# End-to-End Motion Classification Using Smartwatch Sensor Data

Torben Windler, Junaid Ahmed Ghauri, Muhammad Usman Syed, Tamara Belostotskaya, Valerie Chikukwa and Rafael Rêgo Drumond

**Abstract** Analysis of smart devices' sensor data for the classification of human activities has become increasingly targeted by industry and motion research. With the popularization of smartwatches, this data becomes available to everyone. The user's data from accelerometers and gyroscopes is conventionally analyzed as a multivariate time series to obtain reliable information about the user's activity at a specific moment. Due to the particular sampling rate instabilities of each device, previous approaches mainly work with feature extraction methods to generalize the information independently of the gear, which requires a lot of time and expertise. To overcome this problem, we present an end-to-end model for activity classification based on convolutional neural networks of different dimensions without extensive feature extraction. The data preprocessing is not computationally intensive and the model can deal with the irregularities of the data. By representing the input as twofold – both, interpolated 1D time series and encoded time series as images with the help of Gramian Angular Summation Fields – the use of computer vision techniques is enabled. In addition, an online prediction is possible and the accuracy is comparable to feature extraction

Torben Windler* ✉ windler@uni-hildesheim.de
Junaid Ahmed Ghauri* ✉ ghauri@uni-hildesheim.de
Muhammad Usman Syed* ✉ syedmu@uni-hildesheim.de
Tamara Belostotskaya* ✉ belostot@uni-hildesheim.de
Valerie Chikukwa* ✉ chikukwa@uni-hildesheim.de
Rafael Rêgo Drumond ✉ radrumond@ismll.uni-hildesheim.de

*the authors contributed equally to this paper

Information Systems and Machine Learning Lab, University of Hildesheim, Germany

Archives of Data Science, Series A
(Online First)
KIT Scientific Publishing
Vol. 6, No. 1, 2020

DOI: 10.5445/KSP/1000098011/12
ISSN 2363-9881

approaches. The model is validated with random 10-fold and leave-one-user-out cross-validation showing improvement regarding the generalization of the task.

# 1 Introduction

Today's possibilities in recording and expressing human motion as data has led to a rapidly growing interest in automatically recognizing and classifying these movements' data as activities using different machine learning techniques. Various approaches in research show that the monitoring and recognition of human activities can be applied to diverse contexts, such as the gaming industry, where motion data of the player can be used for human-computer-interaction or in healthcare, where knowledge about a patient's motions can be included in the research for bionics or neuro-engineering.

The data available for this purpose can come from different sources and in various formats, where the two most common ones are images or videos from cameras and some time series data from sensors. Research has already been conducted in classifying a user's activity from images, where all of them require a specific set up area with cameras installed. The inevitable condition in such a setting is that the user stays in that spatially limited area in the visual range of the cameras.

Since more and more users have access to devices like smartphones or smartwatches featuring sensors like accelerometer or gyroscope, the collection of data of a user's motion is easily accessible in everyday life. In contrast to working with cameras, sensors in wearable devices allow the user to move freely and monitoring is not bound to the visual range of the devices. According to our knowledge and literature review, early research which has been done in this field mostly revolves around the methodology of feature extraction to classify sensor data which comes under the category of time series data classification.

Deep neural networks or Convolutional Neural Networks (CNNs) are capable to classify such time series data coming from different sources like sensors from an electronic device. These models can perform classification tasks with comparable or better accuracy for different classes like activities in our case. Now a research question "Is feature extraction still necessary in the era of

deep learning along with advanced machine learning techniques" needs to be answered. The objective of the methodology introduced in this paper is to answer the research question. End-to-End learning where feature extraction is not needed allows fast classification with the help of trained models on the incoming data stream created by the devices. The approach presented in this work combined with automated preprocessing techniques to handle irregularities in the time series data thus allows implementing stream processing using minor adjustments in future work.

We propose an end-to-end CNN model with different dimensions for human activity classification. Convolutional kernels in CNNs allow the model to learn varieties in the data and are a better choice compared to simple Multi-Layer Perceptrons. Experiments are conducted on both 1D-CNN and 2D-CNN. The results from both models were then merged yielding a better accuracy than each model alone. Due to the structural nature of the data, which consists of one-dimensional time series, a 1D-CNN is used first. The work of Wang and Oates (2015a) is adopted to build the 2D-CNN, where the time series are encoded into 2D images using Gramian Angular Summation Fields (GASF). The results obtained from the merged model are slightly better than feature extraction approaches. Hence, the study supports the assumption that deep learning models and especially CNNs can be used to classify time series data with comparable accuracy, also reducing the effort of feature extraction from the data.

