The Impact of Transformed Data Representations on Sensor-Based Human Activity Recognition

Laura Sthefany Colombo[1], **Yan Lucas Dias**[1], **Caetano Mazzoni Ranieri**[1]

Instituto de Geociências e Ciências Exatas (IGCE)
Universidade Estadual Paulista (UNESP)
Rio Claro, SP, Brasil

{laura.colombo, yan.dias, cm.ranieri}@unesp.br

Abstract. *Human Activity Recognition (HAR) aims to identify human activities from sensor data and has been extensively studied in recent years. However, the impact of different data representation strategies for inertial and ambient sensor signals remains insufficiently understood. This work investigates the use of feature engineering and transformation techniques, including MiniRocket and recurrence plots, in combination with machine learning models for HAR. The results show that MiniRocket consistently achieved the best performance on inertial datasets, and tree-based models benefited the most from feature engineering. Overall, the findings highlight the role of data representation in HAR and indicate that multimodal and highly imbalanced datasets remain challenging.*

1. Introduction

Human Activity Recognition (HAR) is an important research area driven by advances in Artificial Intelligence (AI) and machine learning [Demrozi et al. 2020]. By analyzing data collected from wearable sensors, ambient sensors, or video systems [Wang et al. 2019], signal processing and machine learning techniques can identify human activities and enable applications in healthcare, elderly monitoring, sports, security, and smart environments [Kulsoom et al. 2022].

However, there is still a lack of studies that compare, under equivalent conditions, the impact of different data representations, such as feature engineering, MiniRocket, and Recurrence Plots, on the performance of HAR systems. In this context, this study investigates the impact of different data representations on HAR systems, including raw data, feature engineering, and transformed representations generated by MiniRocket and Recurrence Plots. To evaluate these representations, both machine learning and deep learning models are employed, including Random Forest, XGBoost, 1D CNN, and ResNet. Experiments are conducted on five widely used public HAR datasets encompassing different sensing modalities and acquisition scenarios: PAMAP2 [Reiss and Stricker 2012], HWU-USP [Ranieri et al. 2021], OPPORTUNITY [Chavarriaga et al. 2013], SBHAR [Anguita et al. 2013], and UTD-MHAD [Chen et al. 2015]. Vision-based modalities were not considered due to the scope of this study.

The main contributions of this study are: (i) the establishment of a unified experimental protocol for evaluating and comparing different classification models; (ii) the assessment of data transformation techniques prior to training machine learning and deep learning models; and (iii) the analysis of scenarios involving ambient sensors and highly imbalanced datasets.

2. Related Works

The HAR literature presents various approaches for recognizing human activities from inertial and ambient sensor data. Despite that, challenges such as sensor heterogeneity, inter-subject variability, and class imbalance continue to hinder the generalization of HAR models.[Chen et al. 2021b].

Random Forest and XGBoost, both machine learning techniques based on ensembles of decision trees, can handle noisy and nonlinear data while maintaining strong performance in high-dimensional settings, making them widely used in HAR applications [Chen et al. 2021a]. For these models, feature engineering based on statistical attributes extracted from the time and frequency domains remains a common strategy [Chen et al. 2021a].

Convolutional Neural Networks (CNNs) have been widely employed for the automatic extraction of temporal features from sensor time series. Early studies [Zeng et al. 2014] proposed 1D CNN architectures for HAR using smartphone sensors, demonstrating that hierarchical pattern learning can outperform manual feature engineering. Hammerla et al. [Hammerla et al. 2016] compared deep, convolutional, and recurrent models for HAR, showing that 1D CNNs are particularly well suited for short and well-defined activities. More recent studies have shown that hybrid architectures, such as CNN-LSTM and CNN-GRU, can achieve higher accuracy on certain datasets, albeit at the cost of increased computational complexity [Brery et al. 2026].

