On the Role of Features in Human Activity Recognition

Harish Haresamudram
School of Electrical and
Computer Engineering,
Georgia Institute of Technology
Atlanta, GA, USA
hharesamudram3@gatech.edu

David V. Anderson
School of Electrical and
Computer Engineering,
Georgia Institute of Technology
Atlanta, GA, USA
anderson@gatech.edu

Thomas Plötz
School of
Interactive Computing,
Georgia Institute of Technology
Atlanta, GA, USA
thomas.ploetz@gatech.edu

ABSTRACT

Traditionally, the sliding window based activity recognition chain (ARC) has been dominating practical applications, in which features are carefully optimized towards scenario specifics. Recently, end-to-end, deep learning methods, that do not discriminate between representation learning and classifier optimization, have become very popular also for HAR using wearables, promising "out-of-the-box" modeling with superior recognition capabilities. In this paper, we revisit and analyze specifically the role feature representations play in HAR using wearables. In a systematic exploration we evaluate eight different feature extraction methods, including conventional heuristics and recent representation learning methods, and assess their capabilities for effective activity recognition on five benchmarks. Optimized feature learning integrated into the conventional ARC leads to comparable if not better recognition results as if using end-to-end learning methods, while at the same time offering practitioners more flexibility to optimize their systems towards specifics of wearables and their constraints and limitations.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**; • **Computing methodologies → Artificial intelligence**; **Supervised learning by classification**.

KEYWORDS

Activity Recognition; Feature Extraction; Machine Learning

ISWC '19, September 9–13, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6870-4/19/09...$15.00
https://doi.org/10.1145/3341163.3347727

ACM Reference Format:
Harish Haresamudram, David V. Anderson, and Thomas Plötz. 2019. On the Role of Features in Human Activity Recognition. In *ISWC 2019: International Symposium on Wearable Computers, September 11–13, 2019, London, UK.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3341163.3347727

1 INTRODUCTION

Human activity recognition (HAR) – which involves the automated inference of what people do and when – constitutes a central aspect of wearable computing. Since its inception, a multitude of sensing modalities have been explored and are used to capture human activities. Essentially, HAR utilizes body-worn sensors to record movement data, which are then analyzed through a combination of signal processing and machine learning techniques. The goal is to recognize–i.e., segment and classify–either finite sets of activities of interest, or to study them in an open-ended manner.

In [6], the Activity Recognition Chain (ARC) is defined as a "sequence of signal processing, pattern recognition and machine learning techniques that implements a specific activity recognition system behavior". Traditionally, HAR has been based on (variants of) the ARC, which describes a processing pipeline from sensory data to the classification of portions (segments) of subsequent readings into activities of interest (or the null class). In recent years, end-to-end learning approaches have been adopted for HAR using wearables, primarily due to their promise of integrated learning – which would effectively eliminate manual crafting and tuning of suitable data representations. This advantage, coupled with astonishing classification capabilities and transfer learning has made end-to-end learning models the de facto approach studied in the last few years. Although end-to-end learning models seem to outperform the conventional ARC, the performance boost comes with the price of requiring substantial computational resources (even though considerable progress has been made in the efficient deployment of deep learning models on resource-constrained devices [1, 5, 11, 26]), and considerable training sets of *annotated* sample data.

No clear consensus exists regarding the *gold standard* feature representation for HAR using wearables, yet the state-of-the-art can broadly be summarized in three categories: *i) Statistical features*, which include a host of heuristics such as the mean and standard deviation, or spectral properties that describe the underlying signal in general, but do not carry any domain related knowledge, often requiring tweaking across tasks and domains [23, 37]; *ii) Distribution-based representations*, which completely abstract away from the application domain and instead focus on compact signal representations, thereby minimizing reconstruction loss [19]; and *iii) Learned features*, that derive representations directly from raw sensor data themselves, typically using unsupervised or supervised learning methods and dimensionality reduction techniques [2, 17, 37].

With the arrival of end-to-end learning approaches in the field of HAR, the central question this work seeks to answer is: *What role do the data representations play in human activity recognition?* With a view on state-of-the-art representation learning methods, we explore to what extent the representation (rather than the classification backend) still contributes to the overall effectiveness of HAR methods, thereby not limiting our analysis to either the more conventional ARC approach, or to Deep Learning based methods. Exploring this question enables us to draw conclusions that impact system design for wearable computing scenarios that typically come with substantial resource constraints, and challenging data qualities due to the nature of the recording apparatus.

We study HAR from a feature perspective, and include the three aforementioned categories of representations into our analysis. By considering factors central to wearable computing, such as memory footprint, and feature dimensionality, we draw conclusions regarding the properties of the representations from our experiments. We offer insights as well as guidelines to practitioners and researchers in HAR. The conclusions are drawn based on the recognition accuracy demonstrated by the representations, the impact on resources they cause, and the suitability of the representations for different activities. The recognition accuracy studies the absolute performance of the representations, while the impact on resources analyzes the applicability of the representations in the resource constrained scenarios prevalent in HAR. Finally, the suitability of representations examines which representations work for target activities. Put together, these considerations allow us to obtain optimal representations for specific recognition tasks (and activities) by accounting for the trade-off between resource requirements and performance. The contributions of this paper are as follows:

- We introduce novel convolutional and recurrent autoencoder models to explore the extent to which unsupervised sequence modeling methods can advance the state-of-the-art in feature learning for HAR [37].
- Feature representations are evaluated on standard benchmark datasets that are diverse in terms of the number of subjects, kinds of activities performed, goals, settings, and the number of samples.
- By considering factors integral to wearable computing (memory footprint, amount of training data required, number of trainable parameters), we develop insights into designing more optimal recognition systems.
- Through our experiments we demonstrate that optimized feature learning integrated into the conventional ARC leads to comparable, if not better activity recognition offered by end-to-end learning models yet with more flexibility for optimization w.r.t. constraints and limitations inherent to wearable computing.

