Efficient Deep Clustering of Human Activities and How to Improve Evaluation

Louis Mahon LOUIS.MAHON@CS.OX.AC.UK
Institute for Language, Cognition and Computation, University of Edinburgh, UK
Department of Computer Science, University of Oxford, UK

Thomas Lukasiewicz THOMAS.LUKASIEWICZ@CS.OX.AC.UK
Institute of Logic and Computation, TU Wien, Austria
Department of Computer Science, University of Oxford, UK

Editors: Emtiyaz Khan and Mehmet Gönen

Abstract

There has been much recent research on human activity recognition (HAR), due to the proliferation of wearable sensors in watches and phones, and the advances of deep learning methods, which avoid the need to manually extract features from raw sensor signals. A significant disadvantage of deep learning applied to HAR is the need for manually labelled training data, which is especially difficult to obtain for HAR datasets. Progress is starting to be made in the unsupervised setting, in the form of deep HAR clustering models, which can assign labels to data without having been given any labels to train on, but there are problems with evaluating deep HAR clustering models, which makes assessing the field and devising new methods difficult. In this paper, we highlight several distinct problems with how deep HAR clustering models are evaluated, describing these problems in detail and conducting careful experiments to explicate the effect that they can have on results. Additionally, we present a new deep clustering model for HAR. When tested under our proposed settings, our model performs better than (or on par with) existing models, while also being more efficient and scalable by avoiding the need for an autoencoder.

1. Introduction

Human activity recognition (HAR), the task of automatically determining the activity that a person is performing based on recorded data, has a number of important applications. It is of interest to healthcare research, as it can provide a direct measure of exercise frequency and intensity. The World Health Organization lists inactivity as the fourth leading risk factor for mortality, and estimates that over 30% of adults are insufficiently active.[1] However, such estimations are difficult. Self-report does not give a reliable measure of exercise, as patients tend to significantly over-report (McConnell et al., 2018), so being able to directly monitor human activity is desirable. HAR is also used in wearable sports technology. Sports watches, for example, provide users with a breakdown of how much time they spend sitting, standing, and walking. Globally, the wearable technology market was valued at $41bn in 2019, and it is forecasted to grow to $114bn by 2028.[2]

1. https://www.who.int/data/gho/indicator-metadata-registry/imr-details/3416
2. https://www.grandviewresearch.com/industry-analysis/wearable-technology-market

The recorded data on which human activity recognition is based can come from three different types of device: video recorders, ambient sensors, and wearable sensors. These data are then input to a recognition model, to infer the activity being performed. If video recorders are used, then the task is one of computer vision, if ambient or wearable sensors are used, then the task is a form of signal processing. The difference between ambient and wearable sensors is that the former stays at a fixed location in the environment, and the latter is attached to the human performing the activity. In this paper, we focus primarily on HAR from wearable sensors. There are two types of wearable sensors, accelerometers, which measure acceleration in three spatial dimensions, and gyroscopes, which measure orientation and angular momentum.

Raw sensor data cannot always be easily interpreted by human inspection, which has two important consequences. Firstly, it can make feature engineering difficult. For example, in the case of gyroscope readings, we do not know, a priori, what the relevant differences are between the signals for certain activities, especially those that are similar, such as walking upstairs vs. downstairs. While engineered features have been used with some success (see Section 2), these are mostly statistical features, rather than features that leverage domain knowledge. Deep learning, which can learn to automatically extract features, is therefore an attractive approach to HAR. The second important consequence is that HAR data are very difficult to label. Labelled data are always more expensive and time-consuming to obtain than unlabelled data, but this is especially the case for sensor-based HAR data, because humans cannot provide annotations just by looking at the sensor readings. Instead, annotators must directly observe a subject, which requires taking them into the lab, or be given a video of the performed actions, which requires subjects to remember to film themselves while using the sensors outside the lab. There is therefore a need for models that can operate without labelled data, as has been noted in two recent survey papers (Wang et al., 2019; Chen et al., 2021). This is one reason why HAR clustering is of value. If it was solved very accurately, so that all instances of the same activity were clustered together, and all instances of different activities were clustered separately, then the HAR classification problem would also have been solved, and its solution would not have required any labelled data at all. Another advantage of HAR clustering is that, even in the absence of a very accurate solution, it can shed light on the most appropriate set of classes into which activities should be partitioned. For example, some datasets distinguish between 'walking' and 'fast walking', while some others just use a single class 'walking'; similarly for 'running' and 'jogging'. If clustering shows there to be a significant difference between walking and fast walking, this is evidence that such a distinction is warranted. This use is not explored further here, though it has been in previous works (Mejia-Ricart et al., 2017).

For these reasons, HAR clustering has received significant research attention. This paper focuses on deep HAR clustering, i.e., clustering HAR data using a deep neural network for feature extraction. Recent years have seen some works applying deep learning to HAR clustering (McConville et al., 2021; Sheng and Huber, 2020; Ma et al., 2021). However, progress has been obstructed in deep HAR clustering, and HAR clustering more generally, by a lack of agreed evaluation standards. Different works test on different datasets, many of them private, and some crucial details are left out when describing the exact evaluation settings. In particular, the distinction between subject-dependent and subject-independent clustering is often not made explicit, even though it greatly affects the results.

