University of Coimbra

Course: ECAC

Analysis and Feature Extraction in Sensor Data — ECAC Project

Authors:
Fábio Fernandes — 2023230805
Sebastián Rivera — 2025155216

2025

ENGLISH VERSION

Note: Portuguese version starts on page 19.

Introduction

This study uses the FORTH-TRACE dataset, composed of data from five sensors (left/right wrist, chest, right upper leg, left lower leg) from 15 participants, with 16 different activities.

All items in the statement were implemented in mainActivity.py, focusing on:

- Analysis and treatment of outliers (IQR, Z-Score, KMeans, DBSCAN)
- Use of 5-seconds time windows with 50% overlap to segment the data
- Extraction of temporal/spectral features using windows (Zhang & Sawchuk article)
- Reduction of the dimensionality of the dataset (PCA)
- Implementation of Fisher Score and ReliefF for selecting the best features.

3. Analysis and treatment of outliers

In this stage, an analysis of potential *outliers* in the sensor data was carried out.
The detection of extreme values was based on statistical metrics such as the **Z-score** and the **Interquartile Range (IQR)**, which allowed the identification of observations that deviate significantly from the typical behavior of the measurements. Subsequently, the **K-Means clustering algorithm** was used to identify anomalous data segments based on their distance to the cluster centroids.

Samples whose distance exceeded the upper interquartile threshold (Q3 + 1.5 × IQR) were labeled as potential outliers.

3.2 Analyze and comment on the density of outliers in the transformed dataset.

The outlier density analysis, performed using the IQR (Tukey) method on the right-wrist sensor modules, revealed clear differences between sensor types and activity levels.

For the accelerometer, outlier densities ranged from less than 1% in static activities (e.g., IDs 2–4) to over 15–20% in highly dynamic activities such as 8, 9, 10, and 11. This behavior reflects the strong variability in acceleration patterns during rapid or forceful motions, indicating that many of these "outliers" correspond to legitimate high-intensity movements.

The gyroscope exhibited moderate outlier densities (1–12%), with higher values during rotationally intense activities (9–11), consistent with larger angular velocity variations.
The magnetometer, on the other hand, remained largely stable across all activities, with outlier densities mostly below 7%, confirming that magnetic field readings are less affected by motion.

outliers_density_device2_IRQ - DataFrame

Índice	device_id	module	ıctivity_labe	lier_density
0	2	accel_module	1	4.19554
1	2	accel_module	2	0.215558
2	2	accel_module	3	0.513851
3	2	accel_module	4	3.53095
4	2	accel_module	5	3.47766
5	2	accel_module	6	5.16568
6	2	accel_module	7	4.51463
7	2	accel_module	8	15.4447
8	2	accel_module	9	19.7976
9	2	accel_module	10	15.3245
10	2	accel_module	11	18.3222
11	2	accel_module	12	10.9186
12	2	accel_module	13	4.79003
13	2	accel_module	14	14.1732
14	2	accel_module	15	4.93438
15	2	accel_module	16	4.98688
16	2	gyro_module	1	9.48493
17	2	gyro_module	2	6.64658
18	2	gyro_module	3	9.11746
19	2	gyro_module	4	1.62076
20	2	gyro_module	5	1.42801
21	2	gyro_module	6	1.49845
22	2	gyro_module	7	2.07349
23	2	gyro_module	8	7.46695
24	2	gyro_module	9	10.7777
25	2	gyro_module	10	10.3365
26	2	gyro_module	11	11.8676
27	2	gyro_module	12	3.30709
28	2	gyro_module	13	8.25459
29	2	gyro_module	14	2.3622
30	2	gyro_module	15	2.3622
31	2	gyro_module	16	2.04724
32	2	mag_module	1	0
33	2	mag_module	2	6.67278
34	2	mag_module	3	4.76636
35	2	mag_module	4	1.45068
36	2	mag_module	5	1.3388
37	2	mag_module	6	0.390559
38	2	mag_module	7	0.221507
39	2	mag_module	8	3.48558
40	2	mag_module	9	7.35085
41	2	mag_module	10	0.841346
42	2	mag_module	11	6.82664
43	2	mag_module	12	0.0262467
44	2	mag_module	13	0
45	2	mag_module	14	6.66667
46	2	mag_module	15	6.66667
47	2	mag_module	16	1.5748

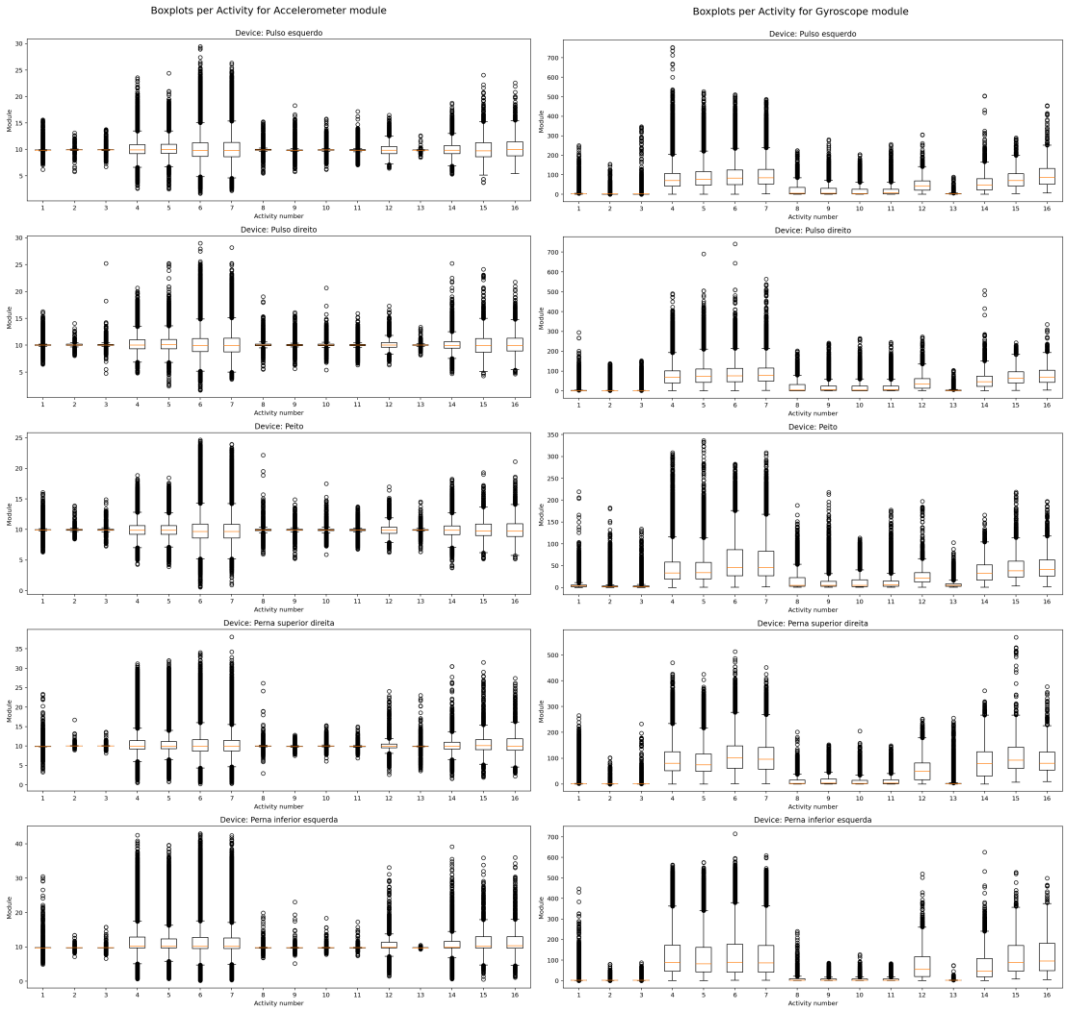
Overall, the results show that outlier density correlates strongly with movement intensity, suggesting that these apparent anomalies might reflect genuine physiological motion variability rather than measuring noise.

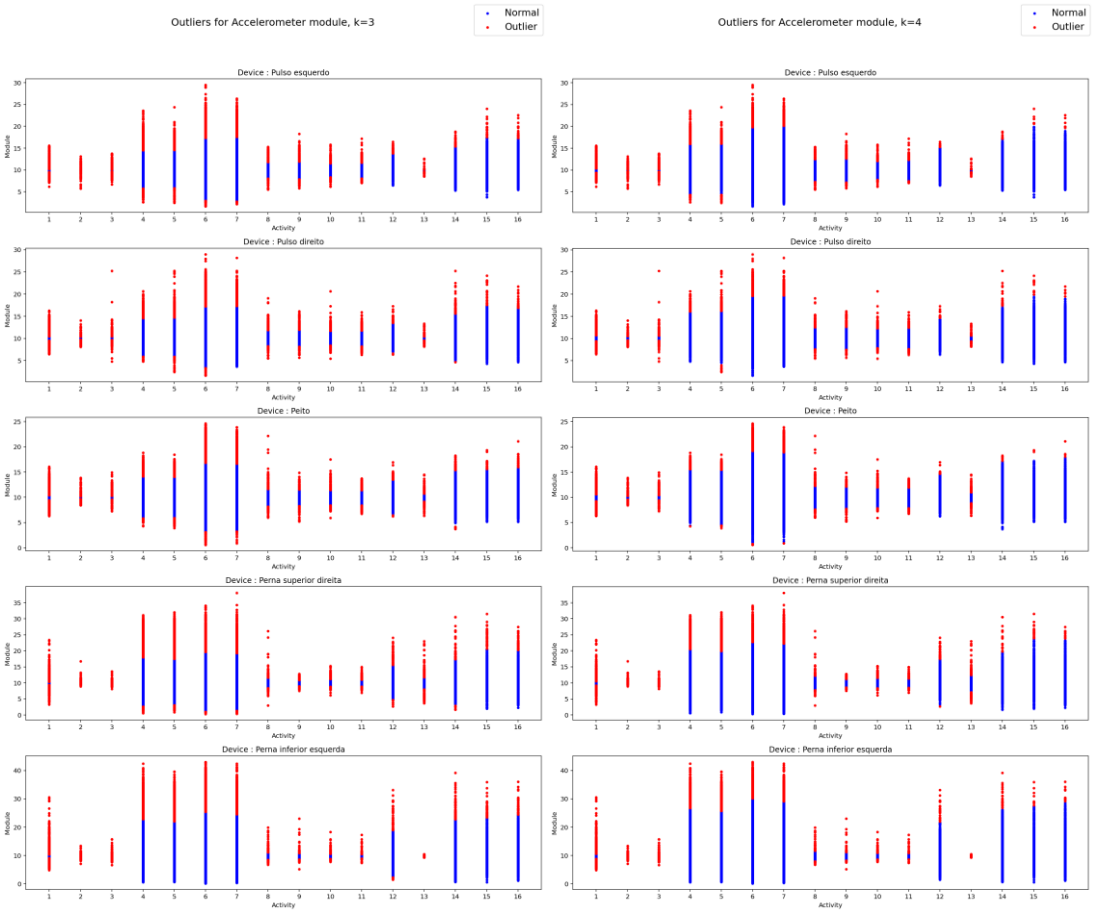
3.5 Compare and discuss results obtained in 3.1 and 3.4

Both methods identify outliers in very similar regions of the data distribution; however, each technique captures different aspects of the signal, and these methodological differences explain the variations observed across devices and activities. In the boxplots (IQR), outliers appear as isolated black circles beyond the whiskers, whereas in the Z-score plots these same values are marked as red points. Visual inspection shows that most of the points flagged by the Z-score coincide with those suggested by the boxplots, especially in activities with high dispersion such as activities 4, 5, 6, and 7.

A key difference between the two approaches lies in the underlying assumptions. The IQR method is non-parametric and robust, meaning it does not depend on the shape of the distribution or the magnitude of extreme values. It simply identifies points that fall outside 1.5 IQR from the quartiles.

