



## Inertial measurement unit based human action recognition for soft-robotic exoskeletons

Jan Kuschan\*, Moritz Burgdorff, Hristo Filaretov and Jörg Krüger

Fraunhofer Institute for Production Systems and Design Technology IPK, Germany

Received 15 June 2021, accepted 14 July 2021, available online 17 November 2021

© 2021 Authors. This is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>).

**Abstract.** Absence from work caused by overloading the musculoskeletal system lowers the life quality of the worker and entails unnecessary costs for both the employer and the health system. Soft-robotic exoskeletons offer a possibility to overcome these problems by increasing the system flexibility, not limiting the supported Degrees of Freedom and being simultaneously an actuator and a joint. Since such exoskeletons can only be designed for using power when supporting the wearer, battery lifetime can be increased by covering only those actions for which support is needed. As regards controls, a major difficulty lies in finding a compromise between saving energy and supporting the wearer. However, an action-dependent control can reduce the supported actions to only relevant ones and increase battery lifetime. The system conditions include detection of user actions in real time and distinguishing between actions requiring and not requiring support. We contributed an analysis and modification of human action recognition (HAR) benchmark algorithms from activities of daily living, transferred them onto industrial use cases and made the models compatible with embedded computers for real-time recognition on soft exoskeletons. We identified the most common challenges for inertial measurement unit based HAR and compared the best-performing algorithms using a newly recorded dataset of overhead car assembly for industrial relevance. By introducing orientation estimation, F<sub>1</sub>-scores could be increased by up to 0.04. With an overall F<sub>1</sub>-score without a *Null* class of up to 0.883, we were able to lay the foundation for using HAR for action dependent force support.

**Key words:** machine learning, human action recognition, wearables, soft-robotic exoskeletons, assembly, inertial measurement unit.

### 1. INTRODUCTION

Physically demanding tasks in manufacturing, logistics, handicraft and service are vital contributors to early damage of the musculoskeletal system, especially the spine [1]. This stress leads to a reduced quality of life and decreased capacity for work [2,3]. Exoskeletons can present a solution. Typically, such systems struggle with stiffness and discomfort, and primarily with the lack of battery lifetime [4,5]. Soft-robotic exoskeletons offer a possibility to overcome these problems by increasing the system flexibility, while not limiting the supported Degree of Freedom (DoF) and being simultaneously an actuator and a joint [6,7]. Since such exoskeletons can

only be designed for using power when supporting the wearer, battery lifetime can be increased by covering only those actions for which support is needed. Use cases with Inertial Measurement Unit (IMU) based Human Activity Recognition (HAR) can be reasonable to avoid common vision limitations such as occlusion, multiple persons in the field of view, interfering contours or even data protection. HAR can be applied to predict movements and support every individually classified action. However, it can be constrained to only support task related actions. The combination of IMU based HAR and soft exoskeletons is therefore predestined to create an action-based prediction of current or future activities as early and as accurately as possible.

\* Corresponding author, [jan.kuschan@ipk.fraunhofer.de](mailto:jan.kuschan@ipk.fraunhofer.de)

The idea behind Human Activity Recognition is that characteristic sensor signals directly correspond to specific body movements [8] 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” [8]. 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 implementing techniques such as Deep Belief Networks (DBNs) [9] or Hidden Markov Models (HMMs) above Restricted Boltzmann Machine (RBM) layers [10]. While combining hand-crafted feature approaches with machine learning methods can lead to systems that perform well in certain scenarios [11], 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

An active textile upper arm, elbow and hand exoskeleton for workers’ support was presented in [12]. Kuschán 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

When working with wearable systems like soft-robotic exoskeletons, it is reasonable to use the existing sensors such as 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 such as recognizing sports activities [13], elderly care [14] or daily activities [15]. Chavarriaga et al. [16] 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 one is 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 consists 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 performed five ADL sessions, during which they only followed a high-level description of the task. Therefore, they could interpret freely how to achieve the goal rather than following 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 using a large number of sensors of different modalities that were shared between the environment, the objects and the subjects. The dataset is around 6 hours long. The subjects’ body-worn sensors included 7 IMUs and 12 3D acceleration sensors with a total number of relevant onbody 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 [17], Decision Trees [18] or Joint Boosting [19] 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 research, the models were selected according to their reported classification performance on the Opportunity dataset and the perceived complexity of their structure. Three models were selected (hereinafter referred to by the names in the respective publications: Deep Convolutional Long Short-Term Memory (DC-LSTM) [8], Cascaded Bidirectional and Unidirectional LSTM-based Deep Recurrent Neural Network (C-DRNN) [20] and CNN-2 [21]).

