

# Too Good To Be True: accuracy overestimation in (re)current practices for Human Activity Recognition

Andrés Tello  
*Bernoulli Institute*  
*University of Groningen*  
Groningen, The Netherlands  
andres.tello@rug.nl

Victoria Degeler  
*Informatics Institute*  
*University of Amsterdam*  
Amsterdam, The Netherlands  
v.o.degeler@uva.nl

Alexander Lazovik  
*Bernoulli Institute*  
*University of Groningen*  
Groningen, The Netherlands  
a.lazovik@rug.nl

**Abstract**—Today, there are standard and well established procedures within the Human Activity Recognition (HAR) pipeline. However, some of these conventional approaches lead to accuracy overestimation. In particular, sliding windows for data segmentation followed by standard random k-fold cross validation, produce biased results. An analysis of previous literature and present-day studies, surprisingly, shows that these are common approaches in state-of-the-art studies on HAR. It is important to raise awareness in the scientific community about this problem, whose negative effects are being overlooked. Otherwise, publications of biased results lead to papers that report lower accuracies, with correct unbiased methods, harder to publish. Several experiments with different types of datasets and different types of classification models allow us to exhibit the problem and show it persists independently of the method or dataset.

**Index Terms**—Performance Overestimation, Biased Accuracy, Human Activity Recognition, Random K-Fold Cross-Validation

## I. INTRODUCTION

Human Activity Recognition is an ongoing research topic in the fields of ubiquitous and pervasive computing, health-care, ambient assisted living, among others. Several methods has been proposed for HAR, from traditional machine learning algorithms [1]–[3], to current Deep Learning approaches [4]–[7]. With either approach, supervised learning is commonly used to learn models that classify activities based on annotated sensor data collected from an instrumented testing environment, e.g., a smart home, wearable IMU sensors. Generally, HAR implementations include data collection, pre-processing, data segmentation, feature extraction, and classification.

Some previous studies on HAR noticed that conventional methods used within the HAR pipeline can lead to accuracy overestimation. Hammerla and Plötz [8] proved that standard k-fold cross validation (CV) are biased due to statistical dependence between data samples, specially when using sliding windows for data segmentation. The problem was also mentioned in [9]–[13], although these papers focused on different goals, and did not go into analyzing the problem in detail.

Surprisingly, it is a widely-used ongoing practice within the HAR domain. Table I shows a list of recent studies where *sliding windows segmentation* and *random K-fold CV* are used in the HAR pipeline. While not exhaustive, this list includes remarkable works from previous years, and a growing number of present-day studies. These works have been published in top journals or presented at top conferences which have created a high impact in the HAR community.

In this work, we evaluated the effect of sliding windows data segmentation and random splitting on model accuracy. We used datasets with different data modality: CASAS [32] binary motion sensors, MHEALTH [33] and PAMAP2 [34] on-body inertial sensors. Likewise, classification models of different nature: Random Forest (RF), Graph Neural Networks (GNNs) were applied. The results show that independently of the type of data and the chosen model the reported accuracy is highly overestimated following the aforementioned approach.

The main contribution of this paper is two-fold: (1) It provides an extensive survey of recent papers in the HAR domain that employ the flawed methodology, showing the importance of warning the community. (2) It provides a set of experiments using different types of HAR datasets and different types of supervised machine learning approaches, evidencing that the problem persists in all cases and configurations.

The remainder of this document is as follows. Section II describes the methodological issues in conventional approaches for HAR. Section III presents an extensive recent related work on HAR incurring in this problem. Section IV describes our experiments, presents the results and discussion focused on showing the effects of the problem. Section V presents the lessons learned. Finally, section VI presents the conclusions.

## II. MODEL PERFORMANCE OVERESTIMATION

Data segmentation following some windowing technique is the conventional approach for HAR [35], [36], and sliding windows are the most widely adopted [37] approach. The windows can be overlapping or non-overlapping, where the length is defined in  $t$  seconds or  $s$  number of sensor readings.

The input stream of sensor readings is split into windows of equal size. Then, a set of features is calculated from each window, which are used as input for the classification models. Fig. 1 shows the windows-based data segmentation.

One way to evaluate the performance of the classifier is using a hold out part of the entire dataset, i.e., the test set. The conventional approach is randomly splitting the dataset into training/test subsets at some predefined ratio (e.g., 80:20). The training set can be split further, obtaining a validation subset which is used for model selection and/or hyper-parameter optimization. The most used technique to assess the performance of the classifiers is k-fold CV [9]. First, the data is split into  $k$  disjoint subsets of equal size. Then, the model is trained on  $k-1$  subsets and evaluated on the  $k^{th}$ . This process is repeated  $k$  times with a different subset. The final performance of the model is the mean of all runs. The CV method assumes data samples to be Independent and Identically Distributed (i.i.d.) [38]; then, the way of choosing samples does not affect the classifier’s performance. This is where the problem on (re)current practices for HAR lies. *Using sliding windows for data segmentation and feature extraction, the statistical independence assumption does not hold anymore. Therefore, random training/test split does affect the performance of the model because of contiguous windows are assigned one to training and the previous and(or) next to the test set.*