In contrast, the Z-score method assumes that values follow an approximately Gaussian distribution, and it measures how many standard deviations a point deviates from the mean. Therefore, its sensitivity depends directly on the standard deviation (σ) of the signal. Sections 3.4 explored different thresholds (k = 3, 3.5, 4), illustrating how changing k modifies the strictness of the method: smaller k values detect more outliers, while larger ones detect fewer. Additionally, the Z-score visualizations make outlier detection more explicit, since normal and anomalous data are color-coded rather than summarized.





Despite the leg sensors presenting the highest overall dispersion in the boxplots, the Z-score plots reveal a greater density of red points (outliers) in the wrist sensors. This occurs because wrist movements generate a noisier and more irregular signal with a smaller standard deviation, causing a larger proportion of values to exceed the Z-score threshold. Conversely, while the leg sensors exhibit larger amplitudes and variability, their correspondingly large standard deviation reduces the number of values classified as outliers by the Z-score criterion.

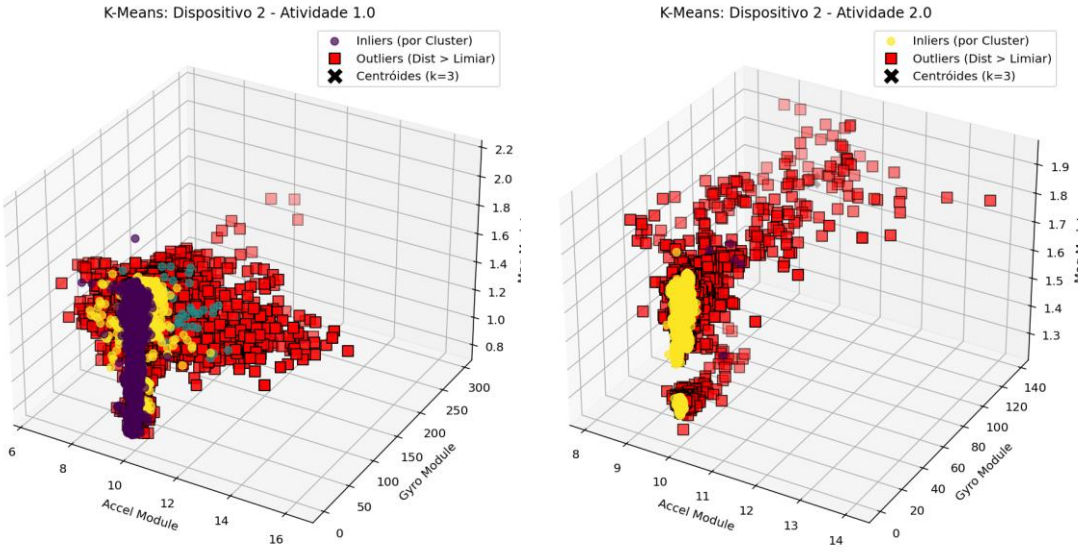
However, the two methods differ substantially in the amount of detected outliers. The IQR approach is more sensitive to abrupt changes and identifies a much higher density of outliers, particularly in the accelerometer and gyroscope data. In contrast, the Z-score method is more conservative: because the wrist signals have a relatively large standard deviation, only a small fraction of values exceed the k-sigma threshold, even with k = 3. As a result, many points considered "extreme" by the IQR method fall within the normal variability range of the Z-score method. Overall, while both approaches highlight similar regions of data variability, the IQR method captures amplitude-based extremes, whereas the Z-score method emphasizes statistical deviation from the mean, leading to consistently lower outlier densities.

3.7 K-means clustering and comparison with the results of 3.4

Outlier detection was additionally performed using the K-Means clustering algorithm, applied to the module space of the accelerometer, gyroscope and magnetometer signals. The algorithm was executed with multiple cluster configurations (k = 3, 4) to assess the stability of the clustering results.

Outliers were defined as samples whose Euclidean distance from their cluster centroid exceeded 1.5 times the interquartile range (IQR) of the corresponding cluster distribution. Compared to the Z-score approach, which identifies statistically extreme values on individual axes, the K-Means method captures multivariate deviations, considering the joint behavior of all three spatial components. This allows for more robust detection of atypical motion patterns, especially in activities involving complex multidirectional movements.

The resulting plots show the distribution of outliers per activity and device. Interestingly, higher outlier densities were observed in static activities such as *Stand*, *Sit*, and *Sit and Talk* (activities 1–3).
This can be explained by the fact that these activities have very compact and low-variance clusters; therefore, even small fluctuations in sensor readings are detected as anomalies.
Conversely, dynamic activities such as *Walk*, *Climb Stairs*, and the various transition movements (activities 4–16) exhibited more dispersed clusters, leading to a lower proportion of detected outliers relative to their natural variability.



The 3D visualizations confirm this behavior, showing that outliers are mainly located at the periphery of dense clusters, distant from the centroids.
These results suggest that the detected outliers correspond to samples with atypical motion magnitudes within otherwise stable activity patterns.

Compared with the Z-score method from section 3.4, K-Means provides a contextual and activity-dependent approach. While Z-score identifies global extremes in the dataset, K-Means detects points that are inconsistent within their own cluster.

Overall, both methods highlight that static activities are more sensitive to small sensor fluctuations, whereas dynamic ones exhibit broader but smoother variability. This reinforces the value of K-Means in uncovering local anomalies specific to each motion context.

4. Extraction of characteristic information

The objective of this stage is to compress the problem space by extracting discriminative characteristic information from the raw sensor data.
This process aims to identify and select meaningful statistical, temporal, and spectral features that best represent the variability of human activities.
By transforming high-dimensional raw measurements into a reduced and more informative feature set, it becomes possible to design more efficient and accurate classification models, capable of distinguishing between different movement patterns and behaviors.

4.1 Kolmogorov-Smirnov test.

To evaluate whether the sensor measurements differed significantly between activities, the normality of the feature distributions was first verified using the Kolmogorov–Smirnov test.
Since most variables did not follow a normal distribution ($p < 0.05$), the non-parametric Kruskal–Wallis test was applied.

The test results were highly significant for all modules: accelerometer ($H = 41\,668.08$, $p < 0.001$), gyroscope ($H = 2\,821\,766.19$, $p < 0.001$), and magnetometer ($H = 608\,088.21$, $p < 0.001$).

These findings indicate that the distributions (or median values) of the features differ markedly across activities.

In particular, the gyroscope features exhibited the largest statistics, suggesting that rotational motion patterns vary most strongly between activity types and thus provide substantial discriminative information.

4.2 Features identification

During Human Activity Recognition (HAR), a set of features can be extracted from sensor information, which provides more data about the activity that is registered. These features help distinguish between different types of activities.

These features are categorized into temporal and spectral, depending on how they are obtained. Temporal features are obtained directly from the raw sensor signals, while the spectral features are computed from transformed information.

Therefore, we computed these features to obtain the best data possible for the recognition, the mean, median, standard deviation, variance, root mean square, average derivative, skewness, kurtosis, interquartile range, mean crossing range, zero crossing range, energy, dominant frequency, averaged intensity, variance of intensity, correlation between acceleration along gravity and heading directions, normalized signal magnitude area, eigenvalues of dominant directions, averaged acceleration energy and averaged rotation energy that got developed according to the provided file.

4.4 PCA component importance

PCA is a data technique, in which the information is transformed and reduced in different components, which are linear combinations of the original data set, in this case the features.

Its objective is to create new variables with variance, and that the first components represent the most data possible.

If you need to represent a specific percentage of information, you can see the accumulated explained variance ratio, which shows the percentage of information that will represent each number of components.

In this case, the first component alone explains approximately 31%, and the first eleven components together account for about 75% of the total variance.

This indicates that a significant amount of information from the original high-dimensional feature space can be effectively represented by a smaller number of components, reducing dimensionality without substantial loss of discriminative power.

4.4.1 Feature obtention related to the PCA

To get the features from this compression, we use the expression X_{PCA}= X_{scaled}* W

where X_{scaled} is the normalized feature matrix, and W is the matrix containing the principal component vectors.
Each row of X_{PCA}represents one observation (or time window) described by the new compressed features.

For example, consider three original features $(x_1, x_2, x_3)$and two principal components defined by the following weight matrix:

$$W = \begin{bmatrix} 0.5 & 0.7 \\ 0.3 & -0.6 \\ 0.8 & 0.4 \end{bmatrix}$$

For one sample with feature values $[x_1, x_2, x_3] = [2.0, 1.0, 3.0]$,
the new PCA features are computed as:

Thus, the original sample $[2.0, 1.0, 3.0]$is represented in the compressed PCA space as $[3.7, 2.2]$.

4.4.2 Advantages and limitations of the PCA

The PCA technique has its own advantages and disadvantages, depending on the use and the focus of the work, whether it will be useful or not that useful.

The advantages identified are the reduction of dimensionality; you need less space for the data, but you keep the most information, which leads to faster processing.

Noise reduction is another advantage; it deletes the features with duplicated information and the ones that are not useful due to its variance.

This reduction brings some limitations with it, these new components are combinations, therefore it has no intuitive use.

When having an enormous amount of data as in this work, even with the reduction, to have a significant representation from the original data it is necessary to use more than 10 components which are hard to interpretate.

4.6 Top 10 features according to Fisher Score and ReliefF

The Fisher Score and ReliefF methods were applied to identify the ten most relevant features for distinguishing between activities.
Both techniques measure how well each feature separates the activity classes but use different principles.

The output generated from these two methods was:

- **Fisher Score - [acce_y_rms, gyro_y_rms, gyro_y_std, acce_z_rms, gyro_x_rms, gyro_x_std, gyro_y_IQR, gyro_x_IQR, acce_y_median, gyro_y_avg_der]**

- **ReliefF – [magne_y_var, acce_z_var, acce_y_rms, acce_z_energy, magne_y_, magne_x_energy, acce_y_median, acce_y_mean, magne_y_mean, acce_z_rms]**

The Fisher Score focuses on features that show large differences between classes and small variation within each class.

The ReliefF method, on the other hand, evaluates the local importance of each feature by comparing neighboring samples from the same and different classes.

The results show that the two methods selected different sets of features, with only three features in common (acce_y_median', 'acce_y_rms', 'acce_z_rms').

4.6.1 Feature obtention according to Fisher Score and ReliefF

After performing these methods, we obtain the best features, if we want to obtain the data set only for specific features, we have the option to call the columns that will be used, it could be performed with a line like this:

df = df[selected_features] where selected features are the names of the columns

This is possible because we can get the name of the columns that we want to keep, and that way the data set will only keep the requested columns.

4.6.2 Advantages and limitations of the scores

These methods as well as the PCA method have advantages and disadvantages. The most important advantage is the dimensionality reduction. The great difference about the PCA method is that this data set is the same that we were using, not mixed, this way it also maintains the interpretability from the original.