DC-LSTM, as well as CNN-2, make use of multiple convolutional layers for feature extraction. However, neither model uses padding, and as such, produces 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 based Deep Recurrent Neural Network (DRNN) from [21] is a cascaded structure. Its first layer is a bidirectional LSTM layer, the output of which is concatenated with a simple summation, followed by unidirectional LSTM layers (bidirectional means here that the layer consists of two LSTMs, one of which processes the data backwards). Even though the reported  $F_1$ -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, performed their test data opti-

mizations on a validation set. Moreover, there is no description of how multiple activity classes were handled during one window. Lastly, no detailed description was presented of their final model structure, and only a few hyperparameters were mentioned.

**DC-LSTM.** DC-LSTM [8] 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. Regarding the Opportunity dataset, all on-body sensors and data splits were used 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 feature maps in the layer and a  $5 \times 1$  kernel, each of which operates along the time axis. Subsequently, the output was fed into two stacked LSTM layers with 128 units each and one softmax layer for classification. For training, the publication proposes a learning rate of  $10e^{-3}$ , which, however, in their implementation was changed to  $10e^{-4}$ . The training was therefore performed with the RMSprop optimiser with the learning rate of  $10e^{-4}$ , a decay factor of  $\rho = 0.9$  and a dropout probability of  $p = 0.5$ . Another inconsistency between their publication and implementation was 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. However, the implementation utilized a window length of 800 ms, which was used by the model that produced the reported  $F_1$ -score.

**CNN-2.** The CNN-2 from [20] consists of two consecutive blocks of two convolutional layers followed by one maxpooling layer, which results in two fully connected and one softmax layer for classification. It was first proposed in [8] as the baseline model and then refined with the use of pooling layers by [20]. Although it only served as a baseline model, it achieved state-of-the-art performance on the Opportunity dataset with  $F_1 = 0.9152$ . The tests adhered to the Opportunity guidelines. It was trained with the RMSprop optimiser with a fixed learning rate of  $10e^{-3}$ , 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 developed from ADL datasets. Moreover, orientation estimation is introduced for IMU based HAR and other varieties of the input data.

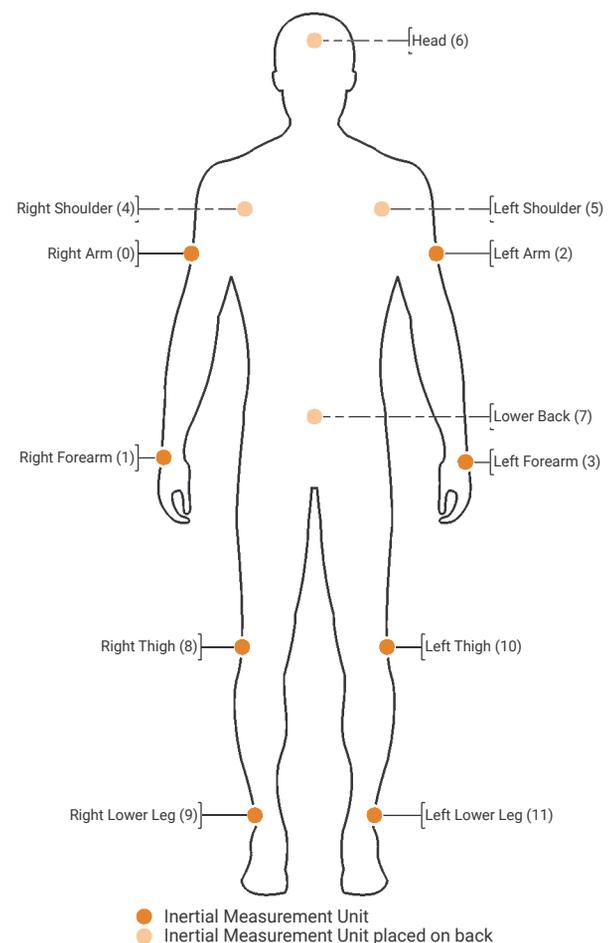
#### 3.1. Overhead car assembly (OCA)

For a dataset that meets our requirements, the OCA dataset was recorded in laboratory environments. OCA covers the mounting and dismounting of a heat capsule under a car. It is a cyclic process taking approximately

75 s. We separated the cycle into six mid-level classes: “Mount Cover Panel”, “Take Panel Down”, “Take Screwdriver”, “Place Screwdriver”, “Screw in Panel” and “Unscrew Panel”, shown in Table 1, and the class distribution is indicated in Table 2. As the lab has no line production, it was necessary to dismount the heat panel after mounting it. Since this is not part of the original process, we decided to handle it as independent classes.

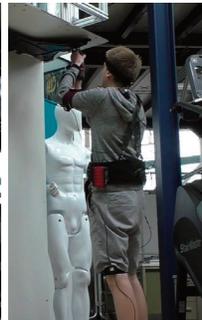
The choice of the actions is related to the wearable robotics use case. OCA contains very dynamic short-term overhead movements, such as mounting the cover panel and taking the cover panel down, without the need of support. However, OCA also contains very static overhead actions such as screwing in and 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 [23]. This vest consists of 12 IMUs, covering all major body segments (Fig.1).



