Bias in Human Activity Recognition: An investigation of different Evaluation techniques on Model Performance

Dimitrios Tsiamouras
dimitrios.tsiamouras@student.uva.nl
University of Amsterdam
Amsterdam, The Netherlands

ABSTRACT

Data contamination, defined as the unintended inclusion of test data in the training process, poses a significant challenge in Machine Learning (ML), particularly within Human Activity Recognition (HAR) domains. Despite its prevalence due to common HAR methodologies, data contamination is often overlooked and inadequately addressed, leading to misleading and unrealistic performance metrics in HAR models. Existing studies have seldom focused on this issue, creating a notable research gap in the field.

This paper investigates the impact of transitioning from conventional, biased evaluation techniques, such as Random Split Train-Test Split (RSTTS), to advanced, unbiased methods like Leave-One-Subject-Out Cross-Validation (LOSO-CV) and Chronological Split (CS) to mitigate bias in HAR model evaluation. Through extensive comparative tests and analyses, the study quantifies the bias introduced by traditional approaches and examines its impact on model accuracy. The findings reveal a potential accuracy drop exceeding 10% when adopting more objective evaluation techniques, emphasizing the need for unbiased assessment. A metric based on JS divergence is proposed to measure accuracy overestimation caused by biased evaluation methods, contributing to more reliable and applicable HAR systems.

KEYWORDS

Human Activity Recognition, Data Contamination, Machine Learning, Evaluation Techniques, Leave-One-Subject-Out Cross-Validation, Chronological Split, Bias Quantification, Model Accuracy, JS Divergence

GITHUB REPOSITORY

https://github.com/dimtsiamouras/measuring_bias_thesis

1 INTRODUCTION

Human Activity Recognition (HAR) is the practice of classifying human activities by analyzing sensor data. It is a continuously evolving tool, utilized in various domains, from healthcare to human-computer interaction and assistive technologies. In the realm of HAR research a standardized pipeline has emerged [7, 14], developed and refined through numerous studies, quickly becoming the de facto framework for conducting such experiments. However, and despite their widespread adoption, these practices keep being unquestioningly followed, perpetuating their inherent imperfections across different studies.

One major problem with recurring HAR methodology is data contamination. This occurs when training and testing data are not successfully separated, despite the fundamental assumption that data samples are Independently and Identically Distributed (i.i.d). More specifically, recurrent practices predominantly utilize sliding windows for data segmentation, followed by random train/test splits or k-fold Cross-Validation. [7, 16] Although splitting the available data in a random manner makes sense in various ML contexts, in HAR specifically, when paired with sliding window techniques, unforeseen issues arise.

First of all, in the case of overlapping windows, adjacent windows with shared features are utilized for both training and testing the model. This issue is not exclusive to overlapping windows, but persists even in the case non-overlapping windows. Here, for long-lasting, repetitive activities similar samples will arise in the data [5, 9].

Consequently, the data segmentation step, which is fundamental for constructing a robust HAR model, violates the assumption of i.i.d. data. The subsequent evaluation of the resulting models is, therefore, unrealistic, as the metrics are inadvertently inflated due to the high degree of similarity between the samples used for training and testing. However, this critical issue is often overlooked, and there is a profound scarcity of literature discussing its inherent limitations [6, 10]. Notably, recent studies, such as the survey presented by Tello et al. [16], highlight this issue by showing a significant amount of recently published papers continue to report results based on biased evaluation methodologies.

Several practices have been developed in order to mitigate the introduction of bias in classification models. Two of the most commonly used approaches are LOSO-CV and GroupK-Fold Cross-Validation (GKF-CV). The former entails the creation of folds based on Subject IDs, guaranteeing the independence between training and test splits. GKF-CV is built on the same principle, but generalizes it further by splitting the sets based on some other parameter (e.g., experiment date, experiment ID). However, these additional parameters are very scarcely included in the data. The datasets utilized in this study did not contain such information beyond the Subject IDs. Consequently, it was not feasible to apply GKF-CV. To address this limitation, a new split method,called the Chronological Split, was developed. This approach involves sorting records chronologically and utilizing the first percentage of them for training, while the rest are used to test the model. This method mirrors how real-life HAR applications would sequentially process and utilize data, providing a more practical evaluation of model performance.

The main issue lies in the sporadic utilization of these methods, as they are not a staple of the aforementioned standard HAR procedures. In addition, even in cases where these techniques are employed, the mitigation of bias results in lower scoring models, since testing data is less similar and more representative of truly unforeseen examples. A surface-level comparison of the performance

between biased and unbiased models gives a false impression of inferiority of the latter. In consequence, papers reporting inflated evaluation metrics receive more favorable treatment and are more likely to be published compared to lower-scoring, unbiased ones.

In light of the challenges introduced by data contamination in HAR research, and recognizing the inherent limitations of recurrent evaluation practices, the need to explore alternative methods that mitigate bias and provide objective assessments of HAR models becomes imperative. Given these considerations, the central focus of this paper can be encapsulated in the following Research Question:

To what extent do more objective data segmentation and splitting techniques, particularly LOSO-CV and CS, impact the performance and accuracy estimation of HAR models, and which methods can be devised to quantify the amount of bias inadvertently introduced by such procedures?

To ensure the successful and appropriate investigation of this overarching question, I will delve into a set of subquestions, each aiming to answer different aspects of model evaluation and comparison. These subquestions are as follows:

1. How much does the performance of HAR models change when transitioning from standard data segmentation (e.g., random train/test split with/without random shuffling, simple k-fold CV) to more deliberately chosen methods (e.g., LOSO-CV, CS)?

2. What methods can be employed to quantify the amount of bias and the subsequent drop in accuracy when transitioning from biased to unbiased evaluation techniques?

3. How do LOSO-CV and CS compare in terms of their effectiveness in mitigating bias and providing a more realistic accuracy estimation of HAR models? Additionally, is it possible to attribute differences in their performance to the variability in how different subjects perform the same activities, as proposed by other works?