The problem can be observed with greater clarity in Fig. 1a. Windows  $w_1$ ,  $w_2$ , and  $w_3$  share the samples  $s_3 - s_6$ . Sample  $s_2$  is shared between  $w_1$  and  $w_2$  and  $s_7$  between  $w_2$  and  $w_3$ . Hence, the features obtained from those windows are drawn from almost the same underlying data samples, breaking the i.i.d. assumption. Thus, if  $w_2$  is randomly chosen for testing and  $w_1$  and  $w_3$  are kept for training, the classifier is tested on data that was already seen. Consequently, the performance of the classification model is overestimated. In the case of non-overlapping windows (Fig. 1b), the dependence between consecutive windows is less evident. However, for long run-

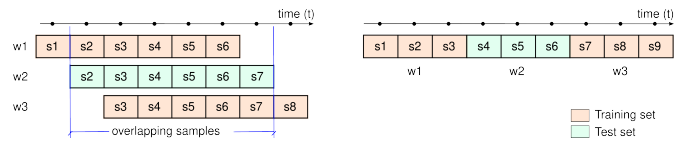


Fig. 1: Overlapping and non-overlapping sliding windows data segmentation

ning activities (e.g., walking, standing, reading), the similarity between samples drawn in a short interval will create a strong correlation between consecutive windows [8], [9]. Hence, it is likely that consecutive windows have similar samples corresponding the same activity. From Fig. 1b, if  $w_2$  is randomly assigned to the test set, it will be almost identically to  $w_1$  and  $w_3$  assigned to the training set. This creates an illusion of perfect accuracy because of overfitting, caused by the strong correlation between consecutive windows and random splitting of training and test sets. The models just memorize the training data instead of learning the patterns that uniquely characterize each activity. They produce the correct label because the same data was seen during training.

Performance overestimation can be avoided ensuring the independence between training and test sets. One option is applying Leave-One-Subject-Out CV (LOSO-CV), a variant of k-fold CV [8], [9]. In LOSO-CV, instead of randomly choosing the samples to include in each fold, the data samples belonging to one subject are used for testing, while the data from the remaining subjects are used for training. This is repeated for each subject in the experiment. This approach is more rigid than traditional k-fold CV, but it ensures the independence between training and test sets [12]. However, a LOSO-CV is not always feasible if the number of subjects in the experiments is either too small or too large [8]. Few users lead to an unrealistic view of the model performance. On the contrary, too many subjects increase the computational

TABLE I: HAR studies following a sliding window data segmentation and random training/test split validation approach.

Authors	Year, (Citations)	Journal/Conference	(Impact Factor)
Khalifa et al. [14]	2017, (159)	IEEE Transactions on Mobile Computing	(6.1)
Micucci et al. [15]	2017, (481)	Applied Sciences	(2.8)
San-Segundo et al. [16]	2018, (101)	Engineering Applications of Artificial Intelligence,	(7.8)
Wang et al. [17]	2018, (29)	Smart Health	(5.1)
Mutegeki and Han [18]	2020, (268)	ICAIC	-
Ni et al. [19]	2020, (20)	Sensors	(3.8)
Gupta [20]	2021, (74)	International Journal of Information Management Data Insights	-
Li et al. [21]	2021, (9)	UBICOMP 2021	-
Mekruksavanich and Jitpattanakul [22]	2021, (196)	Sensors	(3.8)
Bouchabou et al. [23]	2021, (24)	Communications in Computer and Information Science	-
Gómez Ramos et al. [24]	2021, (19)	Sensors	(3.8)
Zimelman and Keefe [25]	2021, (14)	PLOS ONE	-
Yan et al. [26]	2022, (14)	IEEE International Conference on Bioinformatics and Biomedicine	-
Wang et al. [27]	2022, (28)	IEEE Sensors Journal	(4.3)
Huang et al. [28]	2022, (40)	IEEE Transactions on Mobile Computing	(6.1)
Luo et al. [29]	2023, (21)	IEEE Transactions on Mobile Computing	(6.1)
Wu et al. [30]	2023, (10)	Knowledge-Based Systems	(8.1)
Garcia-Gonzalez et al. [31]	2023, (15)	Knowledge-Based Systems	(8.1)

complexity making the use of a model even impractical [8]. Group K-Fold CV<sup>1</sup>, a generalization of LOSO, is a valid and straightforward approach to ensure an unbiased evaluation strategy. In this case, the data samples are grouped by a third-party parameter which can be defined per use-case basis, e.g., collection date, subject-id, etc. Grouped partitions ensure that data samples corresponding to the same group are not represented in both testing and training sets.

