# High-Level Features for Human Activity Recognition and Modeling

Yale Hartmann(✉), Hui Liu, and Tanja Schultz

Cognitive Systems Lab, University of Bremen, Bremen, Germany
{yale.hartmann,hui.liu,tanja.schultz}@uni-bremen.de
http://csl.uni-bremen.de/

**Abstract.** High-Level Features (HLF) are a novel way of describing and processing human activities. Each feature captures an interpretable aspect of activities, and a unique combination of HLFs defines an activity. In this article, we propose and evaluate a concise set of six HLFs on and across the CSL-SHARE and UniMiB SHAR datasets, showing that HLFs can be successfully extracted with machine learning methods and that in this HLF-space activities can be classified across datasets as well as in imbalanced and few-shot learning settings. Furthermore, we illustrate how classification errors can be attributed to specific HLF extractors. In person-independent 5-fold cross-validations, the proposed HLFs are extracted from 68% up to 99% balanced accuracy, and activity classification achieves 89.7% (CSL-SHARE) and 67.3% (UniMiB SHAR) accuracy. Imbalanced and few-shot learning results are promising, with the latter converging quickly. In a person-dependent evaluation across both datasets, 78% accuracy is achieved. These results demonstrate the possibilities and advantages of the proposed high-level, extensible, and interpretable feature space.

**Keywords:** Human activity recognition · High-level features · Interpretable machine learning · Wearable sensors

## 1 Background and Related Works

Human activity recognition (HAR), an important research topic for today's modern life, involves proven machine learning (ML) algorithms for related tasks, including biosignal (pre-)processing, feature extraction and selection, unsupervised data segmentation, and activity modeling approaches, as the classic and the state-of-the-art HAR research pipelines portray [7,28,45]. A suitable algorithm and a high recognition rate are prerequisites for HAR applications to work smoothly [25]. Forefront research works in non-traditional areas, such as device-free HAR [11], the effect of validation methods on HAR performance [6], advanced sensing techniques applicable to HAR [5], and interactive and interpretable online HAR [18], have been emerging.

In most research settings, the domain related to machine learning has received more attention from researchers, focusing on solving the following two problems:

- Which ML approaches are more applicable for the study's objectives, such as activity patterns, application scenarios, and datasets?

A. C. A. Roque et al. (Eds.): BIOSTEC 2022, CCIS 1814, pp. 141–163, 2023.
https://doi.org/10.1007/978-3-031-38854-5_8

- What topological and parametric adjustments should be made to the applied ML approaches to obtain better results (accuracy)?

The first query was fully attended to, and field-related addressed. HAR has been effectively studied on various well-established machine learning models such as Long Short-Term Memory (LSTM) [38,53], Basic and Deep Neural Networks [23,37,50], Convolutional Neural Networks [13,24,38,42,52,55], Recurrent Neural Networks [2,9,21,36,38,44], and Hidden Markov Models (HMM) [29,33,40], among others. More sophisticated modeling schemes based on mature ML models, such as Residual Neural Networks [19,22,32,47] and Hierarchical Hidden Markov Models [54], have also been applied to HAR. With the basic research of these ML models in the field of HAR, the second problem mentioned above has also received favorable attention and is usually solved based on model reformulation, parameter tuning, feature investigation, and iterative optimization [1,2,16,30,41,48].

Noticeably, the vast majority of HAR studies listed above tend to be closely related to, or even just about, ML models. In other words, researchers involved may not care about or cannot effectively investigate the kinematic and physiological implications of "activity" as a study object for human **activity** recognition. In these studies, "human activity" is more of a signal set than a sport or kinesiological phenomenon. Similar deficiencies have been better recognized and addressed in other fields of ML-based pattern recognition. For instance, the mechanics of human voice production has been extensively studied, leading to excellent automatic speech recognition models, such as the three-state HMM-based Bakis-model [3] constructing phonemes by imitating phonetics in segmenting the pronunciation. Each sub-phoneme, represented as one state (begin/middle/end), models a phoneme part, improving the model generalizability and extendability for efficient training and decoding while reflecting the phonetic and biological significance. Moving from audio to video, the physiological model of the human eye's stereo vision has been commonly applied to image/video recognition [10,43].

Looking back at HAR, there is a paucity of literature linking human kinesiology, somatology, physiology, or sportology to ML models. While these connections should be feasible and pivotal, how they are studied requires crossing interdisciplinary divides. A simple attempt in a statistical sense is to derive a reasonable duration based on big data for every single motion in daily life and common sports, providing reference in data segmentation, signal windowing, and modeling [31]. The recently proposed motion units [27] approach is another step forward in bridging the gap from movement science to machine learning, which partitions each human activity into a sequence of shared, meaningful, and activity distinguishing states, analog to phonemes in speech recognition, endowing HAR modeling with operability, interpretability, generalizability, and expandability.

As an exhaustive follow-up to our earlier conference publication [15], this article elaborates on our novel method for practical knowledge incorporation from other fields into HAR, facilitating non-ML experts for developing recognition systems. We propose a feasible approach to allow researchers to design high-level activity properties with their respective possible values, such as Backwards/Neutral/Forwards. Such a set of features, extracted with classifiers, can be optimized or transformed into feature functions in further stages, e.g., with the help of ML experts. A necessary condition for the proposed setup to work well is that each activity is divergent from all other activities on at least one property, which enables the classification to be performed effectively with an easily attributable featuring error discovery and a well interpretable feature space.

The following article includes a further in-depth elaboration of the High-level Features, Error Attribution, and reiterated High-level Features compared to [18]. Furthermore, all experiments have been re-run, and new few-shot experiments were added. All experiments are run on the CSL-SHARE [26], and UniMiB SHAR [35] datasets. These datasets deploy different sensor carriers (knee-bandage and smartphone), sensors and sensor positions (EMG, goniometer, accelerometer, and gyroscope around the knee and accelerometer in different trouser pockets), and mostly different activities with a few shared ones ("jump", "walk", "going up stairs", among others). These properties make the two datasets a perfect fit to demonstrate the utility of High-level Features for Human Activity Recognition.

## 2 High-Level Features

In this work, we propose a set of High-Level Features along with how to extract and utilize them in multiple tasks like classification, few-shot learning, and dataset combination.

### 2.1 Concept

High-Level Features describe properties of human activities independent of the sensor setup used to record them in a human interpretable way. For instance, one HLF entitled *Back/Front* might encode if a person is moving along the frontal direction, e.g., forward, no-movement, or backward. This feature value is assigned based on activity initially, and the association from sensor data to feature value is learned using classification algorithms (see Sect. 3). Therefore, the activity "Walk" will be assigned the *Back/Front* value of *Forward*, while "Walk Back" will be assigned *Backward*, and "Walk sideways" is *Neutral* in the frontal direction.

The feature development is based on the activity target and activity knowledge rather than the sensor data. Therefore, in some sensor setups, they might be impossible to extract. For instance, a sensor setup based on a single IMU can likely not extract the muscle force produced during jumps or bench presses, as the force is dependent not only on the acceleration with which it is moved but also the weight stemmed. Nevertheless, the muscles produce this force during both activities and should be modeled accordingly.

We propose developing these descriptive HLFs such that each activity has a unique combination of feature values. Uniqueness has multiple benefits: activity classification and zero-shot learning are straightforward. Furthermore, this leads to unambiguous activity definitions. Uniqueness means that the one-hot encoded feature value vectors can be looked up in a table for classification. At the same time, uniqueness implies that two activities with identical features must be identical, or another property must distinguish them. Vice versa, a unique combination of feature values implies a specific activity. Both are tremendously helpful for developing HLFs: the first allows thinking about differences between activity pairs, and the latter allows checking how well the features can scale. If most combinations are impossible, it might make sense to re-think the features.

We propose treating the feature extraction as a classification task, thereby learning the relationship between data and defined feature values in a data-driven fashion. One could either use a single classifier to predict all feature values utilizing the possible interdependence between features or treat them as independent by training a different classifier per HLF. Both can be achieved with any machine learning classifier, with different benefits. Here we focus on the latter due to its extensibility and flexibility of adding or removing features by excluding or training classifiers, as well as the ability to attribute errors made during classification to a specific feature extractor.

Feature extraction is sensor and dataset-dependent due to the classification procedure. However, classification in the extracted high-level feature space is sensor-independent due to its knowledge-driven and activity-based design. This independence has multiple advantages: (1) the feature design is not limited to a specific dataset, which results in scalable and meaningful features, (2) in this feature space, we can combine multiple datasets for comparison, classification, and modeling, and (3) the feature extraction itself can be adjusted to each datasets specifics, like sampling-rates, missing values, or sensors.

## 2.2 Proposed High-Level Features

We proposed the HLF concept along with eight specific features in our previous work [18]. Here, we further developed this into a more concise HLF-set utilizing the previously mentioned error propagation (see Sect. 5). We developed six High-Level Features: *Back/Front*, *Left/Right*, *Up/Down*, *Force*, *Knee*, and *Impact*.

***Back/Front.*** describes if the person moved along the sagittal plane. For instance, because they walk forward or sit in a moving car. It can hold three different values: *Back*, *Neutral*, and *Forward*. Note that *Neutral* does not imply the person is sitting still. They could be moving to either side or up and down, just not forward or backward.

***Left/Right.*** similarly describes movement along the frontal plane. However, it can take multiple values, describing the subtlety of the movement: *Left-extreme* (large left movement in short time), *Left*, *Left-slightly* (subtle left movement in long time), *Neutral*, *Right-slightly*, *Right*, *Right-extreme*, and *Any* (indicating there is movement, but not clearly defined). As discussed previously, these are required to distinguish left/right shuffles from left/right curves and slight curves.