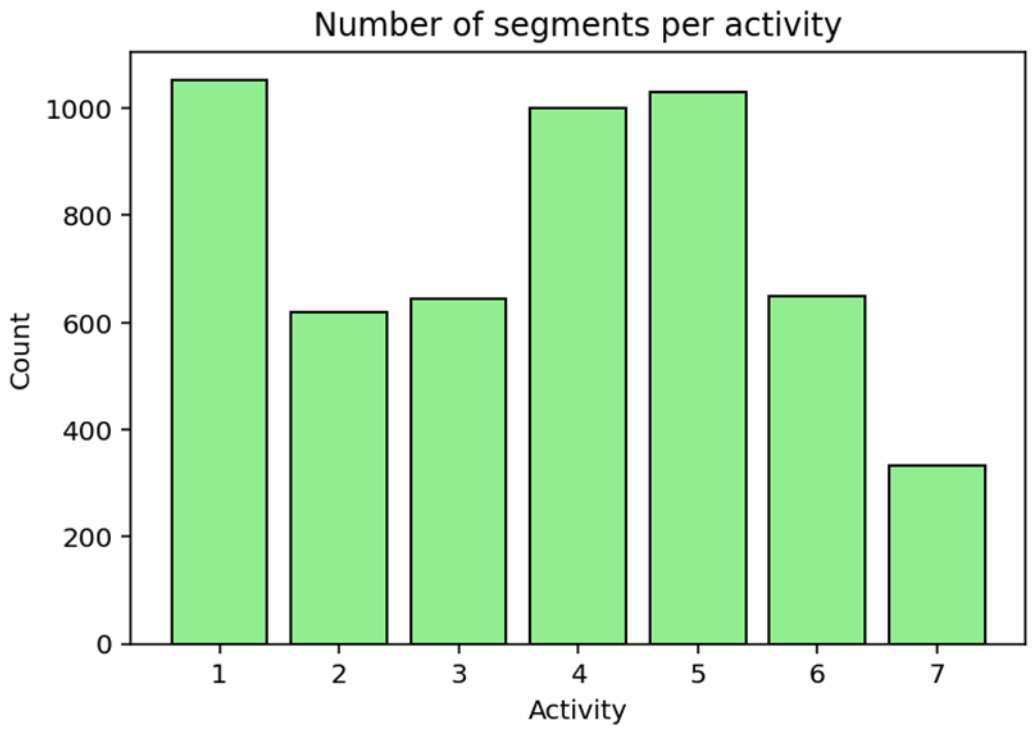
Another advantage is the reduction of noise from the different features, which brings a reduction in computational use; it requires less computation time using these reduced variables.

The main disadvantage from this approach is that the independence between variables may result in the loss of information that some variables have when combined, also another disadvantage is that it may have more computational use during the running of the methods.

GOAL 2

1.1 Analysis of the Balance in the Number of Examples per Activity:

To assess the distribution of examples within the dataset, we generated a bar plot illustrating the number of segments available for each activity. For this analysis, we included only activities labeled from 1 to 7, discarding the remaining activities from the original dataset to maintain a more focused and consistent scope.



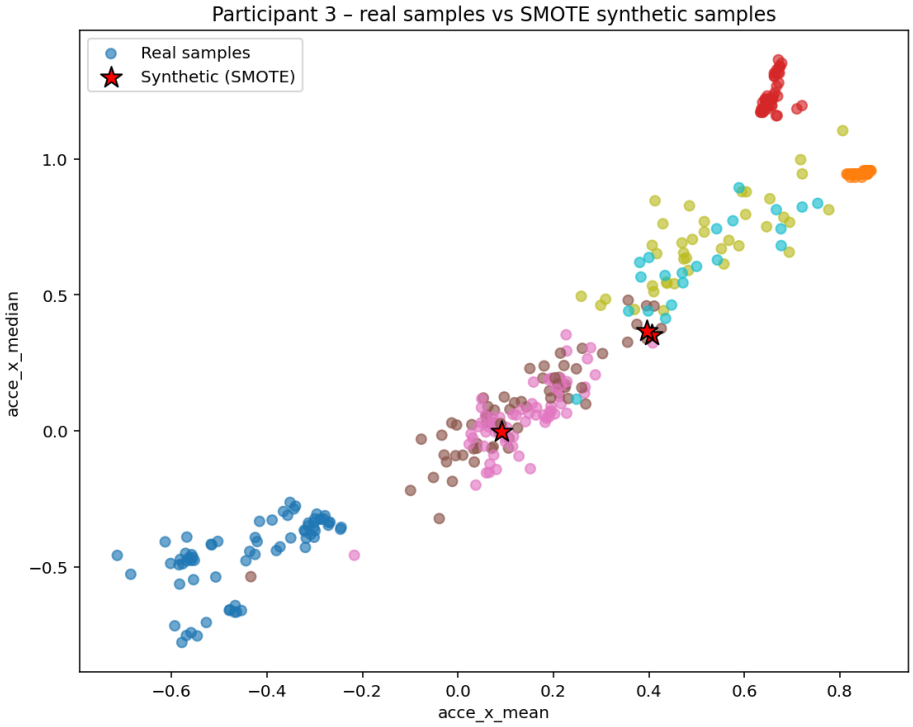
The resulting visualization clearly reveals a significant class imbalance. Activities 1, 4, and 5 are the most represented categories, each containing approximately 1052, 999, 1029 segments respectively, making them the dominant classes in the dataset.

In contrast, the activities 2, 3, and 6 exhibit a moderate number of samples, with roughly 618, 644, 648 examples per class. Although these activities are not severely underrepresented, they still present a noticeable difference compared to the highly populated classes.

The most underrepresented category is activity 7, with only 332 samples, making it the minority class and contributing heavily to the imbalance observed across the dataset.

This uneven distribution may affect model performance, especially in classification settings where the learning algorithm tends to favor majority classes. Depending on the modeling strategy, it may be necessary to apply balancing techniques such as class weighting, oversampling, undersampling, or synthetic data generation (SMOTE).

1.3 Scatter plot for Participant 3 – Activity 4 (real samples vs SMOTE synthetic samples)



The scatter plot illustrates the distribution of samples from activity 4 for participant 3, using only the first two features of the dataset. The colored dots represent the real observations, with each color corresponding to a different activity present in the participant's data. The synthetic samples generated using the SMOTE algorithm are highlighted with red star markers.

In this visualization, a total of three synthetic samples were produced, as required. One of them appears near the central region of the feature space, while the other two are positioned toward the right side and relatively close to each other. This behavior is expected from SMOTE, since it generates new samples by interpolating between existing minority-class neighbors, leading to synthetic points that follow the local structure of nearby real samples. Because the algorithm uses random selection during the interpolation process, the exact location of these synthetic samples may vary slightly from one execution to another.

3.3 Discussion of the differences between the two splitting strategies.

Within-participant split (60/20/20 for each participant):

In this approach, each participant contributes 60% of their data to the training set, 20% to the validation set, and 20% to the test set. As a result, all three sets contain data from every participant. This produces large and well-balanced splits, and

typically leads to higher performance, because the model is evaluated using data from participants it has already seen during training.

Between-participant split (9–3–3 participants):

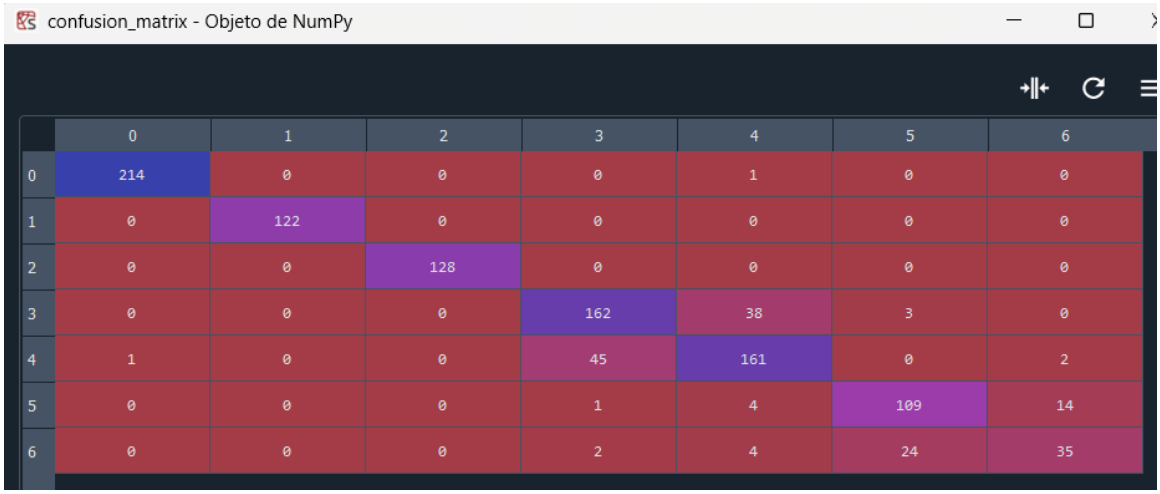
In this second strategy, the participant IDs are randomly shuffled, and then divided into 9 participants for training, 3 for validation, and 3 for testing. Here, each split contains completely different participants, which means the model must classify activity patterns from individuals whose data were never used during training.

Although the within-participant split often yields better accuracy, it does not reflect the performance of the model when applied to a new participant. This is because the model is tested on individuals whose movement patterns were seen during training, leading to an optimistic performance estimate.

On the other hand, the between participant split usually results in lower accuracy, but it provides a much more realistic estimate of how well the model will generalize to entirely new users. Since real world applications require recognizing activities from unseen individuals, the between-participant strategy is ultimately the more appropriate evaluation method.

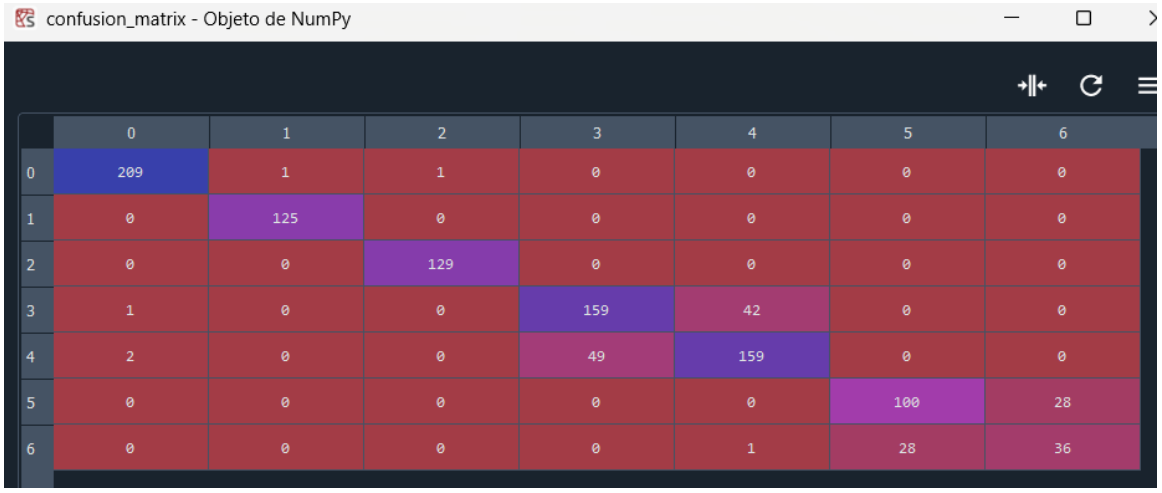
5.2 Overall Comparison and Best Model Analysis

After evaluating all models across different feature sets (features vs. embeddings), dimensionality reduction techniques (PCA and relief f), and data-splitting strategies (within subject vs. between subject), the results indicate that the relief f within features model tends to achieve the best overall performance. It is followed closely by the All within-features approach, which in several experiments showed only a slightly lower accuracy compared to the relief f within features model, these results are shown in the next figures.