The transformation of sensor time series into two-dimensional images through Recurrence Plots (RPs), a technique originally proposed for the analysis of dynamical systems, has been adopted in HAR to leverage well-established computer vision architectures. Penatti and Santos [Penatti and Santos 2017] demonstrated that converting accelerometer data into RPs followed by visual descriptor extraction can outperform approaches based on temporal and spectral features. Bertini et al. [Bertini et al. 2020] employed a similar strategy, transforming inertial data into images and using 2D CNN-LSTM architectures. Hatami and Gavet [Hatami et al. 2018] investigated the use of deep CNNs for time-series classification based on image representations. More recently, Lew et al. [Lew et al. 2023] demonstrated the direct use of a pre-trained ResNet-50 on RPs for activity classification, achieving high accuracy on three benchmark datasets.

ROCKET [Dempster et al. 2020] introduced a time-series classification approach based on transformations generated by random convolutional kernels followed by a linear classifier, achieving state-of-the-art accuracy with low computational cost. MiniRocket [Dempster et al. 2021] reformulated the original ROCKET, making it up to 75 times faster, nearly deterministic, while maintaining comparable accuracy. Despite these advances and the application of MiniRocket to HAR tasks, studies that systematically evaluate its performance across multiple datasets and compare it with tree-based methods, CNNs, and Recurrence Plot-based representations remain limited in the literature.

This study evaluates and compares, under a unified experimental protocol, four HAR strategies: (i) tree-based models with hyperparameter optimization via Optuna [Akiba et al. 2019]; (ii) 1D CNN; (iii) MiniRocket; and (iv) Recurrence Plots with a ResNet50 CNN. Although these approaches have been widely investigated individually, systematic comparisons among them remain limited, particularly in scenarios involving

different sensor modalities and levels of complexity. To address this gap, five public datasets comprising inertial and ambient sensors are evaluated under different data representations (raw data, feature engineering, MiniRocket, and Recurrence Plots). In addition, the impact of hyperparameter optimization is analyzed through bootstrap-based statistical validation.

3. Methods

For this study, several representative HAR datasets were used, covering diverse sensor modalities and activity complexity levels. These datasets enabled a comparative analysis of the different approaches and supported the identification of the most suitable strategies for each recognition scenario.

3.1. Datasets

Five public Human Activity Recognition (HAR) datasets were employed, covering a range of scenarios, sensor modalities, and complexity levels.

A constant physical time of 2.5 seconds was adopted for the segmentation window to capture full cycles of typical rhythmic activities (e.g., walking) while balancing temporal resolution. Consequently, the window size (in number of samples) varied across each dataset according to its respective original sampling frequency (f_s), operating with a 50% overlap. The UTD-MHAD dataset was the sole exception to this criterion. Since it consists of short-duration discrete actions, segmentation would have cut the movements in half; therefore, the entire action was preserved, and the various sensors were synchronized via linear interpolation.

- **HWU-USP [Ranieri et al. 2021]:** two inertial units and ambient sensors of an intelligent kitchen were employed in this study, although RGB-D videos are also available in the dataset.
- **PAMAP2 [Reiss and Stricker 2012]:** contains data from three IMUs (hand, chest, ankle) and a heart rate monitor.
- **Opportunity [Chavarriaga et al. 2013]:** comprises a network of wearable and ambient sensors. The null class (periods without relevant activity) was maintained to simulate real-world scenarios. This class accounts for approximately 21,000 samples, whereas the remaining classes contain around 1,000 samples each.
- **SBHAR [Anguita et al. 2013]:** is composed of data from a smartphone inertial sensors.
- **UTD-MHAD [Chen et al. 2015]:** only IMU data from two sensors and 3D skeleton coordinates (Kinect) were used in this study, although RGB-D data is also available.

3.2. Preprocessing

The common pipeline was applied to all datasets, comprising the handling of missing values when necessary, by means of linear interpolation, forward fill, and backward fill, as well as the removal of samples associated with null labels (absence of activity of interest), except for the OPPORTUNITY dataset. Finally, *z-score* normalization was employed, as defined by Equation 1.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Where x represents a signal sample (an individual *feature* value), μ is the mean of the training samples, and σ is the respective standard deviation.