***Up/Down.*** describes movement along the transverse plane. It can take the values *Down*, *Neutral*, *Up*, and *Updown*. The latter is required to describe jumps, during which the body initially goes up and sinks back down again. Currently, the HLFs are extracted as one value for the whole segmented activity and, therefore, a trade-off in modeling is required for jumps. One solution would be to aim for *Up/Down* to model the starting movement of an activity, in which case jump would be assigned *Up*. The same argument goes for the ending of a movement. Neither truly captures the movement of the jump activity, which is why we opted to use the special case *UpDown* for now. The best solution would be to model HLFs in sequences which values may change over time. We discuss this further in Sect. 8.

***Force.*** describes the force the upper (right) leg muscles need to generate during this activity. It can take the values *N.a./None* (no notable usage of the muscles), *Low*, *Medium*, and *High*. This HLF is primarily required to distinguish jumping with one and jumping with two legs but also supports distinguishing multiple other activities.

***Knee.*** describes the movement or posture of the knees during the activity and is mainly required to distinguish the only two static activities in the two used datasets: stand and sit. It can take the values *Bent* or *Straight* (for postures) as well as *Move* and *Falter* (for movements). Additionally, it can take the values *Left knee first* and *Right knee first*, which brings the total different values for the *Knee* HLF up to six. The latter two are technically a special case of the *Move* value and are currently required due to activity distinctions in the CSL-SHARE dataset.

***Impact.*** describes the amount or quality of impacts broadly. For falls, it takes the value *One*, as in one larger impact when hitting the ground. Special cases are if the person is hitting something during the fall but still hitting the ground (UniMiB SHAR: "HittingObstacle"), in which case it can take the value *Two*. Similarly, the person falling might soften the fall by extending their arms, for which the value *Softened* is chosen. During static activities, the feature may take the value *N.a./None*, and during gait-related activities *Several*. Therefore, resulting in five different values for *Impact*.

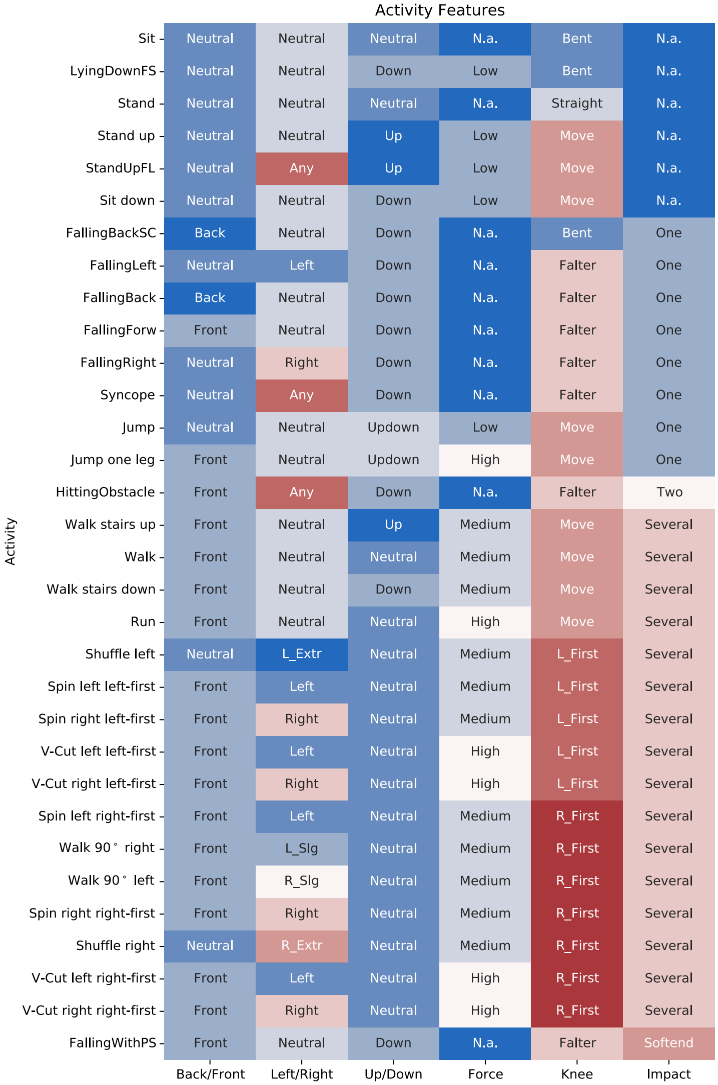
### 2.3 HLF Assignments

The full HLF assignment table across both datasets is depicted in Fig. 1. Each column displays an activity's unique feature value combination, while each row displays the feature values across all activities. For easier readability, the table is color coded. Note that the color is only consistent in each row. *Back* in *Back/Front* is coded dark blue for all *Backs* in this feature, while dark blue for *Up/Down* is used for all *Ups*.

Figure 1 shows that each activity in this feature space is uniquely coded and that multiple clusters of activities exist with similar feature values. For instance, the gait-related activities share most features except *Left/Right* and *Knee*.

Plotting the distance between activities in this new feature space reveals these clusters further. Figure 2 shows the distance between each pair of activities, with red regions indicating low distance and blue regions high distance. The distance is measured as the number of feature values in which a pair of activities differ, which correlates to the euclidean distance of the one-hot encoded feature values per activity.

The main clusters emerge for falls and gait-based activities. Furthermore, it can be observed that the V-Cut and Spin activities create their own clusters as well as share most of their features with each other. More interesting is that the falls are less dissimilar to the more static activities like sitting, standing, or sitting down than the gait and sports-related activities. Intuitively this makes sense, as falls are no active activities and thus

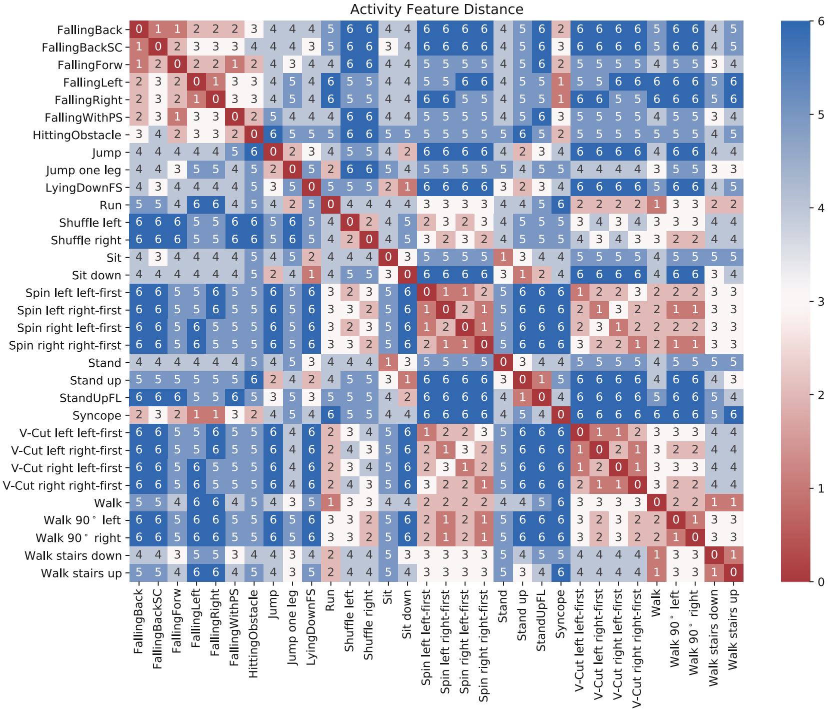


Activity Features

| Activity | Back/Front | Left/Right | Up/Down | Force | Knee | Impact |
|---|---|---|---|---|---|---|
| Sit | Neutral | Neutral | Neutral | N.a. | Bent | N.a. |
| LyingDownFS | Neutral | Neutral | Down | Low | Bent | N.a. |
| Stand | Neutral | Neutral | Neutral | N.a. | Straight | N.a. |
| Stand up | Neutral | Neutral | Up | Low | Move | N.a. |
| StandUpFL | Neutral | Any | Up | Low | Move | N.a. |
| Sit down | Neutral | Neutral | Down | Low | Move | N.a. |
| FallingBackSC | Back | Neutral | Down | N.a. | Bent | One |
| FallingLeft | Neutral | Left | Down | N.a. | Falter | One |
| FallingBack | Back | Neutral | Down | N.a. | Falter | One |
| FallingForw | Front | Neutral | Down | N.a. | Falter | One |
| FallingRight | Neutral | Right | Down | N.a. | Falter | One |
| Syncope | Neutral | Any | Down | N.a. | Falter | One |
| Jump | Neutral | Neutral | Updown | Low | Move | One |
| Jump one leg | Front | Neutral | Updown | High | Move | One |
| HittingObstacle | Front | Any | Down | N.a. | Falter | Two |
| Walk stairs up | Front | Neutral | Up | Medium | Move | Several |
| Walk | Front | Neutral | Neutral | Medium | Move | Several |
| Walk stairs down | Front | Neutral | Down | Medium | Move | Several |
| Run | Front | Neutral | Neutral | High | Move | Several |
| Shuffle left | Neutral | L_Extr | Neutral | Medium | L_First | Several |
| Spin left left-first | Front | Left | Neutral | Medium | L_First | Several |
| Spin right left-first | Front | Right | Neutral | Medium | L_First | Several |
| V-Cut left left-first | Front | Left | Neutral | High | L_First | Several |
| V-Cut right left-first | Front | Right | Neutral | High | L_First | Several |
| Spin left right-first | Front | Left | Neutral | Medium | R_First | Several |
| Walk 90° right | Front | L_Slg | Neutral | Medium | R_First | Several |
| Walk 90° left | Front | R_Slg | Neutral | Medium | R_First | Several |
| Spin right right-first | Front | Right | Neutral | Medium | R_First | Several |
| Shuffle right | Neutral | R_Extr | Neutral | Medium | R_First | Several |
| V-Cut left right-first | Front | Left | Neutral | High | R_First | Several |
| V-Cut right right-first | Front | Right | Neutral | High | R_First | Several |
| FallingWithPS | Front | Neutral | Down | N.a. | Falter | Softend |

