Analysis of online and offline classification algorithms for human activity recognition using IMU sensors

Brena Rodrigues Machado [1], Regis Pires Magalhães [1], Lívia Almada Cruz [1], Criston Pereira de Souza [1], César Lincoln Cavalcante Mattos [2], José Antônio Fernandes de Macedo [2]

[1]Universidade Federal do Ceará - Campus Quixadá (UFC)
63.902-580 – Quixadá – CE – Brazil

[2]Departamento de Computação – Universidade Federal do Ceará - Campus Pici (UFC)
60.440-900 – Fortaleza – CE – Brazil

{regismagalhaes,livia.almada,criston,jose.macedo}@ufc.br

brenarodrigues@alu.ufc.br, cesarlincoln@dc.ufc.br

Abstract. *Physical activity monitoring through machine learning, using data collected from wearable devices equipped with motion sensors and vital signs monitoring, such as heart rate, temperature, and blood oxygenation, has gained significant attention in sports and medical fields. This advancement enables real-time performance tracking and early detection of motor conditions. While offline classifiers achieve high accuracy, they cannot adapt to novel motion patterns; online (incremental) learners overcome this limitation. Although there are online learning algorithms, their application to Human Activity Recognition (HAR) remains limited. The challenge for offline and online approaches is generalizing activity detection based on time-series segments, relying solely on sensor data without additional information. This study analyzes the performance of offline and online algorithms for HAR by applying segmentation and feature extraction techniques and evaluating the adaptability of incremental learning models over time. The research follows a quantitative and prescriptive approach, employing various classifiers to address the problem of HAR. The results highlight the performance of the offline Autogluon Predictor applied to the PAMAP2 dataset, which the five best algorithms nearly tied and achieved average scores of 94% accuracy, 93% F1-score, 94% precision, and 94% recall, followed by CIF with lower results. For online learning the algorithms XGBoost Incremental performed best with 78% accuracy, 77% f1-score, 81% precision and 78% recall. While Adaptive Random Forest and BiLSTM also nearly tied in the online setting, achieving lower results compared to XGBI, this study shows promising results for both scenarios. It highlights that determining the most suitable learning paradigm (online or offline) is a decision for the data scientist, guided by the particularities of the problem and the necessary dynamism required for the scenario. By exploring different learning approaches for HAR and evaluating their effectiveness, this research contributes to the development of more adaptive and personalized systems with applications in health monitoring, sports, and medical diagnostics, fostering advancements in continuous and adaptive user data analysis.*

1. Introduction

The advancement of the Internet of Things (IoT) and the widespread use of smart wearable devices led to a large volume of sensory data transferred via streaming. As a result, human activity recognition (HAR) from temporal data has become a widely studied area of machine learning in recent years. Applications of HAR include monitoring the elderly and diseases that affect the motor system, and detecting physical activities [Helmi et al. 2021].

Machine learning techniques are used to recognize activity patterns to build a system capable of performing HAR. These techniques can be divided into two more general groups: online algorithms, also known as incremental, which learn continuously and process data in streaming format, and offline algorithms, which learn only once and require all available training data.

Given the context of physical activity recognition, each user will have slightly different movement patterns that offline learning might not detect effectively if trained on data from another user [Guo et al. 2022]. This happens due to differences in activity movements among users, making algorithms that can either be tailored to a specific user or adapt to detect variations between users more effective. Furthermore, offline learning requires a large amount of data and a large group of users to be effective for any user not seen by the trained model. These algorithms can be slower in training and prediction. On the other hand, they offer the benefit of consistent prediction efficiency for all learned classes due to their continuous access to the entire dataset. Conversely, a significant drawback is the necessity of full retraining to accommodate new classes or evolving intra-class patterns, a process that demands considerable time and effort.

Thus, online learning appears advantageous for this context due to its adaptability to user-specific data. As the user modifies their movement patterns, such as through training progression or response to medical treatment, the algorithm adapts accordingly. However, the field of incremental HAR is relatively new and faces a significant challenge: catastrophic forgetting, which occurs when the model forgets previously learned activities when adjusting to new activities [Bukhari et al. 2024].

Although online and offline learning techniques share conceptual similarities, they are designed for different learning scenarios and are not directly comparable. This study presents an exploratory evaluation of both approaches in the context of Human Activity Recognition (HAR), using the same dataset and experimental protocol. The main objective is to analyze how each approach performs under similar conditions, highlighting their respective strengths, limitations, and potential for real-world applications such as real-time recognition or batch analysis. Furthermore, the specific objectives are: (i) to select representative online and offline learning algorithms suitable for multivariate time series classification; (ii) to review the literature on online and offline classification techniques applied to Human Activity Recognition (HAR) using inertial sensor data; and (iii) to apply the selected algorithms to inertial sensor data for performance evaluation in the context of HAR. The following sections present the problem definition, related work, data description, and the methodology adopted in this study. Subsequently, the experiments and results are discussed, followed by potential threats to validity are addressed. Finally, the conclusion and directions for future work are presented.