The ultimate objective is that, by addressing these research questions, the efficacy of different evaluation methodologies in reducing data contamination will be better understood. Additionally, I aim to quantify the extent to which performance changes when employing the aforementioned unbiased techniques, potentially yielding a metric that predicts the drop in accuracy due to bias reduction. Lastly, I strive to shed light on the contextual applicability of different evaluation methods, providing insights into how they are to be selected and implemented for effective model assessment in different scenarios.

2 RELATED WORK

This section provides an exploration of existing literature surrounding HAR methodologies, with the particular aim to highlight challenges that emerge in these contexts. Emphasis is placed on the presentation of prominent techniques in the HAR pipeline, while simultaneously identifying inherent flaws. The objective of this discourse is to elucidate gaps in previous research, showcase my sources of inspiration and underline the profound necessity of the work at hand.

The methodological exploration begins with an examination of recent review articles, which offer a comprehensive overview of prevalent methodologies in HAR research. As highlighted by Straczkiewicz et al. [14], the most commonly employed data segmentation approach involves employing the sliding window technique paired with a random train/test split. The authors explicitly state that the intent behind this combination is for the data to be separated in such a manner that the test data will offer a realistic representation of the model's accuracy on unforeseen samples. It is also worth pointing out that, even though the review also mentions instances of cross-validation, these are limited to Subject-independent K-Fold and Leave-One-Out Cross-Validation (LOO-CV). Similar findings are corroborated by Gupta et al. [7], who provide a more extensive description of evaluation methodology, briefly mentioning LOSO-CV without delving into contextual motivations for its use or its potential impact on model accuracy.

Expanding on the insights provided from narrative reviews, further exploration transitions into specific studies that utilize the methodology of interest. For instance, Wang et al. [17] conducted experiments utilizing both a random split and, notably, a 5-fold CV with sliding windows of various overlaps across various benchmark datasets. This subject-independent CV lead to a significant drop in F1-score, the cause of which was not investigated further. Nevertheless, as suggested by Tello et al. [16], this severe overestimation in accuracy and the subsequent drop is consistent with potential data contamination introduced from biased segmentation techniques, underlining the imperative need for thorough investigation.

In another study, Wang et al. introduced an eating detection method based on data encompassing 8 different activities [18]. Therein they conducted tests and comparisons using three distinct approaches: simple 10-fold CV, self-test [where only the subject's own samples are used to form the dataset] and LOSO-CV. Crucially -perhaps unsurprisingly to a discerning reader, the LOSO-CV performed significantly worse than the other two methods, with the F1-score for User 4 being merely 55.4%. Unfortunately, the authors attributed this discrepancy solely to sensor positioning and the way in which the subject performed the activity, without further investigation. Tello et al. [16], however, disagree with this interpretation, stressing the fact that this is a symptom of substantial bias behind the model. According to them, this bias occurs due to "statistical dependence between consecutive windows and random training/test splits" and is a systematic problem that is repeated due to the prevalent HAR methodology.

This sentiment is shared by Hammerla and Plötz [9], who highlight the flawed assumption of user-dependent performance serving as an upper boundary in HAR research. To further support their claim, they demonstrate that, even when employing train-test splits, the assumption of independent and identical distribution of samples often does not hold. However, since this assumption is crucial for cross-validation to serve its purpose, its violation inherently results in data contamination and, consequently, in inflated accuracy metrics. Tello et al. [16] expand upon this, showcasing how overlap or lack-thereof has practically no effect on mitigating data contamination, as adjacent segments are inherently statistically dependent. Dehnghani et al. [6] also highlight challenges in prior papers as these pertain to subject-dependent cross-validation. In their discussion they present solid arguments against these practices, clearly pushing for subject independent approaches. Nevertheless, these studies do not present LOSO-CV as a panacea, as they underscore

possible limitations of LOSO-CV, particularly concerning datasets containing either too many or too few subjects.

This pattern of perpetuating flawed methodology is further exemplified in the work of Tapia et al. [15]. Here, they employed overlapping sliding windows with both subject-dependent and subject-independent cross-validation, with the latter performing nearly 40% worse. In a manner similar to [18], this finding was attributed solely to the characteristics of individual subjects. A similar example of inadvertent contamination not being properly addressed can be observed in the study by Ni et al. [13] in their efforts to address class imbalance. In their methodology they utilized the standard overlapping sliding window technique followed by a 6:2:2 train/-validation/test split. However, a critical and fundamental oversight occurred during the standardization process, where data normalization was performed before partitioning the dataset. Specifically, the normalization parameters where obtained from the entirety of the dataset, rather than exclusively from the training set, inadvertently contaminating the data [16]. The correct approach would entail learning the normalization parameters solely from the training data and then normalizing each set accordingly based on those.

A comprehensive analysis of recent papers utilizing the aforementioned problematic methodology is presented by Tello et al. [16] in an effort to emphasize the need for a paradigm shift in HAR research. By documenting the widespread use of flawed, bias-inducing techniques and how they influence model assessment across various datasets and machine learning algorithms, the authors underscore the importance of addressing these flaws in order to progress the field of HAR. Significantly, many of the studies highlighted in this analysis are highly cited, indicating that these flawed methodologies are not just common but are also propagated by influential research. This widespread issue in reputable papers exacerbates the problem, as it sets a precedent for future research to follow the same biased practices. Similarly, Braganca et al. [4] proposed explainable methods, such as utilizing the SHAP framework, as a way to uncover bias problems inherent in validation strategies. Their findings further support the idea that common practices tend to overestimate model accuracy, thereby reinforcing the need for methodological reform.

In summary, the studies mentioned above showcase how standard methodology persists across various papers and studies, underscoring the perpetuation of its inherent challenges in HAR research. Crucially, the cited studies highlight clear gaps in the existing literature, which the paper at hand aims to address.

