Ayala Raanan, ID: 040474934, Gil Levy, ID: 029378577

AML Final Project Report – AttnSense model for HAR

1. Introduction

Our field of research is Human Activity Recognition (HAR). The source of the data for HAR applications is often multiple sensors, positioned on a human body and reporting the motion of the participant while performing various activities (such as running or climbing steps). The sensors used often include IMUs with accelerometers, gyroscopes, and magnetometers. Such sensors produce continuous time-based data. The goal of HAR models is to classify a time window of data to the right activity.

Our work focuses on the paper “AttnSense: Multi-level Attention Mechanism for Multimodal Human Activity Recognition” by Haojie Ma et al. from 2019. We follow the paper to reconstruct the proposed model and reach similar results. We evaluate the contribution and necessity of different components used in the model and check the robustness of the reported scores.

The conventional HAR model tries to classify an activity within a defined window of time. Each window is split into shorter non-overlapping segments called time-steps. First, data from all sensors within a time-step is processed into an intermediate representation of the time-step. Then, additional processing considers results from all time-steps to produce a prediction. There are numerous variations on the details involved in each of the processing steps. The AttnSense model by Ma et al. was one of the first to suggest using attention to better capture spatial and temporal dependencies in the data:

- *Spatial dependencies* – Not all sensors contribute equally to the classification of activities. AttnSense uses an attention mechanism to combine data from different sensors. The model learns what weight to give each sensor for the sake of the classification task.

- *Temporal dependencies* – Perhaps not all time-steps within a time window contribute equally. Some parts of the activity profile may be more salient than others. The paper uses another attention mechanism applied to the time-step output representations.

Using Attention mechanisms may improve interpretability of the model by letting us see for example which sensors are more important (which helps in designing real commercial systems).

2. Related work

Early work in the field was based on engineered features, extracted as input to a machine learning model, such as SVM [Bulling et al., 2011], K-NN, decision trees, random forest [Stisen et al., 2015], etc. Later models used deep learning neural networks ranging from DBNs to CNNs to extract features and classify activities. RNNs were introduced and utilized for temporal analysis. Hammerla et al.[2016] compared LSTM to CNNs showing benefits of each approach.

Another modification along the way by Alsheikh et al. [2016] suggested to convert the problem to the frequency domain by applying FFT (fast Fourier transform) to the data. This method both improved results and effectively halved the dimensionality of the data.

The state-of-the-art model prior to this paper (Deepsense, [Yao et al., 2017]) used CNNs as subnets for feature extraction from each sensor. All axes from a single sensor enter one subnet (SF approach). An additional CNN was then applied for fusing different sensors together. Finally, GRU layers combined data from all time-steps for a downstream task representation.

As can be expected, there are countless approaches to preprocessing the data: noise cleaning, data augmentation, handling segmentation, data sensor fusion and many more [Chen et al. 2021 review paper].

AttnSense, as well as others [Murahari et al., 2018], suggested using attention for both sensor data fusion and time-step data combination. The model details will be described in section 5.

3. Data description

We chose to focus on the PAMAP2 dataset for results reconstruction. This dataset is available, explored in many papers as reference, and provides synchronized time-based sampling from all sensors, which is convenient to work with. The PAMAP2 dataset consists of 9 subjects. Each of the subjects had to follow a protocol, containing 12 different activities. Data was collected with:

- 3 IMUs (internal measurements units) sampling at 100Hz: One IMU over the wrist on the dominant ARM, one IMU on the chest and one IMU on the dominant side's ankle. Each sensory data contains the following information: Temperature (°C), 3D-acceleration data (ms-2), scale: ±16g, resolution: 13-bit, 3D-acceleration data (ms-2), scale: ±6g, resolution: 13-bit, 3D-gyroscope data (rad/s), 3D-magnetometer data (μT), orientation (invalid in this data collection)

- HR (heart rate) monitor with sampling rate of 9Hz

Altogether, over 8 hours of data were collected, while ~30% of the data relates to idle transients. To comply with the paper, our model used the 3 IMUs, without orientation and without temperature and HR. We can see the nature of the raw data in Fig.1. Fig. 2 shows the distribution of raw samples by subject ID, and fig. 3 shows the distribution of raw samples by activity. More information on the dataset can be found in the PAMAP2_data_statistics notebook.