### III. (RE)CURRENT PRACTICES IN HAR

The methodological issues that lead to accuracy overestimation is an ongoing practice within the HAR community, as presented in Table I. Some of those studies applied traditional machine learning algorithms for HAR [14]–[17], [25]. The most common algorithms are Decision Trees (DT), K-Nearest Neighbours (kNN), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), Hidden Markov Models (HMM), and Multi Layer Perceptrons (MLP). They use different datasets, e.g. UniMiB-SHAR [15], HHAR [39], containing 3-axial accelerometer and/or gyroscope data collected using wearables or smartphones while people performing different activities. All these studies used sliding windows for data segmentation with different degrees of overlap. Then, they evaluated their models following the standard K-Fold CV, and some of these studies also applied LOSO evaluation [15]–[17]. The results with LOSO show a significant drop in performance. However, such a drop is attributed solely to the variability on the way that different subjects perform the same activities, while the bias because of statistical dependence between consecutive windows and random training/test splits is overlooked. The bias introduced is independent of the input features (Fig. 2), feature extraction, and the normalization technique (Fig. 3).

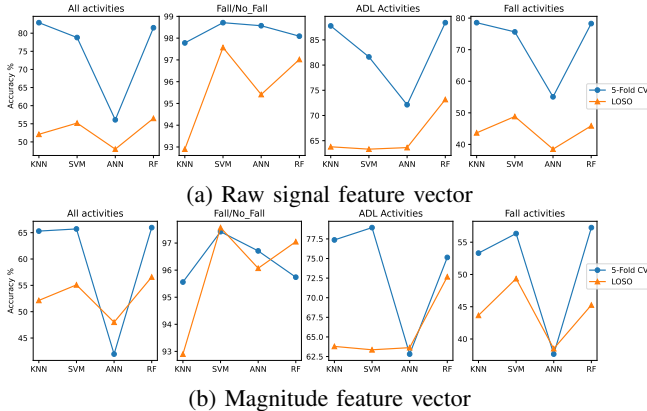


Fig. 2: 5-fold CV vs LOSO: reported accuracy comparison from Micucci et al., [15]

Other studies rely on Deep Learning approaches for HAR [18]–[22]. The most common approaches are CNNs, LSTMs, or the combination of both. The datasets used in

<sup>1</sup>[https://scikit-learn.org/stable/modules/cross\\_validation.html#cross-validation-iterators-for-grouped-data](https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation-iterators-for-grouped-data)

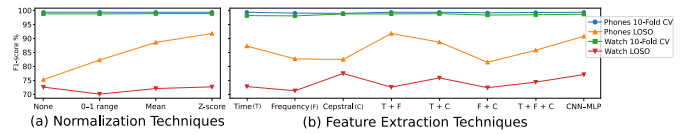


Fig. 3: Reported F1-Score of a RF classifier from San-Segundo et al. [16].

this studies are UCI-HAR [40], WISDM [41], which also contains 3-axial acceleration and gyroscope data. They use sliding windows and evaluate their models using random training/validation/test splits, K-Fold CV, and LOSO. The results also show a significant drop in performance using LOSO with respect to random partitioning and standard K-Fold methods. It is important to note, in [22], that the performance is overestimated using both, overlapping and non-overlapping sliding windows (Fig. 4).

The works of Bouchabou et al. [23] and Gómez Ramos et al. [24] use a different type of data, binary motion and contact sensors, from the CASAS benchmark dataset. Specifically, the Aruba and Milan datasets are used in those studies. Both approaches applied overlapping sliding windows of different sizes and random data partitioning with different ratios. Although both works use different approaches their results are comparable, f-score above 95%. In [23], the authors claim a “better generalization” obtained by means of a random shuffle before splitting, overlooking the negative effect of random splits after sliding windows data segmentation.

Yan et al. [26] propose a HAR model based on GNNs. They used data from the MHEALTH, PAMAP2 and their own dataset TND-A-HAR. The raw input data is transformed into a graph representation based on the Pearson correlation coefficient between the sensors channels signals. Each channel represents a graph vertex, and a correlation of a pair of vertices above 0.2 implies an edge. A GNN model is trained to encode the graph, followed by two fully connected layers as the final classifier. The authors report accuracies of 98.18% for PAMAP2, and 99.07% for MHEALTH. But those accuracies are overestimated due to the use of random training/test sets split, which is seen in the source code provided by the authors.

Wang et al. [27] compared the effect on performance of different data augmentation methods using UCI-HAR, USC-HAD [42], MotionSense [43] and MobiAct [44] datasets.

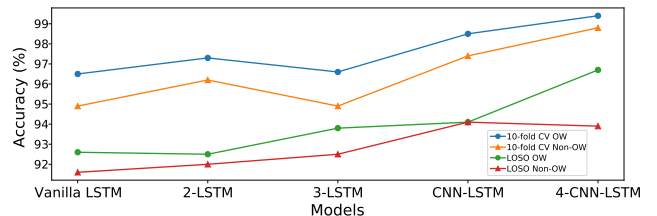


Fig. 4: Reported results from Mekruksavanich and Jitpatanakul [22].