2. Problem definition

HAR using machine learning algorithms is a method that has been increasingly studied due to its wide application in smart devices. This paper examines two classes of these algorithms: online and offline.

The conceptual differences between the techniques do not prevent them from performing the same task; however, adapting the data for each solution is necessary. The online algorithm trains itself by adapting to data arriving in streaming format, making it advantageous in dynamic scenarios, as it can learn new classes over time without requiring the complete dataset for training. Additionally, it is fast and lightweight. However, it faces the challenge of catastrophic forgetting. The offline learning method receives the complete training set, enabling it to capture patterns efficiently. However, it cannot incorporate new classes without performing a full retraining. Moreover, its training process is slower and more computationally expensive.

Both approaches use data from different individuals in the same training process. Thus, the challenge for both classes of algorithms lies in generalizing detection so that the algorithm can identify a specific activity from a segment of a multivariate time series or its representation without any additional information beyond the sensor data.

3. Related Works

Many studies have been developed in human activity recognition (HAR) in recent years, exploring different types of machine learning algorithms, including classifiers and predictors. As related works, this study focuses on classifiers employing traditional machine learning techniques, both incremental and non-incremental, for classifying time series data.

The work of [Tseng and Wen 2023] uses hybrid algorithms, combining different types of offline learning algorithms such as traditional ones, deep learning, and transfer learning for HAR using inertial sensor data. Furthermore, the system structure is defined by four steps, which are: 1) Network detection and data sampling from body area network system (BAN) and Wi-Fi network; 2) Data preprocessing where they proposed a new way of representing the data by transforming them into images, and then performed feature engineering with PCA (Principal Component Analysis) and mRMR (minimum Redundancy and Maximum Relevance); 3) Activity classification where 11 traditional machine learning classifiers were used, five neural networks for classification and four types of transfer learning algorithms making use of the pre-trained models; 4) Data generation through generative models for data augmentation and correction.

Similarly, the work of [Tahir et al. 2022] also used traditional offline machine learning algorithms for HAR, namely Random Forest, SVM-RBF, and AdaBoost. Like this work, [Tahir et al. 2022] used PAMAP2 as one of the datasets. The transformations applied to the data in the preprocessing stage include noise reduction and feature augmentation to obtain features of greater importance through statistical and stochastic transformers, and feature selection was performed using stochastic gradient descent (SGD).

Related to incremental learning of human activities, the work of [Liu et al. 2024] proposed a new framework for HAR that uses Kolmogorov-Arnold Networks (KAN) as a basis, adapted for incremental learning with the approach of including feature branching for specific tasks and a feature distribution layer. PAMAP2 was one of the employed datasets. The approach used for cross-validation in the PAMAP2 dataset was Leave-Two-Subjects-Out (LTSO), that is, leaving two participants out of the training, different from this work, which used the same cross-validation approach, Leave-One-Subject-Out (LOSO), for all machine learning techniques.

3.1. Comparative analysis

Table 1 presents important information about each related work. The works propose solutions to the same HAR problem. However, various techniques are used, such as incremental learning, traditional learning, neural network learning, and transfer learning. The works used similar metrics such as F1-score, accuracy, precision, and recall, which are widely used metrics. However, [Tseng and Wen 2023] used only one metric, accuracy, which can lead to classification bias by not considering the proportion of correct classifications of each class. Regarding the datasets used, all works used datasets from inertial sensors containing time series data from accelerometer, gyroscope, and magnetometer.

Table 1. Summary of the related works.

Article	Algorithms	Datasets	Metrics
[Tseng and Wen 2023]	11 Traditional classifiers, 5 neural networks for classification and 4 transfer learning	Private	Accuracy
[Tahir et al. 2022]	RandomForest, SVM-RBF and Adaboost	IM-WSHA, PAMAP-2, UCI HAR, MobiAct, and MOTIONSENSE	Precision, recall, e F1-score
[Liu et al. 2024]	iKAN	WISDM, MotionSense, MM-Fit, and PAMAP2	Accuracy and F1-score
This work	CIF, BILSTM and ARF Incremental XGBoost, Autogluon tabular	PAMAP2	Accuracy, F1-score, recall, precision

4. Data and Methods

Machine learning techniques were chosen for the experiment for time series classification, tabular data classification, and incremental classification of tabular data aimed at detecting physical activities. A well-known dataset for monitoring physical activities was also used. This section describes the data, the preprocessing performed, the classification techniques, and the cross-validation strategy employed.