This paper has the following contributions:

- We combine two different convolutional architectures for the time series classification problem.
- They are able to deal with complex heterogeneous data.
- The result is an end-to-end model performing slightly better as the state of the art, but without the need for feature extraction.

Our work is structured as follows: Section 2 introduces the related work that serves as a baseline for generating and preprocessing the time series data coming from sensors as well as setting up the models used. Section 3 describes the data itself in more detail and shows which challenges occurred working with it that

led to the necessary preprocessing procedures shown in Sect. 4. The proposed model that is mainly an adjustment and extension of the technique proposed by Wang and Oates (2015b), is presented in Sect. 5. Sections 6 and 7 describe the experimental setup and display the results. Finally, Sect. 8 gives a conclusion on the techniques examined in this work and shows how they can be employed for future research and applications.

## 2 Literature Overview

There has been a significant development in the study of sensor-based human activity recognition in the past years, and extensive research has been undertaken to select and develop reasoning algorithms to infer activities from wearable sensor data. Bulling et al (2014) provide an extensive introduction to the problem. This has also shown am ubiquitous increase in the use of smart devices like smartwatches for the detection of a user's current activity. Smartwatches and smartphones are embedded with a rich set of sensors such as accelerometer, gyroscope, GPS, microphone, and camera. However, the diversity in smartwatch devices and sensor hardware also has huge impacts, e.g., different accelerometer sensors frequently experience various biases and hence vary in density and exactness (Stisen et al, 2015). Combined with the sampling rate instability of each device, this problem poses a lot of challenges for the human action recognition system design (Blunck et al, 2016).

When dealing with classification-oriented problems like activity recognition, typical approaches involve computing suitable features which are derived from raw sensor data. Past approaches involved solving this by extraction of segments from an input sequence followed by the computation of hand-crafted features which are then used for predicting class labels. Statistical features like maximum, minimum, median, or mean are computed for each data segment. The calculated statistical elements are segmented from the input signal using a sliding window approach (Feldhorst et al, 2016; Blunck et al, 2016). These extracted features are then fed into a classifier for training and to predict class labels (Yao et al, 2017; Jiang et al, 2017). Classifiers like Random Forest (Breiman, 2001) or Support Vector Machines (SVMs, Smola and Schölkopf, 2004) are commonly used in the literature in order to do this. However, this classification workflow requires extensive expertise and is most likely time consuming because of the diverse

device settings like sensor heterogeneity or noise (Stisen et al, 2015). Huynh and Schiele (2005) also show that overfitting can be a problem when using approaches that include handcrafted features. Hence, this requires traversing through different combinations of features to be selected for the training process.

Recent techniques from the deep learning area have significantly transformed human activity recognition and computer vision research. The main advantage of using deep learning is the identification and analysis of representative features from enomous data (Najafabadi et al, 2015). An earlier approach to human activity recognition based on deep learning is developed by Alsheikh et al (2015). Here spectrogram images are generated from an inertial signal to feed images to a CNN, therefore, discarding the need to reshape the signals in a suitable format. Through learning with weight sharing strategies and local connectivity, CNNs can exploit translational invariances making them a widely used approach in several deep learning tasks (Lecun et al, 1998; Krizhevsky et al, 2012). However, this step of generating spectrogram images adds initial overhead to the training of the network by replacing the process of feature extraction. Even though work by Alsheikh et al (2015) provides better recognition accuracy for human activities by focusing on triaxial accelerometers, it does not consider sensor heterogeneity or some type of validation like cross-validation to evalute the model proposed, which can have a huge impact on the data and the results obtained.

Wang and Oates (2015a) suggest a similar technique. In their work, they propose a method of encoding time series data as images using specific imaging methods such as Gramian Angular Fields (GAF) which are then fed into a CNN. This approach permits the use of computer vision techniques for time series classification. Their method has two main steps: The first step involves normalization and transformation of the time series into a polar coordinate system. Afterward, the angular perspective is exploited by taking into account the trigonometric difference for each point within different time intervals to distinguish the temporal correlation. Given a time series of size $n$, the resulting image will be a matrix of size $n \times n$. For larger time series, its size can be reduced and smoothed using Piecewise Aggregation Approximation (PAA, Keogh and Pazzani, 2000). However, this work is not evaluated on complex datasets, for example a dataset from different sensors for human activity recognition. The

proposed model also does not take adavantage of ensemble learning (Opitz and Maclin, 1999), which helps to improve machine learning results by combining several models.