As well as highlighting these problems and describing more rigorous evaluation settings that can address them, we propose a new deep HAR clustering model and test it under these settings, showing that it outperforms existing methods (insofar as a comparison can be made, with respect to the above points). We then present ablation studies on its main components. Ours is the first deep HAR clustering model not to require the reconstruction loss of an autoencoder, making it more efficient and better able to scale to more complex datasets. Below is a brief summary of our contributions.

- We explicate differences in evaluation procedures for deep HAR clustering, and show empirically that these differences can affect performance metrics. While our main focus is on deep HAR clustering, much of our analysis, including the important distinction between subject-dependence and subject-independence, applies to HAR clustering in general.
- We discuss suitable evaluation settings to use for HAR clustering. Adoption of our recommendations by future works will enable a direct comparison and benchmarking of deep HAR clustering models (and HAR clustering models more generally), and thus accelerate progress in the field.
- We describe a streamlined and scalable deep HAR clustering model. On six public datasets, this model performs better than or on par with existing models (insofar as they can be compared). We also present ablation studies showing the contributions of its components.

The rest of this paper is organized as follows. Section 2 provides an overview of related work. Section 3 explicates the shortcomings of existing evaluation methods for HAR clustering. In Section 4, we describe our new method for HAR. Section 5 then presents the results of our method under our proposed evaluation settings, and Section 6 summarizes our work. Code is available at https://github.com/Lou1sM/HAR.

2. Related Work

Machine learning has been identified as a promising approach to HAR since at least 2000 (Hongeng et al., 2000), with early works using, e.g., naive Bayes (Tapia et al., 2004), support vector machines (He and Jin, 2009), and generalized discriminant analysis (Khan et al., 2010). These machine learning algorithms require feature engineering, and in the case of HAR, this is commonly done by taking statistical quantities such as mean and higher moments of the raw signal, in both the time and frequency domains. These can be combined with more bespoke features (M. Zhang, 2011; He and Jin, 2009).

Deep learning is a form of machine learning that does not require hand-crafted features, but rather can learn to extract features from a raw input. Applied to HAR, not only does deep learning avoid the need for feature engineering, but has also been shown to achieve better accuracy than feature-engineered models (Alsheikh et al., 2016; Ferrari et al., 2019). Network architectures include convolutional neural networks (CNNs) (Yang et al., 2015; Chen and Xue, 2015; Jiang and Yin, 2015; Ronao and Cho, 2015) and recurrent neural networks (RNNs) (Inoue et al., 2018; Singh et al., 2017). See (Hammerla et al., 2016) for

an empirical comparison of CNNs and RNNs for HAR. See (Chen et al., 2021; Wang et al., 2019) for summaries of recent deep HAR models.

There has been increasing interest in reducing the supervision needed for HAR. The first efforts in this direction were semi-supervised methods trained on unlabelled data alongside labelled data (Li and Dustdar, 2011), which used unlabelled data for pretraining (Li et al., 2014; Alsheikh et al., 2016), or which investigated the optimality of the label set by comparing to cluster labels (Mejia-Ricart et al., 2017). However, training these semi-supervised models still requires some labelled data. By a recent survey on HAR, the need for fully unsupervised models is urgent, due to the difficulty of obtaining labels (Wang et al., 2019). Another (Chen et al., 2021) discussed the advantages of unsupervised HAR models but notes the disadvantage that most, performing only feature extraction, cannot produce labels. Deep clustering models redress this problem by interpreting cluster membership as labels.

As is the case for supervised models, most HAR clustering models begin with a feature extraction stage. While some works apply clustering algorithms directly to the raw sensor signal (Trabelsi et al., 2013), the most common approach is to first extract features from the sensor signal, and then cluster the extracted features. Initial HAR clustering models performed feature extraction by computing statistical quantities (Kwon et al., 2014; Machado et al., 2015; Lu et al., 2017). In (He et al., 2017), statistical features are replaced by the discrete wavelet packet transform followed by principal components analysis for dimensionality reduction. Clustering is performed using fuzzy c-means with a novel initialization method based on cosine similarity. Another HAR clustering model is proposed in (Sheng and Huber, 2020), which includes a deep autoencoder in the feature extraction stage with two additional loss terms to encourage locality and temporal consistency. Statistical feature extraction is dispensed with completely in (Ma et al., 2021), replaced with a CNN-BiLSTM autoencoder plus pseudo-label training (Caron et al., 2018). In (McConville et al., 2021), a deep autoencoder is combined with UMAP (McInnes et al., 2018), for dimensionality reduction, followed by clustering using a Gaussian mixture model (GMM).

3. Problems with Existing Literature

We identify three problems with the existing field of HAR clustering, shown with respect to existing works in Table 1.

- Different works often report on different datasets using different metrics. Many works report results on their own new dataset, and so cannot compare to prior works. Additionally, the datasets are often private.
- The exact evaluation criteria are unclear. There are multiple ways of evaluating the performance of a model, which can give significantly different results. Of particular importance is whether each subject's data are clustered individually or whether all data are clustered together.
- Code is not released, making reproducibility difficult or impossible.

These problems make it hard for new researchers in the field to assess the best existing models from which to build, and difficult for them to determine whether they have improved over these existing models. Consequently, progress is held back.

In the following sections, we address these issues. Section 3.1 describes our choices of datasets on which to evaluate performance, and the reasoning behind these choices. Section 3.2 discusses the effect of different evaluation settings on performance metrics, and demonstrates these effects empirically, by showing that the same model can produce significantly different results in different evaluation settings.