4.1. Dataset

The dataset used in our experiments is the Physical Activity Monitoring (PAMAP2)[Reiss 2012], which consists of multivariate time series collected from 18 physical activities performed by 9 participants using 3 inertial sensors and a heart rate sensor. The sampling frequency of the motion sensors was 100 Hz, and the sampling frequency of the heart rate was approximately 9 Hz. Of those 18 activities, only the first 12 activities were performed by 8 users. Since the experiment methodology uses data from each user as test and validation data in cross-validation, it was decided to use only the first 12 activities and only 8 users who had all 12 activities in common. Those activities are basic activities like lying, sitting, standing, ironing clothes, vacuuming, ascending and descending stairs, walking, Nordic walking, riding a bicycle, running, rope jumping, watching TV, working on a computer, driving a car, folding laundry, house cleaning, and playing soccer. Each row in the data files represents a single instance of sensory data, labeled with a timestamp and an activity classification. The files contain 54 columns, including the timestamp, the ground-truth activity label, and 52 attributes derived from raw sensor readings. The dataset has a total of 3,850,505 instances. These sensors must be strategically placed on the body to capture the activities performed. In PAMAP2, the sensors were positioned on the wrist of the dominant arm, chest, and ankle of the dominant leg. It is important to note that the dataset contains missing values. Thus, data cleaning was carried out during the preprocessing phase.

4.2. Preprocessing

The data preprocessing involves three main tasks: data cleaning, segmentation, and feature extraction.

Data Cleaning. The dataset contains missing heart rate and temperature values due to the sampling difference between inertial and vital sensors. In addition, transient activity intervals, i.e., intervals when the user switches between activities, were removed. Transient activities are a problem because they do not represent an actual activity, and their inclusion in the training data would lead to the detection of a class that is irrelevant to the problem. Therefore, it is necessary to remove the transient activity data, concatenate the valid activities, and correct the timestamps. For missing data, artificial data must be inserted to fill gaps using linear interpolation. Filling in the gaps is essential because the timestamp information must remain linear for the model to interpret the activity.

Data Segmentation. Traditional machine learning algorithms perform best when trained with tabular data, i.e., each record refers to complete information about a class, but when the data is time series, several records need to be considered to identify a class. Thus, the processed data was segmented into non-overlapping 10-second windows with 1000 samples per segment. The non-overlapping strategy was chosen because the initial objective was to minimize preprocessing while classifying the incoming segments in a streaming setting. Each window is related to an activity, and time windows that had more than one activity were discarded. According to [Kwapisz et al. 2011], 10-second segments effectively provide a broad context while reducing variability caused by transient motion or sensor noise. Too short segments may be dominated by noise or capture only part of an activity, which could confuse classification algorithms. A 10-second interval helps smooth out the impact of momentary fluctuations, ensuring that the extracted features better represent the overall activity.

Feature extraction. The catch22 (CAnonical Time-series CHaracteristics) [Lubba et al. 2019] tool, a widely used and efficient method for feature extraction in time series analysis, was chosen due to its balance between computational efficiency and robustness in capturing diverse statistical and dynamical properties of the data.

Unlike more complex feature extraction methods, catch22 provides a curated set of 22 highly interpretable features optimized for general time-series classification tasks. It enables the transformation of each time-series segment into a single representative record, ensuring compatibility with traditional machine learning algorithms while maintaining the integrity of the temporal patterns. Given its proven effectiveness across multiple domains, the studies in [Valerio et al. 2024] and [Sousa et al. 2025] employed this tool for health monitoring and activity recognition, respectively, achieving remarkable results. This motivated us to adopt the same tool in our study.

4.3. Classification algorithms

This study evaluated nine classification algorithms to compare offline and online learning approaches for Human Activity Recognition (HAR) using data from inertial sensors. The Canonical Interval Forest (CIF) was selected because of its strong results in time series classification tasks and its ability to capture important temporal patterns, which are critical in HAR. For the offline tabular models, we used the top five algorithms identified by AutoGluon's Tabular Predictor, as it provides a robust and diverse set of high-performing models without the need for manual tuning. On the online side, we included Incremental XGBoost to explore how a well-known batch learning algorithm would perform when updated incrementally with new data. We also used Adaptive Random Forest (ARF), which is specifically designed for streaming data and can handle changes in the data over time, something common in HAR scenarios. Lastly, BiLSTM was included due to its frequent use in HAR research and its ability to learn sequential patterns, even though in our case it was applied in batch mode. This combination of models allowed us to explore a wide range of techniques and better understand the strengths and limitations of each approach.