3 METHODOLOGY

In this section, a comprehensive break-down of the methodology will be presented. This spans from data collection and preprocessing to the ML model selection and the implementation of the distinct experimental setups.

3.1 Data Collection

All datasets used in this study are publicly accessible and widely utilized in HAR research.

a) PAMAP2: This dataset consists of data collected from 9 subjects engaged in a wide array of everyday, household and sports activities. During these activities, participants wore Inertial Measurement Units (IMUs) attached to their wrists, chests, and ankles to capture motion-related data at 100Hz. Each IMU sensor comprises two accelerometers, one gyroscope and one magnetometer, capturing 3-axial data. Notably, in contrast to most datasets, PAMAP2 includes heart-rate and temperature measurements. For a more comprehensive description of the dataset, the reader can refer to the work of Arif et al. [1].

b) MHEALTH:. This dataset features data collected from 10 participants performing various physical activities in an out-of-lab environment. Subjects wore sensors positioned on their chest, wrist and ankle, recording 3-axial information. However, unlike PAMAP2's sampling rate of 100Hz, MHEALTH contains data sampled at 50Hz. For a detailed description of the dataset, refer to the work of Banos et al. [2].

3.2 Exploratory Data Analysis & Preprocessing

Upon acquisition, datasets underwent rigorous pre-processing to uncover their unique characteristics, feature distributions and address potential issues. This stage was crucial in order to ensure high quality of available data and convert them into a form suitable for analysis.

a) PAMAP2: Exploratory Data Analysis (EDA) revealed a significant number of missing values in the dataset, necessitating imputation. Specifically, missing values were replaced with the mean value of the corresponding feature for each activity. In keeping with the suggestions of the dataset's providers, Orientation-labeled columns and transient-activity records (coded with 0) were excluded from the analysis. Subject 9 emerged as the singular outlier, displaying unique activity distribution patterns, as well as peculiar heart-rate and temperature values, thus justifying its exclusion. After these adjustments, the dataset comprised 1,936,481 samples across 21 features, involving 8 subjects and 11 activities, with a balanced distribution of samples per subject and activity. Features related to heart-rate and temperature were retained due to their observed correlation with activity types, potentially enhancing a model's discrimination capabilities of different activities.

b) MHEALTH:. Conversely, the MHEALTH dataset, devoid of missing values, required minimal intervention aside from addressing class imbalance, particularly with activity 0 being more prevalent than others. A resampling strategy was employed to ensure equal representation of all activities for each subject, mitigating biases in subsequent analyses. The resulting dataset consists of 373,915 samples with 23 features, exhibiting a balanced distribution per activity and subject.

3.3 Data Segmentation

In this section, the different approaches used for segmenting the data into training, validation, and test sets are described.

a) ***Random Shuffled Train/Validation/Test Split (RSTTS) :*** This is the most commonly employed method of data segmentation in ML tasks. For the purposes of this study, it involved randomly shuffling the entire dataset and then partitioning it into training, validation, and test sets in a 6:2:2 ratio. This is accomplished by utilizing the train_test_split function from the Scikit-Learn python

library. However, for HAR tasks, where temporal information between adjacent data windows is crucial, this random segmentation may lead to data contamination.

b) ***LOSO-CV:*** LOSO-CV involves using data samples from one subject as the test set, while using data from the remaining subjects for training. This process is repeated for each subject, generating a number of folds equal to the number of unique subjects. This approach reflects a realistic scenario for HAR models, where the model is applied to subjects not present in the training data. For this study, this segmentation split is accomplished by grouping records based on subject id and subsequently assigning one subject's records to the test set, while the rest are allocated to the train set. (See Figure 1)

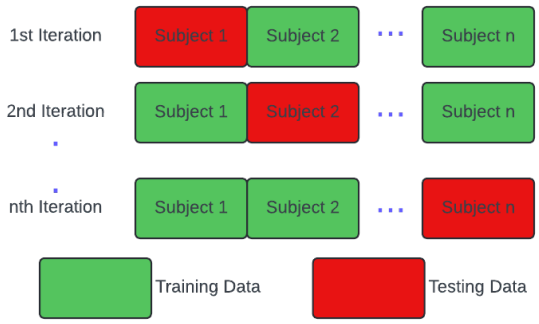

Figure 1: Leave-One-Subject-Out Cross-Validation

c) ***Chronological Split:*** Data segmentation in a chronological manner involves sorting records based on their timestamp and using the earliest records for training and the subsequent records for testing. Similar to LOSO-CV, this method mimics real-world HAR applications, where models are tasked with classifying newer data instances. In practice, this is accomplished by ordering the samples in a chronological fashion, utilizing the timestamp feature, and allocating the first 80% of the data to train the model and the rest to test it. (See Figure 2)

Note: Before fitting the datasets to a ML model, standard scaling is applied. This is performed after segmentation, using parameters learnt exclusively from the train set, rather than the whole dataset.

Figure 2: Chronological Split

3.4 Classification Model

As discussed above, the primary purpose of this paper is to investigate the extent to which more advanced data segmentation techniques impact the performance of HAR models compared to standard randomly shuffled train/test splits. For that reason, rather than implementing HAR pipelines from scratch, this research leveraged existing HAR applications known to utilize the latter technique. The key modification lies in adjusting the data segmentation method so as to study the effects of this change on various performance metrics. The author kindly acknowledges the contributions of Gupta [8], Bouchabou et al. [3] And Micucci et al. [12] for providing their implementations, greatly expediting this research.

