Inertial Measurement Unit based Human Action Recognition for Soft-Robotic Exoskeleton

Jan Kuschan, Moritz Burgdorff, Hristo Filaretov, Jörg Krüger
Fraunhofer Institute for Production Systems and Design Technology IPK, Germany
E-mail: jan.kuschan@ipk.fraunhofer.de

Abstract. Absence from work caused by overloading the musculoskeletal system lowers the life quality of the worker and gains unnecessary costs for both the employer and the health system. Exoskeletons can present a solution. Typically, such systems struggle with stiffness and discomfort and primarily a lack of battery lifetime. Soft-robotic exoskeletons offer a possibility to overcome these problems by increasing the system flexibility, not limiting the supported DoF and being actuator and joint together. Since soft-robotic exoskeletons can be designed only using power when supporting the wearer, it is possible to increase the battery lifetime by only acting on those actions for which the wearer needs support. Dealing with controls for soft-robotic exoskeleton one major difficulty is to find a compromise between saving energy and supporting the wearer. Having an action depending control can reduce the supported actions to cover only relevant ones and increase the lifetime of the battery. The system conditions are to detect the user actions in real-time and distinguish between actions which require support and those which do not. We contribute an analysis and modification of human action recognition (HAR) benchmark algorithms from activities of the daily living, transferred them onto industrial use cases containing short and mid-term action and reduce the models to be compatible using embedded computers for real-time recognition on soft exoskeletons. We identified the most common challenges for inertial measurement units based HAR and compare the best-performing algorithms using a newly recorded data set overhead car assembly for industrial relevance. As a benchmark data set we focused on the "Opportunity" data set. By introducing orientation estimation, we were able to increase the F1 scores by up to 0.04. With an overall F1 score without a Null-class of up to 0.883, we were able to lay the foundation to use HAR for action dependent force support.

1. Introduction
Physically demanding tasks in manufacturing, logistics, handcraft and service are vital contributors to early damage of the musculo-skeletal system and especially the spine [11]. This stress leads to a reduced quality of life and also to a decreased capacity for work [10, 16]. Exoskeletons can present a solution. Typically, such systems struggle with stiffness and discomfort and primarily a lack of battery lifetime [9, 7]. Soft-robotic exoskeletons offer a possibility to overcome these problems by increasing the system flexibility, not limiting the supported Degree of Freedom (DoF) and being actuator and joint together [8, 20]. Since soft-robotic exoskeletons can be designed only using power when supporting the wearer, it is possible to increase the battery lifetime by only acting on those actions for which the wearer needs support. Human Activity Recognition (HAR) can be used to predict movements and support every individually classified action, but can also be constrained to only support task-
related actions. The combination of Inertial Measurement Unit (IMU) based HAR and soft exoskeletons are therefore predestined to create an action-based prediction of the current or future activities as early and accurate as possible.

The idea behind HAR is that characteristic sensor signals directly correspond to specific body movements [18] which can, therefore, be detected and classified from a time series of sensor data. Traditional approaches to HAR rely on hand-crafted or heuristic information, where expert knowledge is used to identify relevant features. This method is highly restrictive and leads to difficulties with recognizing high-level behaviours, as engineered features are only “convenient mathematical operations” and “do not relate to units of behaviour” [18]. An additional problem is the general transfer of this explicit knowledge to different application domains. More recent methods however make use of machine learning with earlier works using techniques such as Deep Belief Networks (DBNs) ([19]) or Hidden Markov Models (HMMs) above Restricted Boltzmann Machines (RBM) layers ([1]). While combining hand-crafted feature approaches with machine learning methods can lead to systems that perform well in certain scenarios ([6]), machine learning and more specifically deep learning methods still offer major advantages, such as more robustness to a large variety of actions and different people.

2. Related Work

In [14] an active textile upper arm, elbow and hand exoskeleton for workers support was presented. Kuschan et al. described how an angular control could be used to achieve a gravity compensation of the arm. Since this supports mainly static arm poses but due to the stiffness of the system also interferes with dynamic movements of the user, it is necessary to detect current and future movements. The system consists of five IMUs — one on the chest, two on the upper arm, one on the lower limb and one on the hand.