**Fig. 1.** HLFs per activity. The color is coded per feature values per row, with no intentional meaning of color between rows.

do not require muscle force and are always *Neutral* in either *Back/Front* or *Left/Right*, while the gait-based activities will seldom be *Neutral* in the *Back/Front* HLF. One could roughly group activities based on these HLFs into three categories: fall, gait-based, and lounge. Note that these observations span two separate datasets because the HLFs design is based on the activities rather than the sensor data of these two datasets.



| | FallingBack | FallingBackSC | FallingForw | FallingLeft | FallingRight | FallingWithPS | HittingObstacle | Jump | Jump one leg | LyingDownFS | Run | Shuffle left | Shuffle right | Sit | Sit down | Spin left left-first | Spin left right-first | Spin right left-first | Spin right right-first | Stand | Stand up | StandUpFL | Syncope | V-Cut left left-first | V-Cut left right-first | V-Cut right left-first | V-Cut right right-first | Walk | Walk 90 ° left | Walk 90 ° right | Walk stairs down | Walk stairs up |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FallingBack | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4 | 4 | 5 | 6 | 6 | 4 | 4 | 6 | 6 | 6 | 6 | 4 | 5 | 6 | 2 | 6 | 6 | 6 | 6 | 5 | 6 | 6 | 4 | 5 |
| FallingBackSC | 1 | 0 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 3 | 5 | 6 | 6 | 3 | 4 | 6 | 6 | 6 | 6 | 4 | 5 | 6 | 3 | 6 | 6 | 6 | 6 | 5 | 6 | 6 | 4 | 5 |
| FallingForw | 1 | 2 | 0 | 2 | 2 | 1 | 2 | 4 | 3 | 4 | 4 | 6 | 6 | 4 | 4 | 5 | 5 | 5 | 5 | 4 | 5 | 6 | 2 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 3 | 4 |
| FallingLeft | 2 | 3 | 2 | 0 | 1 | 3 | 3 | 4 | 5 | 4 | 6 | 5 | 5 | 4 | 4 | 5 | 5 | 6 | 6 | 4 | 5 | 5 | 1 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 5 | 6 |
| FallingRight | 2 | 3 | 2 | 1 | 0 | 3 | 3 | 4 | 5 | 4 | 6 | 5 | 5 | 4 | 4 | 6 | 6 | 5 | 5 | 4 | 5 | 5 | 1 | 6 | 6 | 5 | 5 | 6 | 6 | 6 | 5 | 6 |
| FallingWithPS | 2 | 3 | 1 | 3 | 3 | 0 | 2 | 5 | 4 | 4 | 4 | 6 | 6 | 4 | 4 | 5 | 5 | 5 | 5 | 4 | 5 | 6 | 3 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 3 | 4 |
| HittingObstacle | 3 | 4 | 2 | 3 | 3 | 2 | 0 | 6 | 5 | 5 | 5 | 6 | 6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 5 | 2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 |
| Jump | 4 | 4 | 4 | 4 | 4 | 5 | 6 | 0 | 2 | 3 | 4 | 5 | 5 | 4 | 2 | 6 | 6 | 6 | 6 | 4 | 2 | 3 | 4 | 6 | 6 | 6 | 6 | 4 | 6 | 6 | 4 | 4 |
| Jump one leg | 4 | 4 | 3 | 5 | 5 | 4 | 5 | 2 | 0 | 5 | 2 | 6 | 6 | 5 | 4 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 4 | 4 | 4 | 4 | 3 | 5 | 5 | 3 | 3 |
| LyingDownFS | 4 | 3 | 4 | 4 | 4 | 4 | 5 | 3 | 5 | 0 | 5 | 5 | 5 | 2 | 1 | 6 | 6 | 6 | 6 | 3 | 2 | 3 | 4 | 6 | 6 | 6 | 6 | 5 | 6 | 6 | 4 | 5 |
| Run | 5 | 5 | 4 | 6 | 6 | 4 | 5 | 4 | 2 | 5 | 0 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 4 | 4 | 5 | 6 | 2 | 2 | 2 | 2 | 1 | 3 | 3 | 2 | 2 |
| Shuffle left | 6 | 6 | 6 | 5 | 5 | 6 | 6 | 5 | 6 | 5 | 4 | 0 | 2 | 4 | 5 | 2 | 3 | 2 | 3 | 4 | 5 | 5 | 5 | 3 | 4 | 3 | 4 | 3 | 3 | 3 | 4 | 4 |
| Shuffle right | 6 | 6 | 6 | 5 | 5 | 6 | 6 | 5 | 6 | 5 | 4 | 2 | 0 | 4 | 5 | 3 | 2 | 3 | 2 | 4 | 5 | 5 | 5 | 4 | 3 | 4 | 3 | 3 | 2 | 2 | 4 | 4 |
| Sit | 4 | 3 | 4 | 4 | 4 | 4 | 5 | 4 | 5 | 2 | 4 | 4 | 4 | 0 | 3 | 5 | 5 | 5 | 5 | 1 | 3 | 4 | 4 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 5 | 5 |
| Sit down | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 2 | 4 | 1 | 4 | 5 | 5 | 3 | 0 | 6 | 6 | 6 | 6 | 3 | 1 | 2 | 4 | 6 | 6 | 6 | 6 | 4 | 6 | 6 | 3 | 4 |
| Spin left left-first | 6 | 6 | 5 | 5 | 6 | 5 | 5 | 6 | 5 | 6 | 3 | 2 | 3 | 5 | 6 | 0 | 1 | 1 | 2 | 5 | 6 | 6 | 6 | 1 | 2 | 2 | 3 | 2 | 2 | 2 | 3 | 3 |
| Spin left right-first | 6 | 6 | 5 | 5 | 6 | 5 | 5 | 6 | 5 | 6 | 3 | 3 | 2 | 5 | 6 | 1 | 0 | 2 | 1 | 5 | 6 | 6 | 6 | 2 | 1 | 3 | 2 | 2 | 1 | 1 | 3 | 3 |
| Spin right left-first | 6 | 6 | 5 | 6 | 5 | 5 | 5 | 6 | 5 | 6 | 3 | 2 | 3 | 5 | 6 | 1 | 2 | 0 | 1 | 5 | 6 | 6 | 6 | 2 | 3 | 1 | 2 | 2 | 2 | 2 | 3 | 3 |
| Spin right right-first | 6 | 6 | 5 | 6 | 5 | 5 | 5 | 6 | 5 | 6 | 3 | 3 | 2 | 5 | 6 | 2 | 1 | 1 | 0 | 5 | 6 | 6 | 6 | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 3 | 3 |
| Stand | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 4 | 5 | 3 | 4 | 4 | 4 | 1 | 3 | 5 | 5 | 5 | 5 | 0 | 3 | 4 | 4 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 5 | 5 |
| Stand up | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 2 | 4 | 2 | 4 | 5 | 5 | 3 | 1 | 6 | 6 | 6 | 6 | 3 | 0 | 1 | 5 | 6 | 6 | 6 | 6 | 4 | 6 | 6 | 4 | 3 |
| StandUpFL | 6 | 6 | 6 | 5 | 5 | 6 | 5 | 3 | 5 | 3 | 5 | 5 | 5 | 4 | 2 | 6 | 6 | 6 | 6 | 4 | 1 | 0 | 4 | 6 | 6 | 6 | 6 | 5 | 6 | 6 | 5 | 4 |
| Syncope | 2 | 3 | 2 | 1 | 1 | 3 | 2 | 4 | 5 | 4 | 6 | 5 | 5 | 4 | 4 | 6 | 6 | 6 | 6 | 4 | 5 | 4 | 0 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 5 | 6 |
| V-Cut left left-first | 6 | 6 | 5 | 5 | 6 | 5 | 5 | 6 | 4 | 6 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 2 | 3 | 5 | 6 | 6 | 6 | 0 | 1 | 1 | 2 | 3 | 3 | 3 | 4 | 4 |
| V-Cut left right-first | 6 | 6 | 5 | 5 | 6 | 5 | 5 | 6 | 4 | 6 | 2 | 4 | 3 | 5 | 6 | 2 | 1 | 3 | 2 | 5 | 6 | 6 | 6 | 1 | 0 | 2 | 1 | 3 | 2 | 2 | 4 | 4 |
| V-Cut right left-first | 6 | 6 | 5 | 6 | 5 | 5 | 5 | 6 | 4 | 6 | 2 | 3 | 4 | 5 | 6 | 2 | 3 | 1 | 2 | 5 | 6 | 6 | 6 | 1 | 2 | 0 | 1 | 3 | 3 | 3 | 4 | 4 |
| V-Cut right right-first | 6 | 6 | 5 | 6 | 5 | 5 | 5 | 6 | 4 | 6 | 2 | 4 | 3 | 5 | 6 | 3 | 2 | 2 | 1 | 5 | 6 | 6 | 6 | 2 | 1 | 1 | 0 | 3 | 2 | 2 | 4 | 4 |
| Walk | 5 | 5 | 4 | 6 | 6 | 4 | 5 | 4 | 3 | 5 | 1 | 3 | 3 | 4 | 4 | 2 | 2 | 2 | 2 | 4 | 4 | 5 | 6 | 3 | 3 | 3 | 3 | 0 | 2 | 2 | 1 | 1 |
| Walk 90 ° left | 6 | 6 | 5 | 6 | 6 | 5 | 5 | 6 | 5 | 6 | 3 | 3 | 2 | 5 | 6 | 2 | 1 | 2 | 1 | 5 | 6 | 6 | 6 | 3 | 2 | 3 | 2 | 2 | 0 | 1 | 3 | 3 |
| Walk 90 ° right | 6 | 6 | 5 | 6 | 6 | 5 | 5 | 6 | 5 | 6 | 3 | 3 | 2 | 5 | 6 | 2 | 1 | 2 | 1 | 5 | 6 | 6 | 6 | 3 | 2 | 3 | 2 | 2 | 1 | 0 | 3 | 3 |
| Walk stairs down | 4 | 4 | 3 | 5 | 5 | 3 | 4 | 4 | 3 | 4 | 2 | 4 | 4 | 5 | 3 | 3 | 3 | 3 | 3 | 5 | 4 | 5 | 5 | 4 | 4 | 4 | 4 | 1 | 3 | 3 | 0 | 1 |
| Walk stairs up | 5 | 5 | 4 | 6 | 6 | 4 | 5 | 4 | 3 | 5 | 2 | 4 | 4 | 5 | 4 | 3 | 3 | 3 | 3 | 5 | 3 | 4 | 6 | 4 | 4 | 4 | 4 | 1 | 3 | 3 | 1 | 0 |