Table 1: Codes Employed from Previous Studies

Author(s)	Year	Citation
Bouchabou et al.	2020	[3]
Micucci et al.	2017	[12]
Gupta	2021	[8]

a) ***Convolutional-Recurrent Neural Network (CRNN)***

This HAR pipeline was taken from the work of Gupta [8]. The model architecture is a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), specifically leveraging a Gated Recurrent Unit (GRU). The fusion of CNNs and RNNs is a remarkable innovation for the purpose of sequential data analysis as it leverages CNNs inherent ability to capture spatial hierarchies and RNNs capability to effectively model temporal dependencies. The resulting CRNN architecture showcases significant learning abilities at processing sequential data streams for the purposes of classification.

Delving into the specifics of this architecture, implemented within a Sequential model structure, the network begins with a series of TimeDistributed Conv1D layers. The initial Conv1D layer, consisting of 32 filters with a kernel size of 3, employs the Rectified Linear Unit (ReLU) activation function. Subsequently, a MaxPooling1D layer with a pool size of 2 is applied, followed by another Conv1D layer with 128 filters and the same kernel size and activation function. Dropout regularization is incorporated after each convolutional layer, with dropout rates of 0.08 and 0.1, respectively, to mitigate overfitting. The output of these convolutional layers is then flattened into a one-dimensional array using a Flatten layer. The sequential processing then transitions to the recurrent phase, where two GRU layers with 64 units each are stacked. The first GRU layer returns the full sequence of outputs, while the second one returns only the output at the last time step. Finally, a Dense layer with a softmax activation function is employed to produce class probabilities based on the model's output. This architecture is tailored for the classification task at hand, leveraging both the spatial hierarchies captured by the CNN layers and the temporal dependencies modeled by the RNN layers to effectively process sequential data and make accurate predictions.

b) ***Fully Convolutional Network (FCN) with Natural Language Processing (NLP)***

The FCN is a variant of CNNs characterized by its distinct absence of dense layers and its sole reliance on convolutional layers

for the processes of feature extraction and classification. In the work of Bouchabou et al. [3], each sensor event is treated as a word and activity sequences as sentences. NLP techniques are then employed in order to learn vector representations of sensor events, facilitating the treatment of HAR as a sequence classification task.

Feature extraction is a crucial aspect of this approach, involving the conversion of word sequences into numerical representations suitable for ML. This process begins with the segmentation of data into activity sequences representing distinct periods of activity. These activity sequences are then transformed into word sequences, conceptualizing sensor events as "words" within the context of NLP. Following this transformation, the word sequences undergo frequency indexing, assigning numerical representations to each word. Subsequently, utilizing sliding windows, the indexed sequences are split into smaller segments, which capture temporal patterns. Finally, the FCN architecture is employed for activity classification within each temporal segment. Leveraging the features extracted from the indexed sequences, the ML module outputs the most probable activity label for each window.

The FCN architecture this paper utilized, as implemented by Bouchabou et al. [3], comprises hierarchical convolutional layers. The initial layer applies 128 with a kernel size of 8, followed by batch normalization and ReLU activation. Subsequent layers further enhance the process of feature extraction, employing 256 and 128 filters with kernel sizes of 5 and 3, respectively.

The feature maps generated by the convolutional layers are aggregated by means of a global average pooling layer. To mitigate the effects of overfitting, dropout regularization with a rate of 0.5 is applied. Finally, the output layer consists of a dense layer with softmax activation, generating class probabilities based on the extracted features.

c) ***Random Forest Classifier (RFC)***

The RFC is an ensemble learning method renowned for its robustness and effectiveness in handling a variety of classification tasks. This model is an aggregation of multiple decision trees, typically trained using different subsets of the data. Each tree in the forest outputs a class prediction, and the class with the majority votes among the trees is chosen as the final prediction.

For this study, the RandomForestClassifier from the Scikit-Learn library was employed, configured with 100 trees (n_estimators=100). The implementation employed, originally coded in MATLAB as part of the work by Micucci et al. [12], was adapted to Python for compatibility with this study's framework.

d) ***Support Vector Machine (SVM)***

The SVM is a classification technique known for its effectiveness in high-dimensional spaces and its robustness to overfitting. SVM aims to find the hyperplane that best separates the data into different classes by maximizing the margin between the closest points of the classes (support vectors). Similarly to the RFC implementation, the SVM implementation, taken from the same work by Micucci et al. [12], was originally coded in MATLAB and later converted to Python using the SVC class from the Scikit-Learn library.

The SVM model was instantiated with an RBF kernel (kernel='rbf') for non-linear classification and probability estimates (probability=True). The use of an RBF kernel enables the SVM to handle non-linear relationships between features, while probability estimates allow the model to output probabilities for each class, enabling more nuanced predictions and decision-making.

3.5 Experimental Pipeline

Utilizing the aforementioned HAR models, distinct experimental environments were established for each combination with the data segmentation techniques of interest. The only deviation between environments using the same pipeline lies in the data segmentation process. All other parameters and procedures have remained intact across these environments. Therefore, the the treatment of train/validation/test sets is uniform, regardless of the method through which they were generated. In this section, the setup for each pipeline will be detailed.

a) ***CRNN***

The initial stage in this environment involves the segmentation of the time-series data into sliding windows. Each window is generated from 200 frames with a stride of 50, resulting in a 75% overlap between consecutive windows. Through an iterative process, windows are generated from each dataset solely utilizing records that belong to the same activity. Subsequently, the windows are stored in distinct lists of frames, based on their respective datasets.

Following this, by iterating through each list of frames, relevant features and their activity labels are extracted. These features and labels are stored in separate lists, denoted as the feature list (X) and the target list (y) respectively. The aligned lists are subsequently converted into Numpy arrays and Scikit-Learn's StandardScaler is fit on the train set. Using the scaling parameters obtained, all 3 datasets are transformed, mitigating the influence of varying scales.

The datasets are then reshaped into a format suitable for processing by the CRNN model. Specifically, the sets are reshaped into a structure of 4 time steps (n_steps), each consisting of 50 data points (n_length) and 21 features (n_features). With the conclusion of the preprocessing phase , the datasets can be fit into the ML model.

