








## Article

# Comparing Handcrafted Features and Deep Neural Representations for Domain Generalization in Human Activity Recognition

Nuno Bento <sup>1,\*</sup> , Joana Rebelo <sup>1</sup> , Marília Barandas <sup>1,2</sup> , André V. Carreiro <sup>1</sup> , Andrea Campagner <sup>3</sup> ,  
Federico Cabitza <sup>3,4</sup>  and Hugo Gamboa <sup>1,2</sup> 

<sup>1</sup> Associação Fraunhofer Portugal Research, Rua Alfredo Allen 455/461, 4200-135 Porto, Portugal

<sup>2</sup> Laboratório de Instrumentação, Engenharia Biomédica e Física da Radiação (LIBPhys–UNL), Departamento de Física, Faculdade de Ciências e Tecnologia (FCT), Universidade Nova de Lisboa, 2829-516 Caparica, Portugal

<sup>3</sup> Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, 20126 Milan, Italy

<sup>4</sup> IRCCS Istituto Ortopedico Galeazzi, 20161 Milan, Italy

\* Correspondence: nuno.bento@fraunhofer.pt

**Abstract:** Human Activity Recognition (HAR) has been studied extensively, yet current approaches are not capable of generalizing across different domains (i.e., subjects, devices, or datasets) with acceptable performance. This lack of generalization hinders the applicability of these models in real-world environments. As deep neural networks are becoming increasingly popular in recent work, there is a need for an explicit comparison between handcrafted and deep representations in Out-of-Distribution (OOD) settings. This paper compares both approaches in multiple domains using homogenized public datasets. First, we compare several metrics to validate three different OOD settings. In our main experiments, we then verify that even though deep learning initially outperforms models with handcrafted features, the situation is reversed as the distance from the training distribution increases. These findings support the hypothesis that handcrafted features may generalize better across specific domains.

**Keywords:** human activity recognition; deep learning; domain generalization; accelerometer



**Citation:** Bento, N.; Rebelo, J.; Barandas, M.; Carreiro, A.V.; Campagner, A.; Cabitza, F.; Gamboa, H. Comparing Handcrafted Features and Deep Neural Representations for Domain Generalization in Human Activity Recognition. *Sensors* **2022**, *22*, 7324. <https://doi.org/10.3390/s22197324>

Academic Editor: Christian Haubelt

Received: 4 August 2022

Accepted: 23 September 2022

Published: 27 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Human Activity Recognition (HAR) has the objective of automatically recognizing patterns in human movement given sensor-based inputs, namely inertial measurement units (IMUs), currently available in most wearables and smartphones [1]. HAR is an important enabling technology for applications such as remote patient monitoring, locomotor rehabilitation, security, and pedestrian navigation [1].

The IMU itself may contain several sensors, such as accelerometers and gyroscopes, which possess microelectromechanical properties, allowing their capacitance to vary with movement [2]. The accelerometer measures acceleration, while the gyroscope measures angular velocity [3]. Usually, Machine Learning (ML) is applied to enable an association between the signals obtained from these sensors and specific human activities [2]. The typical HAR system comprises the following steps [4]: data acquisition, preprocessing, segmentation, feature extraction, and classification.

Similar to most ML tasks, HAR models perform well when testing on a randomly sampled subset of a carefully acquired dataset (i.e., out-of-sample validation) and struggle in Out-of-Distribution (OOD) settings (i.e., external validation). These settings occur when the source and target domains are different, such as when the models are tested across different datasets or sensor positions [5–7].

Deep learning is becoming increasingly popular in HAR applications [8]. While the typical pipeline includes a feature extraction step before training a classifier, deep neural networks automatically learn and extract features through a continuous minimization of a cost function. In principle, a neural network may have millions of learnable parameters, which translates into a large capacity to learn more complex and discriminative features [9]. These models have potential for HAR applications since sensor signals may have many inherent subtleties that may not be recognized by Handcrafted (HC) features. Although a promising approach, significant limitations have been discussed when deep learning models are deployed in real-world environments. Current methods for training deep neural networks may converge to solutions that rely on spurious correlations [10], resulting in models that lack robustness and fail in test domains that are trivial for humans [11].

On the other hand, HC features in this field are well-studied [1,12], more interpretable, and can reach high performance. In HAR, results with HC features approximate those of deep learning [13,14] even in tasks where the latter thrives, namely when the train and test sets are split by randomly shuffling the data, thus showing similar distributions [15].

Since both methods have advantages and limitations, there is a need for a more detailed comparison between them in various domains. This translates into a need for benchmarks where the similarity between train and test distributions has considerable variability.

As HAR naturally includes many kinds of possible domains, it can be considered an excellent sandbox to study the OOD generalization ability of learning algorithms (Domain Generalization), being previously used for this purpose [16].

This paper compares the performance of learning algorithms based on HC features with deep learning approaches for In-Distribution (ID) and OOD settings. For this comparison, we use five public datasets, homogenized to have the same label space and input shape, so that the models can be easily trained and tested across them. To validate whether the tasks are in fact OOD, several metrics are considered and compared with the purpose of assessing the disparity between train and test sets. To extract HC features, Time Series Feature Extraction Library (TSFEL) [12] was used. We use one-dimensional Convolutional Neural Networks (CNNs) for our deep learning baselines.

In summary, the major contributions of this work are the following:

1. A comparison between different data similarity measures and their relationship to generalization performance.
2. A validation of the hypothesis that models based on HC features can be more robust than deep learning models for several HAR tasks in OOD settings.
3. An empirical demonstration that a hybrid approach between HC features and deep representations can bridge the gap in OOD performance.

## 2. Related Work

Several studies compared classic ML approaches using HC features with deep learning methods. The authors from [13,14,17,18] compare CNNs with models based on support vector machines, multilayer perceptrons, and random forests. In all these studies, deep learning approaches outperformed classic methods. However, in their experiments, data splits were created by randomly shuffling the datasets, so samples from possibly different domains are represented in both the train and test sets with similar data distributions.

In regard to the use of data similarity to quantify the degree of OOD, associated with generalization, this is both an old and important question in the ML literature, as several ML methods implicitly rely on properties related to similarity (e.g., the large margin assumption in SVM learning) to guarantee good generalization performance [19]. The potential relationship between data similarity and the generalization properties of ML models was first investigated from an empirical point of view in [20], where the authors discovered that datasets found to be substantially dissimilar likely stemmed from different distributions. Based on these findings, the authors of [21] demonstrated that information about similarity can be used to understand why a model performs poorly on a validation set, while the same information can be used to understand when and how to successfully perform

domain adaptation (see, for example, the recent review [22]). To that end, several metrics for measuring data similarity have been proposed in the literature. Bousquet et al. [20] developed a measure (Data Agreement Criterion, DAC) based on the Kullback–Leibler divergence, which has since become frequently used to assess the similarity of distributions [23]. More recently, Schat et al. [24] suggested a modification to the DAC measure (Data Representativeness Criterion, DRC), and investigated the link between data similarity and generalization performance. Cabitza et al. [25] proposed instead a different approach based on a multivariate statistical testing procedure to obtain a hypothesis test for OOD data, the Degree of Correspondence (DC), and also studied the correlation between DC scores and the generalization of ML models. By contrast, in the Deep Learning literature, approaches based on the use of statistical divergence measures, such as the Wasserstein distance [26] or the Maximum Mean Discrepancy (MMD) [27], have become increasingly popular to design methods for OOD detection. See also, the recent review by Shen et al. [28].

Deep learning approaches have been explored in OOD settings by testing the models on data from unseen domains [4,29–32]. Gholamiangonabadi et al. [33] verified that the accuracy went from 85.1% when validating using leave-one-subject-out (LOSO) cross-validation to 99.85% when using  $k$ -fold cross-validation. Bragança et al. [34] had similar results with HC features, reporting an accuracy of 85.37% for LOSO and 98% for  $k$ -fold. The most important features used by each model differed significantly. They concluded that LOSO would be a better validation method for generalization. Li et al. [4] and Logajov et al. [30] compared several deep learning models with classic ML pipelines using LOSO validation. As opposed to what was verified in the previous studies involving ID settings, in the context of OOD, classic methods were mostly on par with deep learning approaches, outperforming them in some cases. Still, data acquired from different subjects of the same dataset may not be as diverse as the data encountered by HAR systems in real-world environments since datasets are usually recorded in controlled conditions with similar devices worn in the same positions. In Hoelzemann et al. [7], significant drops in performance were reported when testing on different positions and different datasets, which were then mitigated by the use of transfer learning techniques.

Transfer learning has previously been applied to HAR in cases where feature representations can be used in downstream tasks or across domains [6,35]. These methods leverage information about the target task or domain to approximate the source and target representations [5]. For example, Soleimani et al. [5] used a Generative Adversarial Network (GAN) to adapt the model to each user, outperforming other domain adaptation methods. However, the performance was poor when no transfer learning method was used (see Table 2 of [5]). The same phenomenon can be noticed in [35], where the domain adaptation methods outperformed the baseline model, which did not have access to data from the target domain. These studies illustrate the difficulty of generalizing to different domains, even when using deep learning models.

Gagnon et al. [16] included a HAR dataset in a benchmark to compare domain generalization methods applied to deep neural networks. The results indicate a 9.07% drop in accuracy from 93.35% ID to 84.28% OOD on a dataset where different devices worn in different positions characterize the possible domains. The same study showed that domain generalization techniques [11,36] did not improve results in a significant manner, and that empirical risk minimization (ERM) is still a strong baseline [37].