Canonical Interval Forest (CIF). The choice of the CIF algorithm was based on its ability to classify multivariate time series data by exploiting their features. Furthermore, it is based on the Time Series Forest (TSF) algorithm, known to perform well in classifying time series data. CIF stands out for randomly selecting intervals of different lengths and dimensions and extracting features from the chosen intervals using catch22 [Middlehurst et al. 2020] and 3 TSF features. Decision tree forests trained with subsets of the intervals and features are then constructed.

Autogluon Tabular Predictor. The use of Autogluon [Erickson et al. 2020] for automated machine learning model selection was guided by its ease of use and the ability to execute various types of advanced models automatically and optimally. This library stands out from other automated machine learning (AutoML) libraries in performing hyperparameter optimization and in performing ensemble and stacking of models [Erickson et al. 2020]. The five algorithms that achieved the best performance with AutoGluon were: ExtraTreesGini_BAG_L2, ExtraTreesEntr_BAG_L2, LightGBMXT_BAG_L2, NeuralNetFastAI_BAG_L2 and RandomForestGini_BAG_L2.

Incremental XGBoost. XGBoost was selected for its outstanding efficiency in classification tasks. Its operation includes the boosting technique that sequentially combines decision trees that are weaker models but can recognize patterns efficiently [Ohwosoro et al. 2024]. In addition, another attractive feature of this algorithm is its online batch learning functionality.

4.4. Cross-Validation

In this experiment, Nested Cross-Validation was conducted, using the Leave-One-Subject-Out (LOSO) Cross-Validation strategy to split the data. In nested cross-validation, an outer and an inner cross-validation are performed. The original dataset is split twice: the outer validation creates subsets to test the model, while the inner validation tunes the hyperparameters. When combined with LOSO, nested cross-validation generates subsets based on participant combinations, leaving two participants out of training at each iteration. The participant excluded in the outer validation is used as the test data, while the participant excluded in the inner validation serves as the validation data for hyperparameter optimization. The Figure 1 illustrates the process.

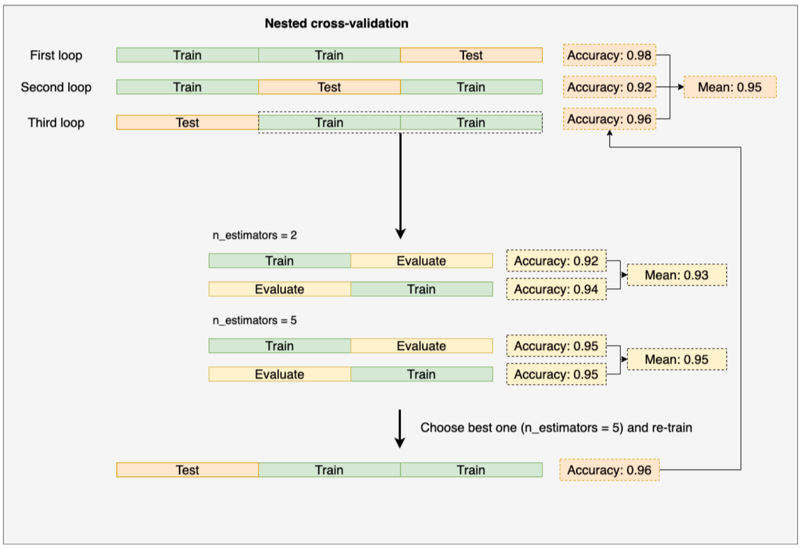


Figure 1. Nested cross-validation: https://ploomber.io/blog/nested-cv/

4.5. Metrics

The metrics used to rank the algorithms during the experiment were accuracy, recall, precision, and F1-score. These are the most commonly used metrics for classification problems. To better understand the importance of each metric in this context, their meanings can be summarized as follows: the accuracy measures the overall correctness of a model by showing the proportion of total predictions that were right. Precision focuses on the quality of positive predictions, indicating how many of the predicted positive cases were actually correct. Recall, on the other hand, measures the model's ability to find all relevant positive cases, showing how many of the actual positives were correctly identified. The F1-score combines both precision and recall into a single metric using their harmonic mean, providing a balanced measure especially useful when there is an uneven class distribution or when both false positives and false negatives carry significant consequences. After the experiment, the mean and standard deviation of each metric were calculated across the outer folds. This experiment, which followed a nested cross-validation process, was executed once. Therefore, each model was trained 56 times (8 outer loops and 7 inner loops). Since some algorithms require a long training time, we were unable to repeat the experiment multiple times.

To compare the models, the Kruskal-Wallis test was used, which is non-parametric and can be used when the objective is to compare three or more groups in some quantitative variable when the assumptions about parametric tests are not met [Niedoba et al. 2023]. After that, the Dunn's post hoc test was applied. Dunn's post hoc statistical test indicates whether there is a significant difference in performance between the algorithms executed. The metric used to compare the algorithms was the F1-score. The verification occurs between pairs of algorithms.