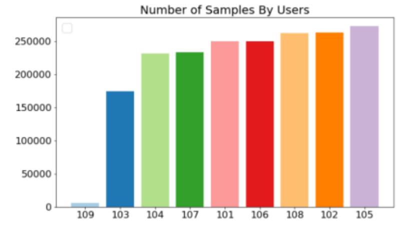


Fig. 2 Distribution of raw samples by user

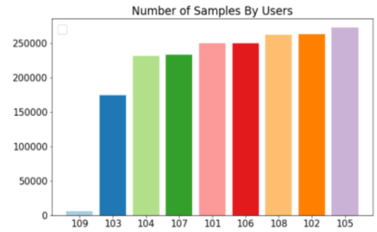


Fig. 3 Distribution of raw samples by activity

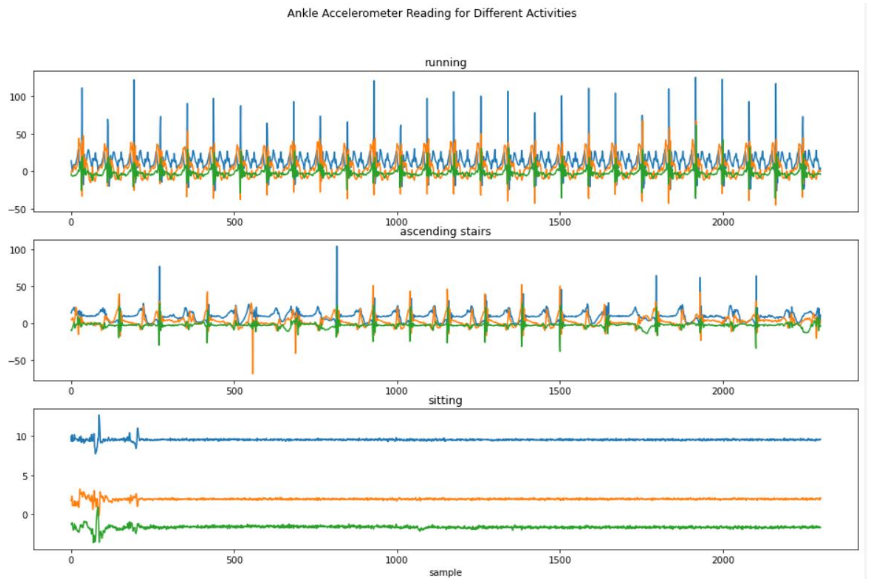


Fig.1 Raw samples during different activities

4. Data preparation and preprocessing

Each sample of the model contains an N-length sequence of time-steps. We assumed samples are segmented, such that all activity labels within a window belong to a single activity. Each time-step contains 25 time-based datapoints, measured at a rate of 100 Hz, thus representing 0.25 sec. We experiment with the sequence length N in section 8.2. When choosing N=20 time-steps, a full sample will represent 5 seconds of an activity.

As in the paper, we keep only data from gyroscopes, magnetometers, and the higher range accelerometers (±16g). This makes an overall of 27 sensors' axes. Since these are all devices of a symmetric range, data normalization included finding the maximal absolute value for each device and scaling by that factor. Note that this was done per device and not per axis. Meaning, that all 3 axes of a specific device were scaled together. This operation is important to preserve the ratios between intensities in difference axes and difference devices, which holds vital information. It is not written in the paper how normalization was carried out. If scaling was done per axis, this may be a reason for our better results.

In further processing, each time-step data is converted from time domain to frequency domain by FFT, where we keep the magnitude of the result and lose the phase. Due to FFT maximal sampling rate, this operation reduces the data length of each time-step from 25 to 13 including frequency 0 term (DC).

In the attempt to make the model more robust to noisy devices, AttnSense also uses data augmentation by adding normally distributed noise to the normalized data (prior to FFT). We will present our analysis for results with and without data augmentation.

5. The AttnSense model

The AttnSense model incorporates the following subnets: (The model schematics are given in figure 4 in the AttnSense paper).

1. A sensor axis based convolutional subnet. Unlike DeepSense which processed together all axes of a device (e.g. 3-D accelerometer), the AttnSense paper does not mention this, so we assumed an individual CNN is applied for feature extraction from each sensor axis.