**Fig. 1.** Front view: sensor placement on the vest for the custom dataset. The numbers indicate the index of the sensor in the dataset. [23]

**Table 1.** High level description of the custom dataset

|   |   |   |  |   |   |
|---|---|---|--|---|---|
|  |  |  |  |  |  |
| <b>Mount Cover Panel</b>  | <b>Screw in Cover Panel</b>   | <b>Take Screwdriver</b>   | <b>Take Cover Panel Down</b>   | <b>Unscrew Cover Panel</b>  | <b>Place Screwdriver Down</b>   |
| 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 the table.   |

**Table 2.** Class labels and their distributions for the Overhead Car Assembly (OCA) dataset. Repetitions and instances show the distribution of the individual classes (across all subjects and sessions) after application of the sliding window

| Name              | Repetitions | Instances | (%)  |
|-------------------|-------------|-----------|------|
| <i>Null</i>       | 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             |             | 13 685    |      |

All IMUs provide 3D acceleration, 3D angular velocity and the sensor’s orientation in quaternions for a total number of  $D_t = 120$  channels. The dataset was recorded at a constant sample rate of 40Hz. It is ca 90 min long and covers 70 cycles. All sessions were video-recorded and retroactively labelled with a synchronization tool. We had five test subjects: persons 1, 2 and 3 each recorded two independent datasets, each dataset ca 10 minutes long; persons 4 and 5 each recorded one dataset, ca 15 minutes each.

The short-duration actions such as “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%. However, 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 will discuss their influence on the model if kept distinct and if combined into one class.

### 3.2. Training

These three models [8,20,21] were chosen because of their best benchmark performance on the Opportunity dataset as well as providing an available code. Only the architecture of the models was implemented, all weights were initialized prior to training and affected only by our own training procedure.

For this research, a few variations on the original C-DRNN were made. Murad and Pyun [21] calculated a prediction for every sample  $\mathbf{y}_c^t$ , where  $t = 1, \dots, T$ , and used a merging step to obtain a class distribution for the entire input window. The prediction of the last sample  $\mathbf{y}_c^T$  was replaced similarly to the final step in DC-LSTM to predict the most recent movement. Additionally, the handling of multiple classes in one window was altered, and the data split from the Opportunity challenge was used.

The model consists of three layers, the first layer is a bidirectional LSTM layer, 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 [24], 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  $\mathbf{y}_c = (y_1, \dots, y_{n_c})$  over all  $n_c$  classes, with  $\sum_i y_i = 1$  and  $y_i \in [0, 1]$ , for each of those matrices.

Table 3 presents the variations of the input data, which is also used for the Results and Discussion sections. The first models were trained using only the acceleration and angular velocity raw data captured by the IMUs. For the subsequent variations, the orientation angles  $\alpha_x, \alpha_y, \alpha_z$  were calculated, using an orientation estimation

$$\alpha_{i,[x|y|z]} = \text{acos}(q_{t,z}), \tag{1}$$

**Table 3.** Variants of the input data. Orientation features were calculated as described in equation 1

| 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 |

where

$$\mathbf{q}_i = \mathbf{q}_i \otimes \mathbf{q}_{[x|y|z]} \otimes \mathbf{q}_i^*, \quad (2)$$

where  $\mathbf{q}_{[x|y|z]} = [0, 1, 0, 0]_{[x]}^T, [0, 0, 1, 0]_{[y]}^T, [0, 0, 0, 1]_{[z]}^T$  and  $\mathbf{q}_i$  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 sufficient to determine the configuration of a kinematic chain (in this case, the human body). Finally, the angular velocity was also removed 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 [16]. As for the later results, we will also discuss the influence of isolating one test subject as designated test data.

### 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 by confusion matrices, as they enable to evaluate the classification outcome on a per-class level. Furthermore, it can 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 since 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.

$$wF_1 = \sum_i 2w_i \frac{p_i \cdot r_i}{p_i + r_i}, \quad (3)$$

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

### 3.4. Tests

The tests for this research were conducted in two parts. Firstly, the models were trained and tested on input variations 1–4 (Table 3) on 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. They 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:

$$\arg \max_{m \in \mathcal{M}} \left( |f_i(m; \mathcal{M}) - f_{val\_acc}(m)| \cdot r(m) \right), \quad (4)$$

where  $\mathcal{M}$  is the set of all tested architectures,  $f_i$  denotes 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 of the given model architecture and  $r$  represents the average complexity reduction in % of the model  $m$  compared to the original model. The logarithmic trend line was chosen to have the form  $f_i = -a \cdot e^{b \cdot x} + c$ , because we expect the models to quickly rise in performance with higher model complexity but they level out. Afterwards, the selected model was trained and tested on input variations 1–4 on both the Opportunity and OCA datasets.