Boyer et al. [38] compared HC features and deep representations on an ID supervised classification task and on an OOD detection task. They concluded that, while a  $k$ -nearest neighbors (KNN) model using deep features as input outperforms the same model using HC features on the ID task, the situation partially reverts for the OOD detection task, where models based on HC features achieve the best results in two out of three datasets. However, the ID and OOD tasks are not directly comparable, since they are of different kinds and use different evaluation methods.

Trabelsi et al. [39] compared three deep learning approaches and a random forest classifier with handcrafted features as input. Similar to the experiments in our work,

the datasets were homogenized by including only common activities and separated the test sets by the user. They concluded that only one of the deep learning approaches outperformed the baseline model with handcrafted features. While they formulated two different domain generalization settings (OOD-U and OOD-MD), the results for each of these settings are not directly comparable since the test sets were combined when reporting the results for the OOD-MD setting.

This paper adds to previous work by explicitly comparing the OOD robustness of HC features and deep representations in four domain generalization settings with different distances between train and test sets.

### 3. Methodology

#### 3.1. Datasets

The datasets used in this study include human activity data recorded using smartphones and wearable inertial measurement units (IMUs). Table 1 contains a detailed description of these publicly available datasets.

**Table 1.** Description of the datasets, including activities, positions, devices, and number of subjects.

Dataset	Description	Devices	Source
PAMAP2—Physical Activity Monitoring	9 subjects; 18 physical activities including sitting, lying, standing, walking, ascending stairs, descending stairs and running.	Heart rate monitor ( $\approx 9$ Hz); 3 inertial measurement units each containing a triaxial accelerometer, a gyroscope and a magnetometer (100 Hz); Positions: wrist, chest and ankle.	[40,41]
Sensors Activity Dataset (SAD)	10 subjects; 7 physical activities: sitting, standing, walking, ascending stairs, descending stairs, running and biking.	5 smartphones containing an accelerometer, a gyroscope and a magnetometer (50 Hz); Positions: jeans pocket, arm, wrist and belt.	[42]
DaLiAc—Daily Life Activities	19 subjects; 13 physical activities including sitting, lying, standing, walking outside, ascending stairs, descending stairs and treadmill running.	4 sensors, each with a triaxial accelerometer and gyroscope (200 Hz); Positions: hip, chest and ankles.	[43]
MHEALTH	10 subjects; 12 physical activities including sitting, lying, standing, walking, climbing/descending stairs, jogging and running.	3 wearable sensors containing an accelerometer, a gyroscope and a magnetometer. One of the sensors also provides 2-lead ECG measurements (50 Hz); Positions: chest, wrist and ankle.	[44,45]
RealWorld (HAR)	15 subjects; 8 physical activities including sitting, lying, standing, walking, ascending stairs, descending stairs and running/jogging.	6 wearable sensors containing accelerometers, gyroscopes and magnetometers (50 Hz). Also includes GPS, light and sound level sensors; Positions: chest, forearm, head, shin, thigh, upper arm, and waist.	[46]

Several criteria were followed to select the datasets for this study. Only datasets with a sampling rate close to or over 50 Hz were considered, to avoid the need for oversampling. The search was restricted to datasets that included most of the main activities seen in the literature (e.g., walk, sit, stand, run, and ascending/descending stairs). For better compatibility and to avoid large drops in performance caused by having considerably

different sensor positions [7], we selected datasets that included overlapping positions with at least one of the other datasets that fulfilled the remaining criteria.

The accelerometer was the selected sensor for this work. The magnitude values were computed as the Euclidean norm of all three axes ( $x$ ,  $y$ , and  $z$ ), as this quantity is invariant to the orientation of the device and can give information that is more stable across domains. The magnitude signal was used along with the signal from each axis, so that all the information given by the accelerometer was retained. From those four channels, five-second windows were extracted without overlap.

All selected datasets were homogenized [47] so that a model trained on a specific dataset could be directly tested in any other. This procedure included resampling all the recordings to 50 Hz and mapping the different activity labels to a common nomenclature: walking, running, sitting, standing, and stairs. Stair-related labels were joined into a general “stairs” label, as having to distinguish between going up and down the stairs would add unnecessary complexity to the task, since it is hard to infer the direction of vertical displacement without access to a barometer [48]. The RealWorld dataset [46] generated considerably more windows than the other datasets, so one-third of these windows was randomly sampled and used in the experiments. The final distribution of windows and activities per dataset is shown in Table 2. This table contains the percentage of samples (five-second windows) of each activity in a given dataset, as well as the total number of samples and corresponding percentage of each activity and dataset. In this table, it can be seen that, while not being very well balanced, the activities have a substantial amount of samples for all the datasets. On the other hand, even with the effort of reducing samples, the RealWorld and SAD datasets have a larger influence in the experiments, which should not be an issue, since the conditions remain the same for both deep and classic approaches.

**Table 2.** Distribution of samples and activity labels per dataset. The # symbol represents the number of samples.

Activity	Datasets (%)					Total		
	PAMAP2	SAD	DaLiAc	MHEALTH	RealWorld	%	#	
Run	10.5	16.9	20.0	33.3	19.1	18.3	7975	
Sit	19.8	16.9	10.6	16.7	17.0	16.3	7102	
Stairs	23.6	32.2	12.3	16.7	30.0	26.3	11,460	
Stand	20.4	16.9	10.6	16.7	16.4	16.2	7047	
Walk	25.7	16.9	46.5	16.7	17.5	22.8	9927	
Total	%	12.7	24.4	15.3	4.96	42.6	-	-
	#	5541	10,620	6644	2160	18,546	-	43,511

### 3.2. Handcrafted Features

To extract HC features, TSFEL [12] was used. This library extracted features directly from the 5-second accelerometer windows generated from each public dataset. To decrease computation time, we removed the features that included individual coefficients, such as Fast Fourier Transform (FFT), empirical Cumulative Distribution Function (eCDF), and histogram values. Nonetheless, the high-level spectral features computed from the FFT were kept. We did not extract wavelet and audio-related features, such as MFCC and LPCC. The total number of features per window was 192.

After the features were computed, samples were split according to each task (see Section 4). Subsequently, features were scaled by subtracting the mean of the train set and dividing by its standard deviation (Z-score normalization). The classifiers used were Logistic Regression (LR) and a Multilayer Perceptron (MLP) with a single hidden layer of 128 neurons and Rectified Linear Unit (ReLU) activation. These classifiers were chosen to enable a fair comparison with deep learning, as they resemble the last layer(s) of a deep neural network, usually responsible for the final prediction after feature learning.

### 3.3. Deep Learning

Convolutional neural networks were the selected deep learning models for this study since they achieved significantly better performance and converged faster when compared with recurrent neural networks (RNN) in preliminary experiments, which was consistent with the literature [49,50]. A scheme of the baseline CNN architectures is presented in Figure 1. We chose three different architectures, which we named CNN-base, CNN-simple, and ResNet. The training process was identical for all the architectures and is explained in Section 4. CNN-simple is a simplified version of the CNN-base with only two convolutional layers and a logistic regression directly applied to the flattened feature maps. ReLU was used as the activation function for the hidden layers of both architectures. The ResNet (Figure 1c) is a residual network inspired by Ferrari et al. [18], with a few modifications. Its convolutional block is represented in Figure 2.

In an attempt to bridge the performance gap between HC features and deep representations, we built a hybrid version of each architecture. There, the HC features are concatenated with the flattened representations of each model and fed to a fusion layer before entering the final classification layer. The number of hidden units for the fusion layer was 128 on both CNN-simple and CNN-base, increasing to 256 for the ResNet. An illustration of the hybrid version of CNN-base is in Figure 3.

For all these models, the input windows were scaled by Z-score normalization, with mean and standard deviation computed across all the windows of the train set.