2 FEATURE EXTRACTION IN HAR

The investigation and understanding of sensor data properties is key to finding a representation that directly captures its core characteristics. In HAR, there is no all-encompassing model that affords the expert-driven design of a universal feature representation that would explain the underlying physical phenomenon that HAR addresses [16]. As such, the state-of-the-art for HAR features comprises more or less descriptive representations of the raw signal, which are often driven by heuristics. Alternatively, recent developments in machine learning, especially deep learning have the potential of overcoming this shortcoming by automatically *learning* relevant feature representations for sensor data.

Conventional feature extraction methods aim at finding compact and descriptive representations of sensor data thereby typically not exploiting any actual domain knowledge. A wealth of heuristics has been developed that aim at extracting either time-domain features, such as statistical moments, or spectral features that directly encode frequency characteristics [9]. Alternatively, direct encodings of temporal aspects of the sensor data have been used, e.g., through time-delay embedding [10], or through discretization of the time-series (e.g., [28, 30]). Apart from these, classic dimensionality reduction techniques such as PCA have been used in HAR [37]. Distribution based approaches represent the state-of-the-art for conventional feature extraction in HAR, where quantiles of the (inverted) empirical cumulative density function represent sensor readings within an analysis frame [19, 25].

In contrast to heuristic feature design, feature learning optimizes an objective function to derive a meaningful data representation. This can broadly be categorized into supervised and unsupervised learning. While we study the performance of both unsupervised and supervised methods, special emphasis is placed on unsupervised methods, as they have the advantage of automatically deriving generalizable features from unlabelled data [21, 37]. Recent progress in general representation learning motivates us to explore the potential of feature learning for HAR beyond the initial introduction using Restricted Boltzmann Machines [37].

Deep networks have the capability to learn meaningful representations without utilizing class labels [21]. This includes approaches such as generative modelling, autoregressive networks, and self-supervised learning [13, 14]. Generative modelling assumes that the characteristics of the data can be discovered by learning how to generate them, and that subsets of the characteristics are then suitable for differentiating between classes [20]. They have been applied, for example, to learn unsupervised representations for video classification [40], for audio scene classification [3], and for obtaining vector representations for words in NLP [29]. Autoregressive networks split high dimensional data into a sequence of small pieces and predict each piece from those before. They have been used, e.g., for generating raw audio [42]. Self-supervised learning utilizes domain expertise to define a prediction task which requires semantic understanding [14]. They are utilized to learn features by predicting geometric transformations, and solving jigsaw puzzles [24].

The state-of-the-art in HAR consists of end-to-end learning approaches that do not explicitly distinguish between representation learning and optimizing a classification function. In [44], convolutional neural networks (CNNs) were developed for multichannel time series data. Similarly, the authors in [46] designed a CNN, which outperformed other representations based on distributions, PCA, statistical metrics, and fully connected neural networks. To utilize the time-series nature of the sensor data, ensembles of recurrent neural networks (RNN) have been used in [15] for performing HAR. The application of recurrent neural networks for HAR is extended in [45], where continuous attention mechanisms over the sensory channels as well as time are developed to improve the performance. DeepConvLSTM [34] uses a combination of convolutional and recurrent layers. Improvements to the DeepConvLSTM have been proposed in [32], where an attention mechanism is added on the long short-term memory (LSTM) network to determine the 'important' time steps. While the supervised models provide excellent performance, they come with the cost of requiring large amounts of *annotated data* to perform learning. This is both time and cost intensive, as it requires domain expert knowledge. We aim at specifically assessing the role of features in the complete HAR pipeline, hence end-to-end learning is discussed here mainly for completeness. However, when discarding the final layer of such complex, deep neural networks the activations of the penultimate layer can be used as features.

3 DATASETS

Our explorations are based on benchmark datasets that represent the state-of-the-field in HAR. All datasets were either recorded with 33Hz sampling rate or downsampled accordingly. Sliding window segmentation obtained frames of 1s length and 50% overlap between subsequent windows.

Opportunity

Opportunity contains recordings from four participants wearing a range of sensors while pursuing kitchen routines [7]. We use annotations that are provided for 18 mid-level activities. For training and evaluation, we employ the same protocol as [18]: The second run from participant 1 is used for validation, while runs 4 and 5 from participants 2 and 3 are used for test. The rest of the data is used for training.

Skoda

This dataset was recorded in a manufacturing scenario aiming to recognize activities of assembly-line workers in a car production environment [41]. Ten quality checks were recorded using 10 body-worn accelerometers ($D = 60$). The training set consists of the first 80% of each class, followed by validation and test sets taking up a remaining 10% each.

PAMAP2

PAMAP2 contains a total of 12 activities of daily living such as domestic activities, and various sportive exercises (nordic walking, running, etc) [39]. Over 10 hours of data were collected using a range of body-worn sensors ($D = 52$). Replicating the protocol from [18], we used runs 1 and 2 from participant 5 for validation and runs 1 and 2 from participant 6 for testing. The remaining data is used for training.

USC-HAD

The USC-HAD dataset [47] was collected on the MotionNode sensing platform and consists of data from 14 subjects. Twelve activities were recorded and they include various walking motions, jumping, sitting, etc. Participants $1 - 10$ form the training set, while participants 11 and 12 form the validation set, and participants 13 and 14 comprise the test set.

Daphnet Freezing of Gait Dataset

This dataset contains data from ten subjects with Parkinson's Disease, who experience freezing of gait (FoG) in daily life [4]. Data were recorded using three body-worn 3D accelerometers ($D = 9$). More than eight hours of data were recorded in which physiotherapists identified 237 FoG events in a post hoc video analysis. Participants 9 and 2 form the validation and test sets, while the rest of the data used for training.