3.1. Datasets

We select six suitable wearable-sensor HAR datasets for measuring clustering performance: Physical Activity Monitoring (PAMAP2) (Reiss and Stricker, 2012), Human Activity Recognition using Smartphones (UCI-Sm) (Anguita et al., 2013), WISDM-v1 (Kwapisz et al., 2011), WISDM-watch (Weiss et al., 2019), Realistic Sensor Displacement (REALDISP) (under the 'ideal placement' setting) (Baños et al., 2012), and Heterogeneous Human Activity Recognition (Stisen et al., 2015). The details of subjects, activities, and sensors for each dataset are shown in Table 3. There are three reasons for selecting these datasets:

- They are all easily accessible in the UCI repository.
- They have been used by some previous works, and so enable comparison (though, as seen in Table 1, there is less consistency in the use of datasets than would be ideal).
- They vary in number of activities, number of data points, number of subjects, and number of sensor channels, helping to measure generalization ability. UCI-Sm and WISDM-v1 are smaller datasets, with only a few channels and activity classes. The other four datasets are more complex. WISDM-watch is unique in having a large number of users, 51, and HHAR is unique in having a large number of data points. REALDISP is a large dataset with many sensors channels. This set of datasets thus tests a model's performance in a range of settings, from a small to a large number of users, from a small to a large number of clusters and from simple hardware to many wearable sensors with a rich array of sensor channels.

3.2. Ambiguous Evaluation Settings

We identify two ambiguities in how HAR clustering models are evaluated, subject-dependent vs. subject-independent, and window-wise vs. point-wise. The former refers to whether clustering was performed on all subjects' data at once (subject-independent) or on each subject's data separately (subject-dependent). For supervised models, specification of the train-test split can disambiguate subject-dependence vs. independence, by specifying that, e.g., data for users X, Y, Z was used for testing. Clustering, however, does not typically use a train-test split, and the question of subject-dependence vs. independence is almost always unclear (see Table 1). The latter refers to whether data points are taken to be the sliding window or each time point. Each time point has one label, but training collates multiple time points into windows. (The window size is 512 in all our experiments.) There is ambiguity as to whether data points should be taken to be the windows or the time points. This type of ambiguity can be present in supervised models as well.

Table 1: Previous HAR clustering models relative to the evaluation criteria outlined in Section 3. There are a number of different datasets and metrics, but none releases their code, and almost all are ambiguous as to subject independence (indicated S-dep below).

Name	Datasets	Metrics	Code Released	S-Dep
Kwon et al. (2014)	own (private)	ACC, NMI	no	unlcear †
Trabelsi et al. (2013)	own (private)	ACC, precision, recall	no	unclear
Lu et al. (2017)	own (private)	ACC, precision, recall, specificity, ARI, FM-index	no	unclear
Machado et al. (2015)	own (private)	ACC	no	both
He et al. (2017)	WISDM-v1	RI, ARI	no	unclear‡
Sheng and Huber (2020)	PAMAP2, SBHAR, REALDISP	ACC, ARI, NMI	no	unclear
He et al. (2018)	DSAD	RI, ARI, FM-index	no	yes
Ma et al. (2021)	HAR, MotionSense,[1] MobiAct,[2] own (private)	precision, recall, F1, NMI	no	unclear
Dobbins and Rawassizadeh (2018)	HHAR	silhouette-index	no	unclear *
ours	PAMAP2, UCI-Sm, WISDM-v1, WISDM-watch, REALDISP, HHAR	ACC, ARI, NMI, F1	yes (on publication)	yes

[1] Altun et al. (2010)
[2] Malekzadeh et al. (2018)
† mentions different cluster sizes for different subjects, implying subject-dependence
‡ displays confusion matrix referring to one subject only
* discusses subject heterogeneity within activity classes, implying subject-independence

We empirically investigate the effect of these two factors, by training and testing a simplified version of our proposed model (described in Section 4) under the four resulting settings. The results are displayed in Table 2.

Columns one and two are subject-dependent. They train a separate clustering model on each subject and average the results across these separate models, weighting each subject by their number of data points. Columns three and four are subject-independent. They train a single clustering model and cluster all subjects' data at once. The subject-dependent models outperform the subject-independent models by a large margin. This is in keeping with results from the supervised domain, where it has been noted that training on some data from the test subject significantly improves performance (Reiss and Stricker, 2012; Suh et al., 2021), which suggests the existence of subject-specific features in how activities are performed. The large difference in results between these two settings, evidenced in Table 2, means that it is crucial that models specify which they are using. Moreover, the two are fundamentally different tasks, one discovering patterns in the activity signal of a specific user, and the other learning generalized activities, independent of who is performing them.