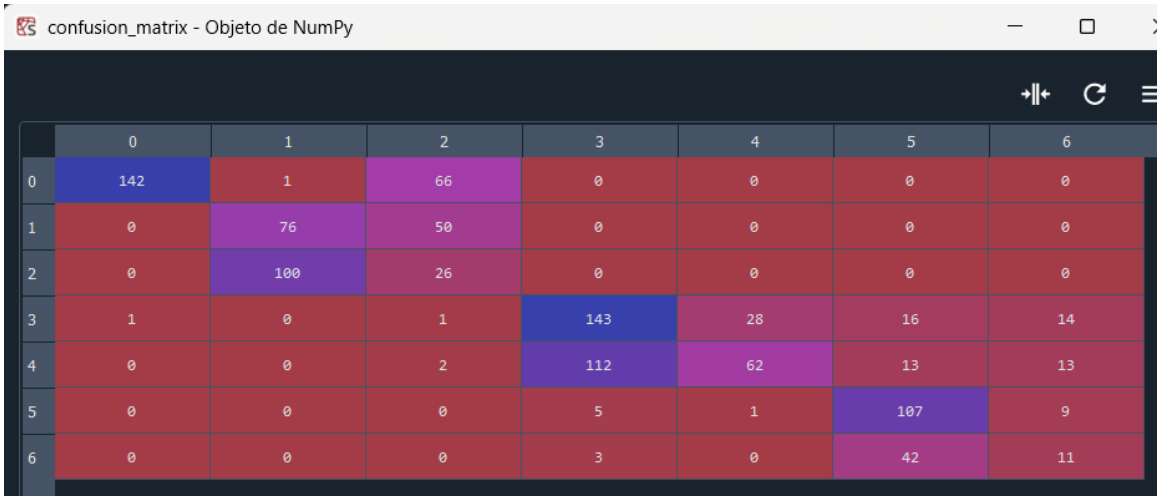
	0	1	2	3	4	5	6
0	214	0	0	0	1	0	0
1	0	122	0	0	0	0	0
2	0	0	128	0	0	0	0
3	0	0	0	162	38	3	0
4	1	0	0	45	161	0	2
5	0	0	0	1	4	109	14
6	0	0	0	2	4	24	35

Fig 1: ReliefF within features confusion matrix



	0	1	2	3	4	5	6
0	209	1	1	0	0	0	0
1	0	125	0	0	0	0	0
2	0	0	129	0	0	0	0
3	1	0	0	159	42	0	0
4	2	0	0	49	159	0	0
5	0	0	0	0	0	100	28
6	0	0	0	0	1	28	36

Fig 2: All within features confusion matrix



	0	1	2	3	4	5	6
0	142	1	66	0	0	0	0
1	0	76	50	0	0	0	0
2	0	100	26	0	0	0	0
3	1	0	1	143	28	16	14
4	0	0	2	112	62	13	13
5	0	0	0	5	1	107	9
6	0	0	0	3	0	42	11

Fig 3: All between features confusion matrix

The relief f within features model achieves near-perfect classification for activities 1, 2 and 3. Activities 4 and 5 (index 3 and 4), although partially confused with each other, maintain high correct prediction numbers. Activity 6 is well classified, with slightly confusion existing with activity 6. Activity 7, the most challenging class across all models, reaches 35 correct classifications, outperforming most models. This activity is confused more often with activity 6. This combination results in the highest diagonal dominance in the confusion matrix, indicating superior overall predictive capability.

When comparing features vs. embeddings, the features methods returned a better classification than embeddings when observed in the confusion matrix. In all models,

the activities classified returned different difficulties; even when they were the same method, the activities' classification resulted differently, sometimes having more trouble with a certain activity than the other model didn't. We observed that sometimes the difference wasn't big, but in some other models the difference was, resulting in all models of features with higher overall accuracies. Another form to confirm this is the use of the accuracy results, performing a sum of the features and getting the media, and doing the same for the embeddings, getting a value of 73% for the features and 62% for the embeddings, confirming that the difference was clear.

The comparison within vs. between presented the same results: a huge superiority of the within over the between method. The comparison was performed with the same models but different forms of splitting. When comparing the confusion matrices, all models resulted in a higher accuracy from the within method, a difference that this comparison has with the features vs. embeddings one is that the best model performed a better consistency when classifying, resulting in a huge dominance of the within method. Another form to confirm this is the use of the accuracy results, performing a sum of the features and getting the media, and doing the same for the embeddings, getting a value of 76% for the within and 59% for the between, resulting in a clear dominance.

For this next comparison, we observed the confusion matrix as well, but in this case, it is not a tuple comparison. This time, we performed a comparison between the four components of each method: the between and within, the features and embeddings for each method, resulting in a comparison of the three methods: the all, the PCA, and the ReliefF. The results showed a near classification performance. While each model struggled to classify activities between them, they were consistent. As mentioned before, the lowest accuracy percentage comes with activity number 7; this is performed due to it being the smallest set of information to train the model. At the end, the method with the most accuracies resulted in the all method. We also performed the accuracy median to observe how the data differ from each other, getting a 63% for the ReliefF, 68% for the PCA, and 70% for the all method.

The reason why the best method is not the combination of these three is because when the comparison is realized 1 vs. 1, the within ReliefF features presented a higher accuracy in most of the runs, but, as mentioned above, the all within features is the second-best method with a slight difference among them.

It is important to note that this model's exceptional performance is partly due to the within-subject evaluation, meaning the model has seen part of each participant's data during training. Therefore, although it achieves the highest accuracy, it does not provide a realistic measure of generalization to completely new participants. Between-subject models, while less accurate numerically, are more indicative of real-world performance on unseen individuals.

5.3 Hypothesis testing

Tecla	Tipo	Tamaño	
RELIEF_withinF	float64	1	np.float64(0.8454205607476636)
ALL_withinF	float64	1	np.float64(0.8429906542056076)
PCA_withinF	float64	1	np.float64(0.8266355140186915)
ALL_withinE	float64	1	np.float64(0.7172897196261683)
PCA_withinE	float64	1	np.float64(0.6640186915887851)
RELIEF_withinE	float64	1	np.float64(0.6383177570093459)
PCA_betweenF	float64	1	np.float64(0.6174305075615556)
ALL_betweenF	float64	1	np.float64(0.6133776688847768)
ALL_betweenE	float64	1	np.float64(0.5747118763624721)
RELIEF_betweenF	float64	1	np.float64(0.5572275401905962)
PCA_betweenE	float64	1	np.float64(0.5515131553968151)
RELIEF_betweenE	float64	1	np.float64(0.5347535523445567)

The Kolmogorov Smirnov test is highly sensitive to sample size, whereas the Shapiro Wilk test provides better accuracy for smaller datasets. For this reason, we used the Shapiro test in our analysis, as we only had 10 splits for evaluation. In small sample scenarios (10 splits), the distribution appeared normal; however, it is not, when using a larger number of splits, the distribution result indicated no normality. That's why we applied a non-parametric test.

Since all comparisons were performed on the same data subsets (same segments), the experiments were paired. Reason why, the Friedman test was employed to assess the presence of significant differences among the models.
The results indicated that significant differences do exist between the methods, confirming that relief f within features consistently demonstrates superior performance.

Example:

> **In the within-split analysis, the best model obtained a p value of 3.0917754466602e-09, indicating a highly significant difference relative to the other models.**
> **In the between-split analysis, the best model had a p value of 0.03178656029465204, showing statistical significance, though not as strong as in the within-split case.**

After the Friedman test, we performed the Nemenyi post hoc test to confirm pairwise statistical differences among the models. For the within split results, the best model was relief f within features, followed by All within features, with a p value of 1, indicating no significant difference between these two models.
For the between-split results, the best model was PCA between features, which obtained a p value of 0.999997 when compared to All between features, also indicating a non-significant difference.

After completing these tests, we used a function to identify the best overall model among the between split evaluations. We examined the mean values obtained from the hypothesis tests and selected the highest one, PCA between features. Then, we retrieved the corresponding data used for this classifier (training set, optimal *k* value, and other relevant parameters) to conduct the final evaluation with the selected model.

The use of this between method is due to its application in real life tasks where unknown subjects will be introduced to the system, making it more realistic and getting a real classifying accurate rate.

7. IMPROVEMENTS

The current work uses k-Nearest Neighbors as the classification model. Being kind of simple and interpretable, KNN is often outperformed by more powerful models. Therefore, one possible improvement is to evaluate alternative classifiers such as Random Forests and Gradient Boosted Trees. These models could capture more complex decision boundaries and potentially improve accuracy, especially for activities that are difficult to distinguish.

Another possibility is to incorporate self-supervised learning strategies directly into the raw signals of this project. Methods such as contrastive learning, masked-signal modeling, or temporal prediction tasks could be applied to pretrain a model on the available segments before performing supervised classification. This would allow the system to discover structure in the wearable data without requiring additional labels, often leading to better feature representations compared to using an external pretrained model.

One possible enhancement is to combine oversampling and undersampling strategies. Undersampling the majority classes while carefully preserving their variability can prevent the classifier from being dominated by high frequency activities such as standing or sitting. When coupled with oversampling methods, this hybrid approach can produce a more uniform training set and reduce bias. Moreover, augmentation can be performed not only in the feature domain but also directly on the raw time-series signals, generating new balanced sample.

The outlier analysis performed in the first milestone could be applied in the current workflow. Removing anomalous segments, either in the raw sensor signals or in the extracted feature space, would help the classifier learn from cleaner and more consistent data. This could improve the model's robustness, reduce bias from extreme values, and potentially enhance overall accuracy.

PORTUGUESE VERSION

Introdução

Este estudo utiliza o conjunto de dados **FORTH-TRACE**, composto por informações de cinco sensores (pulso esquerdo/direito, peito, perna superior direita, perna inferior esquerda) provenientes de 15 participantes, com 16 atividades diferentes.

Todos os itens indicados na proposta foram implementados no ficheiro *mainActivity.py*, com foco em:

- Análise e tratamento de valores atípicos (*outliers*) (IQR, Z-Score, KMeans, DBSCAN)
- Utilização de janelas temporais de 5 segundos com sobreposição de 50% para segmentar os dados
- Extração de características temporais e espectrais utilizando janelas (artigo de Zhang & Sawchuk)
- Redução da dimensionalidade do conjunto de dados (PCA)
- Implementação do *Fisher Score* e do *ReliefF* para seleção das melhores características.

3. Análise e tratamento de *outliers*

Nesta etapa, foi realizada uma análise de potenciais valores atípicos nos dados dos sensores. A deteção de valores extremos baseou-se em métricas estatísticas como o **Z-score** e o **Intervalo Interquartil (IQR)**, o que permitiu identificar observações que se desviam significativamente do comportamento típico das medições.
Posteriormente, foi utilizado o algoritmo de agrupamento **K-Means** para identificar segmentos de dados anómalos com base na distância aos centróides dos grupos (*clusters*).

As amostras cuja distância excedia o limite superior interquartil (Q3 + 1.5 × IQR) foram rotuladas como potenciais *outliers*.

3.2 Análise e comentário sobre a densidade de *outliers* no conjunto de dados transformado

A análise da densidade de *outliers*, realizada através do método IQR (Tukey) sobre os módulos do sensor do pulso direito, revelou diferenças claras entre tipos de sensores e níveis de atividade.

Para o **acelerómetro**, as densidades de *outliers* variaram de menos de 1% em atividades estáticas (por exemplo, IDs 2–4) até mais de 15–20% em atividades altamente dinâmicas, como 8, 9, 10 e 11.
Este comportamento reflete a forte variabilidade nos padrões de aceleração durante

movimentos rápidos ou vigorosos, indicando que muitos destes "*outliers*" correspondem, na verdade, a movimentos legítimos de alta intensidade.

O **giroscópio** apresentou densidades moderadas de *outliers* (1–12%), com valores mais elevados em atividades com forte rotação (9–11), coerente com maiores variações de velocidade angular.