2.1. Datasets

While working with wearable systems like soft-robotic exoskeletons, it makes sense to use the existing sensors like IMUs to avoid the typical problems of working with vision-based HAR. Even if sensor-based HAR is not as common as video or image-based HAR, a large variety of datasets are available. These datasets are often specialized in one specific topic like recognizing sports activities [23], elderly care [2] or daily activities [15]. Chavarriaga et al. [5] created a challenge for HAR based on the publicly available Opportunity dataset. While Activity of Daily Living (ADL) often consists of long- (e.g. relaxing, running), mid- (e.g. cooking, washing dishes) and short-term (e.g. open doors) actions, in industrial use-cases we are often confronted with repetitive processes of mid- (e.g. mounting something) and short-term (e.g. picking up something) actions.

**Opportunity Dataset** Due to its vast documentation, the Opportunity dataset is a very common benchmark for HAR. It consist of $n_c = 18$ numbers of sporadic gesture classes, with the Null class occupying 72%. The activities from the Opportunity dataset were recorded in a home environment and comprised gestures performed during everyday activities. The recordings include four subjects each of which performed five ADL sessions, during which they only followed a high-level description of the task and therefore had free interpretation of how to achieve the goal rather than step-by-step instructions. Every subject also performed a drill session, during which they executed 20 repetitions of a sequence of nine activities. The dataset was recorded at a sample rate of 30 Hz and used a large number of sensors of different modalities that were shared between the environment, the objects and the subjects and is around 6h long. The subjects’ body-worn sensors included 7 IMUs and 12 3D acceleration sensors with a total number of relevant on-body sensor channels of $D_t = 133$. The IMUs provide the 3D acceleration, 3D angular velocity, 3D magnetic field and the sensor orientation in quaternions.
2.2. Classifier
In the domain of HAR, many different machine learning techniques have already been successfully employed. Choices range from shallow models, such as k-Nearest Neighbors (kNN) ([13]), Decision Trees ([3]) or Joint boosting ([4]) paired with automatic or hand-crafted feature extraction, to deep learning models, such as Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), autoencoders or RBM. For this work, the models were selected based on their reported classification performance on the aforementioned “Opportunity” dataset and the perceived complexity of their structure. Three models were selected and will hereinafter be referred to by the names based on their respective publications: Deep Convolutional Long Short-Term Memory (DC-LSTM) [18], Cascaded Bidirectional and Unidirectional LSTM-based Deep Recurrent Neural Network (C-DRNN) [21] and CNN-2 [17].

DC-LSTM, as well as CNN-2, make use of multiple convolutional layers for feature extraction, but neither model uses padding, and as such, they produce increasingly smaller feature maps with every consecutive layer. Therefore, both models can only be applied if the length of the sliding window is above a certain threshold, which is dependent on the amount of convolutional and pooling layers.

C-DRNN The Cascaded Bidirectional and Unidirectional Long Short-Term Memory (LSTM)-based Deep Recurrent Neural Network (DRNN) from [17] is a cascaded structure, in which the first layer is a bidirectional LSTM layer, the output of which gets concatenated with a simple summation and is then followed by unidirectional LSTM layers (where bidirectional means that the layer consists of two LSTMs, one of which processes the data backwards). Even though the reported F1 score indicates state-of-the-art performance, several observations about the methodology have to be considered for its interpretation. First, all datasets were divided with a simple 80/20 split between training and test data with no validation set. This proceeding did not adhere to the Opportunity challenge guidelines and as such did their test data optimizations as opposed to on a validation set. Next, there is no description of how multiple activity classes during one window were handled. And lastly, they present no detailed description of their final model structure, and only a few hyperparameters are mentioned.

| # | Type                   | Layer Size | Parameters       | Opportunity D = 113, n_c = 18 | Overhead Car Assembly (OCA) D = 72, n_c = 7 |
|---|-----------------------|------------|------------------|--------------------------------|------------------------------------------|
| 1 | Bidirectional LSTM    | 64 units   |                  | 91 648                         | 70 656                                   |
| 2-3| LSTM                  | 64 units   |                  | 33 280                         | 33 280                                   |
| 4  | Fully Connected       | n_c units  | (Softmax)        | 1 170                          | 455                                      |
|    |                       |            |                  | Total 159 378                  | 137 671                                  |
|    |                       |            |                  | Total (Relevant Layers) 158 208 | 137 216                                  |

Table 1: Layer configuration and number of parameters per layer in C-DRNN. Number of parameters can vary with the framework used for the implementation. D is the number of features, n_c is the number of classes. Relevant layers are italicised.