5. Experiments and Results

Runtime environments. For the experiments was used 3 virtual environments with different configurations. The first has 1 GPU NVIDIA A100-SXM4-40GB, 16 GB RAM and storage of 128 GB. The second has 1 GPU NVIDIA A100-SXM4-40GB, 16 GB RAM and storage of 64 GB and the third has 1 GPU NVIDIA A100-SXM4-40GB, 32 GB RAM and storage of 128 GB.

Data segmentation. Each time series was divided into 10-second segments, i.e., 1,000 lines of sequential timestamps. After segmentation was completed, the result was 1,216 segments for all users. Since the Autogluon models and XGBoost are used for tabular classification, it was necessary to carry out the feature extraction process. However, in the CIF model, the feature extraction occurs internally during the algorithm execution. Thus, for CIF, the input data was a dataset with all segments (one per line), with a multi-index in the segment identifier and timestamp.

Feature extraction. Twenty-two features were extracted for each of the 31 attributes, generating 682 columns, and each of the 1,216 segments became a record in the dataset.

Training and evaluation of models. Nested cross-validation was performed using the scikit-learn Leave-One-Group-Out (LOGO) [Pedregosa et al. 2011] data splitting strategy to simulate Leave-One-Subject-Out (LOSO), resulting in 56 variations of the training and validation sets. Hyperparameter optimization was performed on each training subset. For hyperparameter optimization, we chose the Optuna tool [Akiba et al. 2019], which dynamically selects the search space and reduces execution time by cutting hyperparameter combinations that do not contribute to the best performance of the algorithm [Akiba et al. 2019]. This tool uses several optimization algorithms, one of which is Bayesian optimization, used for this experiment to maximize the F1-score. Optuna's maximum runtime was limited to 4 hours per algorithm, ensuring that the hyperparameter search would be completed within this time frame. Otherwise, only models trained within this time frame would be considered for ranking. This time limit was chosen due to the high number of iterations in nested cross-validation, significantly increasing the total runtime. The best model from each iteration was retrained using the external validation data at the end of the internal validation step. Finally, all metrics and results were collected for analysis.

5.1. Collection of results

The algorithms were ranked within the CV using accuracy, precision, recall, and F1-score. In the end, for the 8 participants, there was a result for each of these metrics using the test dataset; the mean and standard deviation of the mean of these results were calculated and presented in Table 2. Furthermore, Table 2 presents a categorization of online and offline algorithms, highlighting that comparisons should be made within each respective technique.

Table 2. Average accuracy (ACC), F1-score (F1), precision (P), and recall (R) metrics on the test set.

Technique	Algorithm	ACC	F1	P	R
Offline learning	CIF	0.80 ± 0.053	0.77 ± 0.060	0.81 ± 0.063	0.80 ± 0.053
	ETG	**0.94** ± 0.007	0.93 ± 0.007	0.94 ± 0.007	**0.94** ± 0.007
	ETE	0.93 ± 0.003	0.93 ± 0.007	0.94 ± 0.003	0.93 ± 0.003
	LGBMXT	0.93 ± 0.007	0.93 ± 0.007	0.94 ± 0.007	0.93 ± 0.007
	NNFAI	0.93 ± 0.003	0.93 ± 0.007	0.94 ± 0.003	0.93 ± 0.003
	RFG	0.93 ± 0.007	0.93 ± 0.007	0.94 ± 0.007	0.93 ± 0.007
Online learning	XGBI	**0.78** ± 0.021	**0.77** ± 0.024	**0.81** ± 0.024	**0.78** ± 0.021
	ARF	0.75 ± 0.017	0.72 ± 0.017	0.76 ± 0.014	0.75 ± 0.017
	BiLSTM	0.73 ± 0.028	0.71 ± 0.028	0.73 ± 0.024	0.73 ± 0.028

The final models that performed best are listed with the abbreviation in Table 3. Among the offline learning techniques, the AutoGluon algorithms stood out for their high performance across all metrics and their low standard deviation of the mean. The method with the best results was the Extra Trees Gini classifier, achieving an average of 94%, 93%, 94%, and 94% in accuracy, F1-score, precision, and recall, respectively. However, these algorithms are nearly tied, as their confidence intervals overlap. In contrast, CIF exhibited the lowest performance among the offline algorithms, with average values of 80% for accuracy, 77% for F1-score, 81% for precision, and 80% for recall. As shown in Table 4, CIF presents values greater than 0.05, indicating a significant difference in p-value compared to all other offline algorithms used in this experiment.