The details of the CNN layers are given in the paper. From the schematics it seems like the output spatial dimension is preserved in the CNN, so output size is equal to input size. We compared between a 1-layer CNN and a 4-layers CNN.

2. To reduce dimensionality, data from all sensors is fused together for a single vector representation. AttnSense uses attention to fuse the data with learned weights. A linear transform is applied to all CNN outputs prior to attention. Since the output dimension is not specified, we keep it equal to the input dimension.

3. After the above process is applied to all time-steps in the sequence of length N, the time-steps enter a double-layered GRU for capturing data in the sequence. The GRU hidden size is not specified. We used 50 and confirmed it is close to optimal.

4. Another attention layer is used, where the inputs are the sequence of the GRU representations of each time-step. This layer is supposed to find which time-steps are more important for the classification. The output size is again equal to the input size.

5. Finally, a linear classifier head using softmax produces the activity prediction with argmax.

6. Implementation challenges

We are not aware of an existing implementation. Our model was written from scratch, using all the details given in the paper. There are many details missing from the description of the model, such as the GRU hidden size, internal linear layers sizes, or the magnitude of noise for augmentation. We experiment with some. A single run takes ~1.5 hours without data augmentation and ~5 hours with augmentation on a Tesla P100 GPU. Our extensive experiments required huge amount of processing time on several computers.

Skoda dataset. The missing timestamps, needed to sync the data, were a large overhead. In addition, Skoda dataset is of a single subject. Our work shows that having multiple subject datasets is critical. Therefore, we chose to focus on the PAMAP2.

All our code can be found at: https://github.com/ayalaraanan/AML-Final-Project

7. Training

As in the paper, we keep subject 106 for test data. We split the rest of the data to train set and dev set with a 0.1 ratio. Data augmentation (when used) is based on the train set alone. We use cross-entropy for our loss criteria and Adam optimizer with both learning rate and weight decay at 1e-4. We evaluate loss, accuracy and f1-score on the dev set during training. A deep copy of the model is saved with each f1-score improvement. We run a maximum of 100 epochs with a stopping condition being 10 epochs without dev f1-score improvement.

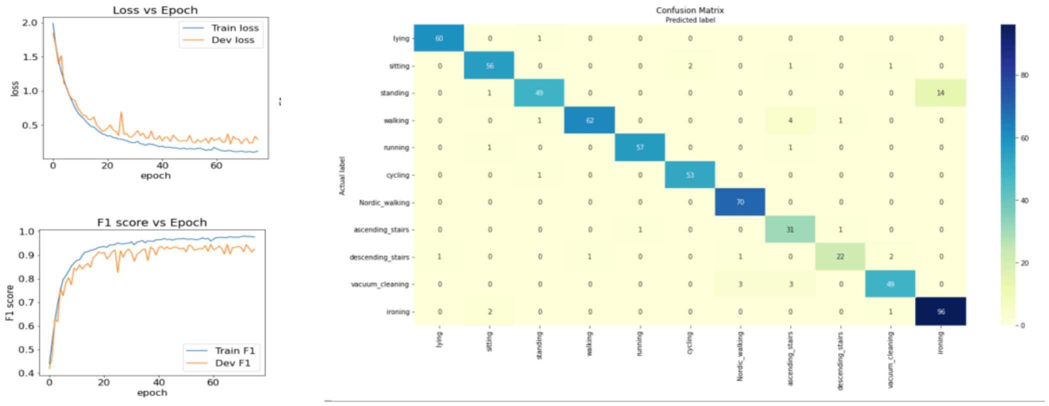
8. Evaluation and additional experiments

At the end of each experiment, we obtain predictions for each labeled test sample. With these predictions we can construct a confusion matrix and calculate precision, recall and f1-score for each label separately and weighted scores for all labels. We report and compare these scores in the different settings of the experiments in table 1 and below.