4 METHODOLOGY

The standard activity recognition chain (ARC, [6]) defines a series of processing steps, where each step has a clearly defined goal. However, substantial manual optimizations are necessary for each part of the pipeline – with known issues such as poor generalization, thereby causing the need for specialized domain knowledge which hinders widespread adoption. On the other hand, this process can be heavily optimized with respect to computational resources (such as memory and processing power). In this manner, the pipeline can be deployed with ease *on* wearables themselves, without the need for offloading the computation to external servers.

In this study, we focus on the fourth step of the pipeline – Feature Extraction. A number of previous works exist, which utilize statistical features, data-driven representations like PCA or distribution-based representations. Papers such as [37] and [2, 43] study deep-learning based RBMs and stacked autoencoders. However, little analysis has been done towards utilizing more powerful variants of autoencoders, which, for example, include convolutional and recurrent layers. Three general types of autoencoders exist: *i)* vanilla; *ii)* convolutional; and *iii)* recurrent. In [43] and [2], vanilla (or stacked) autoencoders have already been studied. In order to capture the temporal aspect of the sensor data, we study recurrent autoencoders. On the other hand, the spatio-temporal aspect of data is exploited using convolutional autoencoders.

The performance of these autoencoder-based feature representations is contrasted to state-of-the-art statistical features, a distribution-based representation (ECDF [19]), and a representation extracted using a supervised classifier – DeepConvLSTM [34]. We also compare overall classification capabilities of the explored feature representations within the ARC paradigm [6] to a state-of-the-art end-to-end HAR system that does not separate representation learning from classifier training (DeepConvLSTM [34]).

Statistical features

We utilize the statistical metrics detailed in [36]. They include the DC mean, variance, correlation, energy and frequency domain entropy. The DC mean is the averaged accelerometer reading in a time interval, while the variance characterizes the stability of the signal. Energy captures the periodicity of the signal and the frequency domain entropy helps discriminate between activities of similar energy. The correlation is computed between all pairwise combinations of axes and captures the correlation between different axes.

Distribution-based representation

Sensor data are time-series, i.e., each reading is contextualized by its temporal neighbors. Distribution-based representations take advantage of these correlations by computing the empirical cumulative distribution (ECDF) of the data in each frame. At the heart of ECDF lies the idea to extract a fixed set of real-valued coefficients that best represents the underlying distribution for each degree of freedom within a frame (i.e., each sensing axis of the accelerometer data) [19]. The ECDF representation f_i for a degree of freedom of analysis frame i is obtained by first estimating the ECDF P_c^i:

$$P_c^i = P(X \leq x), \tag{1}$$

which is quantified by selecting d equally spaced, monotonically increasing points $C = p_1 \dots p_d \in [0 \dots 1]$. For each of those points, the value x_k is estimated, for which $P_c^i(x) = p_k$:

$$\begin{aligned} C &= p_i \in \mathbb{R}^d_{[0,1]}, p_i < p_{i+1} && (2) \\ f_i &= x, \exists j : P_c^i(x) = p_j, && (3) \end{aligned}$$

where cubic interpolation is used to obtain each x. The new representation for each analysis frame i then corresponds to the concatenated ECDF representations of each sensing channel. In effect, this process provides an estimate for the quantile function for each of the selected points in C. The d-dimensional representation f_i fully covers the spatial position of a distribution, as well as its overall shape. The only tunable parameter is the number of points at which the inverse of P_c is interpolated, which controls the granularity for capturing the shape of P_c in the final representation [19].

Autoencoder-based unsupervised representations

An autoencoder is an unsupervised neural network that is trained to reconstruct the input after being passed through a series of layers. Internally, an autoencoder has a hidden part h, that consecutively performs linear as well as non-linear transformations to obtain a latent representation of the input. The network consists of two parts: *i)* an encoder function $h = f(x)$, which transforms input data x; *ii)* and a decoder that produces the reconstruction $r = g(h)$. The network is restricted such that h has lower dimensions than x, thereby creating an intentional bottleneck. Thus, an autoencoder is forced to prioritize some aspects of the input data that need to be copied and thereby learns compact representations [12]. Learning aims at minimizing the loss function:

$$L(x, g(f(x))), \tag{4}$$

where L penalizes $g(f(x))$ for being dissimilar from x. Typically, mean squared error (MSE) is used as the loss function and h is used as the latent representation (or the bottleneck feature) for tasks such as classification and clustering.

Vanilla autoencoders. For the simplest form of autoencoders both the encoder and decoder consist of multi-layer perceptrons (MLP). Excluding the bottleneck layer, the encoder consists of three fully connected layers that consecutively reduce the dimensionality of the representation (in our case containing 2048, 1024 and 512 units, respectively). The decoder mirrors the encoder. One frame of data is vectorized and passed as input to the model.

Convolutional autoencoders. Convolutional autoencoders utilize convolutional layers in lieu of the fully connected layers

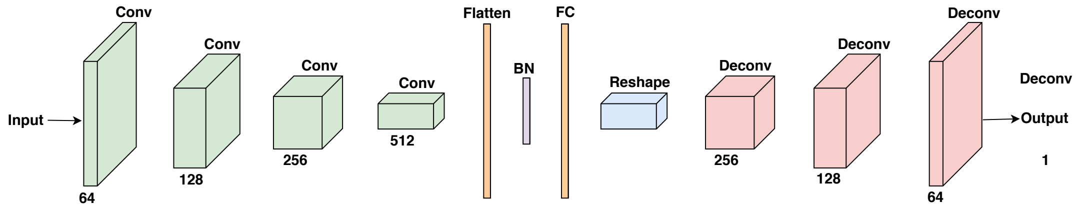

Figure 1: Overview of the convolutional autoencoder used in this study (see text for description).

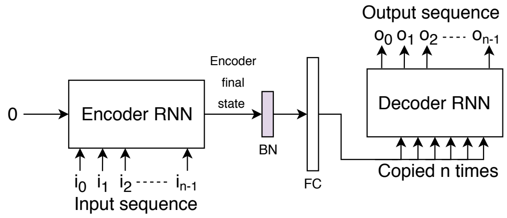

Figure 2: Recurrent autoencoder used in this study.