## 4. RESULTS

Tables 4 and 5 show the weighted  $F_1$ -scores (equation 3) for the three used networks of the Opportunity dataset and the OCA dataset. Different varieties of input data are presented, and the influence of reducing the model complexity is shown. Especially regarding the Opportunity dataset, the weighted  $F_1$ -score is significantly higher than the one without *Null* class. Since a high number of samples has a direct influence on the weighted  $F_1$ -score, the difference to OCA should relate to the class distribution with 79.2% for Opportunity and 42% for OCA. The  $F_1$ -scores without *Null* class perform better for the default and reduced models of C-DRNN and the reduced models of DC-LSTM and CNN-2, compared to the Opportunity dataset. Interestingly, no distinct pattern for the different input data varieties is observable. The best clas-

**Table 4.** Opportunity dataset

| Model   | Variant | C-DRNN       |              | DC-LSTM      |              | CNN-2        |              |
|---------|---------|--------------|--------------|--------------|--------------|--------------|--------------|
|         |         | $wF_1$       | $wF_1^N$     | $wF_1$       | $wF_1^N$     | $wF_1$       | $wF_1^N$     |
| Default | 1       | 0.874        | 0.549        | 0.886        | 0.596        | <b>0.898</b> | <b>0.636</b> |
|         | 2       | <b>0.880</b> | <b>0.573</b> | <b>0.895</b> | <b>0.623</b> | 0.895        | 0.612        |
|         | 3       | 0.864        | 0.517        | 0.879        | 0.560        | 0.893        | 0.597        |
|         | 4       | 0.844        | 0.453        | 0.879        | 0.552        | 0.890        | 0.585        |
| Reduced | 1       | <b>0.881</b> | <b>0.578</b> | <b>0.887</b> | 0.575        | <b>0.899</b> | <b>0.636</b> |
|         | 2       | 0.880        | 0.554        | 0.886        | <b>0.580</b> | 0.890        | 0.607        |
|         | 3       | 0.875        | 0.545        | 0.883        | 0.562        | 0.889        | 0.589        |
|         | 4       | 0.841        | 0.458        | 0.877        | 0.540        | 0.882        | 0.562        |

**Table 5.** Weighted  $F_1$ -score of default and reduced models with all input variants on the respective dataset. Weighted  $F_1$  without the *Null* class ( $wF_1^N$ )

| Model   | Variant | C-DRNN       |              | DC-LSTM      |              | CNN-2        |              |
|---------|---------|--------------|--------------|--------------|--------------|--------------|--------------|
|         |         | $wF_1$       | $wF_1^N$     | $wF_1$       | $wF_1^N$     | $wF_1$       | $wF_1^N$     |
| Default | 1       | 0.741        | 0.617        | <b>0.695</b> | <b>0.569</b> | 0.751        | 0.634        |
|         | 2       | <b>0.766</b> | <b>0.664</b> | 0.673        | 0.517        | <b>0.754</b> | <b>0.638</b> |
|         | 3       | 0.600        | 0.475        | 0.670        | 0.568        | 0.709        | 0.605        |
|         | 4       | 0.425        | 0.256        | 0.439        | 0.247        | 0.478        | 0.291        |
| Reduced | 1       | 0.735        | 0.607        | 0.694        | 0.581        | <b>0.781</b> | <b>0.684</b> |
|         | 2       | <b>0.768</b> | <b>0.664</b> | 0.708        | 0.579        | 0.727        | 0.596        |
|         | 3       | 0.590        | 0.456        | <b>0.723</b> | <b>0.624</b> | 0.703        | 0.604        |
|         | 4       | 0.432        | 0.264        | 0.477        | 0.299        | 0.494        | 0.322        |

sifiers seem to be either variant 2 (acceleration, angular velocity and orientation estimation) or variant 1 (acceleration and angular velocity). The Opportunity dataset has a better classification of variant 2 for the default models and variant 1 for the reduced models. This is probably 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 performance for most of the datasets and models using variant 3. 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.

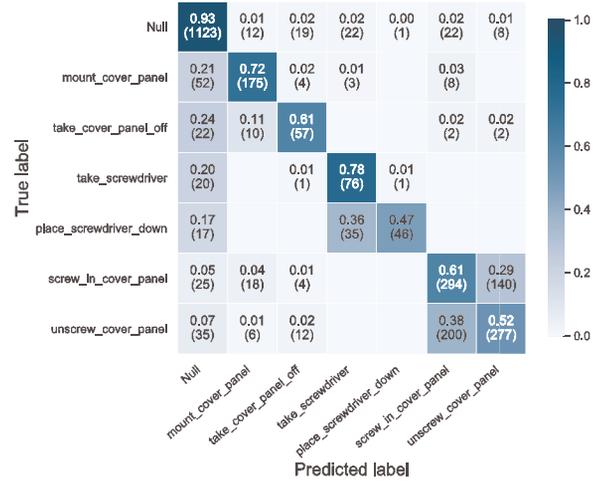
The confusion matrices for the models trained on the OCA dataset were chosen based on the best  $F_1$  presented in Table 5. The confusion matrices for the different models of the OCA dataset in Figs 2, 3, 4 demonstrate that all models display poor results in distinguishing between the “screw in” and “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 C-DRNN on the OCA dataset (Fig. 2) presents the most robust distinguishing between “screw” and “unscrew cover panel” of the three trained models. On the other hand, only C-DRNN seems to have problems with classifying “take screwdriver” and “place screwdriver down” correctly.

