. Qi, and J. Lach, "Piecewise Linear Dynamical Model for Action Clustering from Real-World Deployments of Inertial Body Sensors," *IEEE Trans. Affect. Comput.*, vol. 7, no. 3, pp. 231–242, 2016.
- [23] B. Cvetković, R. Milić, and M. Luštrek, "Estimating Energy Expenditure with Multiple Models using Different Wearable Sensors," *IEEE J. Biomed. Heal. Informatics*, vol. 20, 2016.
- [24] M. D. R. Burkard and S. Martello, "Assignment Problems.," *Soc. Ind. Appl. Math. SIAM*, 2009.
- [25] H. Gjoreski et al., "Competitive live evaluations of activity-recognition systems," *IEEE Pervasive Comput.*, vol. 14, no. 1, pp. 70–77, 2015.