(cf. Figure 1 for architecture overview). In our case, input consists of individual frames (1*s*), and the autoencoder considers it as a single channel image. The encoder contains four convolution blocks, leading to the bottleneck layer. Each of these blocks contains two 3 × 3 convolution layers with the same number of filters. Batch normalization is performed after each layer, and the convolution layers are followed by 2 × 2 max-pooling. The output of the last convolution block in the encoder is flattened into a vector, and is then connected to the bottleneck layer (from which the latent representations are taken). The decoder inverts the encoding operations step-by-step including convolution, upsampling and interpolation, reshaping, and padding operations to match the sizes of the corresponding encoder convolution blocks [33]. Throughout, ReLU activation functions are used, and hyperbolic tangent is used for the output.

Recurrent autoencoders. In a recurrent autoencoder, Vanilla RNNs or more powerful variants such as long short-term memory networks (LSTM) [22], or gated recurrent units (GRU) [8] are used for both encoder and decoder [40]. In our model (Figure 2), both encoder and decoder are initialized with zeros and the input sequence (length: *n*), is passed to the encoder. The final state of the encoder is passed through the bottleneck layer before being connected to another fully-connected (FC) layer. The resulting output is replicated *n* times, and used as input to the decoder. The loss between the input and output, i.e., reconstructed sequences is used to update the model parameters. In this work, we utilize LSTMs and GRUs in the recurrent autoencoder architectures.

DeepConvLSTM-based supervised representations

The DeepConvLSTM architecture (introduced in [34]) consists of four convolutional layers with 64 filters, and a filter size of 5 × 1. The output of these convolutional layers is connected to a two-layer LSTM with 128 hidden units. The last hidden state of the LSTM is connected to the softmax output layer. We explore the effect of representation dimensionality on the performance, for which we add another FC layer after the LSTM whose dimension can be varied. To compute the DeepConvLSTM-based representations, we perform a forward pass until the penultimate fully-connected layer.

DeepConvLSTM utilizes both convolutional and recurrent layers to model the temporal aspects of time-series sensor data. It offers excellent performance on a variety of HAR datasets, and constitutes the state-of-the-art. Its generally superior performance, coupled with the well studied network architecture make it an excellent candidate for a baseline.

Classifier

We explore the role of the data representation in HAR. As such, we fix the classification backend, namely using a state-of-the-art probabilistic classifier – a Multi-Layer Perceptron (MLP). Our MLP classifier has two layers, followed by the softmax output layer. These layers contain 2048 and 512 units respectively, and each layer is followed by batch normalization. The activation function used is ReLU.

The choice of utilizing an MLP classifier is rooted in its superior performance when compared to other classifiers, such as ones based on support vector machines and random forests, especially at higher dimensions. Additionally, popular deep learning frameworks such as Tensorflow and Pytorch facilitate the easy utilization of graphical processing units to massively parallelize computations, thereby reducing the time taken for evaluation (when compared to the other classifiers, which are computed on the CPU).

Performance metrics

The mean f1-score is utilized as the core metric. The datasets used in this study, Opportunity in particular, are imbalanced and hence require performance metrics that are less prone to be negatively affected by biased class distributions [38].

5 RESULTS

Using the methodology outlined in the previous section, we conducted our exploration study on the benchmark datasets

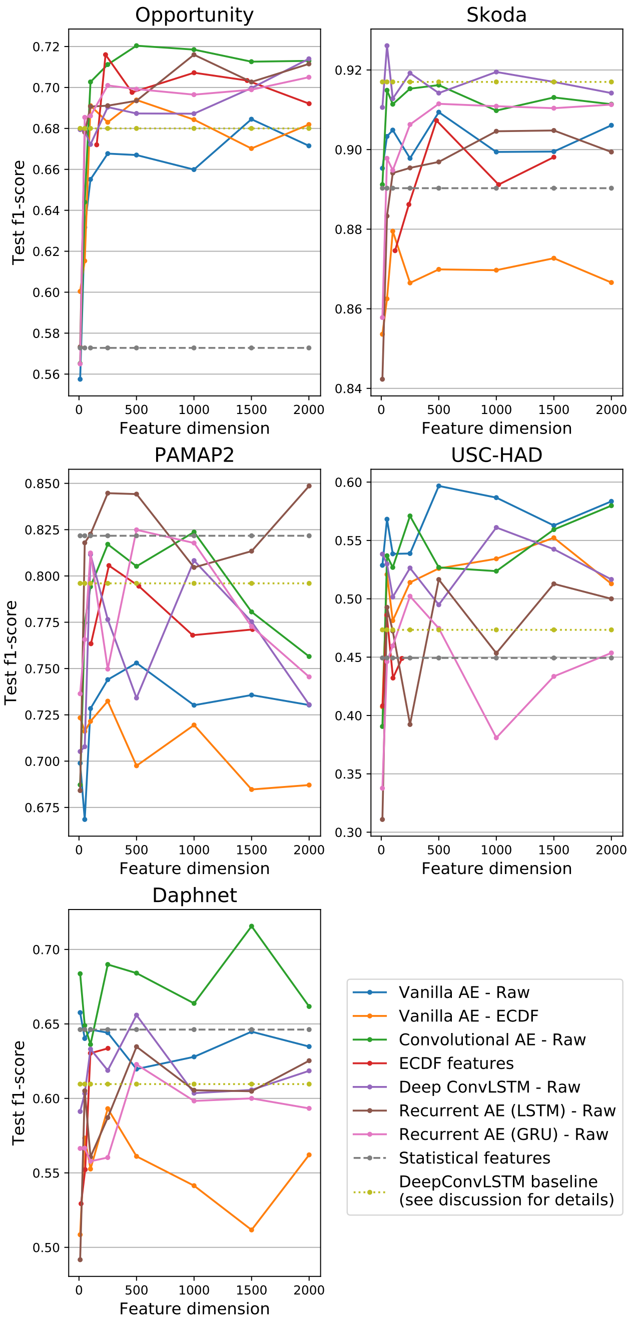

Figure 3: Classification performance (F1) results.