The CRNN model, defined earlier, is trained using the Adam optimization algorithm is chosen coupled with Keras' categorical cross-entropy loss function. The model consumes batches of 128 (batch_size) and optimizes its parameters through 8 epochs.

b) ***FCN with NLP***

This environment begins with the partitioning of activities for each dataset. This is done by means of the devised 'segment_activities' function, which detects transitions between distinct activities, denoted by a change in the activity label. More specifically, an empty list is initialized to store the segmented activity sequences, followed by a Boolean index (potentialIndex) that marks the positions where activity transitions occur between adjacent samples. Subsequently, the function iterates through the identified positions, segmenting the activity sequences by extracting subsets of the dataset that lie between consecutive transitions.

Following the segmentation process, the segmented sequences are transformed into textual representations. The 'sequencesToSentences' function iterates over each sequence, invoking the 'generate_sentence' which iterates over the relevant columns and generates a sentences with this information. Additionally, the corresponding activity label for each sequence is captured, serving as the target label.

Further processing of the sentences involves tokenization using the Keras Tokenizer, coupled with the removal of punctuation and special characters. By fitting the Tokenizer on the generated sentences, a mapping between words and indices is established. These indexed sentences are obtained using the 'texts_to_sequences' method, converting textual representations into integer sequences.

After tokenization, the segmented activity sequences are processed using a sliding window approach. A window size of 50 frames and a step size of 1 are employed, resulting in a 98% overlap. Implemented iteratively, the sliding window function extracts consecutive windows from each sequence, generating input-output pairs that are compatible with the FCN model. Padding is applied to ensure uniformity in input sizes, while output labels are converted into numpy arrays.

Finally, the processed input-output pairs are fed into the FCN model. Parameters such as vocabulary size, embedding dimension, hidden dimension, and output dimension are specified accordingly. Specifically, the model is defined with a vocabulary size obtained from the Tokenizer, an embedding dimension of 100, a hidden dimension of 128, and an output dimension of 25. Training entails 3 epochs with a batch size of 20.

*c) **RFC and SVM***

The experimental setup for the RFC and SVM models follows a similar methodology. Initially, the datasets are partitioned into overlapping sliding windows using the 'sliding_window' function. Each window is generated with a window size of 100 frames and a step size of 50 frames, resulting in a 50% overlap between consecutive windows. The 'sliding_window' function iterates through the data, creating windows of the specified size and appending them to a list. This process is applied separately to the train and test datasets.

The next phase involves extracting features and labels from each window. Specifically, for each window in the training set, features are extracted from all columns except the last two, which contain the activity labels and subject IDs. These features and their corresponding labels are used to train the chosen machine learning (ML) algorithm. The process is encapsulated within a for loop that iterates over each window in the training set.

The for loop iterates over each window in the training set, performing the following sequence: it extracts features and labels from the window, scales the features using the StandardScaler, and then fits the chosen model (RFC or SVM) on the scaled features and labels.

3.6 Evaluation metrics

The primary evaluation metric utilized in this study is Accuracy, primarily chosen due to its prevalent usage in comparable research in prior HAR research. Accuracy provides a comprehensive measure of the model's overall performance by quantifying the proportion of correctly classified instances across all activity classes. Given the balanced nature of the datasets, with equal representation of activities and subjects, Accuracy serves as a robust indicator of model effectiveness, as it equally weights the contribution of each class to the overall performance. Furthermore, the widespread adoption of accuracy facilitates comparisons with models from different studies, providing a standardized metric for assessing overall performance across various methodologies.

3.7 Assessment of Train/Test Set Similarity

To assess the similarity between the train and test sets resulting from each segmentation technique, the Jensen-Shannon divergence (JS) is employed. This method measures the similarity between two probability distributions and is a symmetric and smoothed version of the Kullback-Leibler divergence [11]. The JS divergence value ranges from 0 to 1, where a greater value signifies higher dissimilarity between the distributions in question.

In this study, the JS divergence is calculated for every train and test set combination, providing a measure of how similar the created sets are. This assessment indicates how closely the data the model was trained on matches the data on which it is evaluated. The hypothesis is that a higher similarity between the two sets would likely result in higher model performance. Thus, connecting the JS divergence measure with model accuracy should elucidate how similarity impacts the accuracy of the ML models.

4 RESULTS

In this section, the results of the conducted experiments are presented and analyzed. Each experimental environment yielded a model trained on a train set derived from a specific data segmentation split. Subsequently, the model's predict functionality was employed to generate predictions, which were then compared with the actual labels. The evaluation metrics of interest were acquired, offering insight into the model's performance on data that it had not been directly trained on.

To ensure robustness and account for random variations, the reported results are the averages of three executions, each with different random states. This approach was taken to mitigate the effects of random chance and provide a more reliable assessment of model performance.

By collecting these metrics for each distinct environment, specifically for each data segmentation split for every ML model, comparisons can be made. This approach facilitates the investigation into the effects of these specific changes on the model's ability to provide accurate predictions. The detailed numerical results are provided in Table 2.

4.1 Comparison of Evaluation Metrics

Both the CRNN and the FCN models exhibited high performance in the RSTTS for both datasets, achieving accuracy levels similar to those reported in prior studies. For instance, the CRNN reached 97.71% for PAMAP2 and 95.57% for MHEALTH, while the FCN achieved 99.12% and 95.32%, respectively. This consistency with previous research establishes a strong baseline for evaluating alternative segmentation techniques.

However, when subjected to LOSO-CV, both models showed a noticeable decrease in accuracy. The FCN model's accuracy for PAMAP2 dropped from 99.12% to 85.45%, and the CRNN model's accuracy fell to 88.03%. The CS approach further reduced the models' accuracy, with the CRNN dropping to 81.22% and the FCN to 76.65% for PAMAP2. In the MHEALTH dataset, the CRNN and FCN models' accuracy decreased to 70.11% and 69.21%, respectively.