DC-LSTM DC-LSTM [18] is an architecture that combines multiple convolutional and recurrent layers. It has proven state-of-the-art performance in the related domain of speech recognition.

The model was tested on the Opportunity and Skoda datasets [22] and delivers state-of-the-art performance scores on both of them. Regarding the Opportunity dataset, they used all on-body sensors and data splits as outlined in the Opportunity challenge guidelines.
The model was imported from their publicly available repository without any major changes. It consists of four stacked convolutional layers with $64 \times 5 \times 1$ filters, each of which operates along the time axis. Subsequently, the output is fed into two stacked LSTM layers with 128 units each and one softmax layer for classification, shown in table 2. For training, the publication proposes a learning rate of $10^{-3}$, which, however, in their implementation is changed to $10^{-4}$. The training was therefore performed with the RMSprop optimiser with learning rate of $10^{-4}$, decay factor of $\rho = 0.9$ and a dropout probability of $p = 0.5$. Another inconsistency between their publication and implementation is the difference in the window length. The paper proposes a window length of 500 ms, for which they claim to have obtained the best test results, whereas the implementation uses a window length of 800 ms, which is used by the model that produces the reported $F_1$ score.

**CNN-2** The CNN-2 from [21] can be seen as two consecutive blocks of two convolutional layers followed by one maxpooling layer, which results in two fully connected and one softmax layer for classification.

A detailed listing of the model architecture can be found in table 3. It was first proposed in [18] as the baseline model and then refined with the use of pooling layers by [21], although it still only served as a baseline model, it achieved state-of-the-art performance on the Opportunity dataset with $F_1 = 0.9152$. The tests adhere to the Opportunity guidelines. It was trained with the RMSprop optimiser with a fixed learning rate of $10^{-3}$, a decay factor $\rho = 0.95$ and a dropout probability of $p = 0.5$.

3. Experiments

Overhead work is a typical problem in ergonomic analysis. For this dataset we present in the Experiments section different modifications of the benchmark algorithms we worked out from ADL datasets. We also introduce orientation estimation for IMU based HAR and other varieties of the input data.

### 3.1. OCA

To have a dataset that meets our requirements, we have recorded the OCA dataset in laboratory environments. OCA covers the mounting and dismounting of a heat capsule under a car, a cyclic process which needs approximately 75 s. We separated the cycle in six mid level classes: “Mount
| #  | Type                | Filters @ Patch Size | Parameters | Opportunity | OCA       |
|----|---------------------|----------------------|------------|-------------|-----------|
| 1  | Convolution         | 64 @ 5 × 1           | 384        | 384         |           |
| 2  | Convolution         | 64 @ 5 × 1           | 20 544     | 20 544      |           |
| 3  | MaxPooling          | 2 × 1                |            |             |           |
| 4-5| Convolution         | 64 @ 5 × 1           | 20 544     | 20 544      |           |
| 6  | MaxPooling          | 2 × 1                |            |             |           |
| 7  | Fully Connected     | 512 units            | 22 217 216 | 33 030 656 |           |
| 8  | Fully Connected     | 256 units            | 131 328    | 131 328     |           |
| 9  | Fully Connected     | n_c units            | 4 626      | 1 799       |           |

Total: 22 415 186, 33 225 799

Total (Relevant Layers): 62 016

Table 3: Layer configuration and number of parameters per layer in CNN-2. Relevant layers are italicised.

| Mount Panel | Cover Panel | Screw in | Cover Panel | Take Screwdriver | Take Cover Panel Down | Unscrew Cover Panel | Place Screwdriver Down |
|-------------|-------------|----------|-------------|-------------------|-----------------------|----------------------|------------------------|
| Worker takes the cover panel and loosely mounts it. | Worker screws in all screws for the cover panel. | Worker picks the screwdriver up from the table. | Worker takes down the cover panel and puts it down. | Worker unscrews all screws that hold the cover panel in place. | Worker places his screwdriver on a table. |

Table 4: High level description of the custom data set.