as summarized before. We extracted the variants of features, and then trained classification systems according to the protocols as defined for the respective benchmarks. The choice of data representation is of crucial importance for the effectiveness of a HAR system. First and foremost, the features directly impact its classification capabilities. While the absolute performance of a HAR system is vital, a number of additional factors unique to HAR need to be considered. These factors involve limitations and restrictions of wearable systems such as memory requirements, computation power, computation time, and the amount of training data needed. Any study of the design of wearable recognition systems is incomplete without the consideration of these factors by designers and practitioners as they dictate the practical effectiveness and implementation of these systems *on* wearable devices. In what follows, we analyze the implications of the different feature extraction paradigms on these aspects:

Dimensionality: We study how the classification performance is affected by the dimensionality of data representation. This is an important factor for HAR as lower feature dimensions not only result in lower computational costs and computational time during classification, but also in alleviated demands on size of datasets for training and validating the overall HAR system. In wearable computing it is particularly challenging to obtain large amounts of annotated sample data, and thus reducing dimensionality helps utilizing training data more economically.

Memory footprint: Onboard memory on wearables is limited, and hence, the memory required to store models used to compute the representations is a vital factor.

Number of trainable parameters for deriving data representations: Learned features are the most promising methods with regard to overall classification capabilities and generalization capabilities across different domains. The number of trainable parameters of such feature learners is directly linked to the overall effort required to compute the representations, which has implications specifically for interactive scenarios in which features need to be extracted on-the-fly, and models potentially be adapted in real time. Note that the number of trainable parameters is an indication of the complexity of a model and not necessarily (nor exclusively) linked to the dimensionality of the features.

Dependence on amount of training data: The size of the datasets in HAR is generally smaller than in other domains, which is reasoned by practicalities of how data from wearables is recorded and annotated [31]. Hence, representations which require less data to perform comparably are more suitable for HAR. Some learning methods require more (or less) variability in the training data in order to derive robust representations and as such the evaluation of required sample set size for fixed feature dimensionalities and model complexities is an important aspect of our exploration.

Dimensionality

Figure 3 illustrates the classification capabilities of our MLP-based HAR system for the five considered datasets, utilizing the eight different feature representations. Statistical features have constant dimensionality, which is listed in the diagrams. Dashed lines across the panels are only added for better comparability. The dimensionality of the distribution-based representation corresponds to the number of components computed (Equation 1). We set this number such that the resulting feature dimensions match the dimensions of variants of the learned features . However, the number of ECDF components is limited by the length of the frame and thus the number of samples considered (in our case: 30). Thus, for USC-HAD and Daphnet FoG, the maximum number of dimensions possible is limited to 250 (Figure 3). The plots of the learned feature representations illustrate substantial fluctuation of classification performance depending on feature dimensionality with variation across the different datasets.

For Opportunity, i.e., a dataset that contains non-repetitive activities with substantial variation in duration and overall apperance [27], the Convolutional AutoEncoder (CAE) provides the best performing features with about 500 dimensions. DeepConvLSTM and Recurrent AE (LSTM) only match the performance of the CAE at 2000 dimensions, i.e., at a much higher dimensionality, which renders them less attractive for many wearable computing scenarios with resource constraints. The distribution-based representation obtains similar performance as the CAE at 250 dimensions. Thus, CAE and ECDF provide excellent performance at a fraction of the number of dimensions that other representations require.

The results are very different for Skoda, a dataset with fewer activities and less variability overall [27]. DeepConvLSTM provides the best performance – already with 50-dimensional features. CAE and Recurrent AE provide similar performances at this low dimensionality. The worst performance is shown by the Vanilla AE on the ECDF feature.

For PAMAP2, a dataset that contains short, repetitive activities, relatively low-dimensional representations ($D = 250$) lead to peak performance when using a recurrent AE. The distribution-based representation performs best at a similar dimensionality, but with around 5% lower F1-score. Other representations require $\geq 1,000$ dimensions to perform well.

For USC-HAD, a dataset that contains activities that are not as repetitive as the ones covered by PAMAP2 but at the same time less complex than what is covered by Opportunity, 500-dimensional features learned with a Vanilla AE on raw data lead to best HAR performance. Interestingly, the more complex feature learners require much higher-dimensional features ($D = 2,000$) to achieve comparable performance, and both statistical and distribution-based features lead to substantially worse classification results.

Daphnet-FOG is a dataset that essentially contains only one target class that exhibits substantial variability but is fundamentally different from all other activities performed by the participants. FOG episodes differ in duration but tend to be short overall. DeepConvLSTM and Vanilla AE on raw data achieve similar performances with comparably low-dimensional representations ($D = 100$). ECDF based recognizers also peak at around this dimensionality of the extracted features. In contrast, the Convolutional AE outperforms all other models even with only 10 dimensional representations (yet its peak performance requires much higher-dimensional features, $D = 1000$).

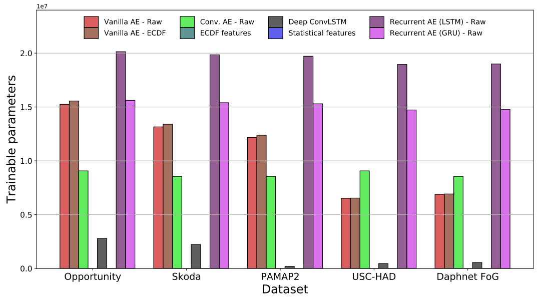

Figure 4: Number of trainable parameters required to compute the representations on different datasets.