Columns one and three in Table 2 are window-wise, while columns two and four are point-wise. The former treats each window as a data point whose label is the most commonly occurring label across all time points in that window. The latter treats each time point as a data point. Using overlapping windows means that each time point appears in multiple windows. In order to produce a single predicted label for each time point, the window-wise setting takes the most commonly occurring label across the multiple windows that contain

Table 2: Evaluation under the four settings corresponding to the two ambiguities described in Section 3.2. Window-wise vs. point-wise does not affect results, but subject-dependent vs. subject-independent does. The subject-dependent settings performs substantially better across all datasets and metrics. It is therefore essential that HAR clustering works specify whether they are testing in the subject-dependent or subject-independent setting.

		window-wise subject-dependent	point-wise subject-dependent	window-wise subject-independent	point-wise subject-independent
PAMAP	ACC	66.28	66.27	48.30	47.35
	NMI	64.95	64.80	46.61	48.14
	ARI	50.57	50.54	30.44	31.31
	F1	65.63	65.57	45.26	45.73
UCI-Sm	ACC	50.57	50.73	38.95	38.93
	NMI	56.36	56.77	35.59	35.57
	ARI	39.57	39.50	23.94	23.92
	F1	46.74	46.88	35.32	35.28
WISDM-v1	ACC	72.15	72.33	50.14	50.04
	NMI	69.44	69.40	38.78	38.80
	ARI	60.05	60.03	33.07	33.06
	F1	64.88	64.95	38.91	38.88
WISDM-watch	ACC	78.40	78.48	25.58	25.56
	NMI	84.68	84.71	28.38	28.36
	ARI	72.22	72.17	12.6	12.59
	F1	77.61	77.67	25.32	25.31
REALDISP	ACC	89.60	89.60	51.37	51.36
	NMI	93.87	93.97	71.80	71.79
	ARI	88.60	88.53	43.91	43.87
	F1	84.36	84.38	47.00	46.99
HHAR	ACC	53.80	53.80	47.93	48.25
	NMI	51.62	51.62	38.42	38.52
	ARI	38.14	38.07	25.60	25.78
	F1	52.57	52.48	49.08	49.36

Table 3: Information on each of the datasets on which we report results. Accel = 3d accelerometer, gyro = 3d gyroscope, and magneto = 3d magnetometer. Total time points = the sum of time points across all users, after discarding those without labels and those with missing data.

Name	Date Released	Number of Activities	Number of subjects	Sensors	Channels	Total Time Points
PAMAP2	2012	12	9	3 x (accel, gyro, magneto)	51	1921431
UCI-Sm	2013	6	30	phone accel and gyro	6	71968
WISDM-v1	2011	5	36	phone accel	3	1085363
WISDM-watch	2019	18	51	phone and watch accel and gyro	12	3635842
REALDISP	2012	33	17	9 x (accel, gyro, magneto, orient)	117	669618
HHAR	2015	6	9	2 x (accel, gyro)	12	11279265

Algorithm 1 Training algorithm

$X \leftarrow$ data;
$Mask, SemiMask \leftarrow X$;
$Final \leftarrow$ empty hash table; **while** $|Final|$ *increases* **do**
  for $i = 1, \ldots, 10$ **do**
    initialize encoder, encode X and UMAP to $\mathbb{R}^2$ cluster using HMM, giving labels $c(x)$ and probabilities $p(x)$ **for** $x \in X$ **do**
      if $p(x) < .95$ *or* $(i > 1$ *and* $c(x) \neq c'(x))$ **then**
        remove x from $Mask$ and $SemiMask$
      else
        add x to $SemiMask$
      end
    end
    for $epoch = 1, \ldots, 5$ **do**
      for $x \in X$ **do**
        if $x \in Mask$ **then**
          train on $x, c(x)$
        else if $x \in SemiMask$ **then**
          train on $x, c(x)$, weighted by 0.5
        end
        $c'(x) \leftarrow c(x)$
      end
    end
  end
  for $x \in Mask$ **do**
    $Final[x] \leftarrow c(x)$
  end
end
for $x \in X \setminus Final$ **do**
  $Final[x] \leftarrow c(x)$
end

that time point. (We also explored other means of combining these multiple labels, with very similar results.) There is essentially no difference between window-wise and point-wise evaluation so, although existing works are ambiguous as to which is being employed, this ambiguity does not prevent a clear assessment of performance. The results presented in Section 5 are all in the point-wise setting.

4. Our Method

Our model is based on the technique of pseudo-label training (Caron et al., 2018), extended with a novel method for selecting points on which to pseudo-label train. Pseudo-label training clusters the output of an encoder, then uses cluster labels as classification targets, and trains on these targets to refine the weights of the encoder. It allows the encoder weights to be iteratively refined. However, the pseudo-labels are noisy and often incorrect. Previous works (Mahon and Lukasiewicz, 2021; Mrabah et al., 2020) have shown that filtering out

the least confident labels can improve performance. We propose a novel method of filtering these labels. At each iteration, we train only on those data points that received the same cluster label as they did in the previous iteration. We also implement a graded label filtering, by double-weighting the training updates for the points that received the same cluster label in every iteration so far. Formally, let X be the space of data points. Elements of X are windowed sequences of sensor readings of length 512. Our encoder network f_θ and clustering model g are then represented by the following functions

$$f_\theta : X \to Z\,,$$
$$g : Z \to \{0, \ldots, K-1\}\,,$$

where the latent space Z is of lower dimension than X, θ denotes the network parameters, and K is the user-defined number of clusters. Let T be the total number of training iterations, and $\theta_j, 1 \leq j \leq T$ be the value of the network parameters at iteration j. Then, we define the full deep clustering model at iteration j as