Among the online algorithms, XGBoost Incremental stands out as the best performer, achieving 78% accuracy, 77% F1-score, 81% precision, and 78% recall. The ARF is second with 75% accuracy, 72% F1-score, 76% precision, and 75% recall. In the context of incremental learning, the performance of BiLSTM shows that even algorithms known for their strength in batch learning can face challenges when not specifically designed for online scenarios. In our study, BiLSTM had the lowest average performance among the evaluated online learning models, reaching the accuracy 73%, the 71% F1 score, the precision 73% and the recall 73% in the test set.

One key aspect is that BiLSTM was trained in batch mode. This has a significant impact, since incremental learning requires models to adapt continuously to data arriving sequentially, which can lead to issues like catastrophic forgetting, where new information overwrites previously learned patterns. Algorithms that are not built to handle this type of dynamic environment often struggle compared to those with native support for continuous adaptation, such as Adaptive Random Forest (ARF), which operates in a fully streaming fashion and performed better than BiLSTM in our evaluation. BiLSTM ranks as the lowest performing online algorithm, achieving 73% precision, 71% F1 score, 73% precision and 73% recall.

5.2. Analysis of results

The Kruskal-Wallis test was used to compare the techniques, which resulted in an H-statistic equal to 29.06 and a p-value equal to 0.00005903, meaning that there is a signifi-

Table 3. Identifier of each classifier.

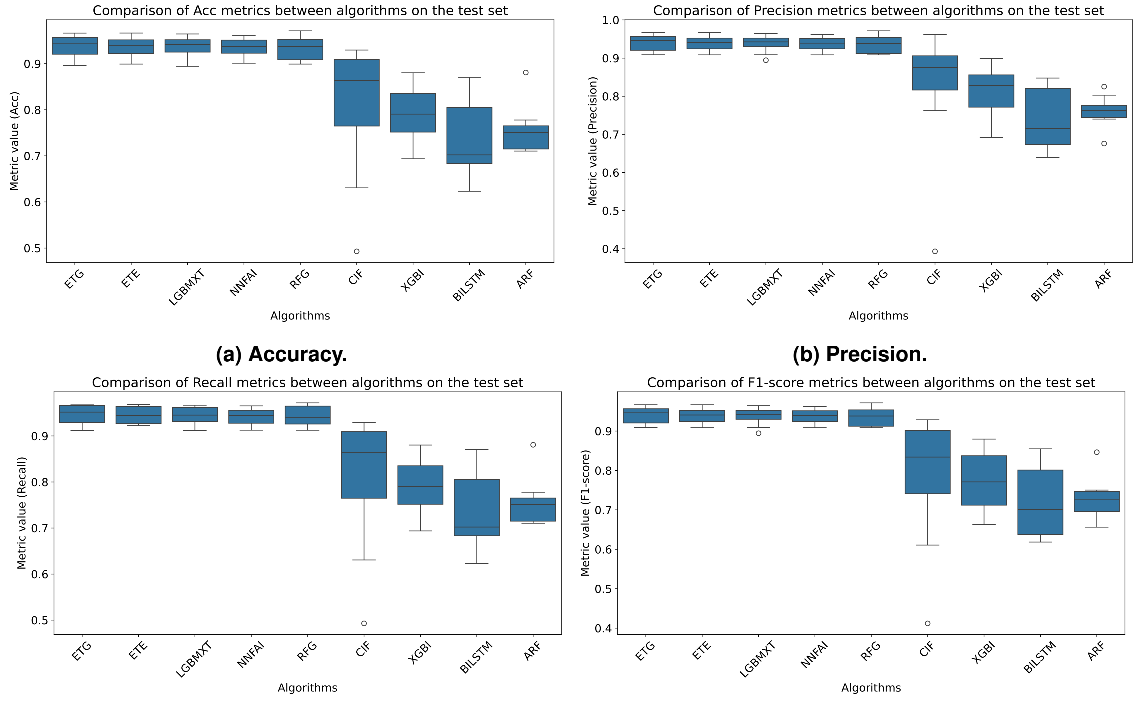
Technique	ID	Algorithm
Offline learning	ETG	ExtraTreesGini_BAG_L2
	ETE	ExtraTreesEntr_BAG_L2
	LGBMXT	LightGBMXT_BAG_L2
	NNFAI	NeuralNetFastAI_BAG_L2
	RFG	RandomForestGini_BAG_L2
	CIF	Canonical Interval Forest
Online learning	XGBI	Incremental XGBoost
	ARF	Adaptive Random Forest
	BiLSTM	Bidirectional LSTM

Table 4. Dunn's post hoc result on the F1-score metric using the test dataset. Values close to zero indicate statistically significant differences.