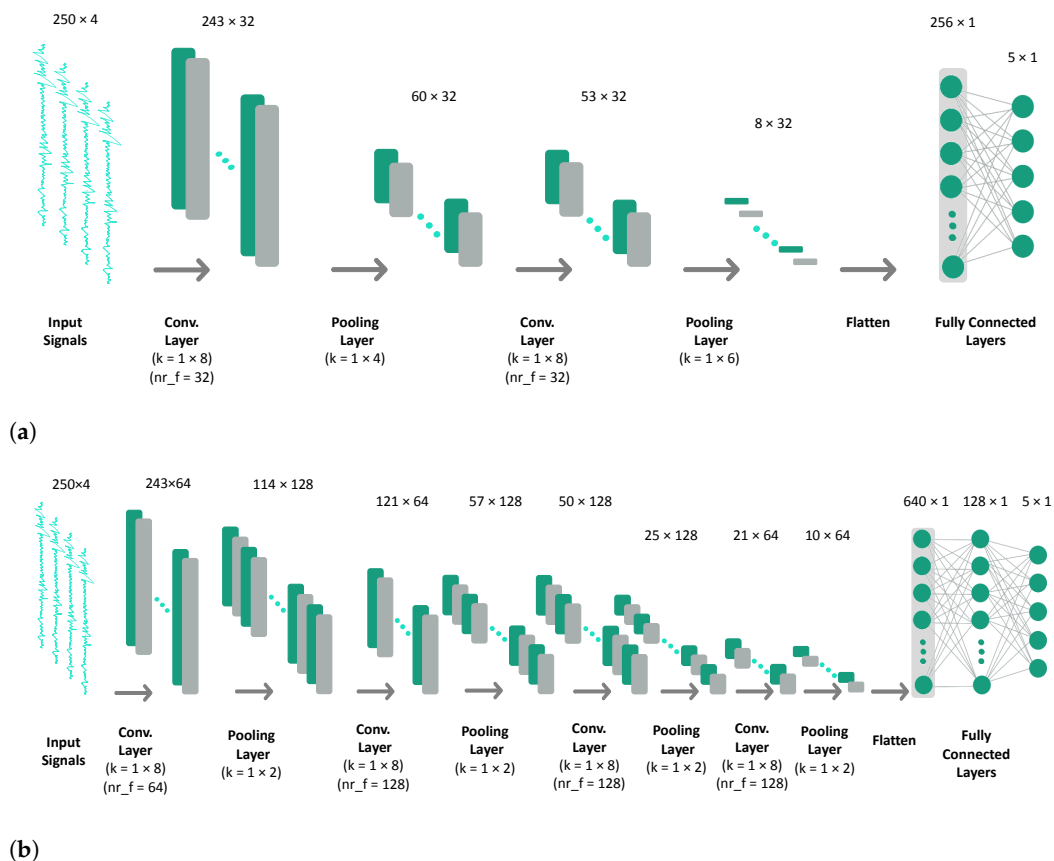
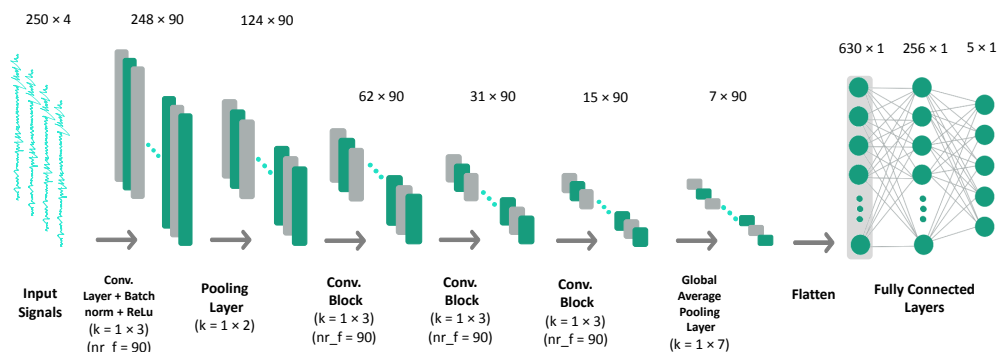
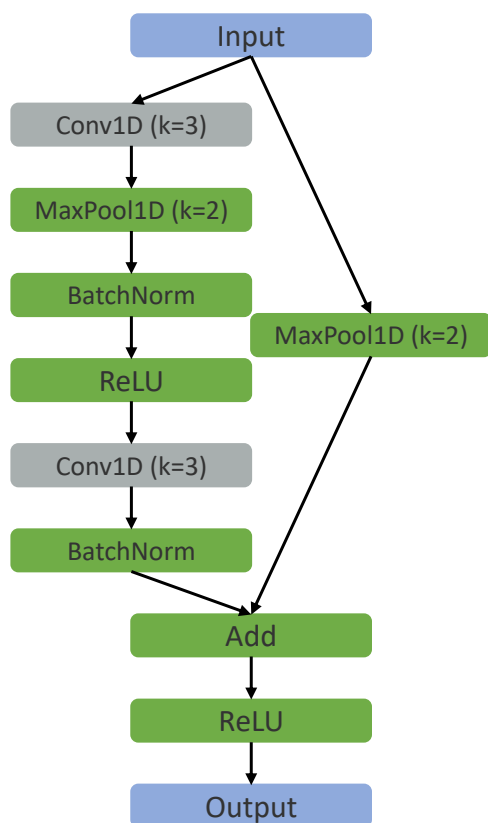


Figure 1. Cont.

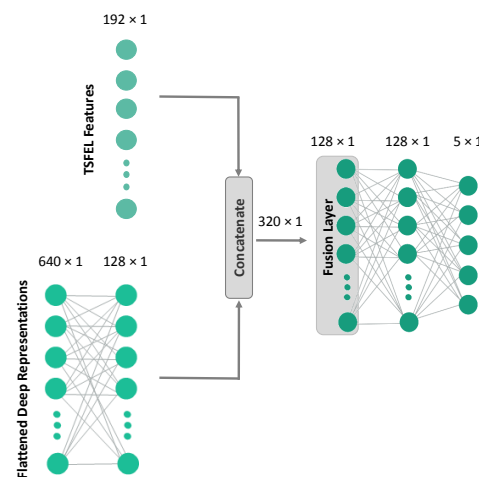


(c)

**Figure 1.** Convolutional neural network architectures. The values above the representation of each feature map indicate their shape (Signal length × Number of channels). Convolutional layers (1D): k = kernel size; nr\_f = number of filters; stride = 1; padding = 0. Max pooling layers: k = kernel size; stride = 1; padding = 0. (a) CNN-simple Architecture. (b) CNN-base Architecture. (c) ResNet Architecture. The convolutional block is depicted in Figure 2.



**Figure 2.** ResNet convolutional block. The letter k stands for “kernel size”.



**Figure 3.** Simplified illustration of the hybrid version of CNN-base (excluding the CNN backbone for ease of visualization).

### 3.4. Evaluation

To quantify the degree to which a test domain is OOD, different metrics were applied, namely Euclidean distance, Cosine similarity, Wasserstein distance, MMD, and DC. Each metric was applied to the representations of each model before the classification stage. Regarding the Wasserstein distance [51], the Wasserstein-1 version was used and is given by:

$$W_1(X, Y) = \inf_{\pi \in \Gamma(X, Y)} \int_{\mathbb{R} \times \mathbb{R}} |x - y| d\pi(x, y), \quad (1)$$

where  $\Gamma(X, Y)$  is the set of distributions whose marginals are  $X$  and  $Y$  on the first and second factors, respectively.  $x$  and  $y$  are samples from each distribution  $\pi(x, y)$  from the set. Intuitively, the distance is given by the optimal cost of moving a distribution until it overlaps with the other. In our experiments,  $x$  and  $y$  are the feature representations of subsets of the train and test data, thus  $W_1$  represents the cost of mapping the distribution of  $x$  into the distribution of  $y$  (or vice versa).

Regarding the MMD, this is a kernel-based statistical procedure that aims at determining whether two given datasets come from the same distribution [52]. Given a fixed kernel function  $k : X \times X \mapsto \mathbb{R}$  and two datasets  $X, Y$  with sizes  $|X| = n$ ,  $|Y| = m$ , the MMD can be estimated as:

$$\text{MMD}(X, Y) = \frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) + \frac{1}{m(m-1)} \sum_{i \neq j} k(y_i, y_j) - \frac{2}{nm} \sum_{i, j} k(x_i, y_j) \quad (2)$$

Intuitively, the MMD measures the distance between  $X$  and  $Y$  by computing the average similarity in  $X$  and  $Y$  separately, and then subtracting the average cross-similarity between the two datasets, where the similarity between two instances is quantified by means of the selected kernel  $k$ . In this work, a simple linear kernel was selected. Furthermore, as for the Wasserstein distance,  $x$  and  $y$  represent the feature representations of subsets of the train and test data. Thus, MMD quantifies the average kernel similarity among instances in  $x$  and  $y$ , discounted by the cross-similarity between the two datasets.

The DC, by contrast, is a multivariate hypothesis testing procedure for the hypothesis that two samples of data come from the same distribution: having fixed a representative data sample, the obtained  $p$ -value, then, can be considered as a measure of how much any other data sample is OOD with respect to the representative one. In particular, scores close to 0 can be interpreted as being most likely OOD (since, assuming the null hypothesis of the two data samples coming from the same distribution, observing a  $p$ -value close to 0 has low probability). While the DC cannot be defined and computed by means of a closed-form



procedure, in [25] a permutation-resampling algorithm (see Algorithm 1) was defined to compute the corresponding  $p$ -value, based on the selection of a base distance metric.

**Algorithm 1** The algorithm procedure to compute the similarity between the two dataset  $T$  and  $V$ , using the Degree of Correspondence (DC).

---

```

procedure DC( $T, V$ : datasets,  $d$ : distance,  $\partial$  distance metrics)
   $d_T = \{d(t, t') : t, t' \in T\}$ 
  For each  $v \in V$ , find  $t_v \in T$ , nearest neighbor of  $v$  in  $T$ 
   $T_{|V} = \{t \in T : \nexists v \in V \text{ s.t. } t = t_v\} \cup V$ 
   $d_{T_{|V}} = \{d(t, t') : t, t' \in T_{|V}\}$ 
   $\delta = \partial(d_T, d_{T_{|V}})$ 
  Compute DC =  $Pr(\delta' \geq \delta)$  using a permutation procedure
  return DC
end procedure

```

---

The selection of the distance metrics  $\partial$  in Algorithm 1 is important to obtain sensible results for the DC. Intuitively,  $\partial$  should represent the appropriate notion of distance in the instance space of interest. In [53], lacking any appropriate definition of distance in the instance space, the authors suggest the use of a general baseline, e.g., the Euclidean or cosine distance, or robust non-parametric deviation metrics, e.g., MMD or Kolmogorov–Smirnov statistics.