They segmented the data using sliding windows with 12.5% to 50% overlap. They applied random training/test splits and additionally a subject-independent 5-fold CV for the MotionSense dataset. The F1-scores reported for all the experiments using randomly partitioned data were: UCI-HAR 98.28, MotionSense 99.35, USC-HAD 92.28 and MobiAct 98.32. In accordance to previous findings, in the experiments with a subject-independent CV for the MotionSense dataset, the F1-score dropped from 99.35 to 92.10.

Huang et al. [28] proposed Channel-Equalization-HAR, a variation of the normal CNNs. They used UCI-HAR, OP-PORTUNITY [45], UniMiB-SHAR, WISDM, PAMAP2, and USC-HAD datasets. The data was segmented using sliding windows of different sizes with 30%, 50%, 78% overlap. Then, they applied a random train/test split using a 7:1:2 ratio. The reported F1-score for UCI-HAR was 97.12%, similar to the ones reported in [22], [27]. Likewise, the reported accuracy on the WISD dataset was 99.04%, even higher than the one reported in [20] which followed the biased approach. The same can be observed for the USC-HAD dataset with a reported F1-score of 98.93%, higher than reported in [27] which also has the problem. With the PAMAP2 dataset, the reported F1-score is 92.18% which is similar to our own experiments using random training/test set splits, shown later in Section IV.

Luo et al. [29] proposed a Binarized Neural Network for HAR, that moves the computation to the edge. They used the Radar HAR dataset [46], UCI-HAR and UniMiB-SHAR datasets, applying sliding windows data segmentation followed by a random training/test split, in a 80:20 ratio for the Radar HAR dataset, and a 70:30 ratio for the UCI-HAR and UniMiB-SHAR datasets. The F1-score reported for the Radar HAR dataset is 98.6%. The reported F1-score for the UCI-HAR dataset is 98.1%, similar to the ones reported in [22], [27], [28] which followed the same biased approach. The reported F1-score for the UniMiB-SHAR dataset is 93.3%, even higher than the one reported in [15] which is also overestimated.

Wu et al. [30] proposed a spatio-temporal LSTM model using data from a pedal wearable device attached to the shoe’s tongue area. The approach combines a GNN model for the spatial features and a LSTM for the temporal patterns to recognize five different activities. A sliding window of 200 samples was used for data segmentation. The segmented data was randomly split into training/test sets. The reported results show a perfect F1-score of 1.0 for the Sitting and Down the Stairs activities, 0.96 and 0.97 for Standing and Walking respectively, and 0.83 for Up the Stairs. Once again, the followed approach shows overestimated results.

Garcia-Gonzalez et al. [31] used their own dataset containing accelerometer, gyroscope, magnetometer and GPS data from smartphones. They evaluated different traditional ML algorithms: SVM, DT, MLP, NB, k-NN, RF and Extreme Gradient Boosting (XGB). Data segmentation and feature extraction were performed using sliding windows from 20 to 90 seconds with 1 second step size, i.e., at least 95%

overlap. Then, a stratified k-fold CV is used to evaluate the different models. While keeping similar distribution of the samples per class, this method does not prevent that consecutive windows are assigned to two different folds. Hence, the reported accuracy, 92%, is overestimated.

#### IV. UNBIASED MODEL EVALUATION

This section shows that the same approach can lead to a considerable difference in accuracy depending on the data segmentation and training/test sets partition strategies.

##### A. Datasets

We evaluate whether the problem described in section II affects in the same way datasets with different data modality. We used the **MILAN** dataset [47] from the CASAS benchmark dataset collection<sup>2</sup>, which contains binary data from motion and contact sensors, and the **PAMAP2** [34] and **MHEALTH** [33] datasets, which contain accelerometer, gyroscope and magnetometer data from wearable on-body sensors.

a) **MILAN**: The sensors mounted in this smart home testbed environment included 28 motion, 3 door contact and 2 temperature sensors. The activities’ “start” and “end” are annotated. Samples which fall outside these markers have been assigned the “*other*” class. This dataset is highly imbalanced, the “*other*” being the dominant class.

b) **PAMAP2**: This dataset contains data collected from 9 subjects doing 12 different physical activities, using IMUs attached to the wrist, chest and ankle while performing everyday, household and sport activities. Each sensor includes two accelerometers, one gyroscope and one magnetometer producing 3-axial data at a sampling rate of 100Hz.

c) **MHEALTH**: This dataset contains data of 10 volunteers performing 12 physical activities. The sensors were placed at subjects’ chest, right wrist and left ankle. The data comprise 3-axis accelerometer, gyroscope and magnetometer signals collected at a sampling rate of 50Hz. The sensor placed at the chest also provides 2-lead ECG measurements, but those data points were not used in the experiments.