**Fig. 2.** Distance between activities within the HLF-space. Measured as the number of distinguishing feature values.

## 3 Feature Extraction

Correct extraction of the High-Level Features from given sensor data is crucial for any further task, including classification and analysis. We treat feature extraction as a classification task for each feature separately. Therefore, the machine learning task is to learn the relationship between the sensory data of an activity to its assigned categorical feature value for each HLF. As discussed in Sect. 2, one could also utilize a single larger extractor. However, we opted for the extensibility, explainability, and the possibility of attributing classification errors to specific feature extractors instead of the single large one when utilizing multiple independent feature extractors.

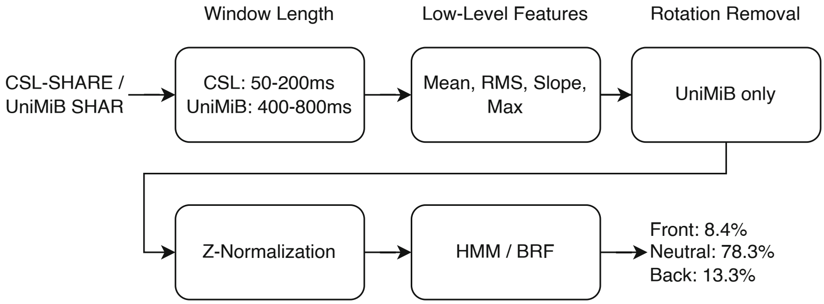


**Fig. 3.** HLF Extractor pipeline with preprocessing and classification stages.

Figure 3 depicts the HLF extractors. The initial sequence is windowed, four features are extracted (Mean, RMS, Slope, Max), and the whole sequence is normalized. In the case of the UniMiB SHAR dataset, a simple rotation removal is employed to mitigate the different phone positions, as described in [17]. Lastly, two classifiers are tested: a Hidden Markov Model and a Balanced Random Forest (BRF) [8]. The former fully supports sequences, and the latter is trained on a fixed vector length of the first 50 low-level features. On the UniMiB SHAR dataset, this entails the full sequence (as all sequences are 3 s long), and on the CSL-SHARE, this translates to the first half second.

Our previous work deployed out-of-the-box Random Forests, BRFs, and HMMs with only slight parameter tuning for each HLF [15]. Very similar, we evaluated HMMs and BRFs here, tuning both slightly to treat them as mostly out-of-the-box. The BRF is chosen here as with the proposed HLF assignment each HLF has one or two dominant and multiple less-represented values due to the datasets' activities, and humans preferring to walk forward instead of backward, as seen in Fig. 4.

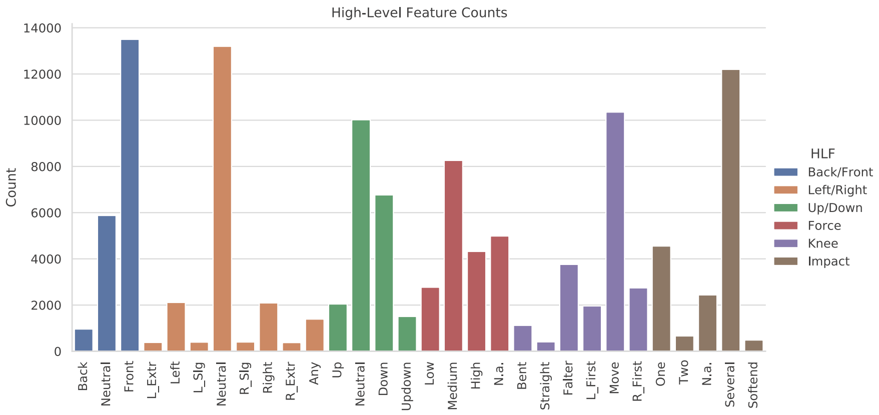


**Fig. 4.** Number of samples for each feature value across datasets. The color indicates the values' corresponding HLF.

The Balanced Random Forests are tuned as to the window length and overlap for each HLF. Tuning resulted in window lengths of 50ms (CSL-SHARE) and 800ms (UniMiB SHAR), with 80% overlap to perform best. The HMMs are tuned for window length and overlap, and two general-purpose topologies were tested. The first topology plainly uses three states for each HLF target value. The second topology has a target-specific second to fourth state and shares its initial and fifth state across all HLF values. This design aims to allow the HMM to pick the most representative 3-phase subsequence for each HLF by allowing the first and last state to take arbitrary portions off of the sequence, leaving the middle states to distinguish between targets. Parameter tuning revealed 100ms and 200ms, depending on HLF (CSL-SHARE), 400ms (UniMiB SHAR) window length and 80% overlap to work best. Both topologies performed well and were mostly on par on the CSL-SHARE dataset. The plain three states outperformed the shared topology on the UniMiB SHAR consistently. This performance contrasts [17] and will be investigated further in future work.

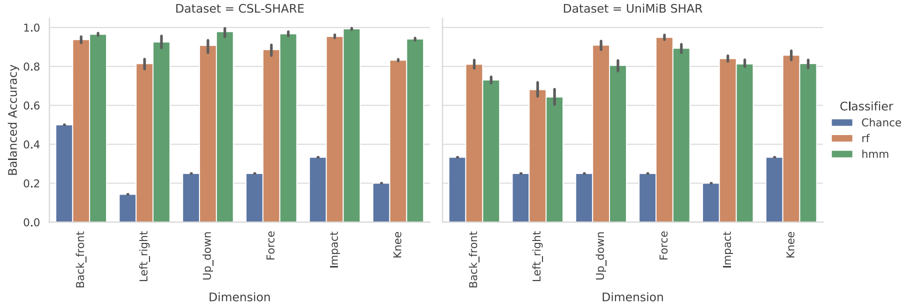


**Fig. 5.** Summarized HLF extraction performance from 5-fold person-independent evaluation, listing HMMs, BRFs, and chance level. Metric is the balanced accuracy.

For each of the six HLFs and each dataset, a 5-Fold person-independent cross-validation was conducted. The balanced accuracy is used as a metric, as each HLF typically contains one dominant and multiple underrepresented feature values, as seen in Fig. 4. All cross-validation results, including the chance level, are summarized in Fig. 5. Both classifiers significantly outperform the chance level. The HMM consistently is better than the BRF on the CSL-SHARE, while the opposite holds on the UniMiB SHAR dataset. Performance on the CSL-SHARE dataset ranges from 92% (*Left/Right*) to 99% (*Impact*) and ranges from 68% (*Left/Right*) to 94% (*Force*) balanced accuracy on the UniMiB SHAR dataset. In both cases, each single feature extraction performance is higher or close to state-of-the-art classification balanced accuracy. While very encouraging, note that error propagates, and the final classification performance is below state-of-the-art.

## 4 Activity Classification

Activity Classification in this HLF space is straightforward. There are two main options: utilizing the unique combination property to look up the activity based on the extracted features or training another classifier in this new space. The former is especially interesting for zero-shot learning or error attribution (see Sect. 5), while the latter has the potential to counter-balance difficulties in extracted HLFs.

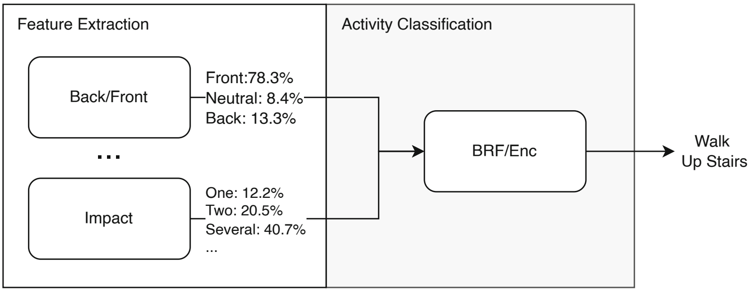


**Fig. 6.** Classifier stages with HLF extractors.

Figure 6 depicts the classification process. The HLFs are extracted as described in Sect. 3 and are configured to return the probability for each possible HLF value, e.g., the *Back/Front* extractor will return how likely the Front/Neutral/Back values are given a sensor data sequence. We found the extra information of the classifier confidences benefited the final classification, but of course, a one-hot encoding from the extractors would also be possible. All HLF probabilities are then stacked into a single feature vector given to the classifier.

The encoding classifier calculates the euclidean distance between the given vector and each activity, represented as stacked one-hot encoding of the assigned HLFs, and picks the closest one. This approach allows zero-shot activity prediction. Given the extracted features and definition of unseen activities, the classifier can predict them. Furthermore, with the encoding classifier, classification errors can be attributed to the different features. If walking was predicted to be walking a curve left, the reason for the misclassification must be the *Left/Right* HLF.