The Random Forest (RF) model also demonstrated high accuracy in the RSTTS, but faced significant challenges under the LOSO-CV split, with accuracy dropping to 55.43% for PAMAP2 and 58.33% for MHEALTH. The CS split proved even more challenging, reducing accuracy to 44.80% for PAMAP2 and 46.23% for MHEALTH.

Similarly, the Support Vector Machine (SVM) model performed well in the RSTTS but struggled with the LOSO-CV split, achieving 65.32% accuracy for PAMAP2 and 69.33% for MHEALTH. The CS split further reduced its performance to 59.51% for PAMAP2 and 63.31% for MHEALTH.

Model	Dataset	Random Split	LOSO-CV	Chronological
CNN GRU	PAMAP2	97.71%	88.03%	81.22%
	MHEALTH	95.57%	89.48%	70.11%
FCN LSTM	PAMAP2	99.12%	85.45%	76.65%
	MHEALTH	95.32%	89.55%	69.21%
RF	PAMAP2	97.32%	55.43%	44.80%
	MHEALTH	95.42%	58.33%	46.23%
SVM	PAMAP2	97.93%	65.32%	59.51%
	MHEALTH	96.69%	69.33%	63.31%

Table 2: Accuracy across Techniques and Datasets

4.2 Train/Test Set Similarity Results

In addition to the accuracy metrics of each environment, the JS divergence score for the train and test sets was measured. Table 3 presents the JS divergence results for each segmentation technique across all datasets and models.

Model	Dataset	Segmentation	JS Divergence
CRNN	PAMAP2	Contaminated	0.38
	PAMAP2	LOSO	0.62
	PAMAP2	Chronological	0.70
	MHEALTH	Contaminated	0.26
	MHEALTH	LOSO	0.35
	MHEALTH	Chronological	0.42
FCN LSTM	PAMAP2	Contaminated	0.08
	PAMAP2	LOSO	0.16
	PAMAP2	Chronological	0.18
	MHEALTH	Contaminated	0.12
	MHEALTH	LOSO	0.18
	MHEALTH	Chronological	0.22
RF, SVM	PAMAP2	Contaminated	0.30
	PAMAP2	LOSO	0.53
	PAMAP2	Chronological	0.60
	MHEALTH	Contaminated	0.23
	MHEALTH	LOSO	0.29
	MHEALTH	Chronological	0.31

Table 3: JS Divergence across Techniques and Datasets

It is evident that the JS divergence varies significantly across different segmentation techniques and datasets. For the CRNN model, the highest divergence was observed with the Chronological split, indicating the greatest dissimilarity between the train and test sets. The contaminated RSTTS, on the other hand, resulted in the lowest JS divergence for both datasets. The same pattern holds true for the FCN model.

Both the RF and SVM models consistently exhibit identical patterns in JS divergence across various segmentation techniques, as they share the same segmentation pipeline. Specifically, for the PAMAP2 dataset, both models show a JS divergence of 0.60 with the Chronological split, 0.53 with LOSO, and 0.30 with the contaminated RSTTS. Similarly, for the MHEALTH dataset, the JS divergence values for both models are 0.31, 0.29, and 0.23 with their respective segmentation techniques.

These results highlight the varying degrees of similarity between the train and test sets created by each segmentation method.

4.3 Relationship Between JS Divergence and Model Accuracy

The choice of data segmentation technique significantly influences JS divergence, a metric indicating the dissimilarity between the training and testing data. Across all test environments, the standard split consistently yielded the lowest JS divergence values. Investigating the relationship between the similarity of the train and test sets and model accuracy is crucial for comprehending the impact of information leakage on the model's generalization capacity.

The findings reveal a clear correlation between the similarity of the train and test sets and the accuracy of the models. Higher JS divergence values, associated with the Chronological and LOSO splits, corresponded to lower accuracy, suggesting that significant differences between training and testing data challenge the models' ability to generalize. Conversely, the standard approach of the RSTTS consistently yielded lower JS divergence scores while recording higher accuracy, indicating that when the training and testing data are more similar, the models perform better.

Figure 3 illustrates this relationship, showing a clear linear trend between JS divergence and model accuracy across different segmentation techniques. Higher JS divergence values are associated with lower accuracy, while lower JS values correlate with higher accuracy. This empirical evidence emphasizes how the segmentation approach affects the similarity between the train and test sets, which, in turn, impacts the evaluation metrics.

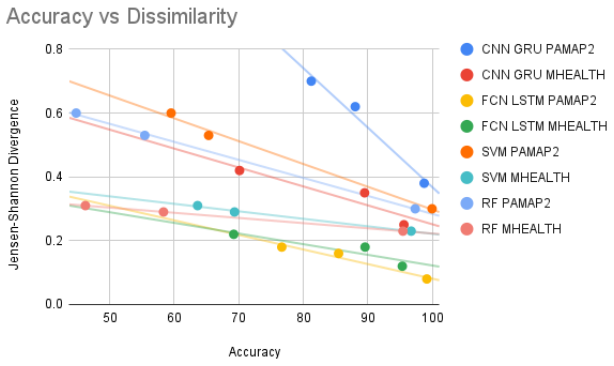

Figure 3: Relationship between JS Divergence and Model Accuracy

5 DISCUSSION

The purpose of this section is to analyze the experiment results previously presented, focusing on comparing evaluation techniques,

assessing model performance, and exploring factors influencing model accuracy in human activity recognition (HAR). Through these analyses, the aim is to provide insights into the complexities of evaluating HAR models and offer directions for future research.

5.1 Impact of Advanced Evaluation Techniques

As illustrated in the Results section, the transition from the standard evaluation technique of RSTTS to the deliberately chosen LOSO-CV or CS approach has a significant impact on model accuracy. Even though the entire pipeline remains the same, the choice of evaluation technique significantly affects the metrics that are reported. This finding is consistent with previous studies, but the fact that the code implementation was derived from studies that reported this inflated accuracy gives merit to the point that papers that report lower accuracy, simply due to using a more objective approach, might be seen as inferior and receive less attention.