O **magnetómetro**, por outro lado, manteve-se amplamente estável em todas as atividades, com densidades de *outliers* abaixo de 7%, confirmando que as leituras do campo magnético são menos afetadas pelo movimento.

outliers_density_device2_IRQ - DataFrame

Índice	device_id	module	ıctivity_labe	lier_density
0	2	accel_module	1	4.19554
1	2	accel_module	2	0.215558
2	2	accel_module	3	0.513851
3	2	accel_module	4	3.53095
4	2	accel_module	5	3.47766
5	2	accel_module	6	5.16568
6	2	accel_module	7	4.51463
7	2	accel_module	8	15.4447
8	2	accel_module	9	19.7976
9	2	accel_module	10	15.3245
10	2	accel_module	11	18.3222
11	2	accel_module	12	10.9186
12	2	accel_module	13	4.79003
13	2	accel_module	14	14.1732
14	2	accel_module	15	4.93438
15	2	accel_module	16	4.98688
16	2	gyro_module	1	9.48493
17	2	gyro_module	2	6.64658
18	2	gyro_module	3	9.11746
19	2	gyro_module	4	1.62076
20	2	gyro_module	5	1.42801
21	2	gyro_module	6	1.49845
22	2	gyro_module	7	2.07349
23	2	gyro_module	8	7.46695
24	2	gyro_module	9	10.7777
25	2	gyro_module	10	10.3365
26	2	gyro_module	11	11.8676
27	2	gyro_module	12	3.30709
28	2	gyro_module	13	8.25459
29	2	gyro_module	14	2.3622
30	2	gyro_module	15	2.3622
31	2	gyro_module	16	2.04724
32	2	mag_module	1	0
33	2	mag_module	2	6.67278
34	2	mag_module	3	4.76636
35	2	mag_module	4	1.45068
36	2	mag_module	5	1.3388
37	2	mag_module	6	0.390559
38	2	mag_module	7	0.221507
39	2	mag_module	8	3.48558
40	2	mag_module	9	7.35085
41	2	mag_module	10	0.841346
42	2	mag_module	11	6.82664
43	2	mag_module	12	0.0262467
44	2	mag_module	13	0
45	2	mag_module	14	6.66667
46	2	mag_module	15	6.66667
47	2	mag_module	16	1.5748

De forma geral, os resultados mostram que a densidade de *outliers* está fortemente correlacionada com a intensidade do movimento, sugerindo que muitas destas aparentes anomalias refletem variabilidade fisiológica genuína em vez de ruído de medição

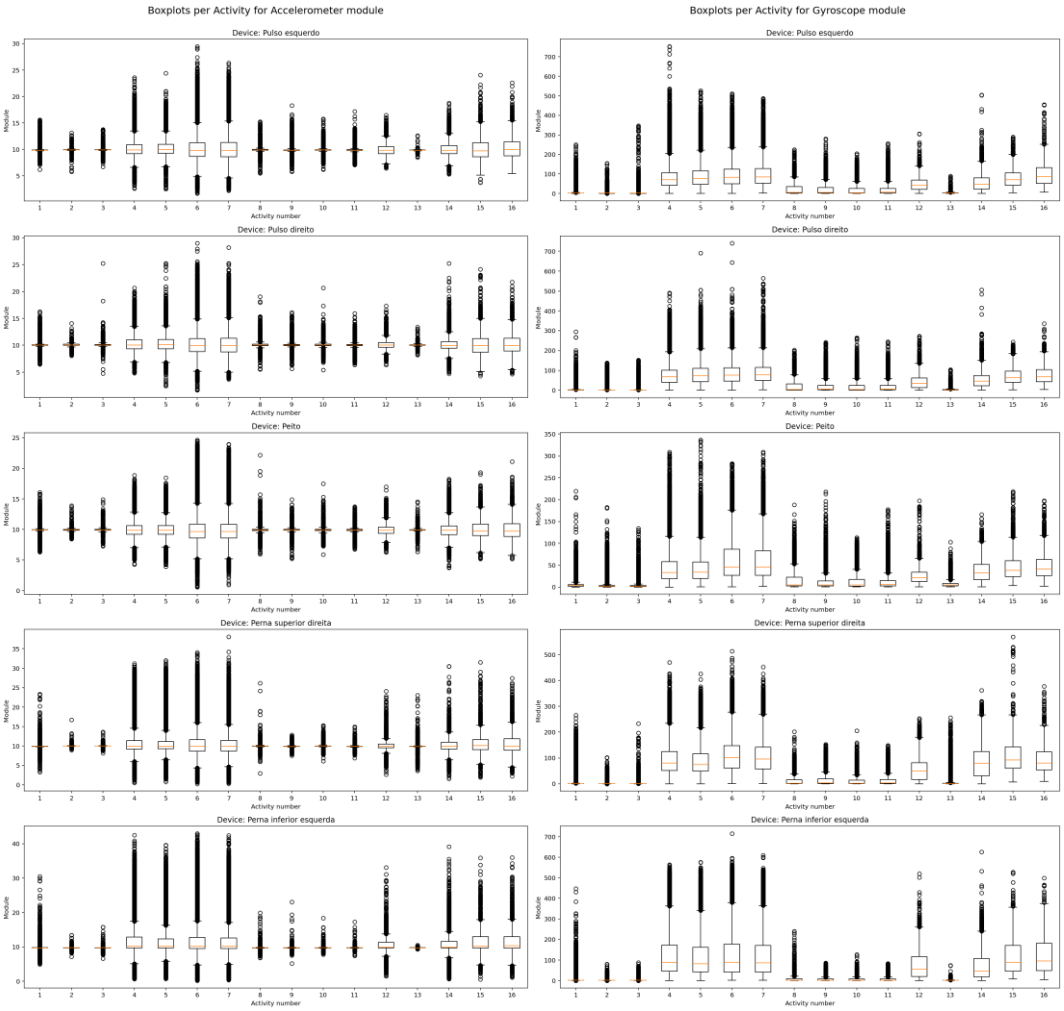
3.5 Comparação e discussão dos resultados obtidos em 3.1 e 3.4

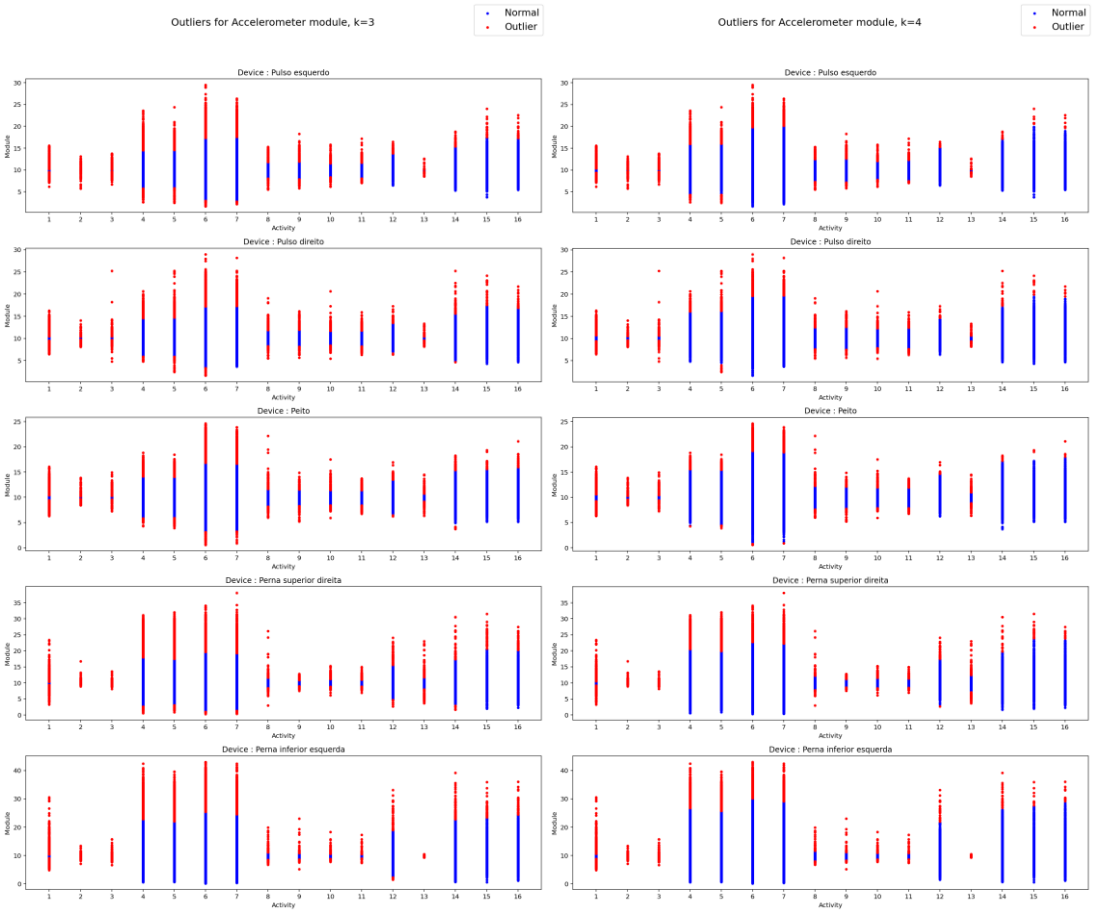
Os gráficos tipo *boxplot* apresentados na secção 3.1 foram criados utilizando o IQR sobre os módulos do acelerómetro, giroscópio e magnetómetro para cada atividade e dispositivo. Estes gráficos permitem observar a forma como os dados estão distribuídos e, assim, identificar facilmente os *outliers*.

Nos gráficos gerados na secção 3.4, foi calculado o **Z-score** com diferentes valores de limiar (k = 3, 3.5, 4), sendo os *outliers* identificados de acordo com o valor de k. Nos gráficos resultantes, os *outliers* são apresentados a vermelho e os dados normais a azul, o que facilita a sua identificação.

Em ambos os casos, os resultados são semelhantes: os *outliers* surgem em posições comparáveis e os dados normais mantêm uma distribuição consistente, sofrendo apenas pequenas variações com diferentes valores de k.
Ainda assim, os gráficos coloridos são mais intuitivos de interpretar, embora os outros também sejam úteis e complementares — a utilização de ambos os métodos conduz a uma identificação mais completa e precisa de *outliers*.

De acordo com ambos os gráficos, o **módulo com maior número de *outliers*** é o do **acelerómetro**, seguido do giroscópio, onde quase todos os dados são classificados como *outliers*.


Boxplots per Activity for Accelerometer module
Boxplots per Activity for Gyroscope module
Device: Pulso esquerdo
Device: Pulso direito
Device: Peito
Device: Perna superior direita
Device: Perna inferior esquerda
Module
Activity number



De forma geral, os **dispositivos colocados nas pernas**, especialmente na **perna inferior esquerda**, apresentam uma grande quantidade de *outliers*, com numerosos pontos vermelhos que se estendem acima e abaixo dos valores normais. Observa-se também uma grande dispersão nas **atividades 4 a 7** em quase todos os dispositivos, indicando maior variabilidade ou movimentos mais bruscos nessas tarefas.

Em contraste, os sensores localizados no **peito** e nos **pulsos** apresentam menos *outliers* e uma distribuição mais concentrada, sugerindo sinais mais estáveis e menos suscetíveis a valores extremos.

Em particular, a **atividade 6** destaca-se por possuir as amplitudes mais elevadas em quase todos os dispositivos, especialmente nas pernas.

O módulo do **magnetómetro** apresenta um número menor de *outliers* em comparação com os outros módulos, embora registe praticamente a mesma distribuição de dados.

3.7 Agrupamento K-Means e comparação com os resultados de 3.4

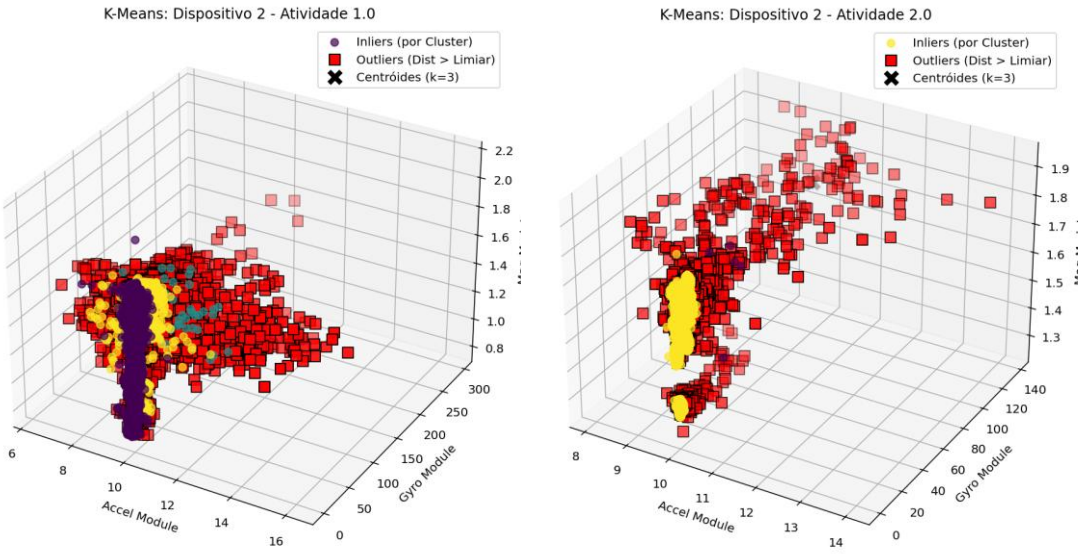
A deteção de *outliers* foi também realizada através do algoritmo de agrupamento **K-Means**, aplicado ao espaço dos módulos dos sinais de acelerómetro, giroscópio e magnetómetro.