In previous work, model performance has been evaluated using metrics such as accuracy, sensitivity, specificity, precision, recall, and f1-score [1]. As class imbalance is common in most publicly available HAR datasets (see Table 2), f1-score is used as the main performance metric since it is more robust than accuracy in these settings [30]. To be able to compare deep learning models and classic models with HC features, the f1-scores are compared in tasks across different OOD scenarios and including five public HAR datasets.

#### 4. Experiments and Results

The main purpose of this paper is to compare the performance of HC features and deep representations in different OOD settings for HAR. A scheme of the full pipeline used for the experiments is presented in Figure 4.

HAR is a classification task that usually involves multiple domains, easily turning into a domain generalization task if the domains are considered when splitting the data. We devise four domain generalization settings, starting with a baseline ID setting where 30% of each dataset is randomly sampled for testing, and three OOD settings: (a) splitting by user within the same dataset, where approximately 30% of the users were assigned to the test set—OOD by user (OOD-U); (b) leaving a dataset out for testing, while including all the others for training—OOD with multiple source datasets (OOD-MD); (c) training on a dataset and leaving another for testing, running all the possible combinations—OOD with a single source dataset (OOD-SD). To obtain a direct comparison, the test set of OOD-U is used as a test set for all the OOD settings. Of the three OOD settings, OOD-U is the one that is expected to be closest to the training distribution since it is drawn from the same dataset, where devices and acquisition conditions are usually similar. It is followed by OOD-MD, since joining all the datasets (except one) for training averages their distributions onto a more general space. Subsequently, as it includes only a single dataset for training, OOD-SD should capture the largest distances between train and test distributions.

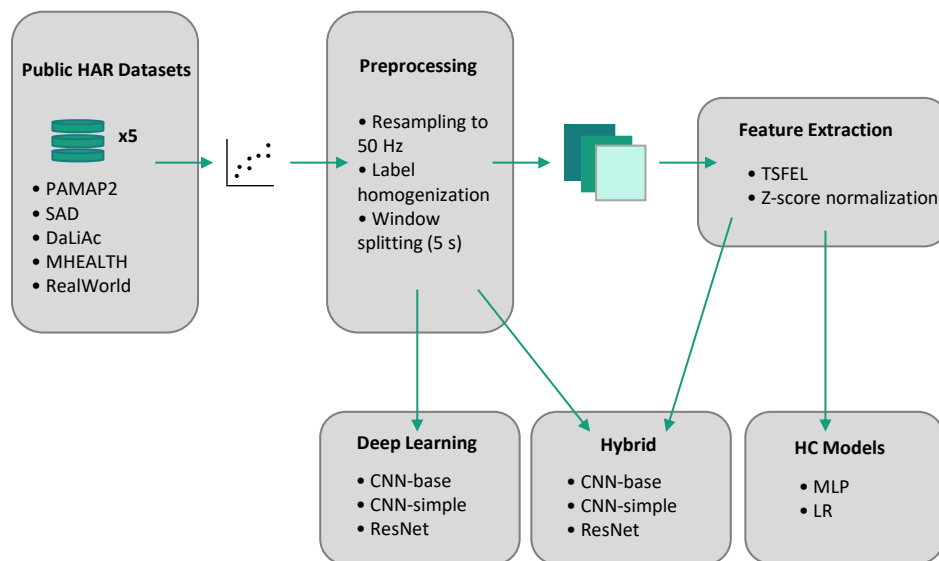


Figure 4. Scheme of the experimental pipeline.

In order to validate our hypothesis about the ordering of the distances between the train and test splits on our four settings, different metrics were applied to the feature representations. This experiment has the following objectives: (1) to validate that our three OOD settings are in fact OOD; and (2) to obtain the best metric for our main experiments, which should output values that agree with our ordering hypothesis for both HC features and deep representations. For models based on HC features, metrics were computed directly from the features. In contrast, for deep models, metrics were calculated from the hidden representations of the last layer before classification.

We note that different distance metrics have different scales, therefore, making their interpretation and comparison more difficult. For this reason, we computed distance ratios instead of raw distances, so as to make the values of the different metrics more consistent across tasks. The distance ratios were computed for each task, i.e., setting/dataset combination, using the following equation:

$$\text{Distance\_ratio} = \frac{\partial(tr_1, ts_1)}{\partial(tr_2, tr_3)}, \quad (3)$$

where  $\partial$  is a distance metric and  $tr_i$  and  $ts_i$  are subsets randomly sampled (with replacement) from the train and test sets, respectively. The sample size is half the minimum of the train and test set lengths. By contrast, for the DC, the raw value without any ratio-based normalization was used, since it is already normalized in the  $[0, 1]$  range and is able to deal with any data representation directly.

A comparison of the considered metrics based on the TSFEL features is presented in Table 3. It is easy to observe that all the metrics agree with the OOD ordering hypothesis stated above. Indeed, the value of all metrics was higher for the OOD-U, OOD-MD, and OOD-SD (respectively, in this order) than for the ID setting. In particular, it can be seen that DC with Euclidean-based metrics saturates to values close to zero for all three OOD settings, indicating that, by the comments above on the interpretation of this score, the test sets are likely to be OOD.

Table 4 shows a comparison of the considered metrics based on the CNN-base representations. In contrast to the case of TSFEL features, the metrics showed a much lower degree of agreement with the OOD ordering hypothesis. First, it can be noted that only Wasserstein and MMD have values that clearly increase with the expected degree of OOD, being in agreement with the results of the TSFEL representations and, consequently, with our OOD ordering hypothesis. Nonetheless, it can be verified that both metrics had a large degree

of variation, with the confidence intervals for the ID, OOD-U, and OOD-MD partially overlapping. In the case of DC Cosine, the score for the OOD datasets was higher than that for the ID one. This seemingly paradoxical behavior may have an intuitive geometric explanation, as it may be a consequence of the transformations that take place during training, which influence the shape of the instance space and possibly make the representations of instances that would otherwise be OOD closer to the training data manifold. In support of this hypothesis, it can be easily observed that most metrics reported a significantly different value for the OOD-SD setting than for the other OOD settings, showing that the training of the deep learning model had an important influence on the natural representation of the data manifold. In this sense, both the Wasserstein and MMD metrics seemed to be more apt at naturally adapting to this change of representation.

**Table 3.** Comparison of metrics over all four domain generalization settings based on the TSFEL feature representations. For each setting, values were averaged over every test set. All metrics are ratios except the ones with (\*).

Metric	Setting				Avg. OOD
	ID	OOD-U	OOD-MD	OOD-SD	
Wasserstein	1.02 ± 0.04	1.42 ± 0.37	2.27 ± 1.25	3.31 ± 2.39	2.33 ± 0.91
MMD	0.95 ± 0.86	30.47 ± 56.25	800.05 ± 1513.29	1072.20 ± 2619.40	634.24 ± 1008.55
Euclidean	1.00 ± 0.01	1.08 ± 0.11	1.33 ± 0.48	1.53 ± 0.73	1.31 ± 0.29
DC Euclidean *	0.55 ± 0.10	0.05 ± 0.08	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.03
Cosine	0.95 ± 0.33	0.85 ± 0.31	0.39 ± 0.52	0.10 ± 0.84	0.45 ± 0.35
DC Cosine *	0.60 ± 0.17	0.32 ± 0.34	0.12 ± 0.16	0.12 ± 0.21	0.19 ± 0.14

**Table 4.** Comparison of metrics over all four domain generalization settings based on the CNN-base representations. For each setting, values were averaged over all the datasets. All metrics are ratios except the ones with (\*).

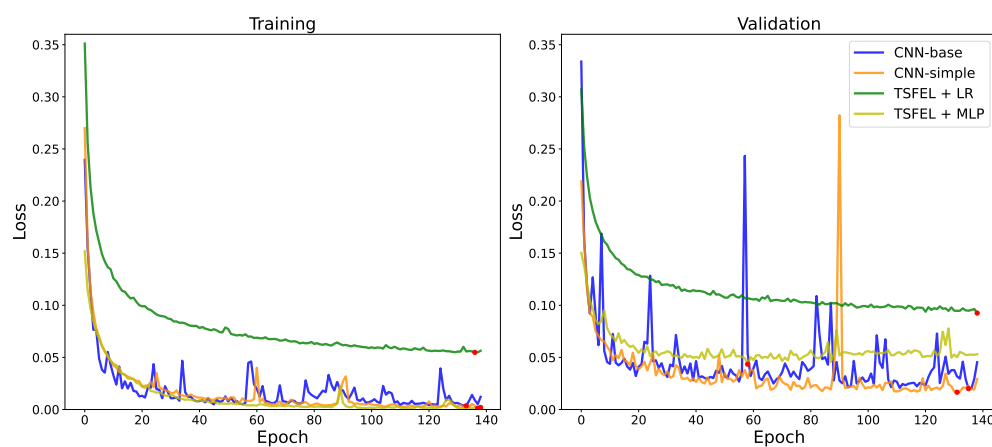
Metric	Setting				Avg. OOD
	ID	OOD-U	OOD-MD	OOD-SD	
Wasserstein	1.06 ± 0.09	1.39 ± 0.27	1.95 ± 0.45	5.71 ± 5.05	3.02 ± 1.69
MMD	1.25 ± 1.00	1.80 ± 0.92	35.23 ± 56.10	245.27 ± 402.93	94.10 ± 135.60
Euclidean	1.00 ± 0.02	1.01 ± 0.05	1.02 ± 0.15	1.12 ± 0.27	1.05 ± 0.11
DC Euclidean *	0.49 ± 0.15	0.51 ± 0.32	0.53 ± 0.45	0.10 ± 0.18	0.38 ± 0.19
Cosine	1.01 ± 0.01	0.98 ± 0.01	0.98 ± 0.03	1.03 ± 0.06	1.00 ± 0.02
DC Cosine *	0.55 ± 0.10	0.92 ± 0.10	0.65 ± 0.43	0.52 ± 0.43	0.70 ± 0.21

Thus, as a consequence of these results, we chose the Wasserstein distance ratio as our main metric to quantify the degree of OOD due to the fact that it agrees with our hypothesis when using both TSFEL features and deep representations as input. This metric has also been applied by Soleimani et al. [5] to compute distances between source and target distributions.