The continuous signals were segmented using sliding windows, which divide the time series into fixed-length segments with overlap between consecutive windows. This approach preserves the temporal dynamics of the activities and increases the number of training samples. The window size was adjusted for each dataset according to its sampling frequency, targeting approximately 2.5 seconds of activity per segment. This resulted in window lengths ranging from 50 to 250 timesteps.

All datasets used strict subject-wise splits to prevent data leakage: PAMAP2 (101-104,107 / 108 / 105-106), HWU-USP (01-08 / 09-10 / 11-16), OPPORTUNITY (LOPO: 1,2 / 3 / 4), SBHAR (predefined groups), UTD-MHAD (1,3,5 / 7 / 2,4,6,8).

3.3. Implementation of Classification Models

To systematically assess the impact of data representation and algorithmic parametrization, a 2×2 experimental protocol was defined, yielding four distinct scenarios for each combination of model and dataset. The two crossed dimensions were: (i) input representation (i.e., raw data versus statistical feature engineering) and (ii) hyperparameter tuning (i.e., default configuration versus Optuna-based optimization). The resulting four scenarios are as follows:

1. **Raw Data (RD)**: original time windows with default hyperparameters;
2. **RD + Optuna**: original windows with hyperparameter optimization;
3. **Feature Engineering (FE)**: statistical attributes (mean, standard deviation, minimum, maximum, etc.) extracted in the time domain for each sensor channel, with default hyperparameters;
4. **FE + Optuna**: statistical attributes with hyperparameter optimization.

For models that require two-dimensional inputs, the raw time windows (a three-dimensional tensor with N samples, t time steps, and c channels) were flattened into one-dimensional vectors of length $t \times c$.

In addition to manual Feature Engineering (FE), two advanced transformations were also investigated as complementary approaches: (i) *MiniRocket*, which extracts ultra-fast features via dilated convolutions and was evaluated under the same four scenarios as the main protocol; and (ii) Recurrence Plots (RP), which convert temporal dynamics into images for processing by 2D convolutional networks. Owing to their image-based nature, the RP pipeline did not include the FE step and was therefore evaluated only in the scenarios with and without optimization. Both techniques aim to capture non-linear dependencies and transient motion patterns, overcoming limitations of manual statistical features.

3.3.1. Random Forest

The Random Forest technique was used as the baseline for comparison with the other algorithms. It was trained directly on either the flattened raw-window vectors or the

extracted statistical features, following the experimental protocol.

Hyperparameter optimization included the number of estimators (`n_estimators`), maximum tree depth (`max_depth`), and the minimum criteria for node splitting and leaf creation (`min_samples_split`) and (`min_samples_leaf`). The number of Optuna trials was adjusted according to the dimensionality of each dataset: for OPPORTUNITY, the high dimensionality required reducing the search to 25 trials to maintain computational feasibility, whereas a more extensive search of 50 trials was performed for the remaining datasets. Notably, the search space for all datasets included the possibility of training trees without a depth limit (`max_depth=None`).

3.3.2. Convolutional Neural Network (CNN)

As a deep learning alternative, a 1D CNN architecture was evaluated, as it is widely employed for time-series analysis [Wang et al. 2017]. A schematic diagram of the network is shown in Figure 1. The architecture consists of two sequential convolutional blocks—the first incorporating temporal downsampling through *MaxPooling1D* and the second using *GlobalAveragePooling1D* to aggregate feature maps—followed by a hidden dense layer with *ReLU* activation and an output layer with *Softmax* activation for multiclass classification. Training was performed using the ADAM optimizer combined with *EarlyStopping*, monitoring the validation loss with a patience ranging from 8 to 10 epochs depending on the dataset. Consistent with the experimental protocol adopted for the other models, hyperparameters such as the learning rate, *Dropout* rate, and the number of neurons in the dense layer were optimized using the Optuna framework.