	ETG	ETE	LGBMXT	NNFAI	RFG	CIF	XGBI	BiLSTM	ARF
ETG	1.00	0.91	0.85	0.87	0.85	0.00	0.00	0.00	0.00
ETE	0.91	1.00	0.93	0.95	0.93	0.00	0.00	0.00	0.00
LGBMXT	0.85	0.93	1.00	0.98	0.99	0.00	0.00	0.00	0.00
NNFAI	0.87	0.95	0.98	1.00	0.98	0.00	0.00	0.00	0.00
RFG	0.85	0.93	0.99	0.98	1.00	0.00	0.00	0.00	0.00
CIF	0.00	0.00	0.00	0.00	0.00	1.00	0.60	0.32	0.37
XGBI	0.00	0.00	0.00	0.00	0.00	0.60	1.00	0.64	0.71
BiLSTM	0.00	0.00	0.00	0.00	0.00	0.32	0.64	1.00	0.92
ARF	0.00	0.00	0.00	0.00	0.00	0.37	0.71	0.92	1.00

cant difference between the models. And with a p-value lower than 0.05, it was possible to run the Dunn's post-hoc test with the results presented in Table 4. The Dunn's test results indicate that the CIF algorithm shows a significant performance difference compared to the other models from the offline technique, as evidenced by the pairwise comparison p-values close to 0. This suggests that this algorithm consistently exhibits behavior distinct from the other methods across the evaluated metrics. However, there are no significant differences at the 0.05 significance level between the best AutoML models. For the online algorithms, the Dunn's test result far from 0 suggests a similarity in their performance. Although XGBI achieves the best results, the difference is not statistically significant compared to the other online learning models. ARF and BiLSTM exhibit a higher degree of similarity to each other compared to XGBI.

In addition to the statistical analysis, the distributions of the obtained accuracy (Figure 2a), precision (Figure 2b), F1-score (Figure 2d), and recall (Figure 2c) for each technique are presented. We can observe that CIF, Incremental XGBoost, and BiLSTM present high variation in their results across all metrics, as indicated by the extended length of the whiskers and boxes for these algorithms. Regarding the results presented in Table 2, ETG stood out as the model with the best performance, achieving the best results in all metrics with averages of 94%, 93%, 94%, 94% in accuracy, F1-score, precision, and recall, respectively. With low standard deviations of the mean in all metrics, ETG managed to maintain a good balance between the correct identification of classes and the consistency of its predictions in different executions and with different data, being



(a) Accuracy.

(b) Precision.

(c) Recall.

(d) F1-Score.

Figure 2. Metrics for online and offline models

therefore the most robust choice for this scenario.

Other algorithms, such as ETE, LGBMXT, NNFAI, and RFG, also presented similar performances, with metrics in the range of 93% to 94%. These results suggest that these algorithms have similar capabilities to deal with the test dataset used, providing reliable and accurate predictions. The most significant difference between these models and ETG can be attributed to specific nuances of optimization or sensitivity to the dataset. In contrast, CIF demonstrated inferior performance, with average accuracy and F1-score on the test set of 80% and 77%, respectively. CIF presented greater variability in its results, with higher standard deviations of the mean, e.g., ± 0.06 for precision, suggesting a lower consistency in its generalization ability. Overall, the results reflect the potential of models such as ETG, ETE, LGBMXT, and NNFAI to achieve high performance in the HAR task.

Incremental algorithms such as XGBI, ARF, and BiLSTM may not achieve the same level of accuracy in offline scenarios, but have advantages in situations where adaptability and continuous learning are essential. This highlights the importance of aligning the choice of algorithm with the specific needs of the scenario in which it will be applied. In the online scenario, XGBI, trained in batch mode, achieved the best performance among the incremental algorithms, while BiLSTM, also trained in batch mode, had the lowest performance. ARF, which operates in a purely streaming fashion—training continuously on one record at a time—ranked second, demonstrating its suitability for real-time learning. Despite XGBI's superior results, Dunn's test revealed no statistically significant differences among the online models, indicating their overall equivalence.

Furthermore, the models generated by Autogluon outperformed CIF in terms of both metrics and uniformity of results for each user left out, showing that their complex transformations and combinations performed on both the data and the models contribute significantly to efficient HAR. For online learning, XGBI outperformed other algorithms, which indicates a relevant choice for HAR in online scenarios. Direct comparison of offline and online models is limited by their distinct operating scenarios. Online learning faces challenges like catastrophic forgetting and user variability, which offline models avoid. Therefore, results must be interpreted within these contexts. Future studies could explore a larger set of online models for a more comprehensive comparison.

6. Threats to validity

During the experiment, measures were taken to mitigate threats to validity; however, given the challenging nature of the problem, the presence of threats is inevitable. Among them, the number of users was a challenge since, with the LOSO technique used, only eight metric results were collected, one for each user left out. Therefore, it was only possible to combine up to 20 results, since this is an upper limit for the execution of more complex statistical tests. In addition, when using Optuna with a limited runtime, the tool may not choose the best combination of hyperparameters among all possible ones. Online classification remains challenging due to catastrophic forgetting: the order of users in training may be decisive in classification ability.