$$C_j \coloneqq g \circ f_{\theta_j}\,,$$
$$C_j : X \to \{0, \ldots, K-1\}\,.$$

The encoder loss at iteration $j+1$ is then given by

$$\mathcal{L}_j = \sum\nolimits_{M_j} CE(h(f_\theta(x_i)), C_j(x_i)) + \sum\nolimits_{S_j} CE(h(f_\theta(x_i)), C_j(x_i))\,,$$

where $h : Z \to (0,1)^K$ is the softmax classifier used for pseudo-label training (it is discarded after training), CE is the categorical cross-entropy loss,

$$S_j \coloneqq \{x \in X | C_j(x) = C_{j-1}(x)\}\,,$$

$$M_j \coloneqq \{x \in X | \forall 1 \leq k < j, C_j(x) = C_k(x)\}\,.$$

Here, S_j and M_j correspond to $SemiMask$ and $Mask$ in Algorithm 1, respectively, and allow for our graded label filtering. Note that $M_j \subset S_j$.

Existing deep HAR clustering models all require an autoencoder to generate feature vectors for clustering (Sheng and Huber, 2020; Ma et al., 2021; McConville et al., 2021). This effectively doubles the time and space requirements, as a decoder must be trained in conjunction with the encoder. Autoencoders can also present some problems for clustering, as they learn to reconstruct every detail of the input, including irrelevant features and noise. This has been well-documented in the case of image clustering, and becomes a greater problem the larger the input is; see (Mrabah et al., 2020) and the references therein for a full discussion. Our method, in contrast, uses a single streamlined loss, which does not require a decoder, and thus can scale better to richer datasets with more sensors, both in terms of computational costs and accuracy. This is supported by the results from Section 5.

Before clustering the latent space, we apply UMAP (uniform manifold approximation (McInnes et al., 2018)), as a second round of dimensionality reduction, reducing the latent dimension from 32 to 2, and cluster with a hidden Markov model (HMM) to capture temporal consistency. As with previous HAR clustering models, the number of clusters is set

manually to be equal to the number of classes in the dataset. The label filtering method described above identifies, each time it is run, a subset of confident labels. It can thus be repeated a number of times. If, in any of the repetitions, a data point received a confident label, then the output of our model for that data point is its most recent confident label. Otherwise, the output of our model is the label from the final repetition. We iterate until the set of points that have ever received a confident label stops increasing.

A final feature of our model is that, for the five smaller datasets, we reduce the step size from 100 (as is standard) to 5, to increase the number of data points. This reduction provides 20 times more data, which improves the training of the encoder, cf. Table 7. However HHAR, having by far the most data points, but comparatively few sensors (12, compared to, e.g., 51 for PAMAP or 117 for REALDISP), does not require additional data, so we keep the original step size of 100. The entire method is described in Algorithm 1.

To evaluate the four settings discussed in Section 3.2 and presented in Table 2, we use a pared-down version of our model. This pared-down version differs from the above in that it removes UMAP and label filtering, and does not decrease the step size. That is, it simply encodes the data using the same convolutional architecture as the main model, with a step size of 100, clusters the encodings using an HMM, and then performs pseudo-label training.

5. Experimental Evaluation

Metrics. We use four metrics: clustering accuracy (ACC), adjusted Rand index (ARI), normalized mutual information (NMI), and macro-F1 (F1). After aligning predicted labels to the ground-truth labels via the alignment that maximizes accuracy, ACC and F1 are computed as in the supervised setting. ARI and NMI are computed by:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right]\Big/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right]\Big/\binom{n}{2}}$$

$$NMI = \frac{2\sum_i\sum_j n_{ij}\log\frac{nn_{ij}}{n_i n_j}}{\sum_i n_i \log\frac{n_i}{n} + \sum_j n_j \log\frac{n_j}{n}},$$

where n_j is the number of data points in ground-truth class j, as indicated by the labels in the dataset, n_i is the number of data points in cluster i, n_{ij} is the number of data points in ground-truth class j and cluster i, and n is the total number of data points.

We choose this set of metrics, because it covers three interpretations of clustering. ARI measures performance with respect to the standard interpretation of finding a partition that maximizes intra-cluster similarity and minimizes inter-cluster similarity. Calling points with the same ground-truth label similar, and those with different labels dissimilar, ARI measures what fraction of the time similar pairs are placed together and different pairs placed separately. NMI is based on the interpretation of clustering as compression. An accurate clustering should be able to encode the maximum amount of useful information about a given data point by specifying its cluster assignment, i.e., the cluster labels act as a compression code, replacing the data for each point with a single integer from $0, \ldots, K-1$, where K is the number of clusters. The ground-truth labels are thought of as the ideal compression code, and we then measure the information-theoretic distance between it and

Table 4: Comparison with (McConville et al., 2021) and various supervised methods. The supervised methods are (Dua et al., 2021) on PAMAP, UCI-Sm, and WISDM-v1, (Aljarrah and Ali, 2021) on REALDISP, and (Qin et al., 2020) on HHAR.