Our experiments were run on an NVIDIA (Santa Clara, CA, USA) A16-8C GPU and an AMD (Santa Clara, CA, USA) Epyc 7302 processor with python version 3.8.12 and Visual Studio Code (Microsoft, Redmond, WA, USA) as the development environment. All the learning models were implemented using the PyTorch library [54]. Adam [55] was adopted as the optimizer used for the training process. To reduce bias [16], results were averaged over nine combinations of three different batch sizes (64, 128, and 256) and three learning rates (0.0008, 0.001, and 0.003). To account for class imbalance, the percentage of instances per class in the training set was given to the cross-entropy loss function as class weights.

To make the experiments as agnostic to the training method as possible, the same procedure was used for training the classifiers based on HC features and the deep learning models. Figure 5 shows the training and validation loss over the course of training for a single task. The chosen task was the OOD-U setting on the SAD dataset, an example of a

task in which there was a verified occurrence of instability in training. One of the ways to handle this instability is by ending the training process earlier—early stopping [56].

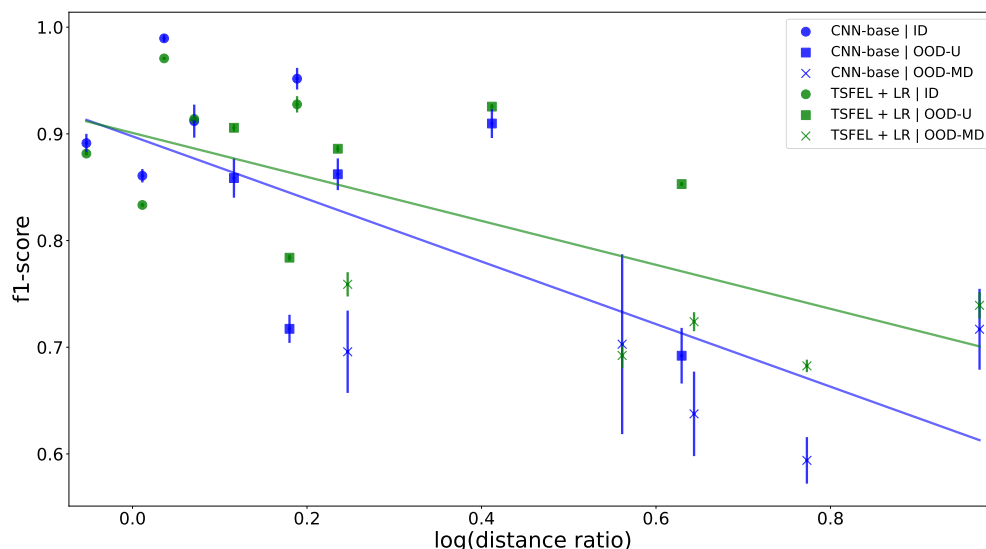


**Figure 5.** Evolution of loss by epoch on SAD dataset in the OOD-U setting. The red dots indicate the minimum loss of each curve.

Over all the tasks, most models reached plateaus on validation performance after 30 to 50 epochs, so the training process was limited to 140 epochs to leave a margin for models to converge, but not so much as to fully overfit the data. For validation, we randomly sampled a 10% subset of the training data without replacement. While training, a checkpoint model was saved every time the validation loss achieved its best value since the start of training. Our early stopping method consisted of stopping training if the validation loss did not improve for 30 epochs in a row, which proved helpful in cases where training was not very stable. In these cases, the validation error oscillates, increasing for a certain number of epochs before decreasing again and, on many occasions, achieving a slightly lower error rate than in any of the previous epochs, which can be seen in the loss curves for the CNN models in Figure 5. This resembles the effects of double descent [57]. In our case, one of the causes of such unstable training may be the fact that these datasets are noisy, due to the diversity in users, devices, and positions, among other factors. It may also be a consequence of overparameterization, as the phenomenon was much more pronounced when training CNNs, which have significantly more parameters than our MLP and LR models. Both these potential causes were documented by Nakkiran et al. [57].

The evolution of the f1-score over the Wasserstein distance ratio for the best performing model of each family (CNN-base and TSFEL+LR) is documented in Figure 6. For each combination of model, dataset, and setting, the average and standard deviation of the f1-score were computed over nine different runs with varying learning rates and batch sizes. The CNN-base embeddings were chosen to compute distance ratios for this figure since they contain less outliers when compared to the distance ratios computed from TSFEL representations (see Figure A2). It can be verified that, initially, the CNN model outperforms the model using HC features. However, as the distance between train and test domains increases, the situation is reverted, with the classic approach outperforming the CNN. This suggests that HC features are more robust to the shifts that occur in OOD data. The regression curves reinforce the idea of OOD stability. As expected, there is a negative correlation between f1-score and distance ratio, meaning that performance decreases as the test data becomes more distant from the distribution seen during training. In general, the distance ratios given by the Wasserstein distance appear to agree with the previously stated OOD ordering hypothesis, with OOD-SD being the most OOD of the three settings, followed by OOD-MD and OOD-U, respectively. Still, a few outliers can be seen in the figure. The higher values of standard deviation for the CNN indicate that these models are more susceptible to the choice of hyperparameters, which is reasonable due to the much larger number of trainable parameters. However, it is not always ideal to have

such variability, as it indicates that the validation loss has become less correlated with the test loss. In practice, an apparently good model may perform surprisingly well in some settings while failing in situations that would otherwise be trivial to a simple model.



**Figure 6.** F1-score vs. log(distance ratio). Each marker represents a different task. Distance ratios are based on the CNN-base embeddings. Error bars represent one standard deviation away from the mean. The natural logarithm was applied to the distance ratios to make the regression curves linear.

More detailed results are presented in Table 5. For each combination of model and setting, the average and standard deviation of the f1-score were computed over all five datasets. The last column represents the average of the three OOD settings, which gives an idea of the overall generalization performance. The significant overturn from the ID to the OOD settings can be noticed in the table. TSFEL + LR, which had the worst ID f1-score (90.54%), turned out to be the best overall in the OOD regime, with an f1-score of 70% for the average of all three OOD settings. Using an MLP instead of LR slightly decreased the overall OOD performance to 69.55%, while increasing the ID performance to 92.87%, becoming closer to the deep learning results. This phenomenon may be related to an increase in the number of trainable parameters. Including HC features as an auxiliary input to deep models improved both ID and OOD results, with the hybrid version of CNN-base being the deep learning model with the strongest generalization performance (average OOD f1-score of 66.95%). However, this improvement is still insufficient to reach the OOD robustness of models solely based on HC features.

**Table 5.** Average f1-score in percentage over all the tasks in a given setting. Values in bold indicate the best performance for each setting.

Model	Setting				Avg. OOD
	ID	OOD-U	OOD-MD	OOD-SD	
CNN-simple	92.09 ± 5.26	79.65 ± 10.75	63.71 ± 3.54	45.21 ± 6.57	62.86 ± 4.36
CNN-base	92.10 ± 5.06	80.79 ± 9.68	66.94 ± 5.19	48.30 ± 5.41	65.34 ± 4.08
ResNet	92.46 ± 4.73	81.16 ± 9.60	67.22 ± 4.89	46.57 ± 4.84	64.98 ± 3.94
CNN-simple hybrid	93.64 ± 4.55	85.13 ± 7.69	66.60 ± 3.31	47.87 ± 2.21	66.53 ± 2.89
CNN-base hybrid	93.48 ± 4.35	85.28 ± 6.64	67.74 ± 3.37	47.84 ± 3.24	66.95 ± 2.71
ResNet hybrid	<b>93.79 ± 4.21</b>	84.71 ± 7.72	67.87 ± 3.40	47.73 ± 2.11	66.77 ± 2.90
TSFEL + MLP	92.87 ± 4.70	<b>87.09 ± 5.35</b>	70.11 ± 3.57	<b>51.45 ± 5.31</b>	69.55 ± 2.78
TSFEL + LR	90.54 ± 5.15	<b>87.08 ± 5.55</b>	<b>71.94 ± 3.19</b>	50.97 ± 3.29	<b>70.00 ± 2.40</b>

Despite being simpler than the ResNet, the CNN-base model achieves a slightly higher generalization performance. On the other hand, CNN-simple, the simplest deep learning

model, did not perform well in OOD tasks. There appears to be an optimal number of parameters, possibly dependent on the architecture, so more studies should be conducted to understand this trade-off.

## 5. Discussion

This work aimed to compare the generalization performance of HC features and deep representations, focusing in particular on generalization in OOD settings.