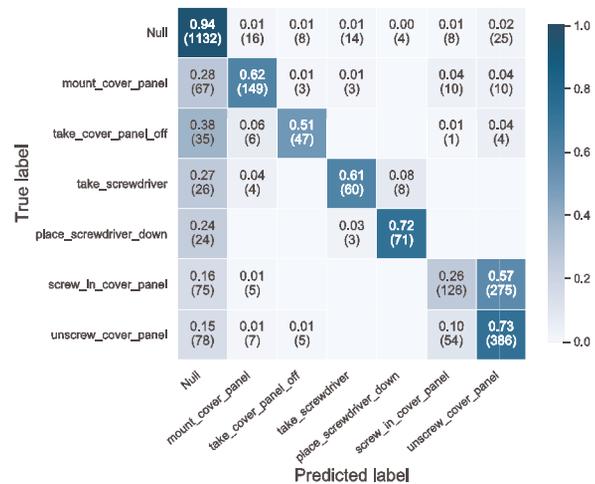
The results of DC-LSTM shown in Fig. 3 and CNN-2 in Fig. 4 are very similar, but the DC-LSTM model differs by displaying the highest false positive rate for *Null* class.

Although CNN-2 does not consider history, which seems to be an advantage for repetitive actions, it appears as a robust classifier. It might be due to the number of repetitions.

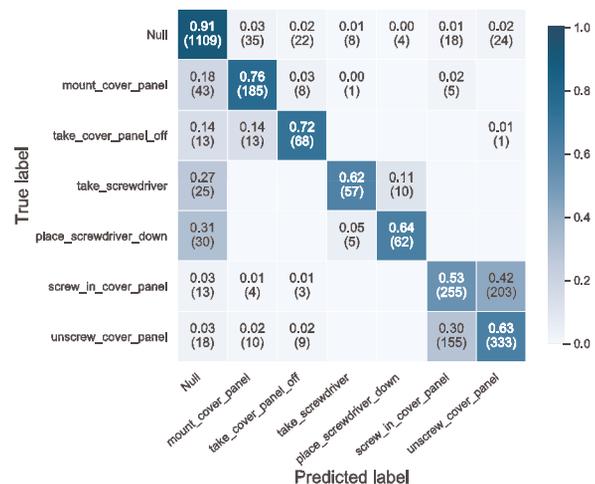
With the results from the confusion matrices of the three trained models, we decided to train the models again, this time with a merged class of “screw” and “unscrew cover panel”. The results are showcased in Fig. 5 and Table 6.



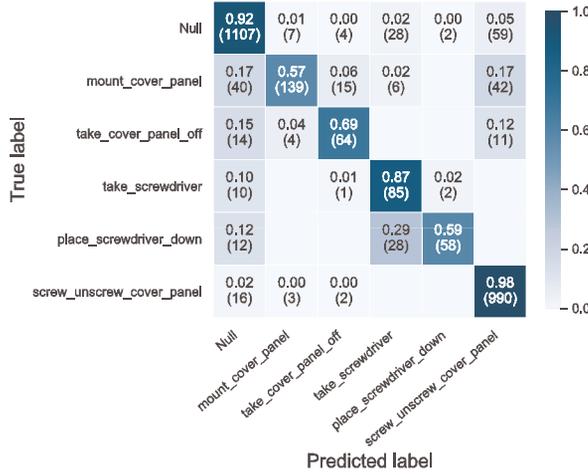
**Fig. 2.** Confusion matrix of C-DRNN on the OCA dataset.



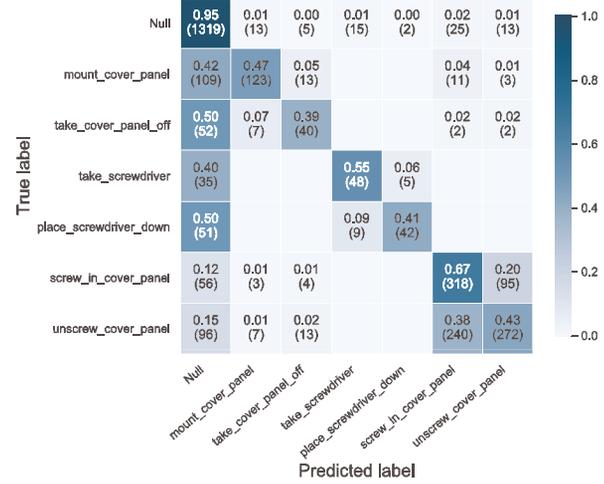
**Fig. 3.** Confusion matrix of Deep Convolutional Long Short-Term Memory (DC-LSTM) on the OCA dataset.