The BRF classifier is simply trained to learn the relation between the stacked probabilities and the activity label. The BRF may learn to counter-balance the difficulties from the extractors by associating and counteracting low confidences of HLF extractions.

Activity classification in this high-level feature space is evaluated in a 5-fold person-independent evaluation using the accuracy and balanced accuracy as metrics. The HLF Extractors are configured with the previously found hyperparameters (see Sect. 3) but are retrained inside the 5-fold cross-validation. Retraining is important to ensure the high-level extractors have not been trained on test samples, thus overestimating the final classification performance. Both a BRF-based and an encoding-based classifier are evaluated.

The final classification performance with the BRF-based classifier is 89.7% (CSL-SHARE) and 67.3% accuracy (UniMiB SHAR). These accuracies trail the previously reported 93.7% (CSL-SHARE) and 77.0% accuracy (UniMiB SHAR) in a leave-one-person-out cross-validation [17]. While 5-fold person-independent cross-validation can underestimate the performance due to a limitation in available training data compared to a leave-one-person-out validation, the results from Sect. 6 indicate that the errors are not made because of too little data. [17] employed (low-level) feature space transformations using a combination of HMMs and an LDA, which should be evaluated with the high-level feature extraction in future work.

The final classification performance with the Encoding-based classifier is 88.1% (CSL-SHARE) and 66.4% accuracy (UniMiB SHAR). These are slightly lower than with the BRF classifier. However, the following analysis mainly focuses on the encoding-based classifier as the interpretation and the relation between final classification and HLFs are solely based on to the HLF extraction.

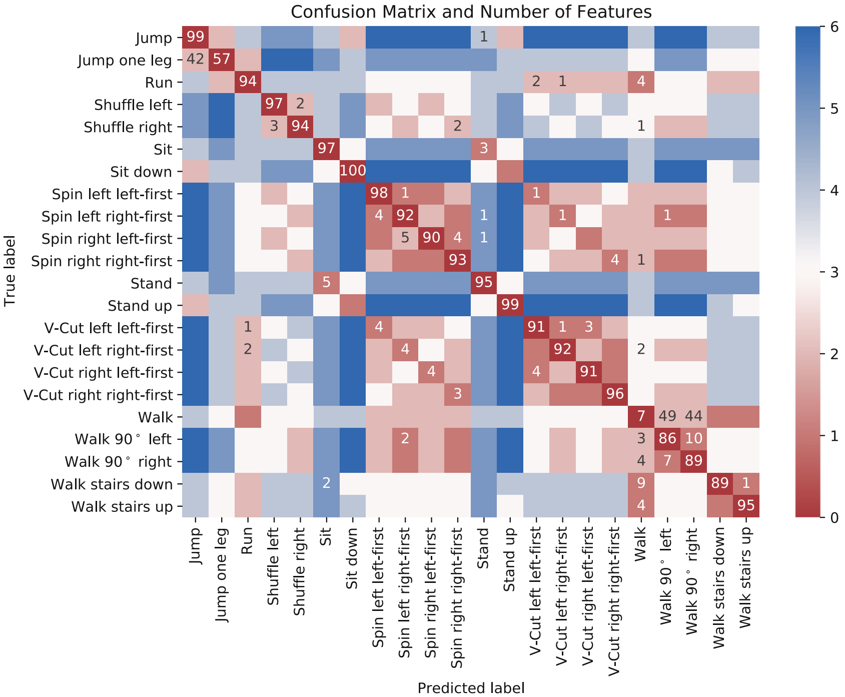


**Fig. 7.** Blended confusion and distance matrix for the Encoding-based classifier on the CSL-SHARE dataset. The color indicates distance, and values indicate the percentage of cases. (Color figure online)

Figure 7 shows the confusion matrix for the CSL-SHARE dataset blended with the distance matrix shown previously. The color indicates feature distance, and the number indicates the percentage of cases. The main error is that walking in a straight line is often predicted to be walking 90° left/right in three steps. The other cluster of errors happens within the spin activities as well as between the spins and their v-cut counterparts. Notably, almost all errors occur between low to medium-distance activities (colored red).

Figure 8 shows the blended matrix for the UniMiB SHAR dataset. Similar to CSL-SHARE, most errors occur between low-distance activities. There are three main clusters of errors: between types of falls, between walking and walking the stairs, and then between in the distinction between sitting and lying. The confusion within the falls on this dataset has been reported previously [17]. Interestingly at no time falling forward is correctly predicted; instead, falling left/right and syncope seem to sink most falls. The last cluster is especially interesting as sitting down is mainly misclassified as lying down from sitting, and standing up is mostly confused with standing up from lying, but standing up (from lying) is misclassified as lying down from sitting.

## 5 Error Attribution

Activity classification is never perfect, and one interesting question is why a misclassification was made. In the High-Level Feature space with unique feature combinations for activities, the classification error can be attributed to the underlying feature extraction by determining the different HLFs distinguishing the actual and predicted activity. For example, if walking is classified as walking left, this error can be attributed to the *Left/Right* and *Knee* HLF, as these are the distinguishing HLFs between the two activities. Encoding-based classifiers render this straightforward, as the distance from the extracted HLF to the definition of walk and between the extracted HLF to walking left is the same in all but the distinguishing HLFs. Therefore, this misclassification occurs due to the erroneous extraction in at least one of these HLFs. The following error attribution analysis is based on the encoding-based classification experiment from Sect. 4.

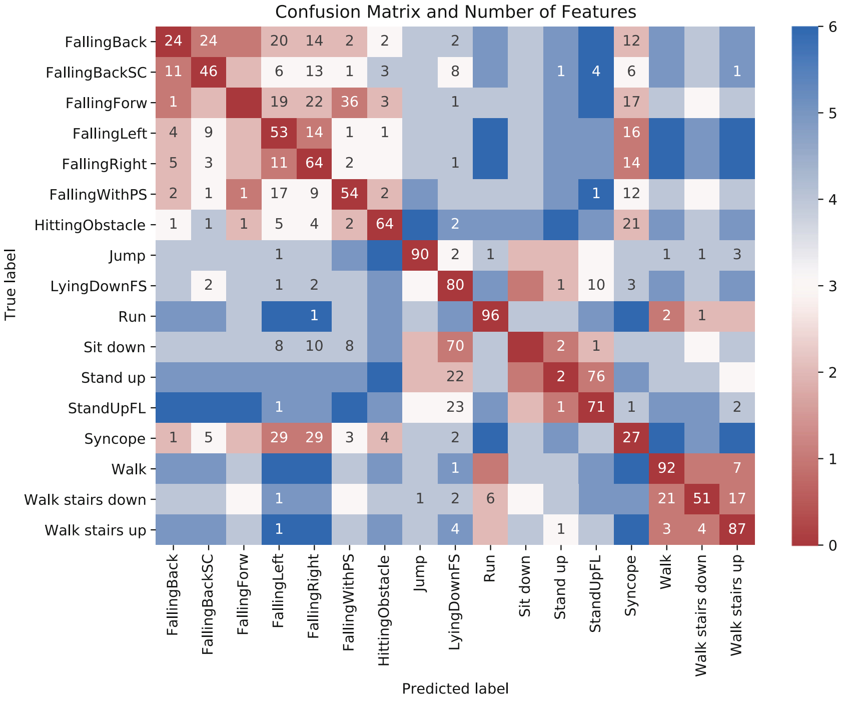


**Fig. 8.** Blended confusion and distance matrix for the Encoding-based classifier on the UniMiB SHAR dataset. The color indicates distance, and values indicate the percentage of cases.

Figure 9 shows the number of times a distinguishing feature was wrongly extracted during the experiment on CSL-SHARE. Distinguishing here means that this HLF value differs between the true and the predicted activity. Wrongly extracted features that are not part of the distinguishing set between the two confused activities are ignored as they did not influence the classification. The color and upper number indicate the number of times the HLF was wrongly extracted divided by the total number of these activities, while the number in brackets divides by the number of errors. That is to say, the former indicates the impact on the global accuracy while the latter indicates how many errors this HLF is partial in. Take jumping with one leg: 41% of its instances are classified as something else due to the *Back/Front* feature not being *Forward*. *Back/Front* is wrongly extracted in 95% of the cases where jumping with one leg was confused.

The two main errors on the CSL-SHARE dataset are the *Left/Right* and *Knee* HLF extractions causing walking to be misclassified in almost all instances, as well as *Back/Front* (in combination with *Force*), causing jumping with one leg to be confused with jumping with two legs.

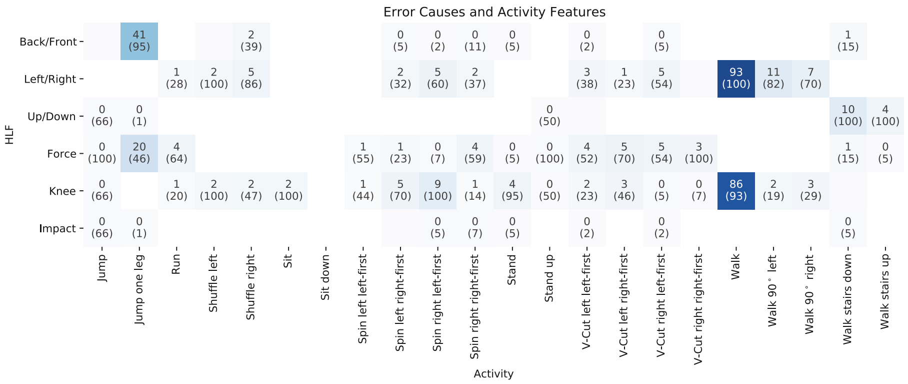


**Fig. 9.** Errors in activity distinguishing HLF extractors on the CSL-SHARE dataset. Color and upper value indicate the percentage of cases of all true activities. Value in brackets indicates the percentage of cases of all misclassified activities.