		ours	n2d	supervised
PAMAP	ACC	**86.3**	86.0	95.3
	NMI	**88.4**	85.4	-
	ARI	**81.4**	78.3	-
	F1	80.1	**83.7**	95.2
UCI-Smartphone	ACC	**65.9**	52.1	96.2
	NMI	**68.6**	55.8	-
	ARI	**56.3**	38.7	-
	F1	**64.4**	48.5	96.2
WISDM-v1	ACC	**75.3**	70.5	97.2
	NMI	**76.0**	69.1	-
	ARI	**65.5**	58.1	-
	F1	**68.9**	63.4	97.2
WISDM-watch	ACC	**91.7**	84.8	-
	NMI	**93.8**	88.9	-
	ARI	**88.1**	79.0	-
	F1	**92.4**	84.5	-
REALDISP	ACC	**91.0**	80.3	99.8
	NMI	**95.2**	90.4	-
	ARI	**88.9**	79.0	-
	F1	**88.0**	72.5	99.8
HHAR	ACC	**62.3**	59.7	96.6
	NMI	**67.9**	60.8	-
	ARI	**50.5**	46.0	-
	F1	59.0	**59.3**	96.6
UCI-full-feats	ACC	**65.5**	64.9	-
	NMI	59.3	**67.0**	-
	ARI	46.3	**55.1**	-
	F1	64.5	**70.4**	-

Table 5: Comparison of our method with that of (Sheng and Huber, 2020). Ours performs better on both datasets and all metrics, with a more significant difference on the more complex dataset, REALDISP.

		ours	S20
PAMAP	ACC	**86.3**	85.4
	NMI	**88.4**	87.3
	ARI	**81.4**	80.2
REALDISP	ACC	**91.0**	68.1
	NMI	**95.2**	60.5
	ARI	**88.9**	80.4

Table 6: Comparison of our method with that of Ma et al. (2021) on the HHAR dataset. We achieve a significantly higher NMI but a lower F1.

	HHAR	
	NMI	F1
ours	**67.9**	59.0
M21	55.0	**65.9**

the compression code produced by the clustering model being evaluated. ACC and F1 treat clustering as unsupervised classification. They are useful because they enable a direct comparison to supervised classifiers, as in Table 4. For these reasons, and to enable comparison, we recommend that future work uses these four metrics.

Network Architecture and Training Parameters. Our encoder network contains four convolutional layers, with batchnorm and max-pooling of size 2 after each. The filter sizes and strides for the convolutional layers are $(50, 2)$, $(40, 2)$, $(7, 1)$, $(4, 1)$, and the number of filters per layer are $4, 8, 16, 32$. In every layer, the convolutional filters are 1D with weight sharing across sensor channels. After the convolutional layers, all channels are combined with a fully connected layer (with input size $32 \times$ the number of sensor channels, and output

Table 7: Ablation studies on the main components of our model.

		ours	no UMAP	no label-filter	GMM	step 100	net. dim.
PAMAP	ACC	**86.3**	56.0	73.3	78.7	81.6	65.06
	NMI	**88.4**	52.9	76.8	81.0	81.5	64.32
	ARI	**81.4**	39.8	62.72	69.7	72.4	50.11
	F1	**80.1**	52.9	71.4	76.7	78.4	63.69
UCI-Sm	ACC	**65.9**	49.8	60.8	58.7	62.5	55.63
	NMI	**68.6**	52.7	63.8	62.4	67.2	56.38
	ARI	**56.3**	36.9	49.3	47.1	52.1	41.71
	F1	**64.4**	44.8	58.9	58.3	60.6	54.52
WISDM-v1	ACC	**75.3**	68.8	68.6	71.5	69.9	65.72
	NMI	**76.0**	63.2	70.2	74.6	69.2	61.2
	ARI	**65.5**	55.1	56.0	60.9	55.9	51.15
	F1	**68.9**	58.6	60.4	63.7	64.9	57.29
WISDM watch	ACC	**91.7**	69.1	87.8	89.2	88.8	62.73
	NMI	**93.8**	80.8	90.5	92.1	92.6	70.31
	ARI	**88.1**	63.6	81.8	85.0	85.5	51.58
	F1	**92.4**	67.0	88.0	90.4	89.5	61.9
REALDISP	ACC	**91.0**	82.4	82.7	87.7	76.8	51.07
	NMI	**95.2**	91.0	92.1	93.0	91.5	68.73
	ARI	**88.9**	82.7	79.9	84.5	79.1	46.53
	F1	**88.0**	76.0	78.8	84.6	81.2	49.3
HHAR	ACC	**62.3**	57.9	58.5	61.2	-	64.84
	NMI	**67.9**	58.4	61.0	66.7	-	65.32
	ARI	**50.5**	45.8	45.4	50.2	-	52.3
	F1	**59.0**	56.4	54.9	58.6	-	63.68

size 32). Weights are updated by Adam (Kingma and Ba, 2014) with learning rate 1e-3, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay of 0. The latent space has dimension 32. UMAP uses two components, minimum distance 0 and n_neighbours 60. Clustering is performed by a hidden Markov model (HMM), whose emission probabilities are Gaussian distributions with no restrictions on the covariance matrices. The transition probabilities are set to $1-p$ on the diagonal and $\frac{p}{K-1}$ on the off-diagonal, where p is the fraction of time points in the dataset that are followed by the same action. The window size is 512 for all datasets. For information on the step size, see the discussion in Section 4. The softmax classifier used for pseudo-label training is a multi-layer perceptron (MLP) with a single hidden layer of 250 units. After the encoder has been pseudo-label trained, we cluster again. We alternate ten times between clustering and pseudo-label training the encoder on the cluster labels, training for five epochs each iteration.