Cover Panel”, “Take Panel Down”, “Take Screwdriver”, “Place Screwdriver”, “Screw in Panel” and “Unscrew Panel”, shown in table 5. Without line production in the lab it was necessary to dismount the heat panel after mounting it and since this is not part of the original process we decided to handle it as independent classes.

The choice of the actions is related to the wearable robotics use-case. OCA contains very dynamic short-term overhead movements, like mounting cover panel and take cover panel down, without the need of support, but also very static overhead actions like screwing in and unscrewing the panel, which requires substantial support. The better the classification rate of the static actions, the better the wearer is supported and the better the classification rate of the dynamic short-term actions the less energy is wasted.

For the recording process subjects were equipped with a sensor vest devised like in [24]. This vest consists of 12 IMUs, covering all major body segments (figure 1).

All IMUs provide 3D acceleration, 3D angular velocity and the sensor’s orientation in
Inertial Measurement Unit placed on back

Figure 1: Front view; Sensor placement on the vest for the custom dataset. The numbers indicate the index of the sensor in the dataset. [24]

quaternions for a total number of $D_t = 120$ channels. The dataset was recorded at a constant sample rate of 40 Hz, is around 90 min long and covers 70 cycles. All sessions were video-recorded and then retroactively labelled with a synchronization tool. Finally, we had five test subjects where person 1, 2 and 3 each recorded two independent datasets, each approximately 10 minutes long, and person 4 and 5 each recorded one dataset, about 15 minutes each.

The short duration actions like “Take Panel Down”, “Take Screwdriver” and “Place Screwdriver” comprise a tiny part of the total dataset, while the Null class is the most dominant with 42%. Still, this is much less than the 71.9% Null class presented in the Opportunity dataset. Since “Screw in Panel” and “Unscrew Panel” are very similar actions, we discuss their influence on the model if they are kept distinct and if they are combined into one class.

3.2. Training
The three models [18, 21, 17] were chosen because they have the best benchmark performance on the Opportunity dataset and also provide available code. Only the architecture of the models was used, all weights are initialized prior to training and affected only by our own training procedure.

For use in this work a few variations on the original C-DRNN have been made. The authors of [17] calculate a prediction for every sample $y^t_c$, where $t = 1, \ldots, T$, and uses a merging step to obtain a class distribution for the entire input window. The prediction of the last sample
was replaced similar to the final step in DC-LSTM to predict the most recent movement. Additionally, the handling of multiple classes in one window was changed, and the data split from the Opportunity challenge was used.

The final model is presented in table 1 and consists of three layers, where the first layer is the bidirectional LSTM layer, which is followed by two unidirectional LSTM layers and one softmax layer for classification. All LSTM layers have 64 units and use the tanh activation function. It was trained with the Adam optimiser [12], with a learning rate of $10e^{-3}$ and a dropout probability of $p = 0.2$.

For training and inference, all models receive the same input structure, a series of $T \times D$ matrices. Likewise, they all produce the same output, which is a probability distribution $y_c = (y_1, \ldots, y_{nc})$ over all $nc$ classes, with $\sum_i y_i = 1$ and $y_i \in [0, 1]$, for every one of those matrices.

In table 6, we present the variations of the input data, which we also use for the Result and Discussion section. The first models were trained only using the acceleration and angular velocity raw data captured by the IMUs. For the subsequent variations, we added an orientation estimation calculated according to

$$q_t = q_i \otimes q_{[x|y|z]} \otimes q^*_i$$

$$\alpha_{i,[x|y|z]} = \text{acos}(q_{t,z})$$

where $q_{[x|y|z]} = [0, 1, 0, 0]^T_{[x]}, [0, 0, 1, 0]^T_{[y]}, [0, 0, 0, 1]^T_{[z]}$ and $q_i$ is the orientation from the acceleration and angular velocity of IMU $i$.

For the third variation of the data, we removed the acceleration, because the orientation and angular velocity are enough to determine the configuration of a kinematic chain (in this case, the human body). Finally, we also removed the angular velocity because it is merely the first-order derivative of the orientation.