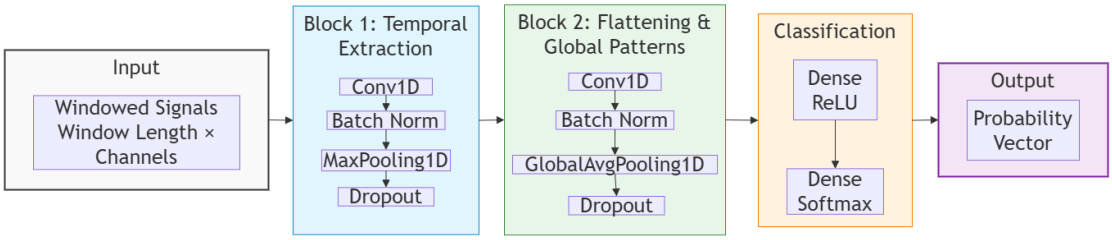


Figure 1. Architecture of the 1D CNN used in this study.

Dataset-specific adaptations (Focal Loss for OPPORTUNITY, class weights for SBHAR, and skewness/kurtosis for XGBoost on SBHAR) were applied solely for training stability and are isolated from the core representation comparison.

In the most severe class imbalance scenario, represented by the OPPORTUNITY dataset, the standard Categorical Cross-Entropy loss function (Equation 2),

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(p_i) \tag{2}$$

where C denotes the number of classes, y_i the ground-truth label, and p_i the predicted probability, was replaced by the Focal Loss function (Equation 3):

$$L_{FL} = -\sum_{i=1}^{C} \alpha_i (1 - p_i)^{\gamma} y_i \log(p_i) \tag{3}$$

where α_i acts as a class-balancing factor for class i, and γ is the focusing parameter. This substitution dynamically reduced the contribution of easily classified instances (such as the dominant null class) and directed the learning process toward the human activities of interest.

3.3.3. XGBoost

The XGBoost algorithm [Chen and Guestrin 2016] was optimized using Optuna, considering the learning rate, maximum tree depth, and subsampling ratios. Instead of fixing the number of estimators, an *Early Stopping* criterion based on the stabilization of the validation loss was employed, ensuring adequate convergence for each dataset. For the SBHAR dataset, feature engineering was extended to include skewness and kurtosis metrics, aiming to capture subtle oscillatory patterns in inertial sensor signals.

3.3.4. MiniRocket

MiniRocket [Dempster et al. 2021] requires input tensors in the (instances, channels, time) format. Therefore, the original tensor of dimensions $N \times T \times C$ (samples, time steps, channels) was transposed to $N \times C \times T$, allowing convolutions to be applied independently to each channel. The final classifier consisted of a linear pipeline combining `StandardScaler` and a `RidgeClassifier` with balanced class weights. Hyperparameter optimization was performed using Optuna, considering the regularization coefficient and solver selection.

In the feature engineering (FE) scenarios, manually extracted statistical features (e.g., mean, standard deviation, skewness, kurtosis, and state transitions) were initially standardized using Z-score normalization. These were then horizontally concatenated with the feature matrices generated by MiniRocket, producing an expanded feature vector. Figure 2 illustrates this procedure. A final joint Z-score normalization was applied to the combined feature set within the classification pipeline, preserving the separation between training, validation, and test partitions to prevent data leakage.

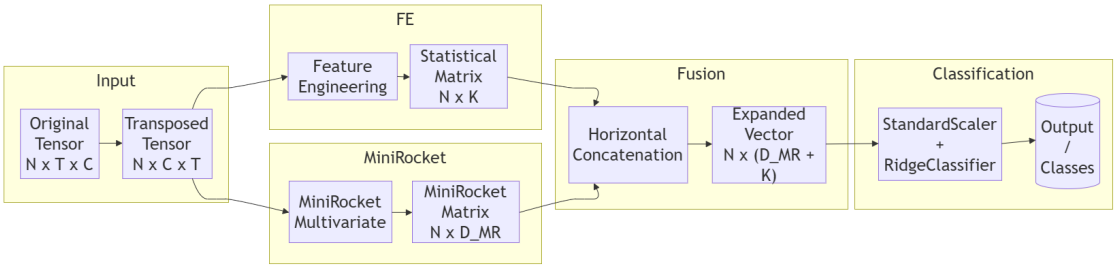


Figure 2. MiniRocket processing pipeline.