##### B. Data segmentation and feature extraction

We perform data segmentation using a fixed-size sliding windows approach, proposed in [35] and expanded in [36].

a) **MILAN**: We used a window of  $k$  sensor events because binary/motion sensors do not fire a sample at a constant rate. The window size was 30 with a step-size of 1, based on empirical evaluation as presented in [35], [48]. The windows were created based on collection date to facilitate the independence between training and test sets during model evaluation. The last sensor event in the window defines the label and the preceding events in the window define its context. Following this approach we can have a prediction every time a new sensor event arrives, achieving a near-real time recognition. Then, feature vectors are calculated for each window following the approach presented in [35], [48].

<sup>2</sup>CASAS benchmark dataset collection: <http://casas.wsu.edu/datasets/>

We transformed the *hour-of-day* and *day-of-week* to sine and cosine pairs to capture the equidistant relation between time-based cyclical values.

b) **PAMAP2**: This dataset was segmented using sliding windows of 5.12 seconds with 1 second shift, following the approach of its original publication [34]. Since the data was sampled at 100Hz, the windows span 512 sample readings with step-size of 100 samples. This data was used to train a classification model based on GNNs. Therefore, the training data was transformed to a graph representation where the vertices correspond to the different channels of the sensors’ signals. The edges are defined by means of the Pearson’s Correlation Coefficient between the channels, where a correlation threshold above 0.2 implies an edge between two channels.

c) **MHEALTH**: Following the protocol in [40], [49], we segmented the data using a sliding window of 2.56 seconds with 50% overlap. Since this dataset was sampled at 50Hz, the windows have 128 samples with 64 samples overlap. For feature creation, we first transformed the window data into a graph representation in the same way as described for PAMAP2 dataset. Then, the activity graphs were used to train a GNN-based classification model.

### C. Classification models and evaluation strategies

To measure the effects of the biased approach using classification models of different nature, we trained a RF and a GNN-based classifiers. The RF classifier uses binary sensor data, and the GNN-based model uses on-body IMU sensors.

a) **MILAN**: After feature extraction, the model was evaluated using the 5-fold CV approach. First, we partitioned the data randomly using the *StratifiedKfold* class, from the scikit-learn python library [50], with the *shuffle* parameter set to *True*. Then, in a second experiment, we used the *StratifiedGroupKfold* CV scheme [50]. This scheme splits the data into folds with non-overlapping groups, preserving the percentage of samples per class. In our case, the groups were determined by the *collection\_date* of the raw data samples.

b) **PAMAP2 and MHEALTH**: Most of the presented studies followed a conventional Deep Learning approach for HAR, e.g., CNN, LSTM or a combination thereof. To check if the overestimation bias affects other Deep Learning models, we trained a 3-layer GNN implemented using the Graph Convolution presented in [51], followed by two fully connected layers and softmax layer for final classification. We split our segmented data in a 6:2:2 ratio used for training, validation, and final model evaluation, respectively. We first partitioned the data randomly, and later on using the *StratifiedGroupKfold* approach, determined by *subject\_id*. We did the same for both, PAMAP2 and MHEALTH datasets. In the case of PAMAP2, subjects 101 and 107 were used for validation, subjects 103 and 105 for testing, and the remaining subjects for training. For MHEALTH dataset, subjects 6 and 10 were used as validation set, subjects 2 and 9 for testing and the rest for training. After hyper-parameter optimization and finding the best parameters combination, the model was updated with the entire training data, including the training

and validation subsets. The model was evaluated using the hold-out test set.

### D. Results and discussion

Table II shows that *window-based data segmentation* and *random data splits* lead to misleading results.

TABLE II: Classification performance comparison on MILAN, PAMAP2 and MHEALTH datasets.

Partitioning	MILAN (RF)		PAMAP2 (GNN)		MHEALTH (GNN)	
	b. acc	f1-score	b. acc	f1-score	b. acc	f1-score
Random (biased)	93.53	86.74	89.36	90.19	98.00	97.99
Grouped (unbiased)	55.59	58.49	81.59	81.73	81.59	81.73

a) **MILAN**: The performance with random splits is higher in all sixteen activities (Fig. 5), in concordance to Bouchabou et al. [23] and by Gómez Ramos et al. [24].

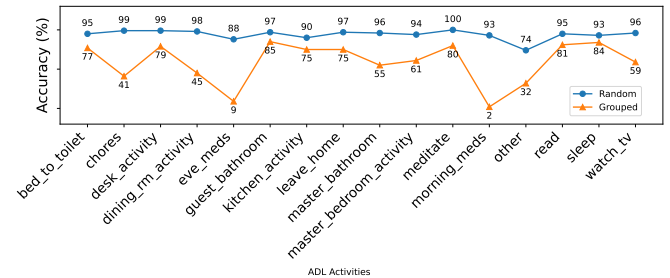


Fig. 5: Accuracy obtained for each activity on the Milan dataset.