Number of Trainable Parameters

All feature learners contain a number of trainable parameters, such as the computation nodes in the various layers of the variants of autoencoders as studied in this paper. The distribution-based representation does not have any trainable parameters, and neither does the statistical feature representation. Trainable parameters can be interpreted as an advantage because they allow for more complex and thus potentially more effective feature extraction, or can be considered a liability because they explicitly require (substantial amounts of) training data, which are often difficult to obtain. They also may be prone to overfitting and poor generalization. Figure 4 gives an overview of the number of trainable parameters in the explored variants of feature extraction methods. Not surprisingly, the recurrent AE comprise the largest number of trainable parameters, followed by the vanilla AE (across all datasets). Interestingly, the complexity of vanilla AE varies significantly and correlates with the complexity of the datasets they are trained for, whereas the number of trainable parameters remains approximately constant for the recurrent and convolutional AE.

Memory footprint

Figure 5 details the amount of memory required, e.g., onboard a wearable, for the representation learning. Since the distribution-based representation and the statistical features are computed directly, the bar plot shows a zero (gap). The Recurrent AE (GRU and LSTM) require the highest amount of memory (around 40 MB in our experiments). In comparison, the Convolutional AE requires only half as much memory. Among the learned features, the DeepConvLSTM

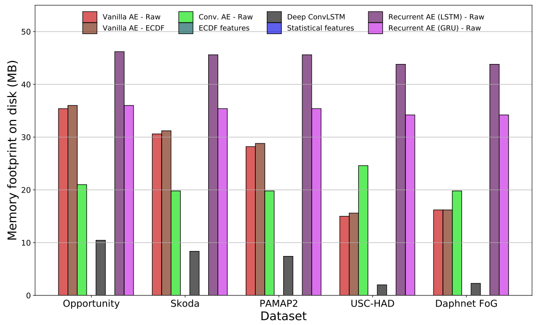

Figure 5: Amount of memory required to store models for computing representations on different datasets.

based features require the least amount of memory. Thus, in terms of memory requirements, the distribution-based and DeepConvLSTM-based representations present the best option. Additionally, while the memory requirement for the Convolutional AE is higher, the potential improvements in performance render it a good alternative.

Dependence on Amount of Training Data

Representations resulting in good HAR performance even when only small datasets are available are critical specifically for wearable computing. As such, the dependency on large amounts of training data for deriving the feature representations is an important aspect of our explorations. For fair comparison, we fix the feature dimensionality for this evaluation to a common number ($D = 500$, which represents a reasonable enough compromise across the different models and datasets) and study how the classification performance changes when reducing the amount of training data. For these evaluations we gradually reduce the amount of training data by randomly removing percentages of samples from the original datasets. We then train the systems on the reduced datasets and report classification performance (F1).

Figure 6 illustrates the results for all five datasets and all eight variants of feature extraction. For the more complex datasets such as Opportunity and Skoda, the general trend is that a strong dependency on large enough datasets exists. All feature extraction methods suffer from smaller training sets as manifested by the substantial drop in classification performance using the resulting HAR systems. For the other datasets that contain more repetitive and shorter target activities (PAMAP2, USC-HAD, Daphnet FOG) such a dependency is not as clear. While there are substantial differences between the various feature extraction methods, the relative changes in resulting classification performance do not change much for the individual systems when reducing the amount of training data. Some outliers to this general trend are noticeable, such as the rather "erratic" behavior of the Vanilla AE trained on ECDF input for both PAMAP2 and Daphnet-FOG. Presumably, the distribution based input representation somewhat counteracts the representation learning process of the autoencoder. Overall, the results are encouraging for application scenarios that target the analysis of less complex activities as it is often the case for wearable computing, such as in automatically logging activities of daily living. Even more complex feature learners

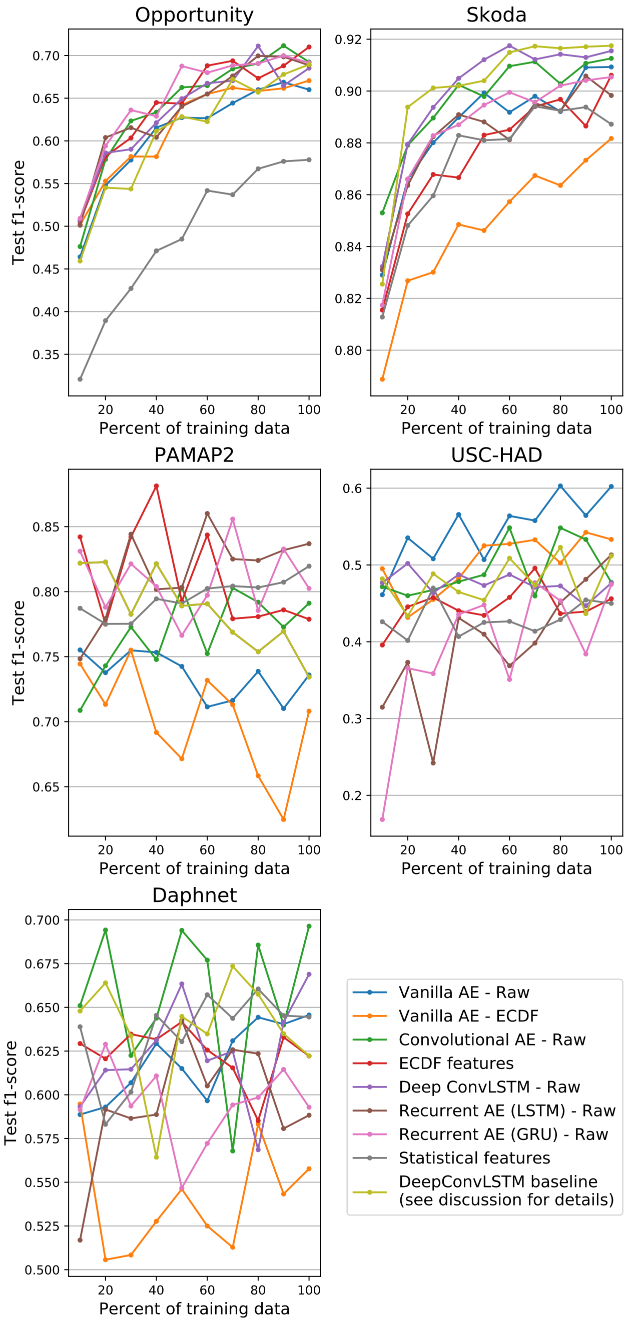

Figure 6: Classification performance (F1) related to percentage of sample data available for deriving features.

can be used–if needed, cf. the excellent recognition capabilities of more conventional features such as the distribution based representation–and do not require substantial amount of training data. For more complex activity recognition (as in Opportunity) substantial amounts of training data are required if feature learners are used. However, it is not clear whether such complex models are actually needed given that non-learned, distribution-based features already lead to effective classification already.

