Activity Recognition Report

Víctor Benito Segura K12416072

December 2025

1 Task Description

This assignment investigates activity recognition using wrist-mounted accelerometer data during screwing operations. Two classification tasks are considered:

- **Screwing Process Recognition**: Classify between screwing and unscrewing activities.
- **Screwing Grip Recognition**: Classify between two different grip types (tight vs loose) used during screwing.

Figure 1: Tight vs Loose Grip [1]

The overall goal is to build an end-to-end machine learning pipeline: prepare the data, extract meaningful features, train several machine learning models, and identify the best classifier for each task.

2 Data Collection

The data was recorded in my dormitory setting using a wrist-mounted screwdriver fixed to the dominant hand (right wrist) with a ponytail. An iphone 13 running the *Phyphox* [2] app captured tri-axial accelerometer data at a sampling rate of 100 Hz.

2.1 Screwing Process Recognition

For the screwing vs unscrewing classification, the following procedure was applied:

- Each activity was recorded for one minute.
- The recordings were automatically started and stopped using the Phyphox app, resulting in around 6000 samples per class.
- The wrist was free to move naturally while performing the screwing operations.

- Raw data was saved into separate CSV files for each class, with columns representing time (s), acceleration x (m/s^2), acceleration y (m/s^2), acceleration z (m/s^2), and absolute acceleration (m/s^2).

2.2 Screwing Grip Recognition

For the tight vs loose grip classification:

- I performed screwing operations using both grips.
- Movements were exaggerated slightly to facilitate classifier differentiation.
- Each grip type was recorded in a separate CSV file with the same format as above.
- Approximately 6000 samples per grip class were collected.

3 Preprocessing

The raw accelerometer data was preprocessed to prepare it for feature extraction and classification. The preprocessing pipeline is very similar for both tasks and therefore the same procedure is applied once, while the results (plots and metrics) are shown for both tasks.

3.1 Filtering

Accelerometer signals contain noise and irrelevant frequency components. To remove these, we tried different filters (low pass, high pass and band pass) and compare which one preserves the best the relevant motion patterns:

- Low-pass filter (15 Hz cutoff)
- High-pass filter (0.5 Hz cutoff)
- Band-pass filter(0.5-15 Hz cutoff)

Figure 2 shows the effect of the filter on the x-axis acceleration for the Screwing task.

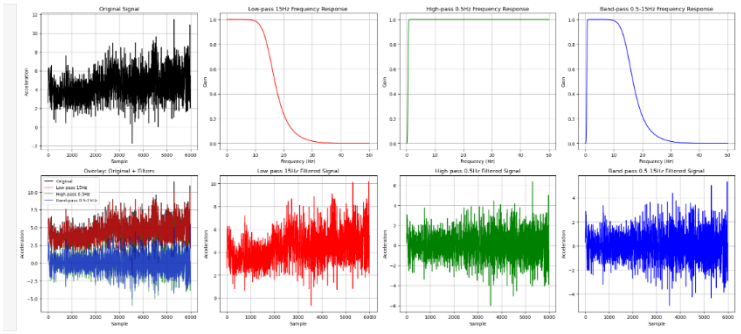

Figure 2: Original and filtered x acceleration signals for Screwing/Unscrewing task. First row: filters, second row: filtered signals

Figure 3 shows the effect of the filter on the x-axis acceleration for Grip type task.

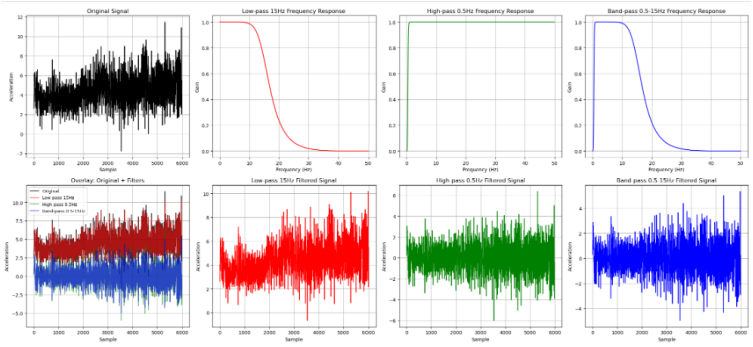

Figure 3: Original and filtered x acceleration signals for Grip type task. First row: filters, second row: filtered signals

The signals were then standardized to zero mean and unit variance.

3.2 Segmentation

After filtering, the accelerometer signals were divided into fixed-length windows to prepare the data for feature extraction. For this project, we used a window size of 2 seconds with 50% overlap.
The choice of window size balances:

- **Capturing sufficient motion dynamics:** Each window includes at least one complete screwing or unscrewing cycle, representing the characteristic motion pattern.
- **Maintaining temporal resolution:** Overlapping windows provide more training samples and ensure subtle transitions between motions are not missed.

3.3 Feature Extraction

For each segmented window, features were computed for all accelerometer axes (x, y, z) and the absolute acceleration:

- **Time-domain features:** mean, standard deviation, variance, maximum, minimum, range, RMS.
- **Frequency-domain features:** dominant frequency, spectral energy, spectral entropy.

These feature vectors summarize the motion characteristics of each window and serve as input to the classifiers.

3.4 Clean Data

The feature vectors from all recordings were concatenated into a single dataset. Each instance contains features for all axes and the corresponding class label. The final datasets were saved as `screw_features.csv` and `grip_features.csv`. Artifacts at the start and end of recordings are minimal in this dataset (automatic start/end by Phyphox), so no additional trimming was necessary.
Both tasks consist in 58 window samples per class, ensuring a balanced classification problem.