For the training, we split the dataset into training, test and validation parts without the leave one out method, following the rules of the Opportunity dataset [5]. But in the later results, we also discuss the influence of isolating one test subject as designated test data.

| Name             | Repetitions | Instances | (%)  |
|------------------|-------------|-----------|------|
| Null             | 791         | 5 749     | 42.0 |
| Mount Panel      | 68          | 1 188     | 8.7  |
| Take Panel Down  | 67          | 535       | 3.9  |
| Take Screwdriver | 130         | 483       | 3.5  |
| Place Screwdriver| 127         | 497       | 3.6  |
| Screw in Panel   | 199         | 2 593     | 18.9 |
| Unscrew Panel    | 200         | 2 640     | 19.3 |
| **Total**        | **13685**   |           |      |

Table 5: Class labels and their distributions for the OCA dataset. Repetitions and instances show the distribution of the individual classes (across all subjects and sessions) after application of the sliding window.
3.3. Performance
In the domain of HAR there are multiple performance measures with individual strengths, such as accuracy, precision, recall, F scores or Receiver Operating Characteristics (ROC) curves. The $F_1$ score belongs to the most commonly used metrics, as it provides a simple yet expressive value that can be used for easy comparison between multiple approaches. More in-depth analysis of a given model can be achieved through the use of confusion matrices, as they offer the ability to evaluate the classification outcome on a per-class level. Furthermore, it can also be employed as a similarity measure between different activities in a dataset, where more misclassifications between two classes can be interpreted as low interclass variability. However, the $F_1$ score suffers in expressiveness when datasets are heavily unbalanced because trivial classifiers that only predict the most prevalent class can achieve high scores. While confusion matrices can be normalized to handle this effect, the $F_1$ score has to be extended to the weighted $F_1$ score, which takes the prevalence of each class into account.

$$wF_1 = \sum_i 2w_i \frac{p_i \cdot r_i}{p_i + r_i}$$  \hspace{1cm} (2)

where $w_i = n_i/N$ is the proportion of number of instances of class $i$ ($n_i$) to the total number of instances ($N$), and $p_i$ and $r_i$ are the recall and precision per class respectively, with $p_i = \frac{TP_i}{TP_i + FP}$ and $r_i = \frac{TP_i}{TP_i + FN}$. For the interpretation of the results, these two metrics will be used.

3.4. Tests
The tests for this work were conducted in two parts. First, the models were trained and tested on the input variations 1–4 (table 6) on both the Opportunity and OCA datasets with the default architectures and hyperparameter settings. Those results serve as a baseline for both classification accuracy and inference time of the original models and explore whether the introduction of model-based features to the input of the unchanged model increases model classification performance. Secondly, all permutations of layer settings for the models were trained on the Opportunity dataset with the default input. Next, the best model was chosen according to the following formula:

$$\arg \max_{m \in \mathcal{M}} \left( |f_t(m; \mathcal{M}) - f_{val\_acc}(m)| \cdot r(m) \right)$$  \hspace{1cm} (3)

where $\mathcal{M}$ is the set of all tested architectures, $f_t$ is a logarithmic trend line for the given set of validation accuracy over the present reduction in model complexity, $f_{val\_acc}$ is the validation accuracy.

| Variant | Features added to the data | Features removed from the data |
|---------|---------------------------|--------------------------------|
| 1       | –                         | –                              |
| 2       | orientation estimation    | –                              |
| 3       | orientation estimation    | acceleration channels          |
| 4       | orientation estimation    | acceleration and angular velocity channels |

Table 6: Variants of the input data. Orientation features were calculated as described in equation 1.
accuracy of the given model architecture and \( r \) is the average complexity reduction in \% of the model \( m \) compared to the original model. The logarithmic trend line was chosen to have the form \( f_t = -a \cdot e^{b \cdot x} + c \), because we expect the models to quickly rise in performance with higher model complexity but level out. Afterwards, the selected model was trained and tested on input variations 1–4 on both the Opportunity and OCA datasets.