The dataset used for this work (Stisen et al, 2015) has achieved (weighted) F1-scores of up to 90% (random 10-fold cross validation) and 67% (leave-one-user-out cross validation). This results are achieved by extensive feature engineering and the use of basic machine learning algorithms like Random Forests, $K$-Nearest Neighbor (Mucherino et al, 2009), or Support Vector Machines. The paper at hand proposes an end-to-end model for this problem using deep learning without the need for manual feature extraction. To the best of our knowledge, the methodology presented in this paper is one of the first approaches to merge CNNs of different dimensions in the field of activity classification. The uniqueness of this research is that we forgo manual feature extraction and focus on automated preprocessing.

## 3 Data Description

This work uses the Heterogeneity Human Activity Recognition (HHAR) dataset created by Stisen et al (2015). It consists of motion sensor data, specifically gyroscope and accelerometer, acquired from smartphones and smartwatches of various device models. The readings are recorded while 9 users execute 6 activities (*bike*, *sit*, *stand*, *stairsdown*, *stairsup*, and *walk*) carrying smartwatches and smartphones. For the scope of this research, only the smartwatch data is used for time series motion classification.

The labels are determined before performing the activity, and the smartwatches are worn on both wrists. Four smartwatches from two different brands (Samsung and LG) are used for this purpose. All devices used differ in their supported maximum sampling rate. For the LG watches the sampling rate is 200 Hz, and for the Samsung Galaxy Gear it is 100 Hz. The data for both sensors (gyroscope and accelerometer) has a similar structure. For each, there are 7 attributes, namely: index, creation time (OS time), arrival time (data logging time), user ($a - i$), model (watch model, e.g., Samsung or LG), device (specific device details, e.g., $lgwatch_1$ or $lgwatch_2$), and "gt" (ground truth, here: activity). The data also consists of three feature axes $x$, $y$, and $z$. These are the values provided by the respective sensor used to build the time series instances in the experiment.

**Table 1:** Number of accelerometer measurements per user-device-activity combination.

| device | user | *bike* | *sit* | *stairsdown* | *stairsup* | *stand* | *walk* |
|---|---|---|---|---|---|---|---|
| $gear_1$ | *a* | 5,547 | 5,643 | 3,574 | 4,532 | 5,061 | 4,825 |
| | *b* | 5,985 | 7,280 | 3,960 | 5,930 | 5,290 | 6,347 |
| | *c* | 5,175 | 4,416 | 6,205 | 7,346 | 4,202 | 5,246 |
| | *d* | 4,385 | 5,990 | 4,631 | 5,988 | 4,952 | 5,698 |
| | *e* | 7,437 | 5,714 | 5,585 | 7,844 | 5,950 | 6,304 |
| | *f* | 9,528 | 0 | 8,971 | 20,068 | 0 | 0 |
| | *g* | 3,820 | 4,160 | 5,060 | 3,857 | 4,516 | 6,757 |
| | *h* | 4,680 | 0 | 29,884 | 27,158 | 0 | 26,977 |
| | *i* | 0 | 0 | 0 | 0 | 0 | 0 |
| $gear_2$ | *a* | 324 | 1,488 | 201 | 193 | 1,822 | 3,533 |
| | *b* | 1,640 | 214 | 206 | 214 | 312 | 228 |
| | *c* | 804 | 824 | 1,075 | 1,128 | 1,140 | 1,290 |
| | *d* | 2,124 | 379 | 108 | 415 | 732 | 225 |
| | *e* | 218 | 866 | 1,644 | 732 | 807 | 1,390 |
| | *f* | 112 | 1,250 | 1,060 | 3,701 | 482 | 3,060 |
| | *g* | 329 | 214 | 198 | 213 | 506 | 2,108 |
| | *h* | 200 | 1,389 | 324 | 293 | 217 | 237 |
| | *i* | 0 | 0 | 0 | 0 | 0 | 0 |
| $lgwatch_1$ | *a* | 60,489 | 62,660 | 53,212 | 45,539 | 60,798 | 55,467 |
| | *b* | 78,049 | 62,567 | 53,876 | 60,298 | 63,978 | 66,758 |
| | *c* | 56,503 | 59,678 | 53,386 | 53,758 | 62,794 | 60,886 |
| | *d* | 51,629 | 60,543 | 47,099 | 54,521 | 58,066 | 63,432 |
| | *e* | 69,512 | 60,098 | 53,984 | 54,339 | 62,959 | 65,605 |
| | *f* | 74,608 | 61,307 | 55,682 | 59,582 | 63,644 | 62,066 |
| | *g* | 1 | 0 | 0 | 0 | 0 | 0 |
| | *h* | 21,826 | 0 | 2,737 | 3,516 | 6,405 | 196 |
| | *i* | 380 | 737 | 20,847 | 11,170 | 797 | 1,488 |
| $lgwatch_2$ | *a* | 21,842 | 1,318 | 2,239 | 1,565 | 23,907 | 20,435 |
| | *b* | 1 | 0 | 0 | 0 | 0 | 0 |
| | *c* | 22,666 | 1,555 | 13,289 | 14,151 | 6,032 | 4,573 |
| | *d* | 9,459 | 0 | 0 | 0 | 0 | 19,884 |
| | *e* | 35,110 | 2,394 | 23,667 | 7,656 | 1,310 | 20,459 |
| | *f* | 15,432 | 7,759 | 8,867 | 1,750 | 1,256 | 23,438 |
| | *g* | 56,375 | 1,566 | 12,880 | 3,288 | 2,208 | 10,179 |
| | *h* | 1 | 0 | 0 | 0 | 0 | 0 |
| | *i* | 9,339 | 1,986 | 11,925 | 13,009 | 1,046 | 670 |
| overall | | 635,530 | 423,995 | 486,376 | 473,754 | 451,189 | 549,761 |