In the first experiment, several metrics were compared to validate and quantify our OOD settings. For TSFEL representations, all the considered metrics were in agreement with our ordering hypothesis. In particular, the DC was able to clearly identify each of the OOD settings as such. In contrast, for the case of deep representations, there was some disagreement among the considered metrics. Still, the MMD and Wasserstein distance ratios remained in agreement with the adopted hypothesis. They were seen as more robust concerning the change of data representation induced by the deep learning model.

In our experiments involving HAR tasks, despite reaching lower f1-scores in the ID setting, models based on HC features were more robust in OOD settings. This difference in OOD performance supporting higher robustness for HC features may be due to their stability since they are fixed a priori based on domain knowledge, which should be valid across tasks. Conversely, deep features are automatically learned and could thus fail to identify generally helpful features, as there are known inefficiencies in the current methods for training neural networks. These are typically biased toward simple solutions [15] and rely on spurious correlations [10] rather than previous knowledge or causal relations.

In regard to the generalizability of our results to other settings, we note that even though we focused on HAR, with minor adaptations, our experiments and analyses could be replicated in a wide range of fields. For example, similar deep learning models and handcrafted features could be used and compared in fields that depend on sensor data, such as fall detection, predictive maintenance, or physiological signal processing (e.g., EEG, EMG, and ECG). Different deep learning architectures and feature extraction libraries would have to be employed for image or video processing.

Concerning practical purposes, HC features, being more robust, appear to be better suited for real-world HAR systems. However, their reimplementation in mobile or edge devices may be an arduous task. CNNs do not show this limitation, as the representations are encoded in weight matrices and can, in principle, be ported to these devices without significant effort [58]. More studies should, thus, be devoted to exploring this trade-off between increased robustness and reimplementation efforts, possibly considering the application of hybrid approaches (such as the ones also considered in this paper), as well as alternative training techniques for CNNs that attempt to improve robustness.

## 6. Conclusions

This paper hypothesizes that models using HC features generalize better than deep learning models across domains in HAR tasks. Three OOD settings were implemented by testing on unseen users and (single or multi-source) datasets. Five public datasets were homogenized so that they could be combined in different ways to create diverse tasks.

Several metrics were used to quantify the degree of OOD of four domain generalization settings. The DC metric was used to validate our OOD settings. In turn, the Wasserstein distance ratio was chosen as our primary metric for the study since it was able to quantify our three OOD settings in the expected order.

In our main experiments, it was verified that, although deep models have better ID performance, they are outperformed in all three OOD settings by shallow models using features that were computed based on domain knowledge. Furthermore, as the drop in f1-score in OOD settings is less accentuated for classic models, it can be inferred that HC are more robust. Hybrid models achieved intermediate results between deep and classic methods, supporting the idea that HC features can stabilize training, which helps to validate our hypothesis.

Acknowledging the limitation of current deep learning techniques in being robust with respect to OOD settings, as compared to models based on HC features, we believe our work could pave the way for further research on the development of novel training methods for making deep learning models more robust and thus bridge the generalization gap toward new, more trustworthy, gold standards in the field of HAR.

**Author Contributions:** Conceptualization, N.B., J.R., M.B. and A.V.C.; data curation, N.B., J.R. and M.B.; methodology, N.B., J.R., M.B., A.V.C. and A.C.; software, N.B. and J.R.; validation, N.B., J.R., M.B., A.V.C. and A.C.; formal analysis, N.B., J.R. and A.C.; writing—original draft preparation, N.B.; writing—review and editing, N.B., J.R., M.B., A.V.C., A.C., F.C. and H.G.; visualization, N.B., J.R. and M.B.; supervision, M.B., A.V.C., F.C. and H.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is financed by national funds through FCT—Fundação para a Ciência e a Tecnologia, I.P., within the scope of SAIFFER project under the Eureka Eurostars program (E!114310).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Supplementary Experiments

Figure A1 shows the behavior of different models over all four domain generalization settings addressed in the study in comparison to TSFEL+LR, the approach with the highest generalization performance. Similarly to the main results, an inversion tendency can be observed from the ID to the OOD regime.

Figure A1a shows a larger gap in performance for the OOD regime. This gap is mitigated in the hybrid model (Figure A1b) and becomes much smaller in Figure A1c, where handcrafted features are the only source of information.

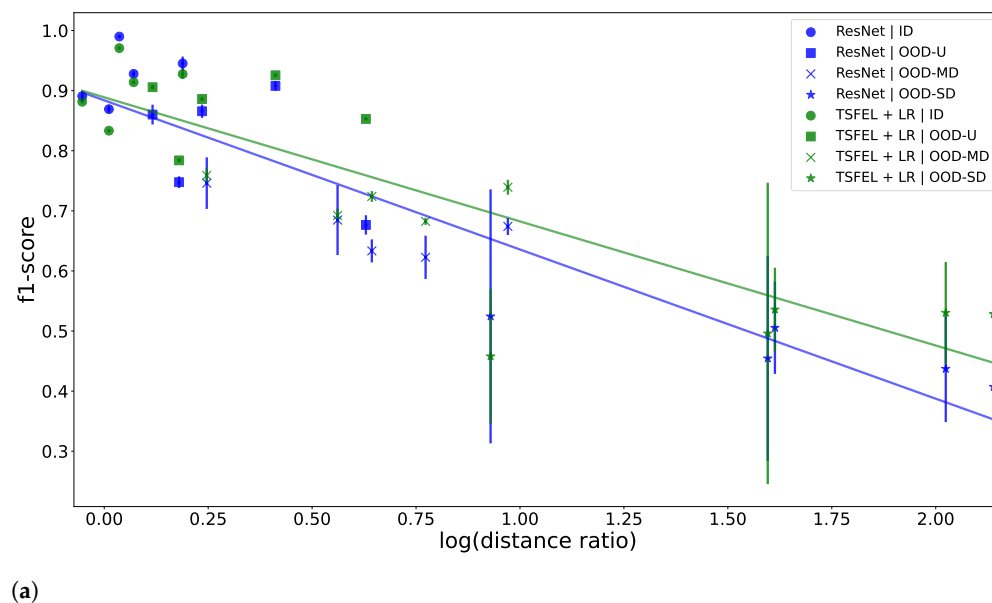
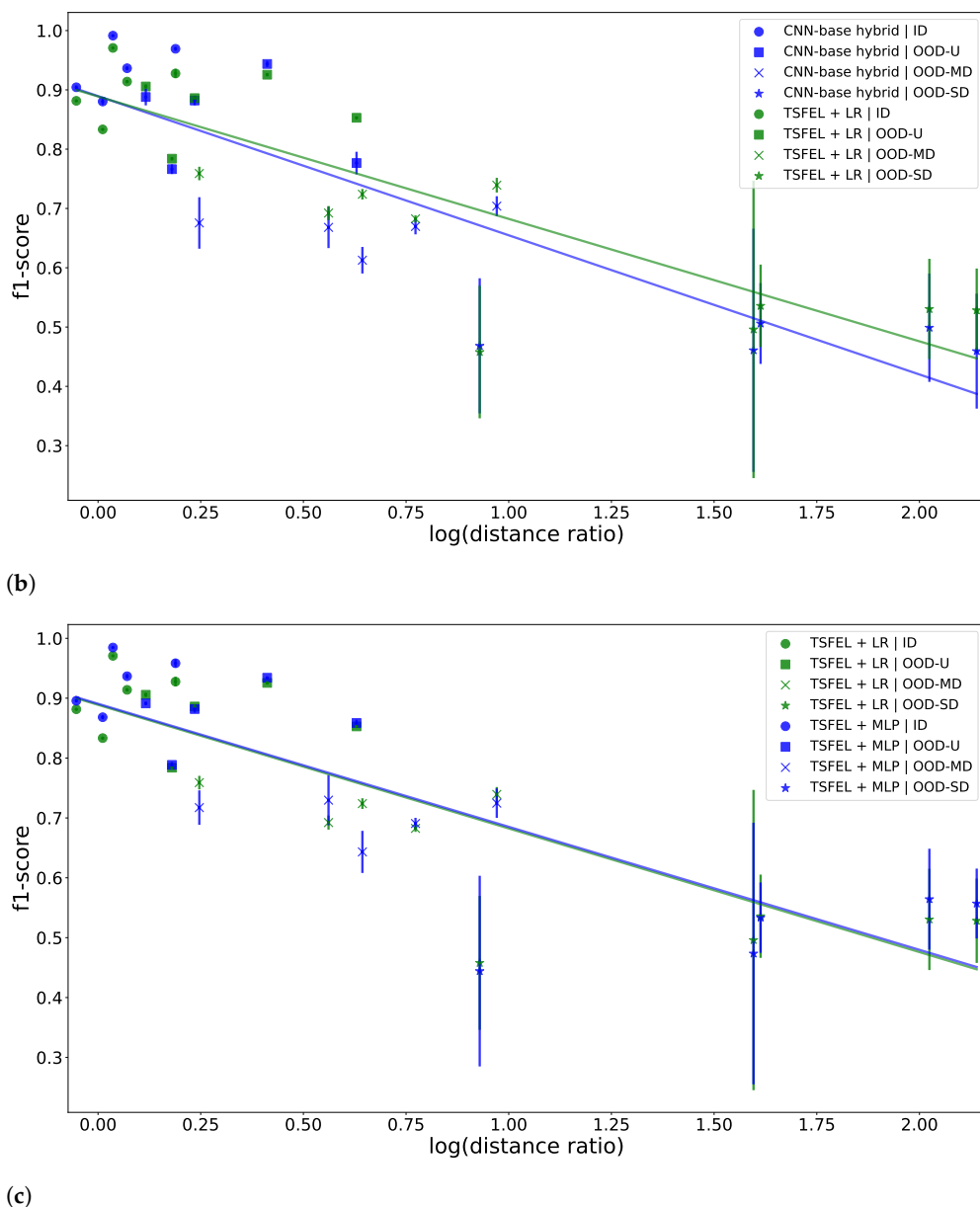


Figure A1. Cont.