**Fig. 4.** Confusion matrix of CNN-2 on the OCA dataset.



**Fig. 5.** Confusion matrix of C-DRNN on the OCA dataset with combined screw and unscrew cover panel.



**Fig. 6.** Confusion matrix of CNN-2 on the OCA dataset, using leave-one-out validation.

**Table 6.** Weighted  $F_1$ -score of default model with all input variants of the OCA dataset with combined screw and unscrew class. Weighted  $F_1$  without the  $Null$  class ( $wF_1^N$ )

| Model   | Variant | C-DRNN       |              | DC-LSTM      |              | CNN-2        |              |
|---------|---------|--------------|--------------|--------------|--------------|--------------|--------------|
|         |         | $wF_1$       | $wF_1^N$     | $wF_1$       | $wF_1^N$     | $wF_1$       | $wF_1^N$     |
| Default | 1       | 0.819        | <b>0.771</b> | 0.824        | 0.803        | <b>0.883</b> | <b>0.873</b> |
|         | 2       | <b>0.848</b> | 0.824        | <b>0.861</b> | <b>0.848</b> | 0.873        | 0.854        |
|         | 3       | 0.740        | 0.702        | 0.761        | 0.724        | 0.784        | 0.752        |
|         | 4       | 0.530        | 0.436        | 0.661        | 0.610        | 0.659        | 0.592        |

Table 7 shows the classification results for the original OCA dataset, but without using the defaults from the Opportunity dataset. In this test, we changed the training, validation and test data using leave-one-out. Therefore, test subjects 1, 2 and 3 were used for training, while test subject 4 was used for testing and 5 for validation. The weighted  $F_1$ -score shows poorer results for all models and nearly all variants. The introduction of the leave-one-out validation leads to a performance reduction between 5 and 12 %.

The confusion matrix in Fig. 6 presents an expected behaviour. Since the test subject from the validation set performs the actions slightly differently, it is not predicted as the correct action class, but instead as the  $Null$

**Table 7.** Weighted  $F_1$ -score of default model with all input variants of the OCA dataset, using leave-one-out validation. Weighted  $F_1$  without the  $Null$  class ( $wF_1^N$ )

| Model   | Variant | C-DRNN       |              | DC-LSTM      |              | CNN-2        |              |
|---------|---------|--------------|--------------|--------------|--------------|--------------|--------------|
|         |         | $wF_1$       | $wF_1^N$     | $wF_1$       | $wF_1^N$     | $wF_1$       | $wF_1^N$     |
| Default | 1       | 0.626        | 0.485        | <b>0.616</b> | 0.431        | <b>0.693</b> | <b>0.562</b> |
|         | 2       | <b>0.655</b> | <b>0.512</b> | 0.543        | 0.377        | 0.655        | 0.487        |
|         | 3       | 0.586        | 0.445        | 0.607        | <b>0.457</b> | 0.286        | 0.000        |
|         | 4       | 0.433        | 0.224        | 0.512        | 0.333        | 0.497        | 0.310        |

class. The rest (e.g. the classification problems between screwing actions) is comparable to the results from the original dataset.

## 5. DISCUSSION

We have analysed and modified algorithms from different neural networks that performed best at HAR in ADL and will learn if they can be applied to support wearable soft robotics control. In order to test the algorithms in a realistic industrial scenario, we recorded a dataset representing a 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. With our OCA dataset 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). However, our set of target classes with 6 classes is smaller than Opportunity’s set with 17 target classes (both excluding the  $Null$  class). Furthermore, the confusion matrices indicate that some of the remaining errors occur between kinetically similar classes: taking/placing down the screwdriver; mounting/unmounting the cover panel. The majority of the misclassifications are  $Null$  class false positives. The difference in performance between the three neural network architectures is much smaller than the difference caused by merging the very similar “Screw” and “Unscrew” classes. Therefore, we 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 favourable data setup. It means that the largest improvements in performance can be achieved by first ensuring that the target classes contain minimal kinetic similarity,