3.3.5. Recurrence Plot with ResNet

The *Recurrence Plot* (RP) [Eckmann et al. 1987] is employed to transform time series into the spatial domain, revealing nonlinear dependencies as two-dimensional matrices. In this study, we propose a hybrid deep learning architecture (Figure 3) based on a *late fusion* strategy, in which visual representations derived from RPs and temporal features extracted from the original signals are processed separately and combined only at the final dense decision layers.

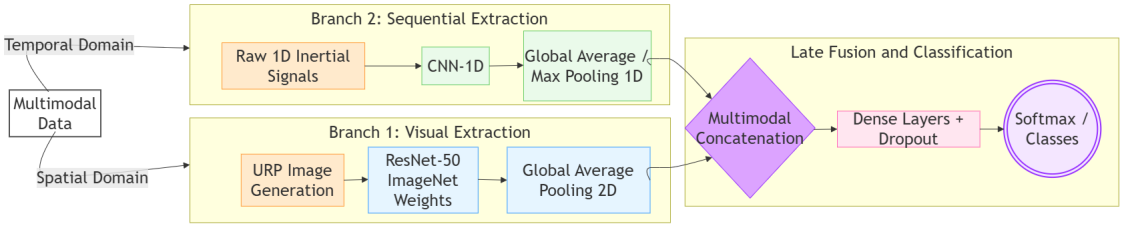


Figure 3. Hybrid *late-fusion* architecture.

The branching scheme was adapted to the characteristics of each dataset. In multimodal datasets, such as HWU-USP and UTD-MHAD, inertial signals were converted into *Unthresholded Recurrence Plots* (URPs) and processed by a pre-trained ResNet-50, while complementary modalities, such as skeletal coordinates and environmental sensors, were processed by a 1D CNN branch. The representations extracted by both branches were subsequently fused for final classification.

Dataset-specific adaptations to the hyperparameters and training protocols were required. For HWU-USP, a static global maximum-distance scaling strategy was adopted during plot generation to preserve the natural amplitude variations of the movements. In OPPORTUNITY, the null-class weights were clipped in the loss function, together with a reduced learning rate, to prevent convergence to trivial solutions. In SBHAR, a similar class-balancing strategy was applied to emphasize minority windows associated with postural transitions. Finally, in UTD-MHAD, the large number of classes (27 categories) required the use of confusion matrices for a detailed performance analysis.

In purely inertial datasets, such as SBHAR, the architecture operated as a representation-fusion framework. The same raw signal was converted into URP images for processing by the ResNet-50 while also being preserved in its original time-series form for the CNN-1D. This design retains temporal information that may be attenuated during the image transformation process. The feature maps from both branches are then flattened, concatenated, and passed to the classification layers.

4. Results

This section presents the results obtained from the evaluation of the predictive models applied to human activity recognition. The performance of each model was analyzed across the five selected datasets (PAMAP2, HWU-USP, OPPORTUNITY, SBHAR, and UTD-MHAD), considering the two input representation scenarios proposed in the methodology: raw data and feature engineering, with and without Optuna-based hyperparameter optimization.

Table 1. Results: CNN-1D - Raw Data (RD) and Feature Engineering (FE)

Dataset	Acc (RD)	Acc (FE)	F1 (RD)	F1 (FE)
HWU-USP	0.443	0.485	0.420	0.472
HWU-USP + Optuna	0.445	0.503	0.411	0.472
Opportunity	0.600	0.706	0.216	0.280
Opportunity + Optuna	0.564	0.740	0.223	0.302
PAMAP2	0.645	0.877	0.598	0.866
PAMAP2 + Optuna	0.598	0.861	0.524	0.803
SBHAR	0.913	0.856	0.845	0.749
SBHAR + Optuna	0.925	0.834	0.848	0.779
UTD-MHAD	0.568	0.678	0.524	0.634
UTD-MHAD + Optuna	0.465	0.609	0.428	0.575