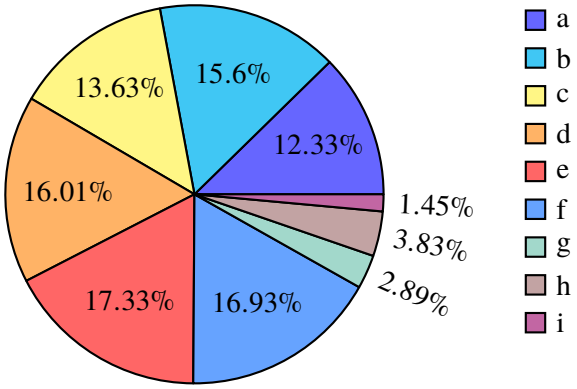


**Figure 1:** Distribution of the proportion of measurements per user.

The dataset contains partially missing data for some user/device-combinations, as it can be seen in Table 1 which shows aggregated instance counts for each device w.r.t. activities and users for the smartwatch accelerometer sensors. Many cells show activities for some particular users where either the sensor failed to record measurements or the OS failed to log it such as in the case for $gear_1$, user $f$, where data is missing for activities *sit*, *stand*, and *walk*. In more dire cases, such as for user $g$ for the device $lgwatch_1$, some users only have one instance per activity. For $gear_1$ and $gear_2$ the accelerometer recordings for user $i$ and some activities are completely missing. In general, the instance counts for users $g$, $h$, and $i$ are relatively low in comparison to other users, as shown in the pie chart in Figure 1. Furthermore, the log rate or the rate at which the OS of the devices timestamped the data instances are often irregular. Training on time series with an irregular time step would become a challenge since large gaps in the data sequence cause poor learning of the algorithms. These gaps in the time series and data sequences are treated through various techniques such as interpolation or Gramian Matrices from GASF which are discussed later in this paper.