assuming that it 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. It must therefore ensure that the system only supports the necessary actions and no *Null* class actions. With a very low false prediction rate at the *Null* class, this can be assured. However, a true positive rate of 98% only for the screwing action when it is combined reveals that improvement on algorithms or more data is needed. This result leads to efficient energy management for soft-robotic exoskeletons, as already those algorithms enable to cover nearly all relevant actions needing support and ignore approximately 66% of actions inappropriate for external support. More data is needed as shown by the leave-one-out validation, where the  $F_1$ -score drops rapidly. Implementing the same setup for different people without customizing it is an important requirement for industrial soft-robotic support systems. The amount of data could also explain why CNN-2 performs similarly to the other algorithms, which should be in advantage due to the repetitive task. A greater error for short-time actions such as taking the screwdriver can be explained by having an overlapping class from the former window class. This affects more short-time than long-term actions. As the processing unit is on the system and not on an edge cloud or similar, it is reasonable for smaller models 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, it should be considered that this will change with an increase in data. The different input variants influence the results. However, using only acceleration and angular velocity and adding orientation estimation show comparable results, only CNN-2 seems to prefer variant 1. Variants 3 and 4, on the other hand, are unable to match variants 1 and 2.

With over 98% of mid-term actions, it seems suitable to set up a HAR based control for soft-robotic wearables. For the more dynamic short-term actions, the results are satisfactory, but not sufficient for setting a robust control. It would be necessary to record more data from different persons to develop a more robust model for people not included in the training set. Since the dataset contains only 90 minutes of working material, it appears to be a manageable effort for introducing such systems in industrial environments.

## 6. CONCLUSIONS AND FUTURE WORK

HAR based control for soft-robotic wearables is not implemented yet due to different challenges. First, it is crucial to detect at which sequential area of the action the classifiers perform worst and identify the underlying reason. If it occurs at the beginning or end of the movements, changing the window sizes might alleviate the problem. If the main errors occur in the middle of the action sequence, a more robust control could resolve the issue. We have established a base for programming a

control for soft-robotic wearables and will port the classifiers onto an embedded computer to be used as the main processing unit. As we generated a vast amount of data and tried not to miss a vital body segment, the sensor system was surely overbuilt. This will be reduced in the future, in accordance with its impact on the classification results. Ultimately, exoskeletons have only a small number of sensors at the supportive areas. Lastly, we were able to show the feasibility even with a small dataset and are planning to increase the amount of data by different persons, more iterations and more varied actions.

## ACKNOWLEDGEMENTS

This research was partly funded by the German Federal Ministry of Education and Research (BMBF) in the project PowerGrasp (16SV7314). The publication costs of this article were covered by the Estonian Academy of Sciences and Tallinn University of Technology.

## REFERENCES

1. Hoy, D., Blyth, F. and Buchbinder, R. The epidemiology of low back pain. *Best Pract. Res. Clin. Rheumatol.*, 2010, **24**(6), 769–781.
2. Hagen, K. B. and Thune, O. Work incapacity from low back pain in the general population. *Spine*, 1998, **23**(19), 2091–2095.
3. Maniadakis, N. and Gray, A. The economic burden of back pain in the UK. *Pain*, 2000, **84**(1), 95–103.
4. Gregorezyk, K. N., Hasselquist, L., Schiffman, J. M., Bensek, C., Obusek, J. P. and Gutekunst, D. J. Effects of a lower-body exoskeleton device on metabolic cost and gait biomechanics during load carriage. *Ergonomics*, 2010, **53**(10), 1263–1275.
5. De Looze, M. P., Bosch, T., Krauze, F., Stadler, K. S. and O’Sullivan, L. Exoskeletons for industrial application and their potential effects on physical work load. *Ergonomics*, 2016, **59**(5), 671–681.
6. Gorissen, B., Reynaerts, D., Konishi, S., Yoshida, K., Kim, J.-W. and De Volder, M. Elastic inflatable actuators for soft robotic applications. *Adv. Mater.*, 2017, **29**(43), 1604977. <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201604977>
7. Polygerinos, P., Correll, N., Morin, S. A., Mosadegh, B., Onal, C. D., Petersen, K. et al. Soft Robotics: review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction. *Adv. Eng. Mater.*, 2017, **19**(12), 1700016. <https://doi.org/10.1002/adem.201700016>
8. Ordóñez, F. J. and Roggen, D. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 2016, **16**(1), 115. <https://doi.org/10.3390/s16010115>
9. Plötz, T., Hammerla, N. Y. and Olivier, P. Feature learning for activity recognition in ubiquitous computing. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence – Volume Two. IJCAI’11, Barcelona*,