These results show that in both experiments the minority classes (*morning\_meds*, *evening\_meds*) are misclassified. The reason is because those activities occur in the same room, triggering the same set of sensors. However, it is important to point out that the biased model classified these activities with an accuracy over 88%. On the contrary, the accuracy on the same classes with the unbiased approach is under 10%. Similarly, the same effect occurs with the majority class, “other”. While the biased model produced an accuracy of 74% on the “other” class, the corrected model just obtains a 32% accuracy on this pseudo activity.

b) **PAMAP2 and MHEALTH**: The results obtained with the GNN-based classification model on PAMAP2 and MHEALTH datasets also show a performance overestimation when the model is trained and evaluated on randomly partitioned data. In the PAMAP2 dataset the accuracy and f1-score dropped  $\approx 9\%$ , when the data is partitioned following a group-based approach. In MHEALTH the performance decreased  $\approx 16\%$  (See Table II). These results confirm that the “perfect” accuracy reported by Yan et al. [26] is overestimated due to the windowing mechanism and random training/test set splits.

The confusion matrices for PAMAP2 and MHEALTH datasets are shown in Fig. 6a and 6b. They show that GNN-based classification models produce consistent results on both datasets. The corrected models misclassify the *standing*

## VI. CONCLUSIONS

Our study reviewed HAR works with approaches that lead to model performance overestimation to raise awareness of the HAR community of this ongoing problem. Due to publications with biased overestimated results, fair approaches, with correct unbiased methods, may be disregarded due to the erroneously perceived low accuracy. We described and explained the downside of using sliding windows for data segmentation and feature extraction, followed by a random k-fold cross validation for model evaluation. We identified previous studies, including present day publications, where the reported performance is overestimated. The findings suggest that this is a recurrent practice with negative effects that are often disregarded by HAR practitioners.

Importantly, we are not implying that sliding windows and k-fold cross validation should not be used in HAR. Those are well established methods whose usage has been empirically justified. However, the data samples assigned to each fold should not be chosen at random. Our experiments used different classification models types and datasets with different distribution, nature and characteristics, proving that the bias introduced by the discussed issue is independent of the data modality and the classification model. The performance drop with an unbiased evaluation is significant to the point where those highly overestimated models would become impractical or even useless in real settings.

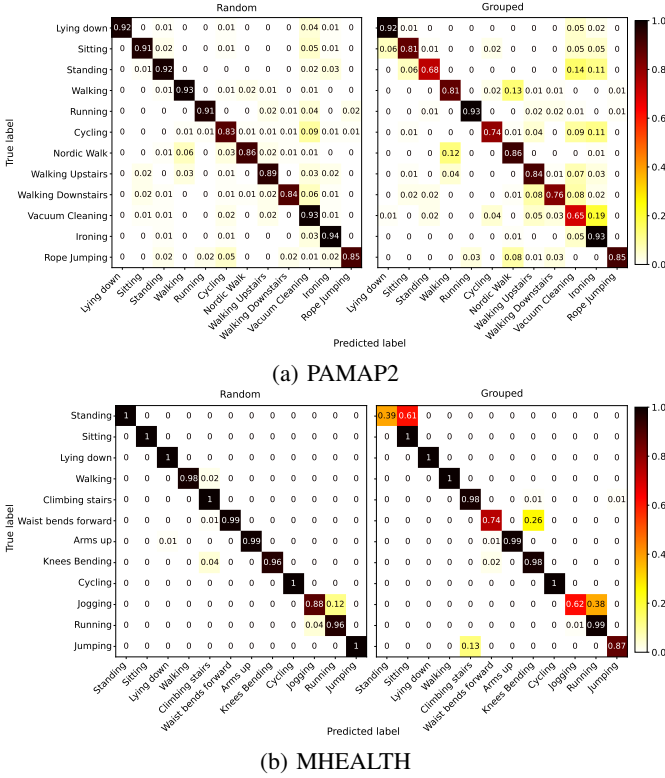


Fig. 6: Confusion matrices of the accuracy on PAMAP2 and MHEALTH datasets.

activity in both, PAMAP2 (68%) and MHEALTH (39%) datasets. Contrary, with the biased approach the accuracy on the *standing* activity increases to 92% for PAMAP2 and a perfect 100% for MHEALTH.

## V. LESSONS LEARNED

Model performance is affected by the data partition strategy that follows the window-based data segmentation. The results show that the performance of the models on randomly partitioned data is overestimated, regardless of the used dataset and classification model. The data imbalance negatively affects the performance of the classifier but its effect is more acute when the independence between training and test sets are guaranteed. Conversely, it may be unnoticed following a biased approach. If data segmentation is performed using a windowing mechanism, the independence between training and test sets, or between folds in CV, must be carefully considered. Hence, data partition must not be performed at random.