The walking activity is most often confused with walking in a slight curve (see Fig. 9). Figure 10 shows that in 93% of all actual walking activities, the *Left/Right* feature was incorrectly extracted and that in 100% of misclassified walks the *Left/Right* feature was incorrect. Indicating that these two features are tough to extract for this activity, which implies that either the HLF assignment needs to be reconsidered, further sensory data included, or further development of the extractor is necessary. This particular difficulty has occurred within other works [17] as well, but on a much smaller scale, meaning that clearer sensor data is likely helpful, but more importantly, the extraction and HLF assignment need to be further investigated. The confusion around jump with one leg mainly stems from the *Back/Front* feature, but *Force* also plays a role. Possibly because jumping with one leg is moving forward, but not the same way as walking is and thus is not far away from *Neutral* either, which includes jumping in place which as an activity is not far away from jumping with two legs.

Figure 10 shows the error causes on the UniMiB SHAR dataset. The leading error is that the extracted *Knee* feature in sit down does not match the defined one. Looking closer reveals multiple clusters: *Left/Right* is misclassified in most errors made on the falls. As these are mostly descriptive on the HLF assignment, this likely is a problem in the extraction being confused by the different device orientations present in the UniMiB SHAR dataset. Similarly, *Left/Right* and *Knee* are very present in the activities in and around standing and sitting. Lastly, *Up/Down* is causing problems in classifying walking down stairs, which makes sense as this is a walking-based activity and the only one in the *Down* group along with all falls with vastly different sensor data making it hard to find commonalities on how to extract these features.

## 6 Few-Shot Learning

Few-shot, zero-shot, and learning with imbalanced data are essential tasks in Human Activity Recognition, as many activities cannot be extensively recorded. A typical example is falling for which little or mostly simulated data exist. Nevertheless, falls are crucial to be recognized accurately. The proposed HLF extraction and classification have a significant advantage here, as data is shared across activities via the HLFs grouping.

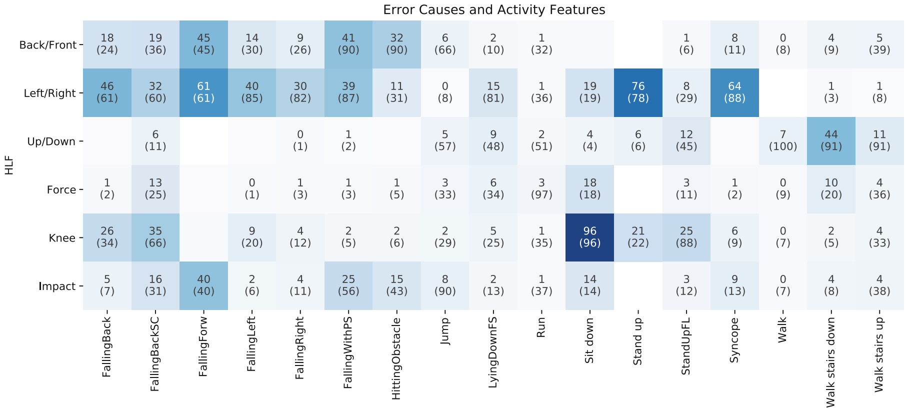


**Fig. 10.** Errors in activity distinguishing HLF extractors on the UniMiB SHAR dataset. Color and upper value indicate the percentage of cases of all true activities. Value in brackets indicates the percentage of cases of all misclassified activities.

### 6.1 Imbalanced Learning

Simulated imbalanced data experiments are adapted from our previous work and re-run in this new feature space and hyperparameters [15]. For each activity, the classifier is evaluated as in Sect. 4, except that from that activity, only a subset of randomly chosen samples is kept simulating a highly imbalanced dataset. Four metrics are recorded: the overall accuracy per fold and three low-resource activity-specific metrics. Namely, the f1 score of that activity only, the percentage of correctly extracted one-hot HLF vectors, and lastly the mean errors in HLF extraction.

The results for the imbalanced experiments are depicted in Fig. 11 and are consistently a few percentage points higher than previously reported [15]. Accordingly, while very promising further improvement in the HLF extraction across sensor data from different activities is required.

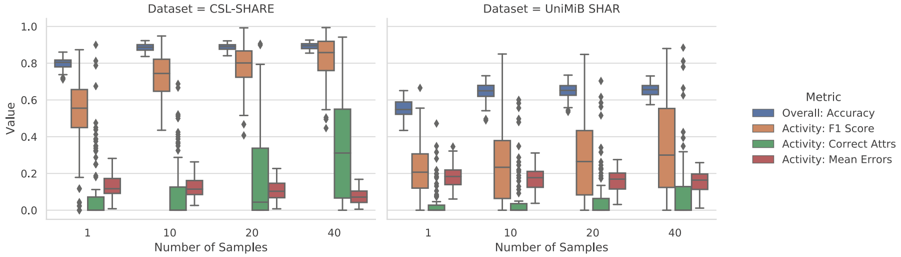


**Fig. 11.** Imbalanced performance on CSL-SHARE and UniMiB SHAR datasets. Averaged across 5-fold person-independent cross-validation and rotating low-resourced activity.

### 6.2 Few-Shot Learning

Few-Shot learning experiments investigate how little data is required to achieve good performance and if fewer than the recorded data might suffice. Similar to the imbalanced experiment, the proposed method has an advantage due to data sharing. The *Back/Fronts* value *Neutral* is associated with multiple activities sensor data.

A 5-fold person-independent cross-validation is run. Similar to the imbalanced case, the training data is subsampled. Here $n \in \{1, 10, 20, 40, 80, 150, 300\}$ sequences are sampled from each activity. In the case of $n = 1$, the full feature extraction and final classification are trained on 22 (CSL-SHARE) and 17 (UniMiB SHAR) samples and evaluated on the full test split. For comparison: in the above classification experiments (see Sect. 4), 310 (CSL-SHARE) and 550 (UniMiB SHAR) training samples existed per activity per fold. Note that the UniMiB SHARE dataset is imbalanced, and some activities might already be fully present in earlier stages.

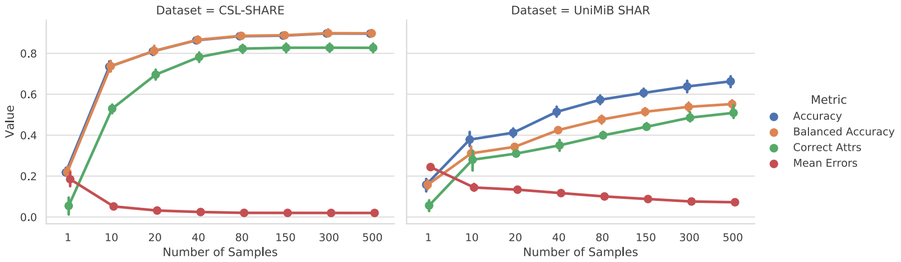


**Fig. 12.** Few-shot performance on CSL-SHARE and UniMiB SHAR datasets. Averaged across 5-fold person-independent cross-validation and for increasing numbers of sampled sequences.

Figure 12 shows the accuracy, balanced accuracy, the percentage of fully correct extracted HLF values, and the mean HLF extraction errors for each evaluated sampled subsets training-fold. The classifier on the CSL-SHARE dataset converges very quickly after around 40 samples per activity, while the classifier on the UniMiB SHAR dataset converges steadily after 40 samples per activity. Note that the percentage of fully correct extracted features in this experiment is calculated over all activities, while in the imbalanced experiments above, it is calculated over the activity in question alone.

The difference is stark: while in the imbalanced experiment, the median of fully extracted feature values was at a maximum of 35% (CSL-SHARE, 40 samples), in the few-shot experiment, it reached a mean of almost 80% after 40 samples while having a much lower standard deviation. Therefore, indicating a problem in feature extraction for these highly imbalanced cases. These results are encouraging for good classification performance on small datasets.

## 7 Dataset Combination

Sensor choices along with feature choices in HAR typically follow the specific activities that should be discriminated. Even comparing the same activity like “walking” across datasets is impossible as soon as different sensors or sensor positions were chosen. However, comparing activities across sensor setups is of high interest and supports understanding the activities as well as their modeling. The following experiment focuses on classification in this high-level feature space, comparison and design are discussed in Sect. 2, and further experiments are planned for future work.

The HLFs are by design sensor independent, allowing to combine multiple datasets in this space and, thus, enabling comparison and classification of the different activities. The transformation into the HLF space is achieved by having two dataset-dependent feature extractions with the same HLFs. The extracted and stacked vectors then have the same dimension across datasets, meaning that the activity classification step is trained on these vectors to predict the original dataset-specific activity labels. Not unlike an interpretable equivalent to swapping the first layers of a Neural Network depending on input while keeping the last layers fixed. The activity-level classification is, therefore, independent of the dataset, which means that now “V-Cuts” (CSL-SHARE) need to be distinguished from “Falls” (UniMiB SHAR). At the same time, both datasets have shared activities like "Walk" resulting in more samples compared to each individual dataset.

The following experiment combines the CSL-SHARE and UniMiB SHAR datasets in a classification experiment. It is not easy to ensure the occurrence of an activity that only exists in one dataset to be in both the training and test set but from different people in a person-independent evaluation across datasets. The main problem is an activity only occurring in the test set as is the case in zero-shot learning. Therefore, a 10-fold person-dependent stratified evaluation scheme is deployed.

The resulting 78% accuracy with both the Encoding and BRF-based HLF classifier is in the middle between the 88% on the CSL-SHARE and the 67% on the UniMiB SHAR dataset (see Sect. 4). This result is noteworthy with the scheme being person-dependent instead of person-independent.