Table 1 shows that the CNN-1D model benefited from feature engineering (FE) on the PAMAP2 dataset, with accuracy increasing from 0.645 to 0.877, and on OPPORTUNITY, where accuracy rose from 0.600 to 0.706, although the F1-score remained low (0.280), reflecting the bias caused by class imbalance. On SBHAR, FE led to performance degradation (accuracy from 0.913 to 0.856; F1 from 0.845 to 0.749), indicating that raw data were already highly discriminative for this dataset. UTD-MHAD showed improvement with FE (accuracy from 0.568 to 0.678; F1 from 0.524 to 0.634), but the addition of Optuna worsened the results (e.g., raw + Optuna: accuracy 0.465, F1 0.428), suggesting that hyperparameter search was not beneficial for this dataset, or may have led to less generalizable solutions.

Table 2. Results: Random Forest - Raw Data (RD) and Feature Engineering (FE)

Dataset	Acc (RD)	Acc (FE)	F1 (RD)	F1 (FE)
HWU-USP	0.416	0.480	0.403	0.479
HWU-USP + Optuna	0.401	0.473	0.392	0.477
Opportunity	0.759	0.763	0.206	0.283
Opportunity + Optuna	0.779	0.839	0.477	0.675
PAMAP2	0.473	0.853	0.386	0.829
PAMAP2 + Optuna	0.561	0.895	0.508	0.893
SBHAR	0.849	0.859	0.726	0.771
SBHAR + Optuna	0.855	0.864	0.763	0.788
UTD-MHAD	0.498	0.729	0.489	0.705
UTD-MHAD + Optuna	0.446	0.782	0.439	0.762

For Random Forest (Table 2), FE yielded performance improvements across most datasets, with the largest gains observed on PAMAP2 (accuracy from 0.473 to 0.853) and UTD-MHAD (from 0.498 to 0.729). The FE + Optuna combination further enhanced performance on PAMAP2 (F1 = 0.893) and OPPORTUNITY (F1 = 0.675), confirming the value of hyperparameter optimization for tree-based models, even under severe class imbalance. On HWU-USP, however, gains were more limited (F1 from 0.403 to 0.479), suggesting that the complexity of domestic activities involving ambient sensors remains a challenging scenario.

Table 3. Results: XGBoost - Raw Data (RD) and Feature Engineering (FE)

Dataset	Acc (RD)	Acc (FE)	F1 (RD)	F1 (FE)
HWU-USP	0.435	0.500	0.424	0.498
HWU-USP + Optuna	0.444	0.507	0.436	0.510
Opportunity	0.823	0.835	0.536	0.572
Opportunity + Optuna	0.830	0.858	0.615	0.697
PAMAP2	0.520	0.758	0.447	0.683
PAMAP2 + Optuna	0.577	0.720	0.508	0.677
SBHAR	0.896	0.873	0.781	0.783
SBHAR + Optuna	0.896	0.885	0.788	0.809
UTD-MHAD	0.460	0.668	0.446	0.637
UTD-MHAD + Optuna	0.507	0.730	0.486	0.715

XGBoost (Table 3) showed contrasting behavior across datasets. On OPPORTUNITY, the FE + Optuna combination improved the F1-score from 0.536 to 0.697, whereas on PAMAP2, Optuna led to a marginal change (F1: 0.683 vs 0.677; accuracy: 0.758 vs 0.720). These findings suggest that hyperparameter optimization does not consistently improve performance; its effectiveness is contingent upon the validation set characteristics and the chosen search space. On SBHAR and HWU-USP, XGBoost proved robust, attaining F1-scores of 0.809 and 0.510, respectively.