O algoritmo foi executado com múltiplas configurações de grupos (k = 3, 4) para avaliar a estabilidade dos resultados.

Os *outliers* foram definidos como amostras cuja distância euclidiana ao centróide do grupo excedia 1,5 vezes o intervalo interquartil (IQR) da respetiva distribuição.
Comparado com a abordagem baseada em Z-score — que identifica valores estatisticamente extremos em eixos individuais —, o K-Means deteta desvios multivariados, considerando o comportamento conjunto das três componentes espaciais.
Isto permite uma deteção mais robusta de padrões de movimento atípicos, especialmente em atividades com movimentos multidirecionais complexos.

Os gráficos resultantes mostram a distribuição de *outliers* por atividade e dispositivo. Curiosamente, observou-se **maior densidade de *outliers* em atividades estáticas** como *Stand*, *Sit* e *Sit and Talk* (atividades 1–3).
Isto deve-se ao facto de estas atividades possuírem grupos muito compactos e de baixa variância; assim, mesmo pequenas flutuações nas leituras dos sensores são detetadas como anomalias.

Por outro lado, atividades dinâmicas como *Walk*, *Climb Stairs* e várias transições (atividades 4–16) apresentaram grupos mais dispersos, resultando numa menor proporção de *outliers* em relação à sua variabilidade natural.



As visualizações 3D confirmam este comportamento, mostrando que os *outliers* se localizam principalmente na periferia dos grupos densos, afastados dos centróides.
Estes resultados sugerem que os *outliers* detetados correspondem a amostras com magnitudes de movimento atípicas dentro de padrões de atividade geralmente estáveis.

Comparando com o método do Z-score (secção 3.4), o **K-Means** oferece uma abordagem contextual e dependente da atividade. Enquanto o Z-score identifica extremos globais no conjunto de dados, o K-Means deteta pontos inconsistentes dentro do seu próprio grupo.

De modo geral, ambos os métodos evidenciam que as atividades estáticas são mais sensíveis a pequenas flutuações dos sensores, enquanto as dinâmicas apresentam variabilidade mais ampla, mas mais suave.
Isto reforça o valor do K-Means na deteção de anomalias locais específicas de cada contexto de movimento.

4. Extração de informação característica

O objetivo desta etapa é **comprimir o espaço do problema** através da extração de informação característica discriminativa dos dados brutos dos sensores.
Este processo visa identificar e selecionar características estatísticas, temporais e espectrais significativas que melhor representem a variabilidade das atividades humanas.
Ao transformar medições de alta dimensionalidade num conjunto reduzido e mais informativo, torna-se possível conceber modelos de classificação mais eficientes e precisos, capazes de distinguir entre diferentes padrões e comportamentos de movimento.

4.1 Teste de Kolmogorov–Smirnov

Para avaliar se as medições dos sensores diferiam significativamente entre atividades, foi primeiro verificada a normalidade das distribuições das características utilizando o teste de **Kolmogorov–Smirnov**.
Como a maioria das variáveis não seguiu uma distribuição normal ($p < 0.05$), aplicou-se o teste não paramétrico **Kruskal–Wallis**.

Os resultados foram altamente significativos para todos os módulos:

- Acelerómetro ($H = 41\,668.08$, $p < 0.001$)
- Giroscópio ($H = 2\,821\,766.19$, $p < 0.001$)
- Magnetómetro ($H = 608\,088.21$, $p < 0.001$).

Estes resultados indicam que as distribuições (ou valores medianos) das características diferem significativamente entre atividades.
Em particular, as características do **giroscópio** apresentaram as maiores estatísticas, sugerindo que os padrões de movimento rotacional variam mais fortemente entre tipos de atividade, fornecendo assim informação discriminativa relevante.

4.2 Identificação de características

Durante o processo de **Reconhecimento de Atividades Humanas (HAR – *Human Activity Recognition*)**, é possível extrair um conjunto de características a partir da informação dos sensores, que fornece mais dados sobre a atividade registada.
Estas características ajudam a distinguir entre diferentes tipos de atividades.

As características podem ser **temporais** ou **espectrais**, dependendo da forma como são obtidas.
As **características temporais** são extraídas diretamente dos sinais brutos dos sensores, enquanto as **características espectrais** são calculadas a partir de informação transformada (por exemplo, via transformada de Fourier).

Deste modo, foram calculadas diversas características para obter os melhores dados possíveis para o reconhecimento, incluindo:

- média, mediana, desvio padrão, variância, raiz quadrada média (*root mean square*), derivada média, assimetria (*skewness*), curtose, intervalo interquartil (IQR), média de cruzamentos, número de cruzamentos por zero, energia, frequência dominante, intensidade média, variância da intensidade, correlação entre aceleração segundo a gravidade e a direção do movimento, área normalizada da magnitude do sinal (*signal magnitude area*), autovalores das direções dominantes, energia média da aceleração e energia média da rotação — todas desenvolvidas de acordo com o ficheiro fornecido.

4.4 Importância das componentes principais (PCA)

A **PCA (Análise de Componentes Principais)** é uma técnica de transformação e redução de dados, na qual a informação é convertida e reduzida em diferentes componentes que são combinações lineares das variáveis originais — neste caso, as características extraídas.

O seu objetivo é criar novas variáveis que preservem a variância dos dados, de forma que as primeiras componentes representem a maior quantidade possível de informação.

Se for necessário representar uma percentagem específica da informação, pode-se observar a **variância explicada acumulada**, que indica a percentagem de informação representada por um determinado número de componentes.

Neste caso, a **primeira componente sozinha explica cerca de 31%**, e as **onze primeiras componentes** juntas representam aproximadamente **75% da variância total**.
Isto indica que uma parte significativa da informação do espaço original de alta dimensionalidade pode ser eficazmente representada por um número reduzido de componentes, diminuindo a dimensionalidade sem perda substancial de poder discriminativo.

4.4.1 Obtenção de características relacionadas com a PCA

Para obter as novas características resultantes desta compressão, utiliza-se a expressão:

$\boldsymbol{X_{PCA}} = \boldsymbol{X_{scaled}} * \mathbf{W}$

onde:

- $\boldsymbol{X_{scaled}}$ é a matriz de características normalizadas, e
- W é a matriz que contém os vetores das componentes principais.

Cada linha de $\boldsymbol{X_{PCA}}$ representa uma observação (ou janela temporal) descrita pelas novas características comprimidas.

Por exemplo, considerando três características originais $(\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3})$, e duas componentes principais definidas pela seguinte matriz de pesos:

$$\boldsymbol{W} = \begin{bmatrix} \mathbf{0.5} & \mathbf{0.7} \\ \mathbf{0.3} & \mathbf{-0.6} \\ \mathbf{0.8} & \mathbf{0.4} \end{bmatrix}$$

Para uma amostra com valores de características $[\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}] = [\mathbf{2.0}, \mathbf{1.0}, \mathbf{3.0}]$, as novas características PCA são calculadas como:

Assim, a amostra original $[\mathbf{2.0}, \mathbf{1.0}, \mathbf{3.0}]$ é representada no espaço comprimido da PCA como $[\mathbf{3.7}, \mathbf{2.2}]$

4.4.2 Vantagens e limitações da PCA

A técnica PCA apresenta diversas **vantagens** e **limitações**, dependendo do seu propósito e contexto de utilização.

Vantagens:

- **Redução da dimensionalidade:** diminui o espaço necessário para armazenar os dados, preservando a maior parte da informação — o que conduz a um processamento mais rápido.
- **Redução de ruído:** elimina características redundantes ou pouco informativas, melhorando a qualidade global dos dados.

Limitações:

- As novas componentes são **combinações lineares** das originais, o que reduz a sua interpretabilidade intuitiva.

- Em conjuntos de dados muito grandes, como neste estudo, mesmo após a redução, podem ser necessárias mais de 10 componentes para representar adequadamente os dados originais, o que dificulta a interpretação direta.

4.6 As 10 principais características segundo Fisher Score e ReliefF

Os métodos **Fisher Score** e **ReliefF** foram aplicados para identificar as **dez características mais relevantes** para distinguir entre as atividades.
Ambas as técnicas medem a capacidade de cada característica em separar as classes de atividade, mas baseiam-se em princípios distintos.

Os resultados obtidos foram:

Fisher Score:
[acce_y_rms, gyro_y_rms, gyro_y_std, acce_z_rms, gyro_x_rms, gyro_x_std, gyro_y_IQR, gyro_x_IQR, acce_y_median, gyro_y_avg_der]

ReliefF:
[magne_y_var, acce_z_var, acce_y_rms, acce_z_energy, magne_y_, magne_x_energy, acce_y_median, acce_y_mean, magne_y_mean, acce_z_rms]

O **Fisher Score** dá prioridade a características que apresentam grandes diferenças entre classes e pouca variação dentro de cada classe.
O **ReliefF**, por sua vez, avalia a importância local de cada característica, comparando amostras vizinhas da mesma classe e de classes diferentes.

Os resultados mostram que os dois métodos selecionaram **conjuntos distintos de características**, com apenas **três características em comum**: *acce_y_median*, *acce_y_rms* e *acce_z_rms*.

4.6.1 Obtenção de características segundo Fisher Score e ReliefF

Após a aplicação destes métodos, é possível obter as melhores características selecionadas.
Se se desejar criar um conjunto de dados apenas com essas características, pode-se utilizar o seguinte comando:

df = df[selected_features]

onde `selected_features` são os nomes das colunas desejadas.

Desta forma, o conjunto de dados mantém apenas as colunas correspondentes às características selecionadas, simplificando o processo de análise posterior.

4.6.2 Vantagens e limitações dos métodos de pontuação

Tal como a PCA, estes métodos também possuem **vantagens** e **limitações**.

Vantagens:

- Permitem **redução da dimensionalidade**, preservando a interpretabilidade das variáveis originais.
- Reduzem o **ruído** entre características redundantes, o que diminui o custo computacional e o tempo de processamento.