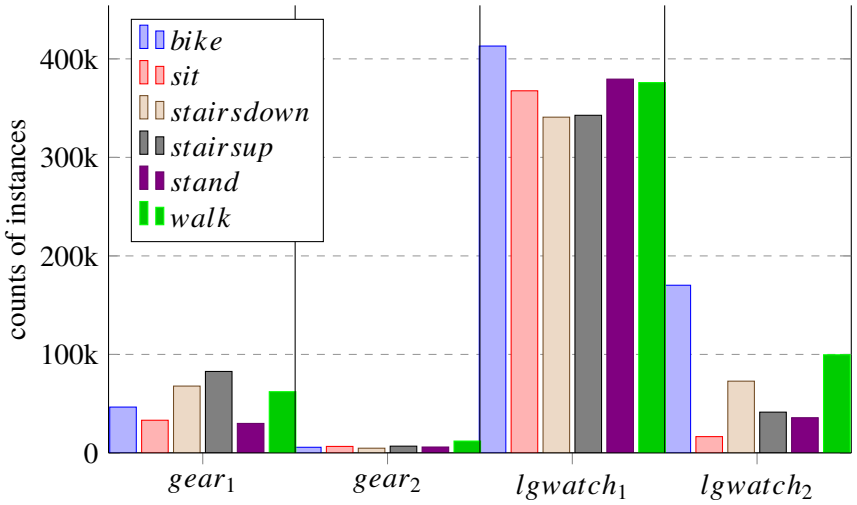


**Figure 2:** Distribution of instance counts per device.

The uneven distribution of the instance counts can be seen in the bar graph in Figure 2. It also shows the disparity in distribution of rows amongst the devices. In case of $gear_1$, the highest number of instances is for the activity *stairsup* with 82k samples across all users. The class with the lowest number of recorded samples is the activity *stand* with 29k frames. There is a big difference between the minimum and maximum number of measurements for all devices. This uneven distribution could cause biased learning by the algorithm for specific users, devices or activities. There is a considerable difference in the case of $lgwatch_2$ where the most instances are recorded for the activity *bike* with 170k frames whereas the least instances are recorded for the activity *sit* with instances a little more than 16k.

Additionally, even within the same device, each activity was recorded over differing timespans. Some activities are executed for a longer time (e.g., 5 minutes) whereas some only last for a shorter period (e.g., 1 minute). A significant difference in the models is that the $lgwatch$ has a higher sampling rate than the $gear$ (200 Hz vs. 100 Hz), but this does not explain the disproportionality of the absent magnitude between the instance counts of the devices. Another interesting finding in the data profiling is that the data from both sensors (accelerometer and

gyroscope) is not aligned in many cases, which impedes the preparation of data for classification. Stisen et al (2015) explain these irregularities with sampling rate heterogeneities and sampling rate instabilities and go into any more detail in their paper. Here they provide specific sections to the data collection process and its characteristics. Because of the properties of the data, some preprocessing techniques should be applied before the classification procedure.

## 4 Data Preprocessing

Due to the described characteristics of the dataset, a complete data preprocessing is required. The first challenge is to align the two sensors as they lack any direct connectivity. Therefore a full outer join is used on the attribute *creation time* per each combination of device, user, and activity. It also handles the instances for some activities within a user that are present for one sensor but missing on the other. The resulting empty rows are imputed using linear interpolation for such cases on both sensors. This data is then saved in 6 separate structures for each sensor axis in the form of time series. The final step is to split the initial time series into smaller subsets of the lengths 16, 32, 64, 128, 256, and 512 using a 50% overlapping sliding window since the activity over the whole timespan behaves periodically. The purpose of this is to augment the dataset by increasing the number of samples.

## 5 Model Structure

For time series data, the use of CNNs is highly beneficial as they can learn directly from raw time series and hence do not require extensive expertise to engineer the input features manually. In this work, a combination of a one dimensional CNN (1D-CNN) and a two dimensional CNN (2D-CNN) is used. The structures of the models are introduced in detail in the following.

## 5.1 1D-CNN

The 1D-CNN consists of one-dimensional layers and filters of order $m$. The primary motivation behind applying this model is the structural nature of the data. As mentioned before, the data is a one-dimensional time series with six axes which are used as input channels $x, y, z$ (accelerometer) and $x, y, z$ (gyroscope).

The input to the first convolutional layer has the shape [*batch size*, *sequence length*, *channels*]. The number of channels is equal to 6 (gyroscope and accelerometer axes). The sequence length for a whole training cycle is the respective length of the subsets. This input is fed to four convolutional layers with the filter size of 12, 24, 48, and 96, respectively, followed by a max-pooling layer with a pooling size of 3 and a stride of 1. After each max-pooling layer batch normalization is applied. The last layer is fully connected with a flattened output that uses the softmax function to classify the given time series as an activity.

## 5.2 2D-CNN

As convolutional networks for image classification advanced in the last years, these techniques are applied to the HHAR dataset as well. To enable the use of techniques from the computer vision domain, we follow the work of Wang and Oates (2015b) by converting the time series into 2D-images. A time series $X = \{x_1, x_2, \ldots, x_n\}$ rescaled to the interval $[-1, 1]$ is used to compute the Gramian Matrix $G$:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cdots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cdots & \cos(\phi_n + \phi_n) \end{bmatrix} \text{ with } \phi_i = \arccos(x_i) \tag{1}$$

This 2D representation of the original time series has the advantages of preserving spatial and temporal dependencies since time increases from top-left to bottom-right (Wang and Oates, 2015a). Moreover, $G$ contains temporal

correlations and the original time series can be reconstructed by the main diagonal in the neural network. A major disadvantage is the squared size $n \times n$ in relation to the original time series $X$ with length $n$.

The data from both sensors is processed in this way, which results in a 4D-tensor with the shape [*batch size*, *sequence length*, *sequence length*, *channels*]. According to the various channels from both sensors the last dimension is also 6. The 2D-CNN used for classification is analogous to the popular Xception model including depth-wise separable convolutions (Chollet, 2016) with the only difference that the size of the input channels is equal to six as opposed to three in the original model for RGB-images.