Table 4. Results: MiniRocket - Raw Data (RD) and Feature Engineering (FE)

Dataset	Acc (RD)	Acc (FE)	F1 (RD)	F1 (FE)
HWU-USP	0.267	0.398	0.257	0.386
HWU-USP + Optuna	0.277	0.403	0.268	0.372
Opportunity	0.740	0.793	0.467	0.602
Opportunity + Optuna	0.704	0.753	0.471	0.581
PAMAP2	0.903	0.886	0.894	0.885
PAMAP2 + Optuna	0.906	0.889	0.903	0.898
SBHAR	0.961	0.961	0.867	0.872
SBHAR + Optuna	0.939	0.958	0.840	0.878
UTD-MHAD	0.887	0.900	0.887	0.898
UTD-MHAD + Optuna	0.878	0.894	0.873	0.892

MiniRocket (Table 4) achieved outstanding performance on raw data, particularly on SBHAR (F1 = 0.867), PAMAP2 (F1 = 0.894), and UTD-MHAD (F1 = 0.887). In contrast to tree-based models, feature engineering (FE) provided only marginal improvements—and in some cases, slight decreases—on these datasets, suggesting that MiniRocket's convolutional transformations are already effective at capturing relevant temporal patterns. The notable exceptions were HWU-USP, where FE was critical (F1 from 0.257 to 0.386), and Opportunity (F1 from 0.467 to 0.602). Hyperparameter optimization with Optuna yielded mostly marginal or neutral impacts overall, with the exception of PAMAP2, where the combination of raw data and Optuna achieved the highest F1-score of 0.903.

The Recurrence Plot approach (Table 5) exhibited high sensitivity to parameter

Table 5. Results: Recurrence Plot + ResNet50 + 1D CNN (Late Fusion) - Without Optuna (NoOpt) and With Optuna (Opt)

Dataset	Acc (NoOpt)	Acc (Opt)	F1 (NoOpt)	F1 (Opt)
HWU-USP	0.324	0.347	0.319	0.330
Opportunity	0.612	0.588	0.099	0.108
PAMAP2	0.357	0.715	0.266	0.618
SBHAR	0.865	0.876	0.865	0.877
UTD-MHAD	0.467	0.493	0.445	0.454

settings. The most substantial improvement with Optuna was observed on PAMAP2, where the F1-score rose from 0.266 to 0.618. Nevertheless, this value remains well below the performance of MiniRocket and Random Forest on the same dataset, both of which achieved F1-scores around 0.89. On OPPORTUNITY, the F1-score stayed at a catastrophic level (≤ 0.11), indicating that the approach failed to capture discriminative patterns relevant to the task.

Notably, the models performed considerably worse on these two datasets, reaching maximum F1-scores of only 0.70 on OPPORTUNITY and 0.51 on HWU-USP. A possible explanation worth investigating is that fixed temporal windows may dilute the occurrence of rare ambient events, and that simply concatenating inertial and ambient sensor streams without attention-based mechanisms may be inadequate.

5. Conclusion

This study investigated the impact of different data representations on Human Activity Recognition (HAR) by evaluating raw data, statistical feature engineering, and transformed representations based on MiniRocket and Recurrence Plots across five public datasets with distinct levels of complexity, sensor modalities, and class imbalance.

The results indicate that MiniRocket achieved the best overall performance on datasets composed exclusively of inertial sensors and structured activities (PAMAP2, SBHAR, and UTD-MHAD), attaining F1-scores above 0.84 even when trained on raw data without extensive optimization. Tree-based models (Random Forest and XGBoost) benefited from feature engineering, particularly when combined with Optuna. The most substantial improvement was observed on OPPORTUNITY, where the Random Forest F1-score increased from 0.206 (raw data) to 0.675 (feature engineering with Optuna), demonstrating the effectiveness of this approach in severely imbalanced scenarios. However, hyperparameter optimization was not universally beneficial: on UTD-MHAD, Optuna degraded the performance of the 1D CNN, while on PAMAP2 it reduced the performance of XGBoost compared with feature engineering alone, highlighting the need to validate hyperparameter search strategies for each model–dataset combination.

The Recurrence Plot-based approach with ResNet exhibited high sensitivity to parameter settings, with performance ranging from an F1-score of 0.618 on PAMAP2 (after optimization) to 0.108 on OPPORTUNITY. Even its best result remained below those achieved by MiniRocket and Random Forest on the same dataset, suggesting that this specific late-fusion approach is sensitive to architectural choices; ablations are needed to isolate the RP effect from the fusion strategy.