4. Results

Table 7 shows the weighted \( F_1 \) scores (equation 2) for the three used networks of both the Opportunity dataset and the OCA dataset. Further, the different varieties of input data are presented, and the influence of reducing the model complexity is shown. Especially for the Opportunity dataset, the weighted \( F_1 \) score is significantly higher than the one without Null class. Since a high amount of samples has a direct influence on the weighted \( F_1 \) score, the difference to the OCA should relate to the class distribution with 79.2\% for Opportunity and 42\% for OCA. The \( F_1 \) scores without Null class performs better for both the default and reduced models of the C-DRNN and reduced models of DC-LSTM and CNN-2 better, than for the Opportunity data set. Interestingly, no distinct pattern for the different input data varieties is observable. The best classifiers seem to be either of variant two (acceleration, angular velocity and orientation estimation) or variant one (acceleration and angular velocity). The Opportunity dataset has a better classification of variant two for the default models and variant one for the reduced models. We assume that this is due to the ratio between reduced layers and an increased amount of input data. As a result, the networks are not able to find a suitable solution for the higher amount of data. Contrary to the expectation that the orientation estimation combined with angular velocity contains all information and the acceleration is not needed, there is a drop in the performance for most of the datasets and models using variant three. Still, there is a lower decrease in performance for the reduced models. Using only the orientation estimation as input data for the models results in the worst classification results.

The confusion matrices for the models trained on the OCA dataset were chosen based on the best \( F_1 \) presented in table 7. Having a look at the confusion matrices for the different models of the OCA dataset in figures 4, 2 and 3 one can see that all models suffer in distinguishing between the “screw in” and “unscrew cover panel” classes. At the current state, it appears more like a random decision between those two actions. The recognition error between the remaining classes is very low and mostly only the Null class has a bigger influence on the false detection rate.

The confusion matrix of the C-DRNN on the OCA dataset, shown in figure 2, presents the most robust distinguishing between “screw” and “unscrew cover panel” of the three trained models. One the other hand, only the C-DRNN seems to have problems to classify “take screwdriver” and “place screwdriver down” correctly.

The results of DC-LSTM shown in figure 3 and CNN-2 shown in figure 4 are very similar, but the DC-LSTM model differs by displaying the highest false positive rate for Null class.

Although the CNN-2 does not consider history, which seems to be an advantage for repetitive actions, it appears as a robust classifier. One can only guess that it is maybe due to the number of repetitions.

Having the results from the confusion matrices of the three trained models, we decided to train the models again, but this time with a merged class of “screw” and “unscrew cover panel”. The results are showcased in figure 5 and table 8.

Table 9 shows the classification results for the original OCA dataset, but not using the defaults from the Opportunity dataset. In this test, we changed the training, validation and test data using leave one out. Therefore, the test subjects one, two and three are used for training, while test subject four is used for testing and five for validation. The weighted \( F_1 \) score shows poorer results for all models and nearly all variants. The best results comparing the
original configuration and leave one out validation dropping between 5 and 12%. The confusion matrix in figure 6 presents an expected behaviour. Since the test subject from the validation set performs the actions a little bit different, it is not predicted as the correct action class, but instead as the Null class. The rest, for example, the classification problems between screwing actions, is comparable to the results from the original dataset.

5. Discussion
We analyzed and modified algorithms from different neural networks that performed best at HAR in ADL and figure out if they can be used to support wearable soft robotics control. In order to test the algorithms in a realistic industrial scenario, we recorded a dataset representing a
Modern Materials and Manufacturing (MMM 2021)

Using our OCA dataset, which only consists of 90 minutes of recorded data, we were able to

Figure 2: Confusion matrix of C-DRNN on the OCA dataset.

Figure 3: Confusion matrix of DC-LSTM on the OCA dataset.

common industrial assembly task. The actions in the dataset have significant kinetic similarities, which makes them difficult to classify. Nevertheless, trivial adaptation of HAR classifiers that exhibit state of the art performance on the Opportunity ADL datasets delivers promising results. Using our OCA dataset, which only consists of 90 minutes of recorded data, we were able to
achieve a comparable $F_1$ score (for the variant with a combined “Screw” and “Unscrew” class). It must be noted, however, that our set of target classes is smaller than Opportunity’s—we have 6 target classes, while Opportunity has 17 (both excluding the Null class). Furthermore, the confusion matrices make it clear that some of the remaining errors occur between kinetically
similar classes—taking/placing down the screwdriver, and mounting/unmounting the cover panel. The majority of the misclassifications are Null class false positives. The difference in performance between the three examined neural network architectures is far smaller than the difference caused by merging the very similar “Screw” and “Unscrew” classes. We therefore conclude that for similar classes of use-cases—HAR problems with target classes composed of a small set of full-body actions, the most important characteristic is a favorable data setup. That is, the largest improvements in performance can be achieved by first ensuring that the target classes contain minimal kinetic similarity, assuming that is not already irrevocably defined by the problem at hand.