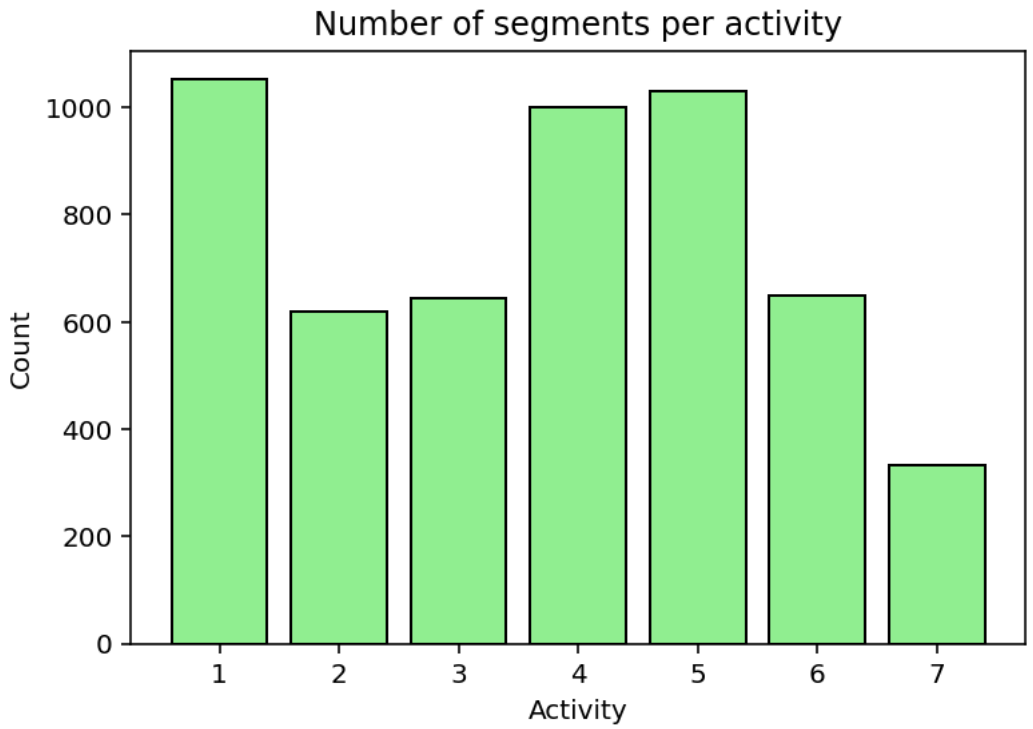
Limitações:

- A suposição de **independência entre variáveis** pode levar à perda de informação presente em combinações de características.
- A execução destes métodos pode exigir **maior custo computacional** durante o cálculo das pontuações.

META 2

1.1 Análise do equilíbrio no número de exemplos por atividade

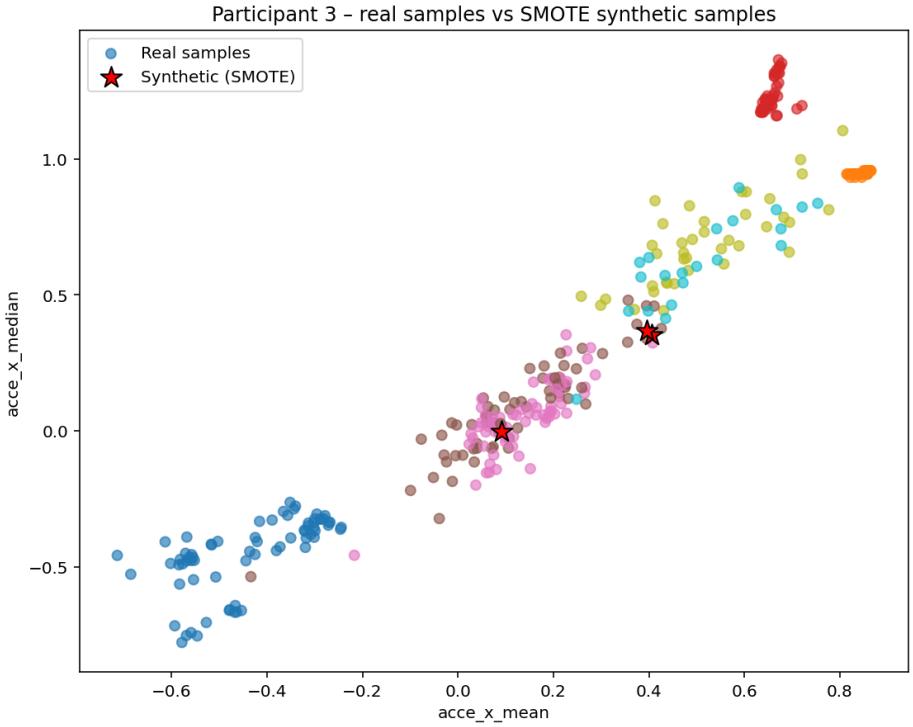
Para avaliar a distribuição de exemplos dentro do conjunto de dados, foi gerado um gráfico de barras que ilustra o número de segmentos disponíveis para cada atividade. Para esta análise, foram incluídas apenas as atividades rotuladas de 1 a 7, descartando as demais atividades do conjunto de dados original para manter um escopo mais focado e consistente.



A visualização resultante revela claramente um desbalanceamento significativo entre as classes. As atividades 1, 4 e 5 são as categorias mais representadas, contendo aproximadamente 1052, 999 e 1029 segmentos, respetivamente, tornando-se as classes dominantes no conjunto de dados.
Em contraste, as atividades 2, 3 e 6 apresentam um número moderado de amostras, com aproximadamente 618, 644 e 648 exemplos por classe. Embora estas atividades não estejam severamente sub-representadas, ainda apresentam uma diferença notável quando comparadas com as classes mais populosas.
A categoria mais sub-representada é a atividade 7, com apenas 332 amostras, tornando-se a classe minoritária e contribuindo fortemente para o desbalanceamento observado no conjunto de dados.

Esta distribuição desigual pode afetar o desempenho do modelo, especialmente em cenários de classificação em que o algoritmo de aprendizagem tende a favorecer as classes majoritárias. Dependendo da estratégia de modelagem, pode ser necessário aplicar técnicas de balanceamento, como ponderação de classes (*class weighting*), *oversampling*, *undersampling* ou geração sintética de dados (SMOTE).

1.3 Gráfico de dispersão para o Participante 3 – Atividade 4 (amostras reais vs. amostras sintéticas SMOTE)



O gráfico de dispersão ilustra a distribuição das amostras da atividade 4 para o participante 3, utilizando apenas as duas primeiras características do conjunto de dados. Os pontos coloridos representam as observações reais, com cada cor correspondendo a uma atividade diferente presente nos dados do participante. As amostras sintéticas geradas pelo algoritmo SMOTE são destacadas com marcadores em forma de estrela vermelha.

Nesta visualização, foram produzidas um total de três amostras sintéticas, conforme necessário. Uma delas aparece próxima da região central do espaço de características, enquanto as outras duas estão posicionadas mais à direita e relativamente próximas entre si. Este comportamento é esperado no SMOTE, uma vez que gera novas amostras por interpolação entre vizinhos da classe minoritária existente, levando a pontos sintéticos que seguem a estrutura local das amostras reais próximas. Como o algoritmo utiliza seleção aleatória durante o processo de interpolação, a localização exata destas amostras sintéticas pode variar ligeiramente de uma execução para outra.

3.3 Discussão das diferenças entre as duas estratégias de divisão

Divisão dentro do participante (60/20/20 por participante):
Nesta abordagem, cada participante contribui com 60% dos seus dados para o conjunto de treino, 20% para o conjunto de validação e 20% para o conjunto de teste. Como resultado, os três conjuntos contêm dados de todos os participantes. Isto produz divisões grandes e

bem balanceadas e, geralmente, conduz a um melhor desempenho, pois o modelo é avaliado utilizando dados de participantes que já foram vistos durante o treino.

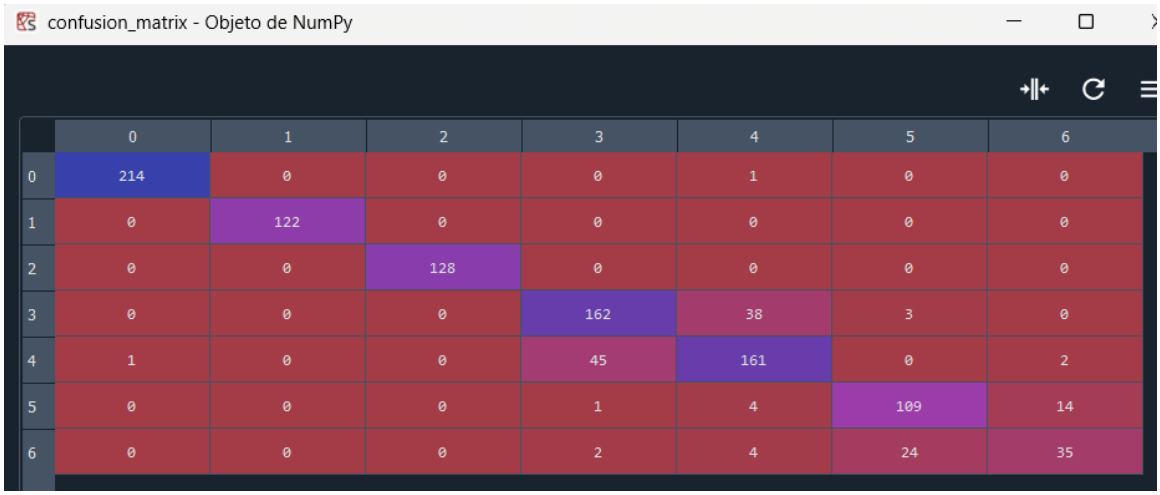
Divisão entre participantes (9–3–3 participantes):
Nesta segunda estratégia, os IDs dos participantes são embaralhados aleatoriamente e depois divididos em 9 participantes para treino, 3 para validação e 3 para teste. Aqui, cada divisão contém participantes completamente diferentes, o que significa que o modelo deve classificar padrões de atividade de indivíduos cujos dados nunca foram utilizados durante o treino.

Embora a divisão dentro do participante frequentemente produza maior acurácia, ela não reflete o desempenho do modelo quando aplicado a um novo participante. Isto ocorre porque o modelo é testado em indivíduos cujos padrões de movimento foram vistos durante o treino, levando a uma estimativa de desempenho otimista.

Por outro lado, a divisão entre participantes normalmente resulta em menor acurácia, mas fornece uma estimativa muito mais realista da capacidade de generalização do modelo para usuários totalmente novos. Como aplicações do mundo real exigem o reconhecimento de atividades de indivíduos não vistos, a estratégia entre participantes é, em última análise, o método de avaliação mais apropriado.

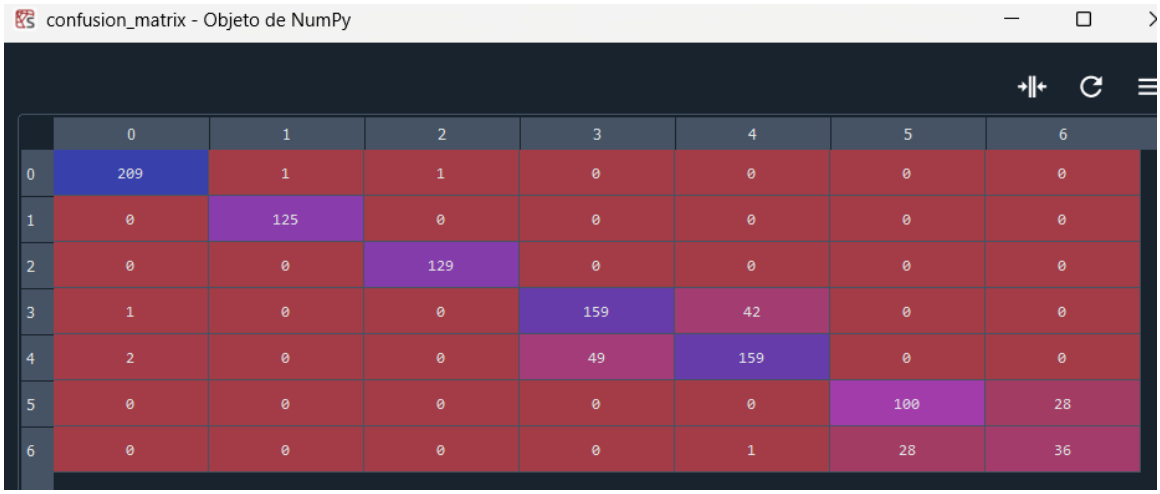
5.2 Comparação geral e análise do melhor modelo

Após a avaliação de todos os modelos em diferentes conjuntos de características (*features* vs. *embeddings*), técnicas de redução de dimensionalidade (PCA e ReliefF) e estratégias de divisão de dados (dentro do sujeito vs. entre sujeitos), os resultados indicam que o modelo ReliefF com features e divisão dentro do participante tende a alcançar o melhor desempenho geral. Ele é seguido de perto pela abordagem All com features e divisão dentro do participante, que em vários experimentos apresentou apenas uma acurácia ligeiramente inferior em comparação com o modelo ReliefF.