Additionally, based on the observed results, it is reasonable to predict that transitioning from standard to advanced techniques could lead to an accuracy drop of over 10%. Such a prediction aligns with the substantial differences in performance metrics observed in this study and underscores the importance of acknowledging and addressing the biases inherent in traditional evaluation methods. By quantifying the potential magnitude of the accuracy drop, researchers can better understand the trade-offs involved in adopting more objective evaluation techniques and anticipate the adjustments needed to account for these changes in reported accuracy.

It is important to note that while Random Forest and SVM models are not considered state-of-the-art in the field of HAR, they are still widely used due to their simplicity and ease of implementation. The results indicate that these models are even more affected by changes in data segmentation techniques, suggesting a higher degree of bias in training when using random splits. This further implies a reduced ability to learn meaningful patterns from the training data compared to neural network models. Such findings emphasize the need for careful selection of evaluation techniques, especially when working with simpler models that may be more prone to performance degradation due to biased training.

5.2 Comparison of LOSO and Chronological Split Accuracy

Table 2 enables a direct comparison between the LOSO-CV and Chronological split methods. It is evident that LOSO consistently outperforms the Chronological split in terms of accuracy. This suggests that subject-specific variability significantly influences model performance.

However, it is important to recognize that in practical applications, one often cannot choose between LOSO and Chronological split methods. In real-world scenarios, chronological splits are inevitable because predictions are typically made based on historical data. This aligns with situations where models need to predict current activities using past data.

Temporal changes in data, known as "data drift" or "concept drift," introduce additional challenges. When significant temporal changes occur, the JS divergence between the training and test sets increases, leading to lower accuracy. Therefore, suggesting that a Chronological split should be used in situations with temporal changes might be misleading. Instead, addressing data drift requires continuously updating the model with new data or employing specialized techniques designed to handle concept drift.

The decision to use a LOSO split depends on the goal of the study. LOSO is valuable when evaluating how well a model generalizes across diverse subjects, particularly when individual variability is crucial. It helps assess the model's performance in predicting activities for new subjects without their own historical data.

In contrast, the Chronological split might be used when the focus is on understanding how a model performs over time, but it is essential to acknowledge that this method may not always provide the best accuracy due to the effects of data drift. Therefore, while chronological splits are inevitable in practice, ensuring the model can handle temporal changes effectively is crucial.

5.3 Impact of Subject-Specific Variability and Temporal Changes

The comparison between the LOSO and Chronological split methods highlights the influence of both subject-specific variability and temporal changes on HAR model performance. As discussed in the Related Work section, the accuracy drop with the LOSO split is often attributed to the unique ways in which subjects perform activities, affecting the model's learning and prediction accuracy.

However, the results show that the LOSO split outperforms the Chronological split, suggesting that temporal factors such as shifts in activity patterns over time, variations in external conditions, or subjects' increasing familiarity with the activities have a more significant impact on model performance than previously assumed. This indicates that while individual differences in activity execution are important, temporal dynamics play a crucial role in determining model accuracy.

These findings underscore the complexity of factors contributing to accuracy drops when transitioning from standard to more advanced evaluation techniques. It becomes clear that attributing drops in accuracy solely to individuality may be an oversimplification of the observed discrepancies. Future studies must recognize the importance of addressing both subject-specific variability and temporal changes in HAR model development and evaluation. By adopting a holistic approach, researchers can create more robust models capable of reliably recognizing activities across diverse subjects and evolving environmental conditions, thereby enhancing the practical applicability of HAR systems.

5.4 JS Divergence as a Predictor for Dataset Changes

Beyond its application in evaluating the impact of different segmentation techniques, JS divergence can also serve as a valuable tool for assessing changes within the dataset itself. This scenario occurs when the segmentation split remains constant, but other factors such as the distribution of activities or the manner in which subjects perform activities change. For example, if subjects become more accustomed to the activities over time or external conditions influence their performance, these variations will be reflected in the JS divergence value.

By measuring the JS divergence between the training and test data during the training phase, researchers can predict how changes

in the dataset will impact model accuracy during the inference phase, assuming all other variables remain constant (e.g., the same ML model, segmentation technique, and dataset sizes are used). This predictive capability is crucial for understanding and anticipating the effects of dataset evolution on model performance. When employed, it can be especially useful in scenarios with limited resources and time, helping to avoid unnecessary retraining and ensuring more efficient deployment strategies.

In this way, JS divergence serves as a versatile metric, not only highlighting differences between training and testing sets due to segmentation techniques but also providing insights into the inherent variability within the dataset.

6 CONCLUSION

In summary, this study addresses the pervasive issue of data contamination in HAR research and investigates the impact of transitioning from standard evaluation techniques to more sophisticated methodologies. By comparing recurrent, traditional practices like RSTTS with more carefully chosen techniques such as LOSO-CV and CS, this research provides valuable insights into the biases inherent in standard approaches and the effectiveness of alternative methods in mitigating these biases.

The findings highlight the significant impact of evaluation techniques on model accuracy, with the transition to LOSO-CV or CS resulting in notable changes in performance metrics. Specifically, a potential accuracy drop of over 10% was observed when transitioning to more objective evaluation techniques, even with the remaining HAR pipeline remaining the same. This significant drop underscores the effect of inherent biases in standard approaches, which often lead to overestimated accuracy in HAR models. For simpler ML models like Random Forest and SVM, this drop in accuracy can be even more exaggerated, indicating a higher degree of bias when using random splits (e.g., RSTTS).