**Figure A1.** F1-score vs.  $\log(\text{distance ratio})$ . Each marker represents a different task. Distance ratios are based on the CNN-base embeddings. Error bars represent one standard deviation away from the mean. (a) TSFEL + LR vs. ResNet. (b) TSFEL + LR vs. CNN-base hybrid. (c) TSFEL + LR vs. TSFEL + MLP.

By using TSFEL features to compute the distance ratios (see Figure A2), we reach the same conclusions. However, the plots in Figures 6 and A1 were based on the CNN-base embeddings, as the distance ratios presented less outliers.

Figure A3 shows the confusion matrices for the ID, OOD-U, and OOD-MD settings of the SAD dataset. It can be verified that, as expected, performance decreased in OOD settings.



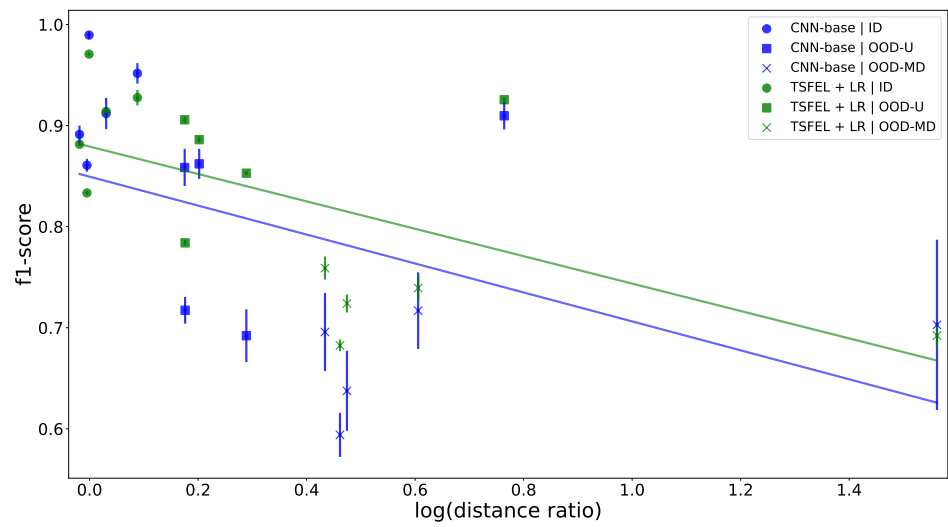
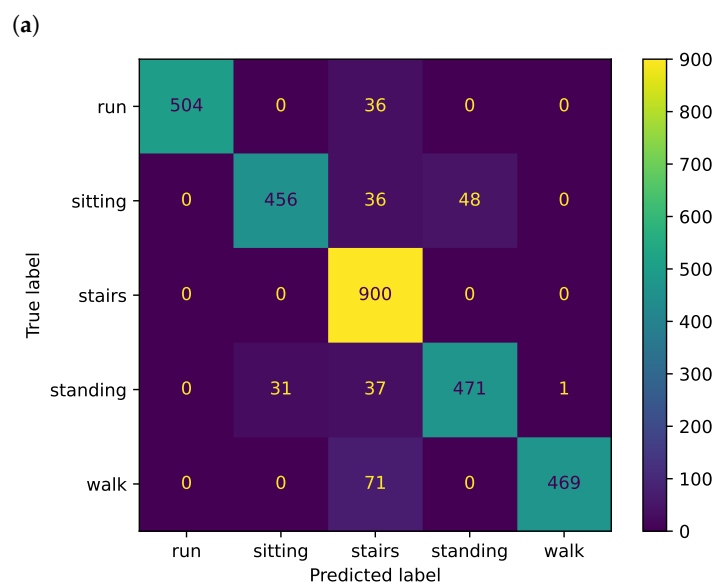
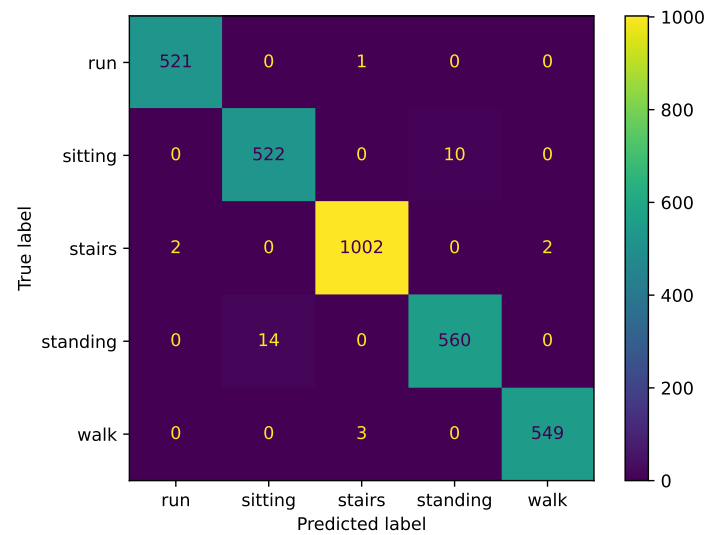


Figure A2. TSFEL + LR vs. CNN-base. Distance ratios are based on TSFEL features.



(b)  
Figure A3. Cont.



(c)

**Figure A3.** Confusion matrices for the SAD dataset. (a) In-distribution (ID). Accuracy: 99.0%, F1-score: 98.9%; (b) Out-of-Distribution leaving users out (OOD-U). Accuracy: 91.5%, F1-score: 91.6%; (c) Out-of-Distribution leaving a dataset out (OOD-MD). Accuracy: 76.0%, F1-score: 73.5%.

## References

- Sousa Lima, W.; Souto, E.; El-Khatib, K.; Jalali, R.; Gama, J. Human activity recognition using inertial sensors in a smartphone: An overview. *Sensors* **2019**, *19*, 3213. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ariza-Colpas, P.P.; Vicario, E.; Oviedo-Carrascal, A.I.; Butt Aziz, S.; Piñeres-Melo, M.A.; Quintero-Linero, A.; Patara, F. Human Activity Recognition Data Analysis: History, Evolutions, and New Trends. *Sensors* **2022**, *22*, 3401. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ahmad, N.; Ghazilla, R.A.R.; Khairi, N.M.; Kasi, V. Reviews on various inertial measurement unit (IMU) sensor applications. *Int. J. Signal Process. Syst.* **2013**, *1*, 256–262. [\[CrossRef\]](#)
- Li, F.; Shirahama, K.; Nisar, M.A.; Köping, L.; Grzegorzec, M. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors* **2018**, *18*, 679. [\[CrossRef\]](#) [\[PubMed\]](#)
- Soleimani, E.; Nazerfard, E. Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. *Neurocomputing* **2021**, *426*, 26–34. [\[CrossRef\]](#)
- Wang, J.; Zheng, V.W.; Chen, Y.; Huang, M. Deep transfer learning for cross-domain activity recognition. In Proceedings of the 3rd International Conference on Crowd Science and Engineering, Singapore, 28–31 July 2018; pp. 1–8.
- Hoelzemann, A.; Van Laerhoven, K. Digging deeper: Towards a better understanding of transfer learning for human activity recognition. In Proceedings of the 2020 International Symposium on Wearable Computers, Virtual, 12–16 September 2020; pp. 50–54.
- Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2019**, *119*, 3–11. [\[CrossRef\]](#)
- Nafea, O.; Abdul, W.; Muhammad, G.; Alsulaiman, M. Sensor-based human activity recognition with spatio-temporal deep learning. *Sensors* **2021**, *21*, 2141. [\[CrossRef\]](#)
- Sagawa, S.; Raghunathan, A.; Koh, P.W.; Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In Proceedings of the 37th International Conference on Machine Learning, Virtual Conference, 13–18 July 2020; pp. 8346–8356.
- Arjovsky, M.; Bottou, L.; Gulrajani, I.; Lopez-Paz, D. Invariant risk minimization. *arXiv* **2019**, arXiv:1907.02893.
- Barandas, M.; Folgado, D.; Fernandes, L.; Santos, S.; Abreu, M.; Bota, P.; Liu, H.; Schultz, T.; Gamboa, H. TSFEL: Time series feature extraction library. *SoftwareX* **2020**, *11*, 100456. [\[CrossRef\]](#)
- Chen, Y.; Xue, Y. A deep learning approach to human activity recognition based on single accelerometer. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 9–12 October 2015; pp. 1488–1492.
- Zebin, T.; Scully, P.J.; Ozanyan, K.B. Human activity recognition with inertial sensors using a deep learning approach. In Proceedings of the 2016 IEEE Sensors, Orlando, FL, USA, 30 October 2016–3 November 2016; pp. 1–3.
- Geirhos, R.; Jacobsen, J.H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; Wichmann, F.A. Shortcut learning in deep neural networks. *Nat. Mach. Intell.* **2020**, *2*, 665–673. [\[CrossRef\]](#)
- Gagnon-Audet, J.C.; Ahuja, K.; Darvishi-Bayazi, M.J.; Dumas, G.; Rish, I. WOODS: Benchmarks for Out-of-Distribution Generalization in Time Series Tasks. *arXiv* **2022**, arXiv:2203.09978.