- Spain, July 16–22, 2011. AAAI Press, 1729–1734. <https://doi.org/10.5591/978-1-57735516-8/IJCAI11-290>
10. Alsheikh, M. A., Selim, A., Niyato, D., Doyle, L., Lin, S. and Tan, H.-P. Deep activity recognition models with triaxial accelerometers. In *CoRR* abs/1511.04664, 2015. <http://arxiv.org/abs/1511.04664>
  11. Chen, Z., Zhang, L., Cao, Z. and Guo, J. Distilling the knowledge from handcrafted features for human activity recognition. *IEEE Trans. Industr. Inform.*, 2018, **14**(10), 4334–4342. <https://doi.org/10.1109/TII.2018.2789925>
  12. Kuschán, J., Goppold, J.-P., Schmidt, H. and Krüger, J. PowerGrasp: Concept for a novel soft-robotic arm support system. In *Proceedings of ISR 2018; 50th International Symposium on Robotics, Munich, Germany, June 20–21, 2018*. VDE, 269–274.
  13. Shoaib, M., Bosch, S., Incel, O. D., Scholten, H. and Havinga, P. J. M. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 2014, **14**(6), 10146–10176. <https://doi.org/10.3390/s140610146>
  14. Banos, O., Villalonga, C., Garcia, R., Saez, A., Damas, M., Holgado-Terriza, J. A. et al. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *BioMedical Engineering OnLine*, 2015, **14**(26). <https://doi.org/10.1186/1475-925X-14-S2-S6>
  15. Leutheuser, H., Schuldhaus, D. and Eskofier, B. M. Hierarchical, multi-sensor based classification of daily life activities: comparison with state-of-the-art algorithms using a benchmark dataset. *PloS One*, 2013, **8**(10), e75196. <https://doi.org/10.1371/journal.pone.0075196>
  16. Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Tröster, G., del R. Millán, J. and Roggen, D. The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. *Pattern Recognit. Lett.*, 2013, **34**(15), 2033–2042. <https://doi.org/10.1016/j.patrec.2012.12.014>
  17. Kunze, K., Barry, M., Heinz, E. A., Lukowicz, P., Majoe, D. and Gutknecht, J. Towards recognizing tai chi an initial experiment using wearable sensors. In *Proceedings of the 2006 3rd International Forum on Applied Wearable Computing*. VDE, 1–6.
  18. Bao, L. and Intille, S. S. Activity recognition from user-annotated acceleration data. In *Pervasive Computing* (Ferscha, A. and Mattern, F., eds). Springer, Berlin, Heidelberg, 2004, 1–17.
  19. Blanke, U. and Schiele, B. Daily routine recognition through activity spotting. In *Location and Context Awareness* (Choudhury, T. et al., eds). Springer, Berlin, Heidelberg, 2009, 192–206.
  20. Rueda, F. M., Grzeszick, R., Fink, G. A., Feldhorst, S. and ten Hompel, M. Convolutional neural networks for human activity recognition using body-worn sensors. *Inform.*, 2018, **5**(2), 26.
  21. Murad, A. and Pyun, J.-Y. Deep recurrent neural networks for human activity recognition. *Sensors*, 2017, **17**(11), 2556. <https://doi.org/10.3390/s17112556>
  22. Skoda Dataset. Human Activity/Context Recognition Datasets. <http://www.har-dataset.org/doku.php?id=wiki:dataset> (accessed 2019-07-26)
  23. Walukiewicz, M. Aufbau und Inbetriebnahme eines Sensoranzugs für die messtechnische Erfassung von Posen und Bewegungen am Menschen. Bachelor thesis. Technical University of Berlin, Germany, 2019.
  24. Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. 2014. arXiv:1412.6980 [cs.LG].

## Inimese luu- ja lihaskonna koormuse inertsiaalsetel mõõtühikutel põhinev tuvastamine pehmerbootilise välisskeleti jaoks

Jan Kuschán, Moritz Burgdorff, Hristo Filaretov ja Jörg Krüger

Luu- ja lihaskonna ülekoormatusest tingitud füüsiline väsimus alandab töötaja elukvaliteeti ning suurendab sellest tulenevaid kulusid nii tööandjale kui ka tervishoiusüsteemile. Toetav ekso- ehk välisskelett võib olla abiks probleemi lahendamisel. Selliste seadmete puuduseks on nende jäikus ja ebamugavus ning akude lühike vastupidavus. Pehmerbootilised eksoskeletid võimaldavad neid probleeme lahendada, suurendades süsteemi paindlikkust, olles samaaegselt nii seadme käitaja kui ka ühenduslüli kasutajaga. Peamine probleem nende projekteerimisel on leida kompromiss energia säästmise ja kandja toetamise vahel. Süsteemi ülesanne on kasutaja tegevuste tuvastamine reaajas ning toetust vajavate liigutuste eristamine vähem olulistest. Antud uuringus analüüsiti ja modifitseeriti 'human activity recognition' (HAR) algoritme inimeste igapäevaste liigutuste alusel. Tulemused kanti üle tööstuslikesse tingimustesse. Uuringus selgitati välja kõige levinumad probleemid inertsiaalsetel mõõtühikutel põhineval HAR-il ja võrreldi kõige paremini toimivaid algoritme. Töö tulemust saab rakendada HAR algoritmide kasutamiseks pehmerbootiliste välisskelettide täiustamiseks.