## 5.3 Merging 1D & 2D-CNN

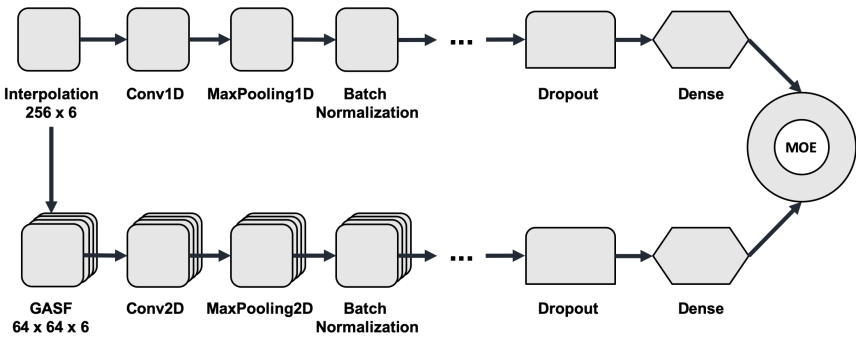


**Figure 3:** General structure of the "hybrid" CNN.

As described later in Sect. 7, the results from both CNN approaches are different. With leave-one-user-out cross-validation, the F1-scores per user and the ranks from best to worst scores differ between 1D and 2D-CNN. To verify that an ensemble of both models would result in better classification since the models' predictions do not correlate, the results of both, 1D & 2D-CNN, are concatenated followed by a Dense Mixture of Experts (MoE) layer (see Figure 3). The new final model consists of 10 feed-forward neural networks (experts) and a trainable

gating network which selects a combination of the experts to process each input (Kaiser et al, 2017).

## 6 Experiments

In the experiments data from all four available smartwatch devices $gear_1$, $gear_2$, $lgwatch_1$, and $lgwatch_2$ is considered. To validate the trained model, two cross-validations (CV) methods are used: random 10-fold CV and leave-one-user-out CV. In the random 10-fold CV, the complete dataset is randomly split into ten subsamples of equal size, where one of them is used as test data, and the remaining nine subsamples are used as training/validation data. This process is performed ten times with each of the subsamples used once for testing. The results of all ten folds are then averaged into a single estimate.

The leave-one-user-out CV is considered for comparison because a model trained with 10-fold CV is likely to learn the classification based on user-specific patterns in the time series. Here, the data is split user-wise, so that each user is used once for testing and the remaining eight users are used for training/validation. This process is also executed nine times with each of the nine users being used once for testing. The results of all nine folds are then averaged by their respective weights (see Figure 1) which is in line with standard HAR research practices (Bieber et al, 2009; Sagha et al, 2011).

The hyper-parameter configuration is as follows: We choose a very small learning rate of $\alpha = 0.00005$, since we have experienced a high variance in validation performance with higher numbers. The training epochs are set to 500, and at the last layer of each submodel is a 40% Dropout layer. With categorical crossentropy as loss function, the model is trained for accuracy and afterwards the respective F1-scores are calculated. The results with the window length of 256 measurements with 50% overlap have shown to be the best within several window lengths (see Sect. 4). These windows are interpolated to an even distribution over time with a rate of 100 Hz. Due to hardware capacities, PAA with a factor 4 is applied on the input windows for the 2D-model. Hence, the sequence length for the 1D-part of the model is equal to 256, whereas the Gramian Matrix $G$ has a dimension of $64 \times 64$ for the 2D-part.

# 7 Results

This section presents the results obtained from the models compared to the F1-scores that are achieved in the baseline paper. As mentioned earlier, the baseline as well as the approach presented in this paper, both use random-10-fold CV and leave-one-user-out CV. Using LSTM networks does not provide a satisfactory result, so this model is not further considered. Using the preprocessing technique on the data that was presented in Sect. 4 and feeding this data to a 1D-CNN improves the average score significantly. However, it is still not comparable to the results mentioned in the baseline using feature extraction.