In our use-case, the HAR classification scheme is meant to be used as part of a soft robotics control scheme and must therefore ensure that the system only supports the necessary actions and no Null class actions. Having only a very low false prediction rate at the Null class, this can be assured. But still having a true positive rate of 98% only for the screwing action when it is combined, shows that improvement on algorithms or amount of data is necessary. This result leads to efficient energy management for soft-robotic exoskeletons, because already with those algorithms, it is possible to cover nearly all relevant actions to support and ignore approximately
66\% for external force support inappropriate actions. That more data is needed is also shown by the leave one out validation, where the $F_1$ score drops rapidly. Using the same setup for different people without specializing it for every wearer is an important requirement for industrial soft robotic support systems. The amount of data could also be a reason why the CNN-2 performs similarly to the other algorithms, which should be in advantage due to the repetitive task. Having a greater error for the short time actions like taking the screwdriver can be explained by having an overlapping class from the former window class. This affects short time actions more than long-term actions. Having the processing unit on the system and not on an edge cloud or the like, smaller models make sense to reduce calculations. Therefore, it is beneficial that the $F_1$ score of the reduced models shows nearly the same results as the default models. Nevertheless, this will change with an increase of data, which must be taken into account. The different input variants influence the results. But, using only acceleration and angular velocity and adding orientation estimation show comparable results, only CNN-2 seems to prefer variant 1. Variant 3 and 4 on the other can not match up to the first and second.

Having over 98\% of mid-term actions seems suitable to set up a HAR based control for soft robotic wearables. For the more dynamic short-term actions, the results are okay, but not sufficient to set a robust control. But it is necessary to record more data from different persons to find a more robust model for people not included in the training set. Considering that the dataset contains only 90 minutes working material, it seems to be a manageable overhead for introducing such systems in industrial environments.

6. Future Work
HAR based control for soft-robotic wearables is due to different challenges not implemented yet. First, it is crucial to figure out at which sequential area of the action the classifiers perform worst and what is the reason for it. If it is at the beginning or end of the movements, perhaps changing the window sizes might alleviate the problem. Having the main errors in the middle of the action sequence, one can think about facing the issue through a more robust control. We now established a base to program a control for soft robotic wearables and will port the classifiers onto an embedded computer to use it as the main processing unit. To generate as much data as possible and not miss a vital body segment, we surely overbuilt the sensor system. This will be reduced in the future, in accordance with its impact on the classification results. In the end, exoskeletons only have a small number of sensors at the supportive areas. Last but not least, we were able to show the feasibility even with a small dataset, and surely we want to increase the amount of data, by different persons, more iterations and also more different actions.

References
[1] Mohammad Abu Alsheikh et al. “Deep Activity Recognition Models with Triaxial Accelerometers”. In: CoRR abs/1511.04664 (2015). arXiv: 1511.04664. URL: http://arxiv.org/abs/1511.04664.
[2] Oresti Banos et al. “Design, implementation and validation of a novel open framework for agile development of mobile health applications”. In: BioMedical Engineering OnLine 14.Suppl 2 (2015), S6. DOI: 10.1186/1475-925X-14-S2-S6.
[3] Ling Bao and Stephen S. Intille. “Activity Recognition from User-Annotation Acceleration Data”. In: Pervasive Computing. Ed. by Alois Ferscha and Friedemann Mattern. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–17. ISBN: 978-3-540-24646-6.
[4] Ulf Blanke and Bernt Schiele. “Daily Routine Recognition through Activity Spotting”. In: Location and Context Awareness. Ed. by Tanzeem Choudhury et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 192–206. ISBN: 978-3-642-01721-6.
[5] Ricardo Chavarriaga et al. “The Opportunity Challenge: A Benchmark Database for On-body Sensor-based Activity Recognition”. In: Pattern Recogn. Lett. 34.15 (Nov. 2013), pp. 2033–2042. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2012.12.014. URL: http://dx.doi.org/10.1016/j.patrec.2012.12.014.