The HWU-USP dataset remained challenging for all models, with the highest F1-score reaching only 0.510 (XGBoost with feature engineering and Optuna). This result, together with the overall low performance observed on OPPORTUNITY (maximum F1-score of 0.70), reinforces the need for more advanced sensor fusion techniques capable of handling the sparse and asynchronous nature of ambient sensors.

Future work may investigate relational reasoning across heterogeneous sensor streams. Graph-based architectures, for instance, could explicitly model the spatial and temporal dependencies between inertial and ambient sensors. Complemented by cross-modal attention or late fusion mechanisms, such frameworks might better interpret sparse environmental events. The deployment of HAR on edge devices also remains a compelling direction. MiniRocket's computational efficiency offers a strong foundation for building lightweight, real-time applications, although further research into resource-aware inference is still needed.

Acknowledgements

This study was funded by the São Paulo Research Foundation (FAPESP), Brazil, grant number 2024/23235-8.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework.

Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J. L., et al. (2013). A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pages 3–4.

Bertini, L., Montesh, T., and Rodrigues, D. (2020). Human activity recognition: An approach 2d cnn-lstm to sequential image representation. *Journal of Intelligent and Robotic Systems*.

Brery, A. F., Gallardo-Antolín, A., Fakhry, M., and González-Carrasco, I. (2026). Deep learning for human activity recognition: A comprehensive review of architectures, performance, and challenges across five sensory datasets. *Expert Systems with Applications*, 323:132415.

Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Tröster, G., Millán, J. d. R., and Roggen, D. (2013). The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042.

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015). Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International conference on image processing (ICIP)*, pages 168–172. IEEE.

Chen, J., Sun, Y., and Sun, S. (2021a). Improving human activity recognition performance by data fusion and feature engineering. *Sensors*, 21(3):692.

Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., and Liu, Y. (2021b). Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Comput. Surv.*, 54(4).

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Dempster, A., Petitjean, F., and Webb, G. I. (2020). Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34:1454–1495.

Dempster, A., Schmidt, D. F., and Webb, G. I. (2021). Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 248–257.

Demrozi, F., Pravadelli, G., Bihorac, A., and Rashidi, P. (2020). Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey. *IEEE Access*, 8:210816–210836.

Eckmann, J.-P., Kamphorst, S. O., and Ruelle, D. (1987). Recurrence plots of dynamical systems. *Europhysics Letters*, 4(9):973–977.

Hammerla, N. Y., Halloran, S., and Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. In *IJCAI*, pages 1533–1540.

Hatami, N., Gavet, Y., and Debayle, J. (2018). Classification of time-series images using deep convolutional neural networks. In *ICMV*.

Kulsoom, F., Narejo, S., Mehmood, Z., Chaudhry, H. N., Butt, A., and Bashir, A. K. (2022). A review of machine learning-based human activity recognition for diverse applications. *Neural Computing and Applications*, 34(21):18289–18324.

Lew, C. H., Lim, K. M., Lee, C. P., and Lim, J. Y. (2023). Human activity classification using recurrence plot and residual network. In *2023 IEEE 11th Conference on Systems, Process & Control (ICSPC)*, pages 78–83.

Penatti, O. A. B. and Santos, M. F. S. (2017). Human activity recognition from mobile inertial sensors using recurrence plots. In *ICIP*, pages 2214–2218.

Ranieri, C. M., MacLeod, S., Dragone, M., Vargas, P. A., and Romero, R. A. F. (2021). Activity recognition for ambient assisted living with videos, inertial units and ambient sensors. *Sensors*, 21(3):768.

Reiss, A. and Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pages 108–109. IEEE.

Wang, J., Chen, Y., Hao, S., Peng, X., and Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters*, 119:3–11.

Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE.

Zeng, M. et al. (2014). Convolutional neural networks for human activity recognition using mobile sensors. In *MobiCASE*, pages 197–205.