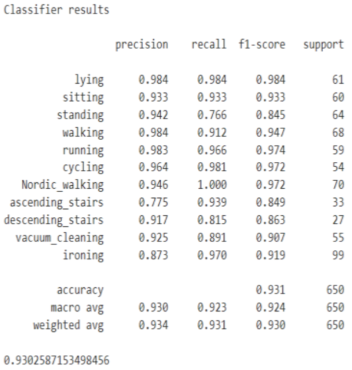
Model Type	AttnSense paper	Our Model	Comments
Base model	0.893 ± 0.013 (w augmentation)	0.891 ± 0.029 (w/o augmentation)	Same best model as reported in the paper: Full preprocessing, N = 20, 4 Conv layers, Attention on both sensors and sequence
Base model with augmentation	0.893 ± 0.013	**0.911** ± 0.018	For details refer to augmentation section below
1 conv layer (Original is 4)	0.82	0.873 ± 0.055	1 conv layer runs faster per epoch but less stable
With attention only on sensors	0.867 ± 0.007	0.875	Paper: "No AG" mode
No FFT	Not reported	0.857 ± 0.026	Longer epochs and worse results – FFT is useful
SVM on preprocessed samples	Not reported	0.872	Applied Sklearn model with default parameters
RF on preprocessed samples	Not reported	**0.924** ± 0.004	Applied Sklearn model with default parameters

Table 1 F1 Scores for various models on PAMAP2. Test on subject 106

8.1 Base model: our anchor model was the model with the best configuration as reported in the paper, but without augmentation. This means a sequence of length N = 20, FFT applied, a 4-layers CNN and both attention layers (sensors + spatial). With this model we obtained a result of 0.891 ± 0.029. This is as high as the paper presented *including augmentation*.



	lying	sitting	standing	walking	running	cycling	Nordic_walking	ascending_stairs	descending_stairs	vacuum_cleaning	ironing
lying	60	0	1	0	0	0	0	0	0	0	0
sitting	0	56	0	0	0	2	0	1	0	1	0
standing	0	1	49	0	0	0	0	0	0	0	14
walking	0	0	1	62	0	0	0	4	1	0	0
running	0	1	0	0	57	0	0	1	0	0	0
cycling	0	0	1	0	0	53	0	0	0	0	0
Nordic_walking	0	0	0	0	0	0	70	0	0	0	0
ascending_stairs	0	0	0	0	1	0	0	31	1	0	0
descending_stairs	1	0	0	1	0	0	1	0	22	2	0
vacuum_cleaning	0	0	0	0	0	0	3	3	0	49	0
ironing	0	2	0	0	0	0	0	0	0	1	96



```
Classifier results

                   precision    recall  f1-score   support

            lying      0.984     0.984     0.984        61
          sitting      0.933     0.933     0.933        60
         standing      0.942     0.766     0.845        64
          walking      0.984     0.912     0.947        68
          running      0.983     0.966     0.974        59
          cycling      0.964     0.981     0.972        54
    Nordic_walking      0.946     1.000     0.972        70
 ascending_stairs      0.775     0.939     0.849        33
descending_stairs      0.917     0.815     0.863        27
  vacuum_cleaning      0.925     0.891     0.907        55
          ironing      0.873     0.970     0.919        99

         accuracy                          0.931       650
        macro avg      0.930     0.923     0.924       650
     weighted avg      0.934     0.931     0.930       650

0.9302587153498456
```

Fig. 5 Base model results for one of the best runs

8.2 Experiments with model architecture and parameters

Basic model with augmentation - For augmentation, we used a factor of 5 - meaning 4 additional copies with added noise. Noise was added relative to the STD found for each type of sensor. We experimented with noise at a scale of 20%, 40% and 200%. With noise at 200%, f1-score was degraded to 0.74, so this was not repeated. 40% and 20% were repeated with average f1-scores of 0.903 and 0.911 respectively. This result is *higher than the paper reported*. We speculate that the difference is in the details not given in the paper. Possible differences may be our scaling method, pure segmentation method, or hidden states sizes. Due to the long run time with augmented set (~5 hours on Tesla P100 GPU), we conducted the rest of our comparison work without augmentation.

Base model w/o FFT - We checked the contribution of the FFT. With this stage removed, each time-step is back to being 25 points long. The f1-score result without FFT was 0.857 ± 0.026. This is lower than the basic model, although we expected a bigger difference. In addition to the score difference, it was noted that this model took longer to run per epoch due to the data dimensionality. We conclude that FFT is a useful operation for HAR.