17. Lee, S.M.; Yoon, S.M.; Cho, H. Human activity recognition from accelerometer data using Convolutional Neural Network. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (Bigcomp), Jeju, Korea, 13–16 February 2017; pp. 131–134.
18. Ferrari, A.; Micucci, D.; Mobilio, M.; Napoletano, P. Hand-crafted features vs residual networks for human activities recognition using accelerometer. In Proceedings of the 2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT), Ancona, Italy, 19–21 June 2019; pp. 153–156.
19. Balcan, M.F.; Blum, A.; Srebro, N. A theory of learning with similarity functions. *Mach. Learn.* **2008**, *72*, 89–112. [[CrossRef](#)]
20. Bousquet, N. Diagnostics of prior-data agreement in applied Bayesian analysis. *J. Appl. Stat.* **2008**, *35*, 1011–1029. [[CrossRef](#)]
21. Kouw, W.M.; Loog, M.; Bartels, L.W.; Mendrik, A.M. Learning an MR acquisition-invariant representation using Siamese neural networks. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; pp. 364–367.
22. Redko, I.; Morvant, E.; Habrard, A.; Sebban, M.; Bennani, Y. *Advances in Domain Adaptation Theory*; Elsevier: Amsterdam, The Netherlands, 2019.
23. Veen, D.; Stoel, D.; Schalken, N.; Mulder, K.; Van de Schoot, R. Using the data agreement criterion to rank experts' beliefs. *Entropy* **2018**, *20*, 592. [[CrossRef](#)] [[PubMed](#)]
24. Schat, E.; van de Schoot, R.; Kouw, W.M.; Veen, D.; Mendrik, A.M. The data representativeness criterion: Predicting the performance of supervised classification based on data set similarity. *PLoS ONE* **2020**, *15*, e0237009. [[CrossRef](#)]
25. Cabitza, F.; Campagner, A.; Soares, F.; de Gadiana-Romualdo, L.G.; Challa, F.; Sulejmani, A.; Seghezzi, M.; Carobene, A. The importance of being external. methodological insights for the external validation of machine learning models in medicine. *Comput. Methods Programs Biomed.* **2021**, *208*, 106288. [[CrossRef](#)]
26. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv* **2014**, arXiv:1412.3474.
27. Zhou, F.; Jiang, Z.; Shui, C.; Wang, B.; Chaib-draa, B. Domain generalization with optimal transport and metric learning. *arXiv* **2020**, arXiv:2007.10573.
28. Shen, Z.; Liu, J.; He, Y.; Zhang, X.; Xu, R.; Yu, H.; Cui, P. Towards out-of-distribution generalization: A survey. *arXiv* **2021**, arXiv:2108.13624.
29. Ronao, C.A.; Cho, S.B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.* **2016**, *59*, 235–244. [[CrossRef](#)]
30. Logacjov, A.; Bach, K.; Kongsvold, A.; Bårdstu, H.B.; Mork, P.J. HARTH: A Human Activity Recognition Dataset for Machine Learning. *Sensors* **2021**, *21*, 7853. [[CrossRef](#)] [[PubMed](#)]
31. Xu, C.; Chai, D.; He, J.; Zhang, X.; Duan, S. InnoHAR: A deep neural network for complex human activity recognition. *IEEE Access* **2019**, *7*, 9893–9902. [[CrossRef](#)]
32. Moreira, D.; Barandas, M.; Rocha, T.; Alves, P.; Santos, R.; Leonardo, R.; Vieira, P.; Gamboa, H. Human Activity Recognition for Indoor Localization Using Smartphone Inertial Sensors. *Sensors* **2021**, *21*, 6316. [[CrossRef](#)] [[PubMed](#)]
33. Gholamiangonabadi, D.; Kiselov, N.; Grolinger, K. Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection. *IEEE Access* **2020**, *8*, 133982–133994. [[CrossRef](#)]
34. Bragança, H.; Colonna, J.G.; Oliveira, H.A.B.F.; Souto, E. How Validation Methodology Influences Human Activity Recognition Mobile Systems. *Sensors* **2022**, *22*, 2360. [[CrossRef](#)]
35. Ding, R.; Li, X.; Nie, L.; Li, J.; Si, X.; Chu, D.; Liu, G.; Zhan, D. Empirical study and improvement on deep transfer learning for human activity recognition. *Sensors* **2018**, *19*, 57. [[CrossRef](#)] [[PubMed](#)]
36. Ahuja, K.; Caballero, E.; Zhang, D.; Gagnon-Audet, J.C.; Bengio, Y.; Mitliagkas, I.; Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 3438–3450.
37. Rosenfeld, E.; Ravikumar, P.; Risteski, A. The risks of invariant risk minimization. *arXiv* **2020**, arXiv:2010.05761.
38. Boyer, P.; Burns, D.; Whyne, C. Out-of-distribution detection of human activity recognition with smartwatch inertial sensors. *Sensors* **2021**, *21*, 1669. [[CrossRef](#)]
39. Trabelsi, I.; Françoise, J.; Bellik, Y. Sensor-based Activity Recognition using Deep Learning: A Comparative Study. In Proceedings of the 8th International Conference on Movement and Computing, Chicago, IL, USA, 22–24 June 2022; pp. 1–8.
40. Reiss, A.; Stricker, D. Introducing a New Benchmarked Dataset for Activity Monitoring. In Proceedings of the 2012 16th Annual International Symposium on Wearable Computers (ISWC), Newcastle, UK, 18–22 June 2012; IEEE Computer Society: Washington, DC, USA, 2012; pp. 108–109. [[CrossRef](#)]
41. Reiss, A.; Stricker, D. Creating and Benchmarking a New Dataset for Physical Activity Monitoring. In *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, Heraklion, Greece, 6–8 July 2012*; Association for Computing Machinery: New York, NY, USA, 2012. [[CrossRef](#)]
42. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H.; Havinga, P.J. Fusion of smartphone motion sensors for physical activity recognition. *Sensors* **2014**, *14*, 10146–10176. [[CrossRef](#)]
43. Leutheuser, H.; Schuldhaus, D.; Eskofier, B.M. Hierarchical, Multi-Sensor Based Classification of Daily Life Activities: Comparison with State-of-the-Art Algorithms Using a Benchmark Dataset. *PLoS ONE* **2013**, *8*. [[CrossRef](#)] [[PubMed](#)]

44. Banos, O.; Garcia, R.; Holgado-Terriza, J.A.; Damas, M.; Pomares, H.; Rojas, I.; Saez, A.; Villalonga, C. mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications. In *Ambient Assisted Living and Daily Activities*; Pecchia, L., Chen, L.L., Nugent, C., Bravo, J., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 91–98. [[CrossRef](#)]
45. Banos, O.; Villalonga, C.; Garcia, R.; Saez, A.; Damas, M.; Holgado-Terriza, J.A.; Lee, S.; Pomares, H.; Rojas, I. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *BioMed. Eng. Online* **2015**, *14*. [[CrossRef](#)] [[PubMed](#)]
46. Szttyler, T.; Stuckenschmidt, H. On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), Sydney, NSW, Australia, 14–19 March 2016; pp. 1–9. [[CrossRef](#)]
47. Ferrari, A.; Mobilio, M.; Micucci, D.; Napoletano, P. On the homogenization of heterogeneous inertial-based databases for human activity recognition. In Proceedings of the 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 8–13 July 2019; Volume 2642, pp. 295–300.
48. Figueira, C.; Matias, R.; Gamboa, H. Body Location Independent Activity Monitoring. In Proceedings of the 9th International Conference on Bio-inspired Systems and Signal Processing, Rome, Italy, 21–23 February 2016.
49. Shakya, S.R.; Zhang, C.; Zhou, Z. Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data. *Int. J. Mach. Learn. Comput.* **2018**, *8*, 577–582.
50. Shiranthika, C.; Premakumara, N.; Chiu, H.L.; Samani, H.; Shyalika, C.; Yang, C.Y. Human Activity Recognition Using CNN & LSTM. In Proceedings of the 2020 5th International Conference on Information Technology Research (ICITR), Online, 2–4 December 2020; pp. 1–6.
51. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.
52. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
53. Cabitza, F.; Campagner, A.; Sconfienza, L.M. As if sand were stone. New concepts and metrics to probe the ground on which to build trustable AI. *BMC Med. Inform. Decis. Mak.* **2020**, *20*, 1–21. [[CrossRef](#)] [[PubMed](#)]
54. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
55. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
56. Heckel, R.; Yilmaz, F.F. Early stopping in deep networks: Double descent and how to eliminate it. *arXiv* **2020**, arXiv:2007.10099.
57. Nakkiran, P.; Kaplun, G.; Bansal, Y.; Yang, T.; Barak, B.; Sutskever, I. Deep double descent: Where bigger models and more data hurt. *J. Stat. Mech. Theory Exp.* **2021**, *2021*, 124003. [[CrossRef](#)]
58. Bai, J.; Lu, F.; Zhang, K. ONNX: Open Neural Network Exchange. Available online: <https://github.com/onnx/onnx> (accessed on 26 September 2022).