The group-based approach, used in this work, is a good alternative for guaranteeing an unbiased model evaluation. Splitting the data by a third-party parameter, chosen per use-case basis, gives the flexibility to create unbiased scenarios for evaluation even for single-subject datasets.

## REFERENCES

- [1] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, “Accurate activity recognition in a home setting,” in *Proc. of the 10th int. conf. on Ubiquitous computing*, 2008, pp. 1–9.
- [2] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.
- [3] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
- [4] Y. Chen and Y. Xue, “A deep learning approach to human activity recognition based on single accelerometer,” in *IEEE int. conf. on systems, man, and cybernetics*. IEEE, 2015, pp. 1488–1492.
- [5] N. Y. Hammerla, S. Halloran, and T. Plötz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” *arXiv preprint arXiv:1604.08880*, 2016.
- [6] F. J. Ordóñez and D. Roggen, “Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, p. 115, 2016.
- [7] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, “Deep learning models for real-time human activity recognition with smartphones,” *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.
- [8] N. Y. Hammerla and T. Plötz, “Let’s (not) stick together: pairwise similarity biases cross-validation in activity recognition,” in *Proc. of the 2015 ACM int. joint conf. on pervasive and ubiquitous computing*, 2015, pp. 1041–1051.
- [9] A. Deghani, T. Glatard, and E. Shihab, “Subject cross validation in human activity recognition,” *arXiv preprint arXiv:1904.02666*, 2019.
- [10] A. Deghani, O. Sarbishei, T. Glatard, and E. Shihab, “A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors,” *Sensors*, vol. 19, no. 22, p. 5026, 2019.
- [11] A. Jordao, A. C. Nazare Jr, J. Sena, and W. R. Schwartz, “Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art,” *arXiv preprint arXiv:1806.05226*, 2018.