The blended confusion matrix for the encoding-based classifier is shown in Fig. 13. The only difference for the activities occurring in only one dataset (like "FallingBack" or "Walk 90° left") is the person-dependent evaluation and, therefore, feature extraction. Accordingly, it is reasonable to attribute the better performance of falling forward/left/right, and walking 90° left/right to the evaluation scheme. Data of a person

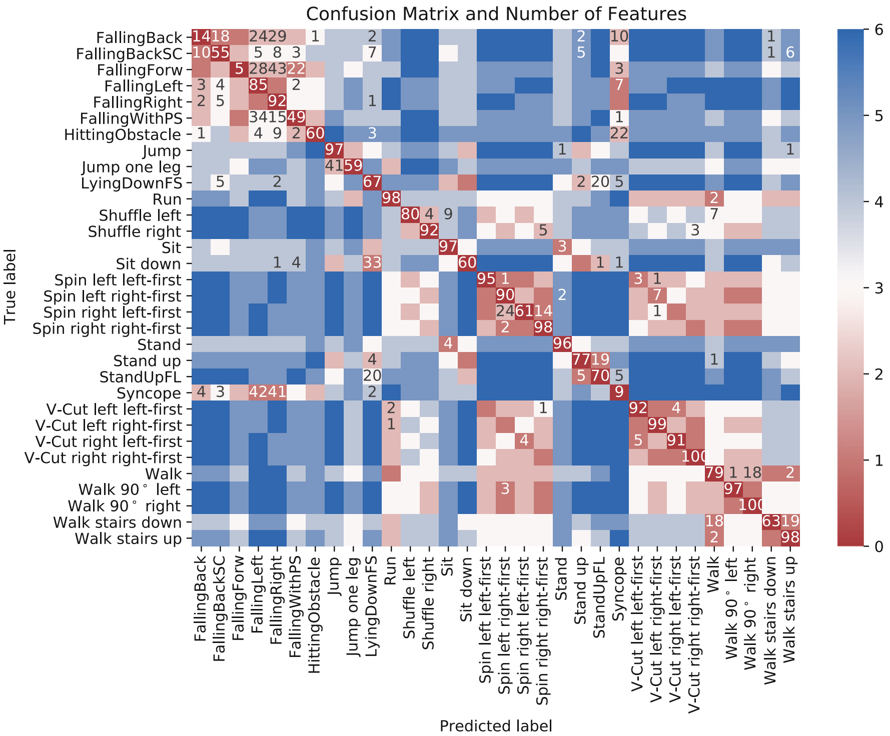


**Fig. 13.** Blended confusion and distance matrix for the Encoding-based classifier on the combined CSL-SHARE and UniMiB SHAR dataset. The color indicates distance, and values indicate the percentage of cases.

might be present in both train and test set, resulting in person-dependent subtleties to be picked up and included in the HLF extraction. However, it is also noteworthy that shuffle left/right, lying down from sitting, and syncope perform significantly worse. This might be due to hyperparameter settings being optimized for independence but requires further investigation. Nevertheless, some activities' misclassifications remain. Especially falling left/right and syncope remain sink states, and lying down from sitting is often confused with standing up from lying. Future work should involve both optimization and analysis of person-dependent HLF extraction and a person-independent dataset combination evaluation. For instance, by pairing participants across datasets to ensure activities in the test set are present in training prior.

This dataset combination demonstrates that combining datasets with this method is possible, which creates a foundation to find commonalities and suited features for activities across different settings.

## 8 Future Work

The results from the HLF extraction and the different classification experiments, on each dataset separately, imbalanced and few-shot, as well as on both datasets combined, are very encouraging and show the feasibility and possibilities with these HLFs and within this feature space. Multiple points should be addressed to improve these results, catch up to state-of-the-art performances and develop the HLF space further.

The proposed HLF feature space has more than eleven thousand unique combinations. Not all of those have to be possible, but investigating this further and naming more of them will further enhance the understanding of activities and HLFs. High-level features will be developed by borrowing from previous HAR work [27], sports knowledge [34], and even utilizing findings and criteria from dance [12] from decades of previous work. Additionally, extending the HLF to work on other datasets with different activities should ensure more robust activity definitions as well as more versatile HLFs.

Another major part for future work is developing these HLFs to return sequences of probabilities rather than compressing a given sequence into a single vector. HLF sequences would have multiple significant advantages, including online recognition and removing the need for edge cases mentioned above. Take jumping, which currently in the *Up/Down* HLF has the value *Updown*, and could then simply be first up and then down. Not only a more sensible way of modeling but also more expressive and precise over time. Similarly, the *Knee* HLF values used to indicate the starting foot would benefit from more precise time information, as this could be transformed to model which knee is in front as the more general case of the starting foot.

There are multiple challenges with extracting HLFs as sequences. The main challenges include learning the sensor data to HLF relationship, different sampling rates across datasets, and reconsidering the uniqueness property and categorical values requirement. Addressing the learning challenge, one approach could be to change the ground truth labeling from being a single value per sequence to a sequence itself. However, this labeling is a lot more time-consuming, and in cases of data recordings without videos might not even be possible. Another approach could be to develop a classifier that implicitly learns the sequence to value relationship. LSTM hidden states come to mind, but it might be hard to ensure interpretability, otherwise, the need for HLF definitions is void. The different sampling rates mainly play a role in comparison across datasets and could be addressed by re-sampling or interpolation. Classification with sequence models is not affected as much, as models like HMMs do model order over time and are not affected by different speeds of HLF changes. Furthermore, the slow nature of HLFs might counteract the sampling rates. The uniqueness property might be addressed by condensing the sequence HLFs by removing duplicates or directly calculating the Levenshtein distance, as long as the HLFs remain categorical. Extracting sequences of HLFs rather than vectors promises more expressive and precise activity definitions, and its implications and modeling requirements will be further investigated.

The HLFs might benefit from supporting ordered or even numerical HLFs in addition to categorical ones. Take the *Left/Right* HLF, which has an ordering and intensity (Extreme Left, Left, Slight Left), which a classifier might explicitly learn. Implicitly this is likely already learned via the returned probabilities. If the classifier cannot surely distinguish between *Extreme Left* and *Left*, both of these probabilities will be high, indicating the true value to be somewhere in between. Nevertheless, explicitly modeling this might further enhance the HLFs. In the same wake, it might make sense to reconsider the *N.a.* and *Any* values present in multiple HLFs.

Further investigating the learned HLF extractors and how they arrive at their predictions is another major point. One of the substantial advantages of the proposed HLFs and their extraction mechanism is the fast and knowledge-based target assignment. The main assumption is that extractors can utilize the extra information by extracting the shared commonality across different activity labels. Some classifiers might be better at this than others, and there is no restriction enforcing this currently. The classifiers might subspace the sensor data by classifying the activity and grouping, say walk and walk left. This does not change the usefulness of combing the datasets via the HLF space, as knowledge engineering still provides multiple benefits. However, the benefit will be greatly improved, especially in the few-shot and imbalanced classification experiments, the more the classifiers can extract the specific attribute.

These and other topics, including estimating a performance ceiling with neural networks and extending to further datasets and modalities, are future work for these high-level features.

## 9 Conclusion

High-Level Features (HLFs) are a novel way of describing human activities via unique combinations of high-level human interpretable features. The nature of HLFs being defined based on activities rather than sensor data makes the extracted features comparable and combinable across datasets with different sensor setups.

In this article, we further elaborated and extended our initial HLFs and proposed six HLFs, *Back/Front*, *Left/Right*, *Up/Down*, *Force*, *Knee*, and *Impact*. We then demonstrated how to extract these with BRFs and HMMs and showed that classification in this feature space works well in standard settings, as well as imbalanced, few-shot settings and across datasets. The feature extraction worked very well with accuracies of at least

92% (CSL-SHARE) and 68% accuracy (UniMiB SHAR) and up to 99% (CSL-SHARE) and 94% accuracy (UniMiB SHAR) in a 5-fold person-independent cross-validation. The classification experiments showed that classification in the HLF-space works very well, with 89.7% (CSL-SHARE) and 67.3% (UniMiB SHAR) accuracy. Although currently behind state-of-the-art (93.7% CSL-SHARE, 77.0% UniMiB SHAR), the results are promising and stand out due to the possibility of understanding why an error was made and to attribute it to a combination of HLF extractors. The imbalanced and few-shot learning experiments showed how sharing data between activities with multiple HLF extractors supports learning with little or imbalanced data. In the case of imbalanced data, improvements can likely be made by further investigating the feature extractors and how they partition the low-level feature space in these cases. At the same time, few-shot learning converged very quickly on the CSL-SHARE dataset and more slowly on the UniMiB SHAR dataset, only requiring 10% (CSL-SHARE) of available data to perform on par with the full data for training. Both experiments indicate that further evaluation of preprocessing techniques and classifiers to only pay attention to the correct portions of the sensor data for this particular HLF should further enhance HLF extraction. The experiment on dataset combination as well as the analysis of activity distances in the HLF-space demonstrated the usefulness for cross-sensor-setup comparison, classification, and HLF design.

The next steps are clear: extend to more datasets and reiterate the activities' definition by adjusting and reconsidering the HLF value assignments. Furthermore, neural networks and further low-level feature processing will need to be evaluated, and the actual learnings of the HLF extractors will be investigated to improve the scalability of these HLFs further. At the same time, we want to extend the HLFs to sequence modeling to enable more precise activity descriptions. All of this is to improve performance and expressive power while not losing the option for non-ML experts to understand the extracted high-level features.

**Acknowledgements.** Many thanks to Steffen Lahrberg for his contributions, ideas, and handcrafted feature functions in the initial paper on High-Level Features [15]. Furthermore, a large thank you to all toolboxes: SciPy [49], NumPy [14], scikit-learn [39], Matplotlib [20], Seaborn [51], TSFEL [4], and our in-house decoder BioKIT [46].