▹ **The following paragraph was changed, based on 5.2. Check if ok** Comparative analyses reveal that LOSO consistently outperforms CS in terms of accuracy, suggesting that subject-specific variability significantly influences model performance. However, in practical applications, the choice between LOSO and CS is often constrained by the nature of the data and study objectives. In real-world scenarios, chronological splits are inevitable because predictions are typically made based on historical data, and temporal changes or "data drift" require specialized techniques. Therefore, while LOSO is valuable for evaluating model generalization across diverse subjects, temporal dynamics must be considered when using CS.

This study also highlights the relationship between JS divergence and model accuracy, demonstrating how data segmentation techniques impact the similarity between training and testing sets, thereby influencing model generalization capacity. Higher JS divergence values, associated with CS and LOSO splits, correspond to lower accuracy, suggesting that significant differences between training and testing data challenge the models' ability to generalize. Conversely, the Standard approach consistently yields lower JS divergence scores and higher accuracy, indicating that when training and testing data are more similar, the models perform better.

Beyond evaluation, JS divergence can be utilized by other researchers to assess the necessity of retraining models. An increase in JS divergence with the same data and pipeline suggests a probable drop in model performance. This predictive capability can save time and resources by indicating when retraining is necessary, thus ensuring efficient deployment strategies.

In conclusion, this research contributes to a deeper understanding of the complexities involved in HAR model evaluation and provides practical insights for selecting appropriate evaluation techniques. By quantifying the extent of bias introduced by standard procedures and proposing metrics to measure accuracy overestimation, we aim to encourage more objective and rigorous evaluation practices in HAR research. This, in turn, enhances the reliability and applicability of HAR systems in real-world scenarios. Future studies should continue exploring advanced evaluation methods and consider incorporating additional metrics, such as JS divergence, to further refine model assessment and development practices.

REFERENCES

[1] M. Arif and A. Kattan. 2015. Physical Activities Monitoring Using Wearable Acceleration Sensors Attached to the Body. *PLOS ONE* (2015). https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130851

[2] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga. 2014. mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications. *International Workshop on Ambient Assisted Living* (2014). https://orestibanos.com/paper_files/banos_iwaal_2014.pdf

[3] D. Bouchabou, S. Nguyen, C. Lohr, B. Leduc, and I. Kanellos. 2020. Fully Convolutional Network Bootstrapped by Word Encoding and Embedding for Activity Recognition in Smart Homes. *Sensors* (2020). https://arxiv.org/abs/2012.02300

[4] H. Bragança, J. G. Colonna, H. A. B. F. Oliveira, and E. Souto. 2022. How Validation Methodology Influences Human Activity Recognition Mobile Systems. https://www.mdpi.com/1424-8220/22/6/2360

[5] A. Dehghani, T. Glatard, and E. Shihab. 2019. Subject Cross Validation in Human Activity Recognition. https://arxiv.org/abs/1904.02666

[6] A. Dehghani, O. Sarbishei, T. Glatard, and E. Shihab. 2019. A Quantitative Comparison of Overlapping and Non-Overlapping Sliding Windows for Human Activity Recognition Using Inertial Sensors. *Sensors* (2019). https://www.mdpi.com/1424-8220/19/22/5026

[7] N. Gupta, S. Gupta, R. Pathak, V. Jain, and P. Rashidi. 2022. Human Activity Recognition in Artificial Intelligence Framework: A Narrative Review. https://link.springer.com/article/10.1007/s10462-021-10116-x

[8] S. Gupta. 2021. Deep Learning Based Human Activity Recognition (HAR) Using Wearable Sensor Data. *International Journal of Information Management Data Insights* 1, 2 (2021), 100046. https://doi.org/10.1016/j.jjimei.2021.100046

[9] N. Hammerla, S. Halloran, and T. Plotz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. https://arxiv.org/abs/1604.08880

[10] A. Jordao, A. C. Nazare Jr., J. Sena, and W. Robson Schwartz. 2018. Human Activity Recognition Based on Wearable Sensor Data: A Standardization of the State-of-the-Art. (2018). https://arxiv.org/abs/1806.05226

[11] J Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory* 37, 1 (1991), 145–151.

[12] D. Micucci, M. Mobilio, and P. Napoletano. 2017. UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones. *Applied Sciences* (2017). https://www.mdpi.com/2076-3417/7/10/1101

[13] Q. Ni, Z. Fan, L. Zhang, C. D. Nugent, I. Cleland, Y. Zhang, and N. Zhou. 2020. Leveraging Wearable Sensors for Human Daily Activity Recognition with Stacked Denoising Autoencoders. *Sensors* (2020). https://www.mdpi.com/1424-8220/20/18/5114

[14] M. Straczkiewicz, P. James, and J. Onnela. 2021. A Systematic Review of Smartphone-Based Human Activity Recognition Methods for Health Research. *npj Digital Medicine* (2021). https://www.nature.com/articles/s41746-021-00514-4#ref-CR110

[15] E. Tapia, S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman. 2007. Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor. (2007). https://www.researchgate.net/publication/221240724_Real-Time_Recognition_of_Physical_Activities_and_Their_Intensities_Using_Wireless_Accelerometers_and_a_Heart_Rate_Monitor

[16] A. Tello, V. Degeler, and A. Lazovik. 2024. Too Good To Be True: accuracy overestimation in (re)current practices for Human Activity Recognition. In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Biarritz, France, 511–517. https://doi.org/10.1109/PerComWorkshops59983.2024.10503465

[17] J. Wang, T. Zhu, J. Gan, L. Chen, H. Ning, and Y. Man. 2022. Sensor Data Augmentation by Resampling for Contrastive Learning for Human Activity Recognition. https://arxiv.org/abs/2109.02054

[18] S. Wang, G. Zhou, Y. Ma, L. Hu, Z. Chen, Y. Chen, H. Zhao, and W. Jung. 2018. Eating Detection and Chews Counting Through Sensing Mastication Muscle Contraction. *Smart Health* (2018). https://www.sciencedirect.com/science/article/pii/S2352648318300394