7. Conclusions and Future Work

This paper compared HAR techniques using algorithms specialized in time series classification, offline classification of tabular data, and online classification of tabular data. Among the techniques used for the offline classification of tabular data with Autogluon, the five best algorithms (ExtraTreesGini, ExtraTreesEntr, LightGBMXT, NeuralNetFastAI, and RandomForestGini) showed high performance across all metrics and were statistically similar based on the F1-score metric. The Canonical Interval Forest (CIF) algorithm performed significantly worse in the offline learning category compared to the other offline techniques evaluated in this study.

For online classification of HAR, while the average performance metrics for XGBI, ARF, and BiLSTM differed, the statistical analysis (Dunn's test) indicated no statistically significant differences among these online models. XGBI achieved the best average results among the online algorithms. Thus, while highly optimized offline methods currently yield superior accuracy on static datasets, online learning is a vital but still developing area for HAR, crucial for adaptive systems despite facing challenges like catastrophic forgetting. The performance differences observed highlight the importance of selecting and potentially adapting algorithms based on the specific requirements and nature of the HAR application. In future work, we will use other newer online and deep learning techniques, such as RNNs, LSTMs, GRUs to enable a more comprehensive comparison with existing approaches. We also intend to use other datasets to ensure the generalization of the HAR model to different domains.

Acknowledgments

Part of the results presented in this work were obtained through the project "CENTER OF EXCELLENCE IN ARTIFICIAL INTELLIGENCE - AI4WELLNESS", funded by Samsung Eletrônica da Amazônia Ltda at Federal University of Ceará, Brazil, under the Information Technology Law Number 8.248/91.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery.

Bukhari, A., Hosseinimotlagh, S., and Kim, H. (2024). Opensense: An open-world sensing framework for incremental learning and dynamic sensor scheduling on embedded edge devices. *IEEE Internet of Things Journal*, 11(15):25880–25894.

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., and Smola, A. (2020). Autogluon-tabular.

Guo, S., Gu, Y., Wen, S., Ma, Y., Chen, Y., Wang, J., and Hu, C. (2022). Kici: A knowledge importance based class incremental learning method for wearable activity recognition. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 646–655, New York, NY, USA. Association for Computing Machinery.

Helmi, A. M., Al-qaness, M. A. A., Dahou, A., Damaševičius, R., Krilavičius , T., and Elaziz, M. A. (2021). A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human activity recognition using smartphone sensors. *Entropy*, 23(8).

Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82.

Liu, M., Bian, S., Zhou, B., and Lukowicz, P. (2024). ikan: Global incremental learning with kan for human activity recognition across heterogeneous datasets. page 89–95.

Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., and Jones, N. S. (2019). catch22: Canonical time-series characteristics. *Data Mining and Knowledge Discovery*, 33(6):1821–1852.

Middlehurst, M., Large, J., and Bagnall, A. (2020). The canonical interval forest (cif) classifier for time series classification. In *2020 IEEE international conference on big data (big data)*, pages 188–195. IEEE.

Niedoba, T., Surowiak, A., Hassanzadeh, A., and Khoshdast, H. (2023). Evaluation of the effects of coal jigging by means of kruskal–wallis and friedman tests. *Energies*, 16(4).

Ohwosoro, I., Edje, A., and Ogeh, C. (2024). A hybrid assault detection system using random forest enabled xgboost-lightgbm technique. *Nigerian Journal of Science and Environment*, 22(2):177–189.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Reiss, A. (2012). PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NW2H.

Sousa, T., Cruz, L., Souza, C., Magalhães, R., and Macêdo, J. (2025). Enhancing har novelty detection with activity confusion analysis and clustering. In *Anais do XXV*

Simpósio Brasileiro de Computação Aplicada à Saúde, pages 12–23, Porto Alegre, RS, Brasil. SBC.

Tahir, S. B. u. d., Dogar, A. B., Fatima, R., Yasin, A., Shafiq, M., Khan, J. A., Assam, M., Mohamed, A., and Attia, E.-A. (2022). Stochastic recognition of human physical activities via augmented feature descriptors and random forest model. *Sensors*, 22(17).

Tseng, Y.-H. and Wen, C.-Y. (2023). Hybrid learning models for imu-based har with feature analysis and data correction. *Sensors*, 23(18).

Valerio, A., Demarchi, D., O'Flynn, B., and Tedesco, S. (2024). Development of a personalized anomaly detection model to detect motion artifacts over ppg data using catch22 features. In *2024 IEEE SENSORS*, pages 1–4.