Basic model with different sequence size N - The sequence size N can greatly influence the prediction accuracy. On one hand, N needs to be long enough to contain enough data for a good prediction. On the other hand, you can expect less long coherent samples from random data. The ratio of non-pure samples was 0.0065 for N=5 and 0.043 for N = 20. Numbers are negligible.

In our specific dataset, the "rope jumping" activity has very little representation in time. So much so, that for sequences of 20, there are no more pure samples of "rope jumping". F1-score is averaged over 11 labels instead of 12.

With each dataset of different N, we ran training and evaluation 5 times. The averaged results are shown in figure 6. This confirms the paper's conclusion that a sequence of 20 is optimal for this dataset and model combination.

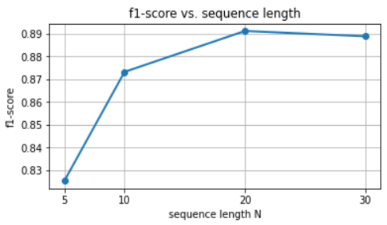


Fig. 6 Optimizing sequence length

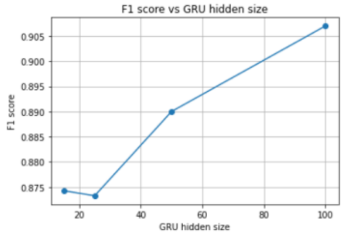


Fig. 7 GRU hidden state size optimization

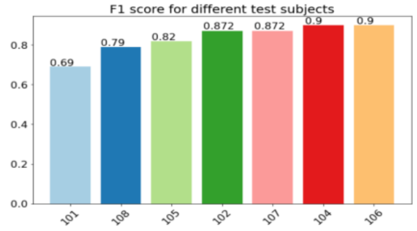


Fig. 8 F1-score for different subjects

Basic model, GRU hidden sizes - The paper does not state the GRU hidden size used. We have explored a few relevant values for this architecture parameter: (15, 25, 50, 100). We expect the hidden size to be in the range of a few input sizes. The input size is 13 (25 timestamps represented by 13 values after FFT). All results reported used GRU hidden size = 50, which are confirmed here to be a reasonable working point.

Different subjects as test - The paper keeps all samples of one subject (106) as a test. This makes sense since real evaluation will probably be done on an unknown subject. However, the paper does not state why this subject was chosen and does not report results for other subjects. We think it makes more sense to report at least an average f1-score over different test subjects. Therefore, we tested the model over all subjects (with a threshold on minimal samples). Fig. 8 shows results for different test subjects. We see major differences when reporting the F1 score for different subjects. We have found a reference in a following paper (Buffelli and Vandin [7]) that reports a much lower f1 score for AttnSense and think it may be related to issues of reporting test scores on different subjects.

Discussion on attention: The attention mechanism used in AttnSense is a simple form of attention. When used for sensor fusion it is forced to accommodate both the difference in the sensors and the difference in activities. For attention to be more efficient, it's possible that a multihead attention, using additional context vectors in parallel, would enable different attention heads to learn specific activities and improve prediction.

Regarding the attention used in the time domain (over GRU outputs), the use of attention with the PAMAP2 database is not very useful. The reason is that all the activities in the dataset are repetitive. Most of them do not have a clear beginning and end defining the activity. During segmentation, we extract windows that are basically shifted in time, but roughly contain the same data. This is not the case for the Skoda dataset for example with activities such as "opening trunk". Hammerla et al. concluded in their comparison of CNNs and RNNs for HAR datasets that "prolonged and repetitive activities like walking or running" are better detected using CNNs over RNNs.

We extracted the GRU attention weights and averaged them over all samples for each label separately (figure 8). We can see that the attention weights pattern for all activities is similar and almost trivial. The last few time-steps receive a greater weight. This can be expected since the last time-steps in the GRU accumulated data from the entire sequence.

Our results are very similar to those reported in Murahari and Ploetz (On Attention Models for Human Activity Recognition) where they show most weight is given to the final hidden states.

Classic ML algorithms - The paper presented base results for Random Forest (RF) and SVM. We assume these algorithms were applied to raw samples - meaning that each sample is a single point in time with readings from all sensors. We wanted to test performance of classic algorithms when applied to the equivalent of our samples. We then took all the data of a single processed sample, containing all sensors after FFT, and all time-steps within a window, and flattened them as new inputs. We applied the scikit-learn tools on this data. The results were completely surprising. **RF achieved better results than AttnSense** for subject 106. Results were also better or comparable on all of the other subjects. The obvious plus is that RF takes less than a minute to run. Figure 10 shows the results.