Comparisons with Prior Work. Tables 4, 5, and 6 compare, respectively, the performance of our model to that of McConville et al. (2021), Sheng and Huber (2020), and Ma et al. (2021). All evaluations are in the pointwise subject-dependent setting. For the latter two, we are restricted to only comparing on certain datasets and metrics, as the authors do not report on all the datasets and metrics that we do, and they do not release their code.

We attempted to reimplement their methods using the details in the respective papers, but were unable to, and where we contacted the authors to ask for access to their code, we received no response. The reasons we test on these six datasets are given in Section 3.1. Table 4 also compares to recent supervised models, to give an indication of the gap between them and clustering models. The supervised models used are shown in the table caption.

We outperform (McConville et al., 2021) on almost all datasets and metrics. The most significant difference is on the most complex dataset, REALDISP (see Table 3). This supports that our method is better able to scale to complex datasets, because it avoids the need for an autoencoder. For comparison with (McConville et al., 2021), we report on both UCI-Sm, which contains the raw sensor signal, and on UCI-feat, which contains statistical features for each window. (Both datasets and their details are available on the UCI repository.) The figures reported by McConville et al. (2021) on UCI-feat are significantly higher than we obtained by running their code, 80.1 and 68.3 for ACC and NMI, respectively. This difference could be due to their reported results using a larger number of clusters than the ground truth, which tends to increase ACC significantly while keeping NMI similar. For fair comparison, we fixed the number of clusters to the ground truth for all methods.

We outperform (Sheng and Huber, 2020) on both PAMAP and REALDISP. The margin is larger for REALDISP, again suggesting that our approach is better able to leverage the more complex information in REALDISP's 117 channels, because it does not use an autoencoder. We outperform (Ma et al., 2021) on NMI but not F1. Partly, this could be due to their reporting micro-F1 (not specified but implied), which tends to be higher. Our micro-F1 score on HHAR is closer at 62.30. This difference between metrics also highlights the value of reporting multiple metrics to capture different aspects of performance.

Ablation Studies. Table 7 shows the results of removing each of the components of our model. UMAP gives an improved performance across all metrics and datasets. This is consistent with its use in image clustering models (Allaoui et al., 2020). We use two different ablation settings, one removes all dimension reduction, the other replaces UMAP with an MLP of hidden size 256, and output size 2. The second setting also shows a general drop of performance, which is larger on datasets with more sensors. Results on HHAR are comparable to using UMAP, likely because, as well as only 12 sensor channels, HHAR has the most data with which the additional network can train. Results on all other datasets are worse. PAMAP and, especially, REALDISP, show a significant drop. This suggests that, for simple datasets, more of the improvement with UMAP is due to it reducing dimension, rather than exactly how it reduces dimension, but for complex datasets with more information to fit into the reduced dimensions, the manner of dimension reduction is more important. Label filtering, the technique of removing likely-incorrect labels from pseudo-label training, so that the less noisy filtered labels can facilitate a better training of the encoder, has been shown to improve clustering performance in prior works (Mrabah et al., 2020; Mahon and Lukasiewicz, 2021). Here, our novel method of label-filtering is also effective, significantly increasing all metrics across all datasets. Clustering with an HMM instead of a GMM markedly improves performance on PAMAP, UCI-Sm, and WISDM-v1. On WISDM-watch and REALDISP, where the metrics are already high, the improvement, though still significant, is less substantial. This suggests that, as the sensor data become richer, the further benefit of temporal information offers less improvement. (The GMM

has full covariance matrices, convergence threshold of .001, maximum EM steps of 100, is warm-started with k-means, and uses five initializations.) The smaller step size, which produces more data points to train the encoder, is effective at improving performance on all datasets. It is most effective on the two datasets with the most sensor channels, PAMAP and REALDISP, as they require larger networks, and hence more training data.

6. Conclusion

In this paper, we articulated and discussed the shortcomings of current evaluation procedures for HAR clustering models. We noted a lack of consistency in previous works' reporting of results, and conducted experiments to show that a common ambiguity in evaluation (namely, subject dependence) can significantly alter the results. We then discussed superior evaluation alternatives. Additionally, we introduced a new deep clustering model for HAR. Tested on six public datasets, under our proposed settings, it performs better than or on par with existing works, while also being more scalable and efficient by avoiding the need for an autoencoder. This paper can serve as a guide for future efforts in deep clustering of human activities, by articulating the requirements for comprehensive evaluation, and by detailing a number of effective techniques with respect to our own model.

Acknowledgments

This work was partially supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1 and by the AXA Research Fund. We also acknowledge the use of the EPSRC-funded Tier 2 facility JADE II (EP/ T022205/1) and of GPU computing support by Scan Computers International Ltd.

References

A. Aljarrah and A. Ali. Human activity recognition by deep convolution neural networks and principal component analysis. *Further Advances in Internet of Things in Biomedical and Cyber Physical Systems*, 2021.

M. Allaoui, M. L. Kherfi, and A. Cheriet. Considerably improving clustering algorithms using UMAP dimensionality reduction technique: A comparative study. In *Proc. International Conference on Image and Signal Processing*, pages 317–325. Springer, 2020.

M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H. P. Tan. Deep activity recognition models with triaxial accelerometers. In *Workshops at AAAI*, 2016.

K. Altun, B. Barshan, and O. Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10), 2010.

D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proc. ESANN*, volume 3, 2013.