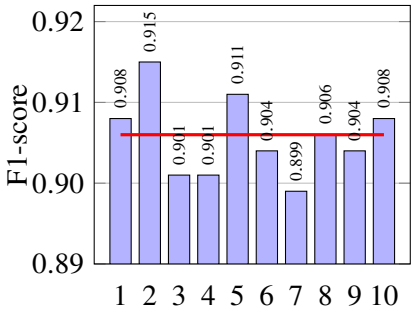


**Figure 4:** F1-scores of random 10-fold CV.

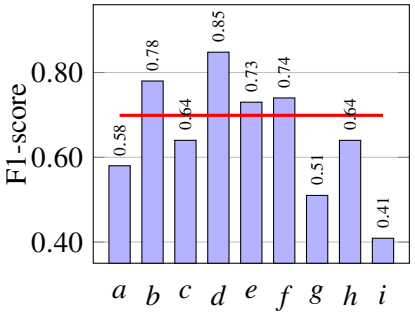


**Figure 5:** F1-scores of leave-one-user-out CV.

The technique presented in Subsection 5.2 using GASF for transforming the time series data to an image and feeding this image to a 2D-CNN achieves a weighted average F1-score of 66.3% using the leave-one-user-out CV, being almost as high as the baseline. To further improve the prediction quality of the model, an ensemble technique was applied, combining the results of the 1D-CNN and 2D-CNN as described in Subsection 5.3.


**Figure 6:** Confusion matrix of an example random fold result from the merged CNN.

**Table 2:** F1-scores of the different approaches.

| Model | Random 10-Fold | Leave-One-User-Out |
|---|---|---|
| Baseline (Feature Extraction) | 90% | 67% |
| 1D-CNN | 90% | 61% |
| 2D-CNN | 90% | 66% |
| 1D & 2D-CNN | **91%** | **70%** |

Figures 4 and 5 show the overall F1-scores per fold that are achieved by this "hybrid" model. The results illustrate that the weighted F1-score achieved by leave-one-user-out CV (69.9%) is significantly lower than the one achieved by random 10-fold CV (90.6%, see Table 2). This score confirms that the time series data is dependent on a respective user and a random 10-fold CV does not generalize the time series for each activity enough. As a consequence, this would not allow the model to be utilized in a wide range of scenarios. The high variation in the F1-scores per user (see Figure 5) further emphasizes the discrepancy in the data quality of the time series as discussed in Sect. 3.

The confusion matrix in Figure 6 shows that the majority of activities are recognized correctly with an acceptable confidence. Due to similar characteristics in the sensor data, the classification among the activities *bike*, *sit*, and *stand* can not always be performed unequivocally. The same can be observed for the distinction between the activities *stairsdown*, *stairsup*, and *walking*.

# 8 Conclusion

In this work, we demonstrate a competitive fusion technique for efficiently classifying human activity using an ensemble of Convolutional Neural Networks with different dimensions. Due to many irregularities in the recordings of the gyroscope and accelerometer sensors, e.g., missing values and model specific gaps, an efficient way of handling the difficulties associated with smartwatch sensor data is presented.

We found out that the measurements are very user-specific and require the leave-one-user-out CV approach to train a model to generalize on activities instead of users or devices. Through the use of interpolated 1D time series and 2D encoded times series as images, the model presented in this work combines two approaches of dealing with time series data, resulting in a better classification score than each model alone and performs slightly better than the baseline, which uses extracted features as input for basic classifiers.

Although our approach covers the area of time series classification with CNN models along with strong evaluation using cross-fold and leave-one-out validation, deep learning has other models as well like Multi-Layer Perceptrons or Recurrent Neural Networks. The task at hand could be trained on the same data to have a comparison with CNNs, although it is well-known that the performance of CNNs is higher than that of the other models for a majority of tasks. Accuracy, performance, and training time of all these models should be compared as well to summarize the study in the deep learning domain in a broader spectrum. Unfortunately, our research scope did not cover that hence leaving room for potential future work.

In addition to the methodology presented in this paper, future work can include adjustments of the proposed model to classify on streaming data and the

use of different ensemble methods. Especially the performance of time series with missing values could be explored. The model could be applied to different datasets from the Human Activity Recognition field as well as to time series datasets of other domains which will prove the methodology to its new heights. Future steps can be to use light deep learning models including CNNs and others as mentioned earlier for embedded devices. As a consequence, these devices do not need to connect to servers where heavy models are deployed. In contrast, they would be able to perform classification tasks on the devices itself. It would be interesting to find out if these computationally light models have an effect on performance or accuracy.

## References

Alsheikh MA, Selim A, Niyato D, Doyle L, Lin S, Tan HP (2015) Deep Activity Recognition Models with Triaxial Accelerometers.

Bieber G, Voskamp J, Urban B (2009) Activity Recognition for Everyday Life on Mobile Phones. In: Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments, Stephanidis C (ed), Springer, Berlin, Lecture Notes in Computer Science, pp. 289–296. DOI: 10.1007/978-3-642-02710-9_32.