6 DISCUSSION

Arguably, proper data representation, i.e., features play an important role in the Activity Recognition Chain (ARC), as it is widely used in wearable computing [6]. In the past much work has been invested into feature design, yet with limited success towards developing a thorough understanding for how to capture the relevant physical phenomena underlying human activities as they are captured using body-worn sensors. This work aimed at systematically exploring and quantifying the impact contemporary feature representations have on human activity recognition scenarios in wearable computing. In what follows, we draw conclusions from our study and offer insights for HAR researchers as well as guidelines for practitioners of HAR in wearable computing. The focus of our considerations is on: *i)* overall recognition accuracy; *ii)* impact on resources (and their constraints); and *iii)* suitability for different categories of target activities.

Recognition Accuracy

The past few years have seen an explosion of research into end-to-end, typically deep, learning methods [12], with spectacular improvements in classification results in very challenging domains such as computer vision or natural language processing. The wearables community also has adopted such techniques and remarkable recognition results on benchmark dataset can now be achieved–offline–using variants of end-to-end learning methods, e.g., [15, 34, 46]. Notably, the focus of model developments here is often on recognition performance alone, ignoring challenges inherent to wearable computing in terms of availability and quality of annotated training data, and substantial resource constraints [35].

As part of our explorations we have compared the various variants of feature extraction schemes and their effectiveness within the conventional activity recognition chain (ARC) to state-of-the-art end-to-end learning. Remarkably, the differences between ARC (using sophisticated feature extraction methods) and ConvLSTM models (as an example of state-of-the-art deep learning approaches) are far less than what one might have expected and often ARC based systems with specifically optimized (learned) features outperform the end-to-end learning systems (see yellow dotted lines in Figures 3 and 6). As a consequence, ARC –with reasonable feature extraction schemes–has its place in wearable computing.

Dealing with Resource Constraints

Typical wearables scenarios have strict resource constraints with regards to memory and computational power, and most importantly battery power, which often prevents the application of deep learning models. For such scenarios statistical and especially distribution-based data representation remain attractive options with recognition capabilities that are comparable to latest feature learning methods as explored in this paper. The distribution-based representation performs better even with very small amounts of sample data.

For less constrained scenarios, feature learning represents an attractive alternative to distribution based representations, outperforming these on most datasets. Interestingly, integrating learned features into the ARC leads to results that are comparable to those achieved when using end-to-end learning methods that do not explicitly discriminate between representation learning and classifier optimization. This is encouraging, because the ARC typically requires substantially fewer resources than end-to-end learned models.

Suitability for Different Categories of Activities

Despite the desire for generalization, so far no single, universally optimal feature extraction method has been developed for HAR using wearables. As such, it remains relevant to explore which representations are suitable for which categories of target activities. Convolutional AE based representation lead to best performance for scenarios with large variations in duration and overall appearance of the target activities (e.g., Opportunity). The convolutional layers are better suited to capture underlying spatio-temporal relations in such data.

For shorter, more repetitive activities (such as in PAMAP2), less complex models can be used for feature learning. Interestingly, state-of-the-art recurrent autoencoders lead to better performing features for such scenarios, which is somewhat in contrast to what has been reported previously [18].

7 CONCLUSION

We have explored the role of data representations in HAR using wearables. We conclude that the conventional activity recognition chain (ARC) in combination with state-ot-the-art feature learning methods lead to recognition systems that show comparable if not better performance to those obtained from latest end-to-end deep learning systems, which highlights the importance of suitable feature representations. This is encouraging, because it allows designers of wearable computing systems to specifically optimize components of the ARC for suitability in a wearable system, which the monolithic end-to-end learning architectures do not. From these systematic studies, we have identified three key aspects of HAR using wearables that allow practitioners to choose suitable feature extraction schemes.

REFERENCES

[1] M. Alizadeh and N. D Lane. 2018. Using Pre-trained Full-Precision Models to Speed Up Training Binary Networks For Mobile Devices. In *Proc. Int. Conf. Mobile Systems, Applications, and Services.*

[2] B. Almaslukh, J. AlMuhtadi, and A. Artoli. 2017. An effective deep autoencoder approach for online smartphone-based human activity recognition. *Int. J. Comput. Sci. Netw. Secur* 17 (2017), 160.

[3] S. Amiriparian, M. Freitag, N. Cummins, and B. Schuller. 2017. Sequence to sequence autoencoders for unsupervised representation learning from audio. In *Proc. DCASE 2017 Workshop.*

[4] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Troster. 2010. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine* 14, 2 (2010), 436–446.

[5] S. Bhattacharya and N. D. Lane. 2016. Sparsification and separation of deep learning layers for constrained resource inference on wearables. In *Proc. ACM Conf. Embedded Network Sensor Systems.*

[6] A. Bulling, U. Blanke, and B. Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *Comput. Surveys* 46, 3 (2014), 33.

[7] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. R. Millán, and D. Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. 2015. Gated feedback recurrent neural networks. In *Proc. Int. Conf. Machine Learning (ICML).*

[9] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso. 2010. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing* 14, 7 (2010), 645–662.