O. Baños, M. Damas, H. Pomares, I. Rojas, M. Tóth, and O. Amft. A benchmark dataset to evaluate sensor displacement in activity recognition. In *Proc. UbiComp*, 2012.

M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proc. ECCV*, 2018.

K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM CSUR*, 2021.

Y. Chen and Y. Xue. A deep learning approach to human activity recognition based on single accelerometer. In *Proc. SMC*, 2015.

N. Dua, S. Shiva Nand, and S. Vijay Bhaskar. Multi-input CNN-GRU based human activity recognition using wearable sensors. *Computing*, 2021.

A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano. Hand-crafted features vs residual networks for human activities recognition using accelerometer. In *Proc. ISCT*, 2019.

N. Hammerla, S. Halloran, and T. Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv:1604.08880*, 2016.

H. He, Y. Tan, and J. Huang. Unsupervised classification of smartphone activities signals using wavelet packet transform and half-cosine fuzzy clustering. In *Proc. FUZZ-IEEE*, 2017.

H. He, Y. Tan, and W. Zhang. A wavelet tensor fuzzy clustering scheme for multi-sensor human activity recognition. *Engineering Applications of Artificial Intelligence*, 2018.

Z. He and L. Jin. Activity recognition from acceleration data based on discrete cosine transform and SVM. In *Proc. SMC*, 2009.

S. Hongeng, F. Bremond, and R. Nevatia. Representation and optimal recognition of human activities. In *Proc. CVPR*, volume 1, 2000.

M. Inoue, S. Inoue, and T. Nishida. Deep recurrent neural network for mobile human activity recognition with high throughput. *AROB*, 23(2), 2018.

W. Jiang and Z. Yin. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proc. ACM Multimedia*, 2015.

A. M. Khan, Y. K. Lee, S. Y. Lee, and T. S. Kim. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *Proc. ICFIT*, 2010.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

J. Kwapisz, G. Weiss, and S. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2), 2011.

Y. Kwon, K. Kang, and C. Bae. Unsupervised learning for human activity recognition using smartphone sensors. *Expert Systems with Applications*, 41(14), 2014.

F. Li and S. Dustdar. Incorporating unsupervised learning in activity recognition. In *Workshops at AAAI*, 2011.

Y. Li, D. Shi, B. Ding, and D. Liu. Unsupervised feature learning for human activity recognition using smartphone sensors. In *Proc. MIKE*, 2014.

Y. Lu, Y. Wei, L. Liu, J. Zhong, L. Sun, and Y. Liu. Towards unsupervised physical activity recognition using smartphone accelerometers. *Multimedia Tools and Applications*, 2017.

A. Sawchuk M. Zhang. A feature selection-based framework for human activity recognition using wearable multimodal sensors. In *BodyNets*, 2011.

H. Ma, Z. Zhang, W. Li, and S. Lu. Unsupervised human activity representation learning with multi-task deep clustering. In *Proc. IMWUT*, 2021.

I. Machado, A. Gomes, H. Gamboa, V. Paixão, and R. Costa. Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization. *Information Processing & Management*, 51(2), 2015.

L. Mahon and T. Lukasiewicz. Selective pseudo-label clustering. In *Proc. KI*, 2021.

M. Malekzadeh, R. Clegg, A. Cavallaro, and H. Haddadi. Protecting sensory data against sensitive inferences. In *Workshops at EuroSys*, 2018.

M. McConnell, M. Turakhia, R. Harrington, A. King, and E. Ashley. Mobile health advances in physical activity, fitness, and atrial fibrillation: Moving hearts. *JACC*, 2018.

R. McConville, R. Santos-Rodriguez, R. J. Piechocki, and I. Craddock. N2d: (Not too) deep clustering via clustering the local manifold of an autoencoded embedding. In *Proc. 2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5145–5152. IEEE, 2021.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2018.

L. Mejia-Ricart, P. Helling, and A. Olmsted. Evaluate action primitives for human activity recognition using unsupervised learning approach. In *Proc. ICITST*, 2017.

N. Mrabah, N. Khan, R. Ksantini, and Z. Lachiri. Deep clustering with a dynamic autoencoder: From reconstruction towards centroids construction. *Neural Networks*, 130, 2020.

Z. Qin, Y. Zhang, S. Meng, Z. Qin, and K.-K. R. Choo. Imaging and fusing time series for wearable sensor-based human activity recognition. *Information Fusion*, 53:80–87, 2020.

A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Proc. ISWC*, 2012.

C. A. Ronao and S.-B. Cho. Deep convolutional neural networks for human activity recognition with smartphone sensors. In *Proc. NeurIPS*, 2015.

T. Sheng and M. Huber. Unsupervised embedding learning for human activity recognition using wearable sensor data. In *Proc. FLAIRS*, 2020.

D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger. Human activity recognition using recurrent neural networks. In *Proc. CD-MAKE*, 2017.

A. Stisen et al. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *Proc. SenSys*, 2015.

S. Suh, V. Rey, and P. Lukowicz. Adversarial deep feature extraction network for user independent human activity recognition. *arXiv preprint arXiv:2110.12163*, 2021.

E. Tapia, S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Proc. PerCom*, 2004.

D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat. An unsupervised approach for automatic activity recognition based on hidden Markov model regression. *T-ASE*, 2013.

J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119, 2019.

G. Weiss, K. Yoneda, and T. Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7, 2019.

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 *Proc. IJCAI*, 2015.