[6] Z. Chen et al. “Distilling the Knowledge From Handcrafted Features for Human Activity Recognition”. In: IEEE Transactions on Industrial Informatics 14.10 (Oct. 2018), pp. 4334–4342. ISSN: 1551-3203. DOI: 10.1109/TII.2018.2789925.

[7] Michiel P De Looze et al. “Exoskeletons for industrial application and their potential effects on physical work load”. In: Ergonomics 59.5 (2016), pp. 671–681.

[8] Benjamin Gorissen et al. “Elastic Inflatable Actuators for Soft Robotic Applications”. In: Advanced Materials 29.43 (2017), p. 1604977. DOI: 10.1002/adma.201604977. Eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201604977. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201604977.

[9] Karen N Gregorczyk et al. “Effects of a lower-body exoskeleton device on metabolic cost and gait biomechanics during load carriage”. In: Ergonomics 53.10 (2010), pp. 1263–1275.

[10] Kåre B Hagen and Ola Thune. “Work incapacity from low back pain in the general population”. In: Spine 23.19 (1998), pp. 2091–2095.

[11] Damian Hoy et al. “The epidemiology of low back pain”. In: Best practice & research Clinical rheumatology 24.6 (2010), pp. 769–781.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2014. arXiv: 1412.6980 [cs.LG].

[13] K. Kunze et al. “Towards Recognizing Tai Chi - An Initial Experiment Using Wearable Sensors”. In: 3rd International Forum on Applied Wearable Computing 2006. Mar. 2006, pp. 1–6.

[14] J. Kuschan et al. “PowerGrasp: Concept for a novel Soft-Robotic Arm Support System”. In: ISR 2018; 50th International Symposium on Robotics. June 2018, pp. 1–6.

[15] Heike Leutheuser, Dominik Schulthaus, and Bjoern M. Eskofier. “Hierarchical, multisensor based classification of daily life activities: comparison with state-of-the-art algorithms using a benchmark dataset”. In: PloS one 8.10 (2013), e75196. DOI: 10.1371/journal.pone.0075196.

[16] Nikolao Maniadakis and Alastair Gray. “The economic burden of back pain in the UK”. In: Pain 84.1 (2000), pp. 95–103.

[17] Abdulmajid Murad and Jae-Young Pyun. “Deep Recurrent Neural Networks for Human Activity Recognition”. In: Sensors 17.11 (2017). ISSN: 1424-8220. DOI: 10.3390/s17112556. URL: https://www.mdpi.com/1424-8220/17/11/2556.

[18] Francisco Javier Ordóñez and Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition”. In: Sensors 16.1 (2016). ISSN: 1424-8220. DOI: 10.3390/s16010115. URL: https://www.mdpi.com/1424-8220/16/1/115.

[19] Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. “Feature Learning for Activity Recognition in Ubiquitous Computing”. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two. IJCAI’11. Barcelona, Catalonia, Spain: AAAI Press, 2011, pp. 1729–1734. ISBN: 978-1-57735-514-4. DOI: 10.5591/978-1-57735-516-8/IJCAI11-290. URL: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-290.
[20] Panagiotis Polygerinos et al. “Soft Robotics: Review of Fluid-Driven Intrinsically Soft Devices; Manufacturing, Sensing, Control, and Applications in Human-Robot Interaction”. In: Advanced Engineering Materials 19.12 (2017), p. 1700016. DOI: 10.1002/adem.201700016. Eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adem.201700016. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/adem.201700016.

[21] Fernando Moya Rueda et al. “Convolutional Neural Networks for Human Activity Recognition Using Body-Worn Sensors”. In: Informatics 5 (2018), p. 26.

[22] Skoda Data Set. http://www.har-dataset.org/doku.php?id=wiki:dataset. Accessed: 2019-07-26. 2008.

[23] Muhammad Shoaib et al. “Fusion of Smartphone Motion Sensors for Physical Activity Recognition”. In: Sensors (Basel, Switzerland) 14.6 (2014), pp. 10146–10176. ISSN: 1424-8220. DOI: 10.3390/s140610146.

[24] Mikolaj Walukiewicz. “Aufbau und Inbetriebnahme eines Sensoranzugs für die messtechnische Erfassung von Posen und Bewegungen am Menschen”. Bachelor Thesis. Germany: Technical University of Berlin, 2019.