- [12] D. Gholamiangonabadi, N. Kiselov, and K. Grolinger, "Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection," *IEEE Access*, vol. 8, pp. 133 982–133 994, 2020.
- [13] M. A. R. Ahad, A. D. Antar, and M. Ahmed, "Iot sensor-based activity recognition," *IoT Sensor-based Activity Recognition*. Springer, 2020.
- [14] S. Khalifa, G. Lan, M. Hassan, A. Seneviratne, and S. K. Das, "Harke: Human activity recognition from kinetic energy harvesting data in wearable devices," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1353–1368, 2017.
- [15] D. Micucci, M. Mobilio, and P. Napolitano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.
- [16] R. San-Segundo, H. Blunck, J. Moreno-Pimentel, A. Stisen, and M. Gil-Martín, "Robust human activity recognition using smartwatches and smartphones," *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 190–202, 2018.
- [17] S. Wang, G. Zhou, Y. Ma, L. Hu, Z. Chen, Y. Chen, H. Zhao, and W. Jung, "Eating detection and chews counting through sensing mastication muscle contraction," *Smart Health*, vol. 9, pp. 179–191, 2018.
- [18] R. Mutegeki and D. S. Han, "A cnn-lstm approach to human activity recognition," in *Int. conf. on Artificial Intelligence in Information and Communication (ICAIIIC)*. IEEE, 2020, pp. 362–366.
- [19] Q. Ni, Z. Fan, L. Zhang, C. D. Nugent, I. Cleland, Y. Zhang, and N. Zhou, "Leveraging wearable sensors for human daily activity recognition with stacked denoising autoencoders," *Sensors*, vol. 20, no. 18, p. 5114, 2020.
- [20] S. Gupta, "Deep learning based human activity recognition (har) using wearable sensor data," *Int. Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100046, 2021.
- [21] J. Li, Z. Wang, Z. Zhao, Y. Jin, J. Yin, S.-L. Huang, and J. Wang, "Tribogait: A deep learning enabled triboelectric gait sensor system for human activity recognition and individual identification," in *Adjunct Proc. of the 2021 ACM Int. Joint conf. on Pervasive and Ubiquitous Computing and Proc. of the 2021 ACM Int. Symposium on Wearable Computers*, 2021, pp. 643–648.
- [22] S. Mekruksavanich and A. Jitpattanakul, "Lstm networks using smartphone data for sensor-based human activity recognition in smart homes," *Sensors*, vol. 21, no. 5, p. 1636, 2021.
- [23] D. Bouchabou, S. M. Nguyen, C. Lohr, B. Leduc, and I. Kanellos, "Fully convolutional network bootstrapped by word encoding and embedding for activity recognition in smart homes," in *Int. Workshop on Deep Learning for Human Activity Recognition*. Springer, 2021, pp. 111–125.
- [24] R. Gómez Ramos, J. Duque Domingo, E. Zalama, and J. Gómez-García-Bermejo, "Daily human activity recognition using non-intrusive sensors," *Sensors*, vol. 21, no. 16, p. 5270, 2021.
- [25] E. G. Zimbelman and R. F. Keefe, "Development and validation of smartwatch-based activity recognition models for rigging crew workers on cable logging operations," *Plos one*, vol. 16-5, p. e0250624, 2021.
- [26] Y. Yan, T. Liao, J. Zhao, J. Wang, L. Ma, W. Lv, J. Xiong, and L. Wang, "Deep transfer learning with graph neural network for sensor-based human activity recognition," *arXiv preprint arXiv:2203.07910*, 2022.
- [27] J. Wang, T. Zhu, J. Gan, L. L. Chen, H. Ning, and Y. Wan, "Sensor data augmentation by resampling in contrastive learning for human activity recognition," *IEEE Sensors Journal*, vol. 22, no. 23, pp. 22 994–23 008, 2022.
- [28] W. Huang, L. Zhang, H. Wu, F. Min, and A. Song, "Channel-equalization-har: a light-weight convolutional neural network for wearable sensor based human activity recognition," *IEEE Transactions on Mobile Computing*, 2022.
- [29] F. Luo, S. Khan, Y. Huang, and K. Wu, "Binarized neural network for edge intelligence of sensor-based human activity recognition," *IEEE Transactions on Mobile Computing*, vol. 22, no. 3, pp. 1356–1368, 2023.
- [30] H. Wu, Z. Zhang, X. Li, K. Shang, Y. Han, Z. Geng, and T. Pan, "A novel pedal musculoskeletal response based on differential spatio-temporal lstm for human activity recognition," *Knowledge-Based Systems*, vol. 261, p. 110187, 2023.
- [31] D. Garcia-Gonzalez, D. Rivero, E. Fernandez-Blanco, and M. R. Luaces, "New machine learning approaches for real-life human activity recognition using smartphone sensor-based data," *Knowledge-Based Systems*, p. 110260, 2023.
- [32] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, 2012.
- [33] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in *Int. workshop on ambient assisted living*. Springer, 2014, pp. 91–98.
- [34] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th int. symposium on wearable computers*. IEEE, 2012, pp. 108–109.
- [35] N. Krishnan and D. Cook, "Activity recognition on streaming sensor data," *Pervasive and mobile computing*, vol. 10, pp. 138–154, 2014.
- [36] B. Quigley, M. Donnelly, G. Moore, and L. Galway, "A comparative analysis of windowing approaches in dense sensing environments," *Multidisciplinary Digital Publishing Institute Proc.*, vol. 2, no. 19, p. 1245, 2018.
- [37] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.
- [38] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, pp. 40 – 79, 2010.
- [39] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition," in *Proc. of the 13th ACM conf. on embedded networked sensor systems*, 2015, pp. 127–140.
- [40] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. of the 21th int. European symposium on artificial neural networks, computational intelligence and machine learning*, 2013, pp. 437–442.
- [41] G. M. Weiss, K. Yoneda, and T. Hayajneh, "Smartphone and smartwatch-based biometrics using activities of daily living," *IEEE Access*, vol. 7, pp. 133 190–133 202, 2019.
- [42] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Protecting sensory data against sensitive inferences," in *Proc. of the 1st Workshop on Privacy by Design in Distributed Systems*, 2018, pp. 1–6.
- [43] M. Zhang and A. A. Sawchuk, "Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proc. ACM conf. on ubiquitous computing*, 2012, pp. 1036–1043.
- [44] C. Chatzaki, M. Padiaditis, G. Vavoulas, and M. Tsiknakis, "Human daily activity and fall recognition using a smartphone's acceleration sensor," in *Information and Communication Technologies for Ageing Well and e-Health: 2nd Int. Conf., ICT4AWE 2016, Revised Selected Papers 2*. Springer, 2017, pp. 100–118.
- [45] D. Roggen, A. Calatroni, M. Rossi, T. Holleczeck, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkel, A. Ferscha *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 7th int. conf. on networked sensing systems (INSS)*. IEEE, 2010, pp. 233–240.
- [46] F. Luo, S. Poslad, and E. Bodanese, "Kitchen activity detection for healthcare using a low-power radar-enabled sensor network," in *IEEE Int. conf. on Communications (ICC)*. IEEE, 2019, pp. 1–7.
- [47] D. J. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods of information in medicine*, vol. 48, no. 05, pp. 480–485, 2009.
- [48] T. Wang and D. J. Cook, "Multi-person activity recognition in continuously monitored smart homes," *IEEE Transactions on Emerging Topics in Computing*, 2021.
- [49] D. Anguita, A. Ghio, L. Oneto, F. X. Llanas Parra, and J. L. Reyes Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," *Journal of universal computer science*, vol. 19, no. 9, pp. 1295–1314, 2013.
- [50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [51] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proc. of the AAAI conf. on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4602–4609.