## References

1. Amma, C., Gehrig, D., Schultz, T.: Airwriting recognition using wearable motion sensors. In: First Augmented Human International Conference, p. 10. ACM (2010)
2. Arifoglu, D., Bouchachia, A.: Activity recognition and abnormal behaviour detection with recurrent neural networks. Procedia Comput. Sci. **110**, 86–93 (2017)
3. Bakis, R.: Continuous speech recognition via centisecond acoustic states. J. Acoust. Soc. Am. **59**(S1), S97–S97 (1976)
4. Barandas, M., et al.: TSFEL: time series feature extraction library. SoftwareX **11**, 100456 (2020)
5. Bian, S., Liu, M., Zhou, B., Lukowicz, P.: The state-of-the-art sensing techniques in human activity recognition: a survey. Sensors **22**(12), 4596 (2022)
6. Bragança, H., Colonna, J.G., Oliveira, H.A.B.F., Souto, E.: How validation methodology influences human activity recognition mobile systems. Sensors **22**(6), 2360 (2022)

7. Bulling, A., Blanke, U., Schiele, B.: A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput. Surv. (CSUR) **46**(3), 1–33 (2014)
8. Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Technical report (2004)
9. Deng, Z., Vahdat, A., Hu, H., Mori, G.: Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4772–4781 (2016)
10. Dickinson, S.J., Leonardis, A., Schiele, B., Tarr, M.J.: Object Categorization: Computer and Human Vision Perspectives. Cambridge University Press, Cambridge (2009)
11. Ding, X., Hu, C., Xie, W., Zhong, Y., Yang, J., Jiang, T.: Device-free multi-location human activity recognition using deep complex network. Sensors **22**(16), 6178 (2022)
12. Guest, A.H.: Labanotation: Or, Kinetography Laban : the System of Analyzing and Recording Movement, no. 27. Taylor & Francis (1977). http://books.google.com/books?id=Tq1YRDuJnvYC&pgis=1
13. Ha, S., Yun, J.M., Choi, S.: Multi-modal convolutional neural networks for activity recognition. In: SMC 2015 - IEEE International Conference on Systems, Man, and Cybernetics, pp. 3017–3022. IEEE (2015)
14. Harris, C.R., et al.: Array programming with NumPy. Nature **585**(7825), 357–362 (2020)
15. Hartmann, Y., Liu, H., Lahrberg, S., Schultz, T.: Interpretable high-level features for human activity recognition. In: Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies, pp. 40–49. SCITEPRESS - Science and Technology Publications (2022)
16. Hartmann, Y., Liu, H., Schultz, T.: Feature space reduction for multimodal human activity recognition. In: Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4: BIOSIGNALS, pp. 135–140. INSTICC, SciTePress (2020)
17. Hartmann, Y., Liu, H., Schultz, T.: Feature space reduction for human activity recognition based on multi-channel biosignals. In: Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies, pp. 215–222. INSTICC, SciTePress (2021)
18. Hartmann, Y., Liu, H., Schultz, T.: Interactive and interpretable online human activity recognition. In: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), pp. 109–111. IEEE, Pisa (2022)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR 2016 - IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
20. Hunter, J.D.: Matplotlib: a 2D graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007). http://ieeexplore.ieee.org/document/4160265/
21. Inoue, M., Inoue, S., Nishida, T.: Deep recurrent neural network for mobile human activity recognition with high throughput. Artif. Life Robot. **23**(2), 173–185 (2018)
22. Keshavarzian, A., Sharifian, S., Seyedin, S.: Modified deep residual network architecture deployed on serverless framework of IoT platform based on human activity recognition application. Futur. Gener. Comput. Syst. **101**, 14–28 (2019)
23. Kwon, Y., Kang, K., Bae, C.: Analysis and evaluation of smartphone-based human activity recognition using a neural network approach. In: IJCNN 2015 - International Joint Conference on Neural Networks, pp. 1–5. IEEE (2015)
24. Lee, S.M., Yoon, S.M., Cho, H.: Human activity recognition from accelerometer data using convolutional neural network. In: BIGCOMP 2017 - IEEE International Conference on Big Data and Smart Computing, pp. 131–134. IEEE (2017)
25. Liu, H.: Biosignal processing and activity modeling for multimodal human activity recognition. Ph.D. thesis, University of Bremen (2021)

26. Liu, H., Hartmann, Y., Schultz, T.: CSL-SHARE: a multimodal wearable sensor-based human activity dataset. Front. Comput. Sci. (2021)
27. Liu, H., Hartmann, Y., Schultz, T.: Motion units: generalized sequence modeling of human activities for sensor-based activity recognition. In: EUSIPCO 2021–29th European Signal Processing Conference. IEEE (2021)
28. Liu, H., Hartmann, Y., Schultz, T.: A practical wearable sensor-based human activity recognition research pipeline. In: Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 5: HEALTHINF, pp. 847–856 (2022)
29. Liu, H., Schultz, T.: ASK: a framework for data acquisition and activity recognition. In: Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 3: BIOSIGNALS, pp. 262–268. INSTICC, SciTePress (2018)
30. Liu, H., Schultz, T.: A wearable real-time human activity recognition system using biosensors integrated into a knee bandage. In: Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 1: BIODEVICES, pp. 47–55. INSTICC, SciTePress (2019)
31. Liu, H., Schultz, T.: How long are various types of daily activities? Statistical analysis of a multimodal wearable sensor-based human activity dataset. In: Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 5: HEALTHINF, pp. 680–688 (2022)
32. Long, J., Sun, W., Yang, Z., Raymond, O.I.: Asymmetric residual neural network for accurate human activity recognition. Information **10**(6), 203 (2019)
33. Lukowicz, P., et al.: Recognizing workshop activity using body worn microphones and accelerometers. In: Ferscha, A., Mattern, F. (eds.) Pervasive 2004. LNCS, vol. 3001, pp. 18–32. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24646-6_2
34. Meinel, K., Schnabel, G.: Bewegungslehre - Sportmotorik: Abriß einer Theorie der sportlichen Motorik unter pädagogischem Aspekt. Meyer & Meyer Verlag, Aachen, 12, ergänzte auflage edn. (1987). https://suche.suub.uni-bremen.de/peid=B80288025
35. Micucci, D., Mobilio, M., Napoletano, P.: UniMiB SHAR: a dataset for human activity recognition using acceleration data from smartphones. Appl. Sci. **7**(10), 1101 (2017)
36. Murad, A., Pyun, J.Y.: Deep recurrent neural networks for human activity recognition. Sensors **17**(11), 2556 (2017)
37. Oniga, S., Sütő, J.: Human activity recognition using neural networks. In: Proceedings of the 15th International Carpathian Control Conference, pp. 403–406. IEEE (2014)
38. Ordóñez, F.J., Roggen, D.: Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors **16**(1), 115 (2016)
39. Pedregosa, F., et al.: Scikit-learn: machine Learning in Python. J. Mach. Learn. Res. **12**(85), 2825–2830 (2011). http://jmlr.org/papers/v12/pedregosa11a.html
40. Ronao, C.A., Cho, S.B.: Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models. In: ICNC 2014–10th International Conference on Natural Computation, pp. 681–686. IEEE (2014)
41. Ronao, C.A., Cho, S.B.: Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst. Appl. **59**, 235–244 (2016)
42. Ronaoo, C.A., Cho, S.B.: Evaluation of deep convolutional neural network architectures for human activity recognition with smartphone sensors. J. Korean Inf. Sci. Soc. 858–860 (2015)
43. Scheirer, W.J., Anthony, S.E., Nakayama, K., Cox, D.D.: Perceptual annotation: measuring human vision to improve computer vision. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1679–1686 (2014)
44. Singh, D., Merdivan, E., Psychoula, I., Kropf, J., Hanke, S., Geist, M., Holzinger, A.: Human activity recognition using recurrent neural networks. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-MAKE 2017. LNCS, vol. 10410, pp. 267–274. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66808-6_18

45. Straczkiewicz, M., James, P., Onnela, J.P.: A systematic review of smartphone-based human activity recognition methods for health research. NPJ Digit. Med. **4**(1), 148 (2021)
46. Telaar, D., et al.: BioKIT - Real-time decoder for biosignal processing. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 2650–2654 (2014)
47. Tuncer, T., Ertam, F., Dogan, S., Aydemir, E., Pławiak, P.: Ensemble residual network-based gender and activity recognition method with signals. J. Supercomput. **76**(3), 2119–2138 (2020). https://doi.org/10.1007/s11227-020-03205-1
48. Uddin, M.Z., Thang, N.D., Kim, J.T., Kim, T.S.: Human activity recognition using body joint-angle features and hidden Markov model. ETRI J. **33**(4), 569–579 (2011)
49. Virtanen, P., et al.: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods **17**(3), 261–272 (2020)
50. Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L.: Deep learning for sensor-based activity recognition: a survey. Pattern Recogn. Lett. **119**, 3–11 (2019)
51. Waskom, M.L.: seaborn: statistical data visualization. J. Open Sour. Softw. **6**(60), 3021 (2021)
52. Yang, J., Nguyen, M.N., San, P.P., Li, X., Krishnaswamy, S.: Deep convolutional neural networks on multichannel time series for human activity recognition. In: IJCAI, vol. 15, pp. 3995–4001. Buenos Aires, Argentina (2015)
53. Yang, S.H., Baek, D.G., Thapa, K.: Semi-supervised adversarial learning using LSTM for human activity recognition. Sensors **22**(13), 4755 (2022)
54. Youngblood, G.M., Cook, D.J.: Data mining for hierarchical model creation. IEEE Trans. Syst. Man Cybern. Part C (Appl. and Rev.) **37**(4), 561–572 (2007)
55. Zeng, M., et al.: Convolutional neural networks for human activity recognition using mobile sensors. In: MOBICASE 2014–6th International Conference on Mobile Computing, Applications and Services, pp. 197–205. IEEE (2014)