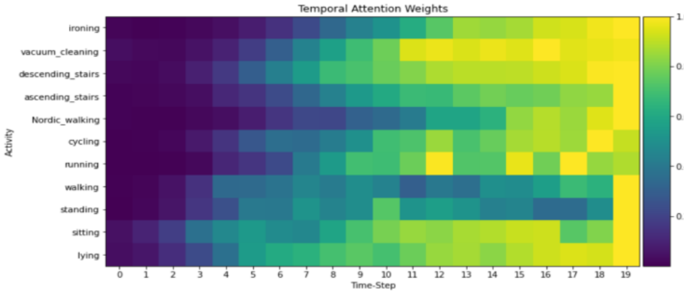


Fig 9 Attention weights for GRU outputs

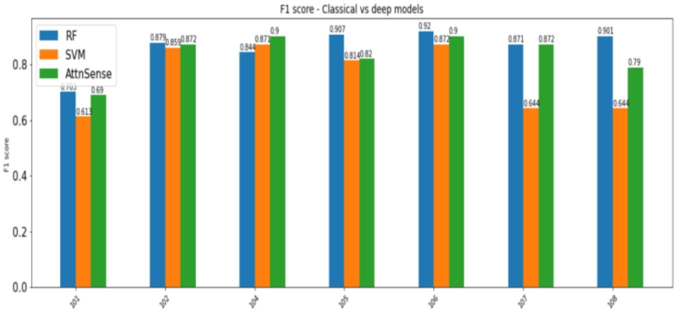


Fig. 10 Comparison to classic ML algorithms

9. Summary

We have achieved a similar score as the paper for the same model parameters without using augmentation. With augmentation we have achieved a slightly higher score. We also achieved similar results to the paper when applying attention only on sensors data. We did not observe score reduction when avoided temporal attention like reported in the paper. We found two interesting issues that were not reported by the paper: a. High difference in F1-score when testing over different subjects. b. Achieved higher results with a RF model applied on the preprocessed samples.

10. Innovation – Next steps

1. *Data augmentation*: As described in detail on the project proposal
2. *Model: Multiple attention vectors* (as in the project proposal). The architecture of the existing model offers only one context attention vector per sensor and one per time. We will explore the impact of multiple attention vectors per sensors and per time
3. *Field calibration* – Adjust the model to a new subject by additional fine tuning. This work is triggered by our latest work.
4. *Model: Subject classification*. If time permit, we plan to design a model that classifies a subject to the closest training subjects and utilize this info for better activity classification. This work is also triggered by our latest work.

References

[1] Haojie Ma, Wenzhong Li, Xiao Zhang, Songcheng Gao, and Sanglu Lu. 2019. AttnSense: Multi-level Attention Mechanism For Multimodal Human Activity Recognition. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019. 3109–3115.

[2] Mohammad Abu Alsheikh, Ahmed Selim, Dusit Niyato, Linda Doyle, Shaowei Lin, and Hwee-Pink Tan. 2016. Deep activity recognition models with triaxial accelerometers. In Workshops at the Thirtieth AAAI Conference on Artificial Intelligence

[2] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 351–360.

[3] Nils Y. Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. In Twenty-Fifth International Joint Conference on Artificial Intelligence. 1533– 1540.

[4] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Troster. Eye movement analysis for activity recognition using electrooculography. IEEE transactions on pattern analysis and machine intelligence, 33(4):741–753, 2011.

[5] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Mikkel Baun Prentow, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 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 (SenSys'15), pages 127–140. ACM

[6] Vishvak S Murahari and Thomas Plötz. 2018. On attention models for human activity recognition. In Proceedings of the 2018 ACM International Symposium on Wearable Computers. ACM, 100–103.

[7] Buffelli, D.; and Vandin, F. 2020. Attention-Based Deep Learning Framework for Human Activity Recognition with User Adaptation. CoRR abs/2006.03820.

[8] Chen, Kaixuan, et al. "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities." ACM Computing Surveys (CSUR) 54.4 (2021): 1-40.