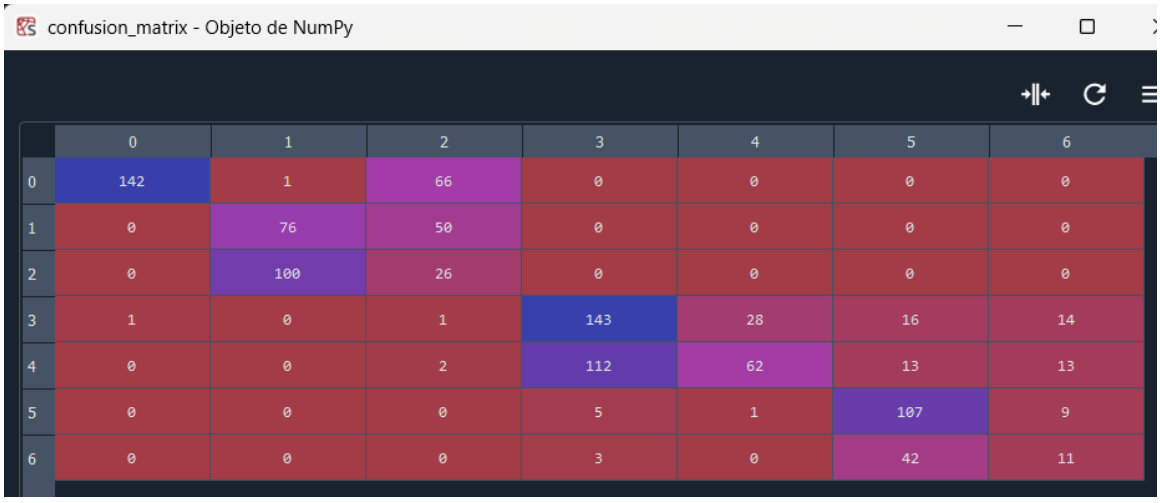
	0	1	2	3	4	5	6
0	214	0	0	0	1	0	0
1	0	122	0	0	0	0	0
2	0	0	128	0	0	0	0
3	0	0	0	162	38	3	0
4	1	0	0	45	161	0	2
5	0	0	0	1	4	109	14
6	0	0	0	2	4	24	35

Fig. 1: Matriz de confusão do ReliefF com features (within)



	0	1	2	3	4	5	6
0	209	1	1	0	0	0	0
1	0	125	0	0	0	0	0
2	0	0	129	0	0	0	0
3	1	0	0	159	42	0	0
4	2	0	0	49	159	0	0
5	0	0	0	0	0	100	28
6	0	0	0	0	1	28	36

Fig. 2: Matriz de confusão do All com features (within)



	0	1	2	3	4	5	6
0	142	1	66	0	0	0	0
1	0	76	50	0	0	0	0
2	0	100	26	0	0	0	0
3	1	0	1	143	28	16	14
4	0	0	2	112	62	13	13
5	0	0	0	5	1	107	9
6	0	0	0	3	0	42	11

Fig. 3: Matriz de confusão do All com features (between)

O modelo ReliefF com features (within) alcança uma classificação quase perfeita para as atividades 1, 2 e 3. As atividades 4 e 5 (índices 3 e 4), apesar de parcialmente confundidas entre si, mantêm elevados números de predições corretas. A atividade 6 é bem classificada, com ligeira confusão ainda existente. A atividade 7, a classe mais desafiadora em todos os modelos, atinge 35 classificações corretas, superando a maioria dos modelos. Esta atividade é mais frequentemente confundida com a atividade 6. Esta combinação resulta na maior dominância diagonal na matriz de confusão, indicando superior capacidade preditiva geral.

Ao comparar *features* vs. *embeddings*, os métodos baseados em *features* retornaram melhor classificação do que os *embeddings* quando observadas as matrizes de confusão. Em todos os modelos, as atividades classificadas apresentaram diferentes níveis de dificuldade; mesmo quando o método era o mesmo, a classificação das atividades resultou de forma diferente, por vezes tendo mais dificuldade com determinada atividade do que o outro modelo. Observou-se que, em alguns casos, a diferença não era grande, mas em outros modelos a diferença foi significativa, resultando em todos os modelos baseados em *features*

com maiores acurácias gerais. Outra forma de confirmar isto é através dos resultados de acurácia, somando os valores dos modelos com *features* e calculando a média, e fazendo o mesmo para os *embeddings*, obtendo-se um valor de 73% para *features* e 62% para *embeddings*, confirmando que a diferença foi clara.

A comparação *within* vs. *between* apresentou os mesmos resultados: uma enorme superioridade do método *within* sobre o *between*. A comparação foi realizada com os mesmos modelos, mas com diferentes formas de divisão. Ao comparar as matrizes de confusão, todos os modelos apresentaram maior acurácia com o método *within*. Uma diferença desta comparação em relação à de *features* vs. *embeddings* é que o melhor modelo apresentou maior consistência na classificação, resultando numa grande dominância do método *within*. Outra forma de confirmar isto é através dos resultados de acurácia, somando os valores e calculando a média, obtendo-se 76% para *within* e 59% para *between*, resultando numa dominância clara.

Para a próxima comparação, também observámos a matriz de confusão, mas neste caso não se trata de uma comparação em pares. Desta vez, foi realizada uma comparação entre os quatro componentes de cada método: *between* e *within*, *features* e *embeddings*, resultando numa comparação dos três métodos: All, PCA e ReliefF. Os resultados mostraram um desempenho de classificação próximo. Embora cada modelo tenha tido dificuldades em classificar as atividades entre si, eles foram consistentes. Conforme mencionado anteriormente, a menor percentagem de acurácia ocorre com a atividade número 7, devido a ser o menor conjunto de informação para treinar o modelo. No final, o método com maior número de acertos foi o método All. Também foi calculada a mediana das acurácias, obtendo-se 63% para ReliefF, 68% para PCA e 70% para All.

A razão pela qual o melhor método não é a combinação dos três é que, quando a comparação é realizada 1 vs. 1, o modelo ReliefF com *features* e divisão *within* apresentou maior acurácia na maioria das execuções. Contudo, conforme mencionado anteriormente, o método All com *features* e divisão *within* é o segundo melhor, com uma ligeira diferença entre eles.

É importante notar que o desempenho excecional deste modelo deve-se em parte à avaliação *within-subject*, o que significa que o modelo viu parte dos dados de cada participante durante o treino. Portanto, embora atinja a maior acurácia, não fornece uma medida realista de generalização para participantes completamente novos. Os modelos *between-subject*, embora numericamente menos precisos, são mais indicativos do desempenho real em indivíduos não vistos.

5.3 Testes de hipótesis

Insertar Tecla	Tipo	Tamaño	
RELIEF_withinF	float64	1	np.float64(0.8454205607476636)
ALL_withinF	float64	1	np.float64(0.8429906542056076)
PCA_withinF	float64	1	np.float64(0.8266355140186915)
ALL_withinE	float64	1	np.float64(0.7172897196261683)
PCA_withinE	float64	1	np.float64(0.6640186915887851)
RELIEF_withinE	float64	1	np.float64(0.6383177570093459)
PCA_betweenF	float64	1	np.float64(0.6174305075615556)
ALL_betweenF	float64	1	np.float64(0.6133776688847768)
ALL_betweenE	float64	1	np.float64(0.5747118763624721)
RELIEF_betweenF	float64	1	np.float64(0.5572275401905962)
PCA_betweenE	float64	1	np.float64(0.5515131553968151)
RELIEF_betweenE	float64	1	np.float64(0.5347535523445567)

O teste de Kolmogorov–Smirnov é altamente sensível ao tamanho da amostra, enquanto o teste de Shapiro–Wilk fornece melhor precisão para conjuntos de dados menores. Por esta razão, utilizámos o teste de Shapiro na nossa análise, uma vez que apenas tínhamos 10 divisões para avaliação. Em cenários com amostras pequenas (10 divisões), a distribuição aparentou ser normal; no entanto, ao utilizar um número maior de divisões, o resultado indicou ausência de normalidade. Por isso, foi aplicado um teste não paramétrico.

Como todas as comparações foram realizadas nos mesmos subconjuntos de dados (mesmos segmentos), os experimentos foram emparelhados, razão pela qual foi utilizado o teste de Friedman para avaliar a existência de diferenças significativas entre os modelos.

Os resultados indicaram que existem diferenças significativas entre os métodos, confirmando que o ReliefF com *features* e divisão *within* demonstra consistentemente desempenho superior.

Exemplo:
Na análise *within-split*, o melhor modelo obteve um valor de p = 3.0917754466602e-09, indicando uma diferença altamente significativa em relação aos outros modelos.
Na análise *between-split*, o melhor modelo obteve um valor de p = 0.03178656029465204, mostrando significância estatística, embora não tão forte quanto no caso *within*.

Após o teste de Friedman, realizámos o teste *post hoc* de Nemenyi para confirmar diferenças estatísticas par a par entre os modelos. Para os resultados da divisão *within*, o melhor

modelo foi ReliefF com *features*, seguido de All com *features*, com p = 1, indicando que não há diferença significativa entre estes dois modelos.
Para os resultados da divisão *between*, o melhor modelo foi PCA com *features*, que obteve p = 0.999997 quando comparado com All com *features*, também indicando ausência de diferença significativa.

Após a conclusão destes testes, utilizámos uma função para identificar o melhor modelo geral entre as avaliações da divisão *between*. Foram examinados os valores médios obtidos nos testes de hipótese e selecionado o mais elevado, PCA com features (between). Em seguida, foram recuperados os dados correspondentes utilizados por este classificador (conjunto de treino, valor ótimo de *k* e outros parâmetros relevantes) para realizar a avaliação final com o modelo selecionado.

A utilização deste método *between* deve-se à sua aplicação em tarefas da vida real, onde sujeitos desconhecidos são introduzidos no sistema, tornando-o mais realista e fornecendo uma taxa de classificação mais próxima da realidade.

7. MELHORIAS

O trabalho atual utiliza o k-Nearest Neighbors (kNN) como modelo de classificação. Por ser simples e interpretável, o kNN é frequentemente superado por modelos mais poderosos. Assim, uma possível melhoria é a avaliação de classificadores alternativos, como Random Forests e Gradient Boosted Trees. Estes modelos podem capturar fronteiras de decisão mais complexas e potencialmente melhorar a acurácia, especialmente para atividades difíceis de distinguir.

Outra possibilidade é incorporar estratégias de aprendizagem auto-supervisionada (*self-supervised learning*) diretamente nos sinais brutos deste projeto. Métodos como aprendizagem contrastiva, modelagem de sinais mascarados ou tarefas de predição temporal podem ser aplicados para pré-treinar um modelo nos segmentos disponíveis antes de realizar a classificação supervisionada. Isto permitiria ao sistema descobrir estruturas nos dados de *wearables* sem exigir rótulos adicionais, conduzindo frequentemente a melhores representações de características quando comparado com o uso de um modelo pré-treinado externo.

Uma possível melhoria é combinar estratégias de oversampling e undersampling. Reduzir as classes majoritárias enquanto se preserva cuidadosamente a sua variabilidade pode evitar que o classificador seja dominado por atividades de alta frequência, como estar em pé ou sentado. Quando combinado com métodos de *oversampling*, esta abordagem híbrida pode produzir um conjunto de treino mais uniforme e reduzir o viés. Além disso, a aumentação de dados pode ser realizada não apenas no domínio das características, mas também diretamente nos sinais brutos de séries temporais, gerando novas amostras balanceadas.

A análise de outliers realizada no primeiro marco do projeto pode ser aplicada no fluxo de trabalho atual. A remoção de segmentos anómalos, tanto nos sinais brutos dos sensores como no espaço de características extraídas, ajudaria o classificador a aprender a partir de dados mais limpos e consistentes. Isto poderia melhorar a robustez do modelo, reduzir o viés de valores extremos e potencialmente aumentar a acurácia global.