Blunck H, Bhattacharya S, Stisen A, Prentow TS, Kjärgaard MB, Dey A, Jensen MM, Sonne T (2016) Activity Recognition on Smart Devices: Dealing with Diversity in the Wild. GetMobile: Mobile Computing and Communications 20(1):34–38. DOI: 10.1145/2972413.2972425.

Breiman L (2001) Random Forests. Machine Learning 45(1):5–32. DOI: 10.1023/A:1010933404324.

Bulling A, Blanke U, Schiele B (2014) A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. ACM Computer Survey 46(3):33–1–33:33. DOI: 10.1145/2499621.

Chollet F (2016) Xception: Deep Learning with Depthwise Separable Convolutions. Tech. Rep., Honlolulu. DOI: 10.1109/CVPR.2017.195, 1610.02357.

Feldhorst S, Masoudenijad M, ten Hompel M, Fink GA (2016) Motion Classification for Analyzing the Order Picking Process Using Mobile Sensors. In: Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods, Marsico MD, di Baja GS, Fred A (eds), SCITEPRESS - Science and Technology Publications, Lda, ICPRAM 2016, pp. 706–713. DOI: 10.5220/0005828407060713.

Huynh T, Schiele B (2005) Analyzing Features for Activity Recognition. In: Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies, Bailly G, Crowley JL (eds), Association for Computing Machinery, New York, sOc-EUSAI '05, pp. 159–163. DOI: 10.1145/1107548.1107591.

Jiang Z, Zheng Y, Tan H, Tang B, Zhou H (2017) Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 1965–1972. DOI: 10.24963/ijcai.2017/273.

Kaiser L, Gomez AN, Shazeer N, Vaswani A, Parmar N, Jones L, Uszkoreit J (2017) One Model To Learn Them All. arXiv: 1706.05137 [cs, stat].

Keogh EJ, Pazzani MJ (2000) Scaling Up Dynamic Time Warping for Datamining Applications. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Ramakrishnan R, Stolfo S, Bayardo RJJ, Parsa I (eds), Association for Computing Machinery, New York, KDD '00, pp. 285–289. DOI: 10.1145/347090.347153.

Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Chien AA (ed), Curran Associates Inc., NIPS'12, pp. 1097–1105. DOI: 10.1145/3065386.

Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE 86(11):2278–2324. DOI: 10.1109/5.726791.

Mucherino A, Papajorgji PJ, Pardalos PM (2009) K-Nearest Neighbor Classification. In: Data Mining in Agriculture, Vol. 34, pp. 83–106. Springer, New York. DOI: 10.1007/978-0-387-88615-2_4.

Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E (2015) Deep Learning Applications and Challenges in Big Data Analytics. Journal of Big Data 2(1):1. DOI: 10.1186/s40537-014-0007-7.

Opitz D, Maclin R (1999) Popular Ensemble Methods: An Empirical Study. Journal of Artificial Intelligence Research 11:169–198. DOI: 10.1613/jair.614.

Sagha H, Digumarti ST, Millán JdR, Chavarriaga R, Calatroni A, Roggen D, Tröster G (2011) Benchmarking Classification Techniques Using the Opportunity Human Activity Dataset. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics, Tunstel E, Nahavandi S (eds), pp. 36–40. DOI: 10.1109/ICSMC.2011.6083628.

Smola AJ, Schölkopf B (2004) A Tutorial on Support Vector Regression. Statistics and Computing 14(3):199–222. DOI: 10.1023/B:STCO.0000035301.49549.88.

Stisen A, Blunck H, Bhattacharya S, Prentow TS, Kjærgaard MB, Dey A, Sonne T, Jensen MM (2015) Smart Devices Are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition. In: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, Association for Computing Machinery, New York, NY, USA, SenSys '15, pp. 127–140. DOI: 10.1145/2809695.2809718.

Wang Z, Oates T (2015a) Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks. In: Workshops at the

Twenty-Ninth AAAI Conference on Artificial Intelligence. URL: https://aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10179.

Wang Z, Oates T (2015b) Imaging Time-Series to Improve Classification and Imputation. In: Proceedings of the 24th International Conference on Artificial Intelligence, AAAI Press, Buenos Aires, pp. 3939–3945.

Yao S, Zhao Y, Zhang A, Su L, Abdelzaher T (2017) DeepIoT: Compressing Deep Neural Network Structures for Sensing Systems with a Compressor-Critic Framework. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, Association for Computing Machinery, New York, pp. 1–14. DOI: 10.1145/3131672.3131675.