[10] J. Frank, S. Mannor, and D. Precup. 2010. Activity and gait recognition with time-delay embeddings. In *Proc. AAAI Conf. on Art. Intelligence (AAAI).*

[11] P. Georgiev, N. D. Lane, C. Mascolo, and D. Chu. 2017. Accelerating mobile audio sensing algorithms through on-chip gpu offloading. In *Proc. Int. Conf. Mobile Systems, Applications, and Services.*

[12] I. Goodfellow, Y. Bengio, and A. Courville. 2016. *Deep Learning.* MIT Press.

[13] A. Graves and M. Ranzato. 2018. Unsupervised Deep Learning Tutorial – Part 1, NeurIPS 2018. https://ranzato.github.io/publications/tutorial_deep_unsup_learning_part1_NeurIPS2018.pdf

[14] A. Graves and M. Ranzato. 2018. Unsupervised Deep Learning Tutorial – Part 2, NeurIPS 2018. https://ranzato.github.io/publications/tutorial_deep_unsup_learning_part2_NeurIPS2018.pdf

[15] Y. Guan and T. Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1, 2 (2017), 11.

[16] N. Hammerla. 2015. Activity recognition in naturalistic environments using body-worn sensors.

[17] N. Hammerla, J. Fisher, P. Andras, L. Rochester, R. Walker, and T. Plötz. 2015. PD disease state assessment in naturalistic environments using deep learning. In *Proc. AAAI Conf. on Art. Intelligence (AAAI).*

[18] N. Hammerla, S. Halloran, and T. Plötz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proc. Int. Joint Conf. on Art. Intelligence (IJCAI).*

[19] N. Hammerla, R. Kirkham, P. Andras, and T. Ploetz. 2013. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *Proc. Int. Symp. Wearable Computing (ISWC).*

[20] G. Hinton. 2007. To recognize shapes, first learn to generate images. *Progress in brain research* 165 (2007), 535–547.

[21] G. Hinton, S. Osindero, and Y. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.

[22] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[23] T. Huynh and B. Schiele. 2005. Analyzing features for activity recognition. In *Proc. Int. Joint Conf. Smart objects and Ambient Intelligence.*

[24] Longlong Jing and Yingli Tian. 2019. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. *arXiv preprint arXiv:1902.06162* (2019).

[25] H. Kwon, G. Abowd, and T. Ploetz. 2018. Adding structural characteristics to distribution-based accelerometer representations for activity recognition using wearables. In *Proc. Int. Symp. Wearable Computing (ISWC).*

[26] N. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, and F. Kawsar. 2017. Squeezing deep learning into mobile and embedded devices. *IEEE Pervasive Computing* 16, 3 (2017), 82–88.

[27] H. Li, G. Abowd, and T. Ploetz. 2018. On specialized window lengths and detector based human activity recognition. In *Proc. Int. Symp. Wearable Computing (ISWC).*

[28] J. Lin, E. Keogh, L. Wei, and S. Lonardi. 2007. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery* 15, 2 (April 2007), 107–144.

[29] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. Advances in Neural Information Processing Systems (NIPS).*

[30] D. Minnen, T. Westeyn, T. Starner, J. Ward, and P. Lukowicz. 2006. Performance metrics and evaluation issues for continuous activity recognition. *Performance metrics for intelligent systems* (2006), 141–148.

[31] Tudor Miu, Paolo Missier, and Thomas Ploetz. 2015. Bootstrapping Personalised Human Activity Recognition Models Using Online Active Learning. *Proc. IUCC* (2015).

[32] V. Murahari and T. Ploetz. 2018. On Attention Models for Human Activity Recognition. In *Proc. Int. Symp. Wearable Computing (ISWC).*

[33] A. Odena, V. Dumoulin, and C. Olah. 2016. Deconvolution and Checkerboard Artifacts. *Distill* (2016). http://distill.pub/2016/deconv-checkerboard

[34] F. Javier Ordóñez and D. Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.

[35] T. Ploetz and Y. Guan. 2018. Deep Learning for Human Activity Recognition in Mobile Computing. *IEEE Computer* 51, 5 (2018), 50–59.

[36] T. Ploetz, P. Moynihan, C. Pham, and P. Olivier. 2010. Activity Recognition and Healthier Food Preparation. In *Activity Recognition in Pervasive Intelligent Environments.* Atlantis Press.

[37] T. Plötz, N. Hammerla, and P. Olivier. 2011. Feature learning for activity recognition in ubiquitous computing. In *Proc. Int. Joint Conf. on Art. Intelligence (IJCAI).*

[38] David Martin Powers. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2011).

[39] A. Reiss and D. Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *Proc. Int. Symp. Wearable Computing (ISWC).*

[40] N. Srivastava, E. Mansimov, and R. Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *Proc. Int. Conf. Machine Learning (ICML).*

[41] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster. 2008. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing* 2 (2008), 42–50.

[42] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *SSW* 125 (2016).

[43] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang. 2017. Device-free wireless localization and activity recognition: A deep learning approach.

IEEE Transactions on Vehicular Technology 66, 7 (2017), 6258–6267.

[44] J. Yang, M. Nguyen, P. San, X. Li, and S. Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition.. In *Proc. Int. Joint Conf. on Art. Intelligence (IJCAI)*.

[45] M. Zeng, H. Gao, T. Yu, O. Mengshoel, H. Langseth, I. Lane, and X. Liu. 2018. Understanding and improving recurrent networks for human activity recognition by continuous attention. In *Proc. Int. Symp. Wearable Computing (ISWC)*.

[46] M. Zeng, L. Nguyen, B. Yu, O. Mengshoel, J. Zhu, P. Wu, and J. Zhang. 2014. Convolutional neural networks for human activity recognition using mobile sensors. In *Proc. Int. Symp. Wearable Computing (ISWC)*.

[47] M. Zhang and A. Sawchuk. 2012. USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proc. Int. Conf. on Ubiquitous Computing (UbiComp)*.