3.5 Filter Comparison

We compared low-pass, high-pass, and band-pass filters using kNN ($k = 5$, 10-fold cross-validation) on the Screwing vs. Unscrewing task:

- Low-pass: 0.914 ± 0.067
- High-pass: 0.662 ± 0.127
- Band-pass: 0.611 ± 0.097

and for the Grip-type task:

- Low-pass: 0.956 ± 0.060
- High-pass: 0.879 ± 0.124
- Band-pass: 0.844 ± 0.108

The low-pass filter achieves the highest accuracy, which justifies its selection for both tasks.

4 Model Comparison and Hyperparameter Tuning

In this study, four different classifiers were trained and evaluated to predict the class label based on the extracted motion features for both tasks: Screwing vs. Unscrewing and Tight vs. Loose Grip. For each model, hyperparameter tuning was performed using 10-fold stratified cross-validation. The approach was to find the best hyperparameters for each model individually and then compare the best-performing models across all classifiers.

4.1 k-Nearest Neighbors (kNN)

The k-Nearest Neighbors algorithm classifies a sample based on the majority class among its k nearest neighbors in the feature space. Hyperparameter tuning was performed for k using the set $[1, 3, 5, 7, 9, 11, 13, 15]$, with 10-fold stratified cross-validation for both tasks.

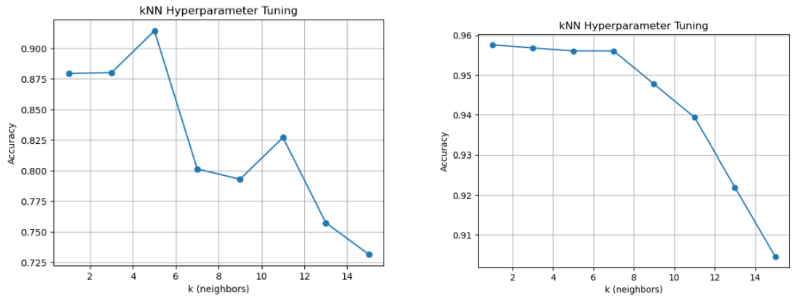

Figure 4: kNN accuracy vs k (neighbors) for Screwing vs. Unscrewing (left) and Tight vs. Loose Grip (right).

For Screwing vs. Unscrewing, the highest cross-validated accuracy was 0.914 for $k = 5$.
For Tight vs. Loose Grip, the highest cross-validated accuracy was 0.958 for $k = 1$.
These parameters were selected for further model comparisons in both tasks.

4.2 Decision Tree (J48)

A Decision Tree classifier was trained to predict the class label. Hyperparameter tuning was performed on the maximum depth, evaluating depths 2, 4, 6, 8, 10, and unlimited (`None`).

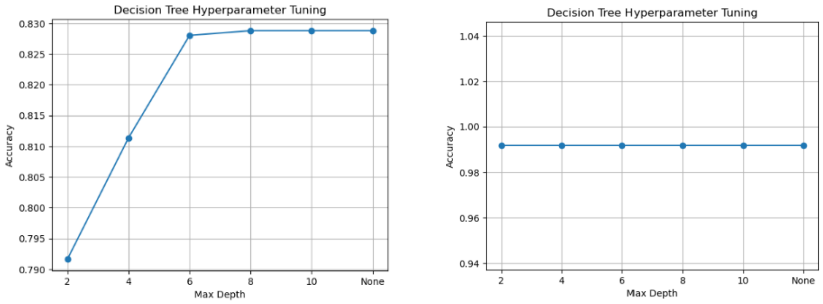

Figure 5: Decision Tree hyperparameter tuning: accuracy vs maximum depth for Screwing vs. Unscrewing (left) and Tight vs. Loose Grip (right).

The best maximum depth for the Screwing task was 8 with an accuracy of 0.829.
The best maximum depth for the Grip type task was 2 with an accuracy of 0.992.

This tree is equivalent to the J48 algorithm in WEKA.

4.3 Naive Bayes

A Gaussian Naive Bayes classifier was used. No hyperparameter tuning was applied because the model has no parameters that significantly affect performance. Cross-validated accuracies: 0.707 (Screwing/Unscrewing) and 0.958 (Tight/Loose Grip).

4.4 Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) is a feedforward neural network capable of capturing nonlinear relationships in the data. Hyperparameter tuning was performed over different hidden layer sizes and regularization values (α) with 10-fold cross-validation:

- (32,), $\alpha = 0.0001$
- (64,), $\alpha = 0.0001$
- (32,64), $\alpha = 0.001$
- (64,32), $\alpha = 0.001$

- (32,64,128), $\alpha = 0.001$
- (128,64,32), $\alpha = 0.001$

The best performing configuration for the screwing task was a single hidden layer with 32 neurons and $\alpha = 0.0001$, and for the grip task a double hidden layer with 32 and 64 neurons and $\alpha = 0.001$ Cross-validated accuracies: 0.921 (Screwing/Unscrewing) and 0.974 (Tight/Loose Grip).

4.5 Models Comparison

In this subsection, we compare the performance of the four tuned classifiers: k-Nearest Neighbors (kNN), Decision Tree (J48), Naive Bayes and Multilayer Perceptron (MLP). The comparison is based on standard evaluation metrics including Accuracy, Precision, Recall, F1-score, ROC-AUC, and computation time for both tasks: Screwing vs. Unscrewing and Tight vs. Loose Grip.

Table 1: Model Performance Summary – Screwing vs. Unscrewing

Model	Accuracy	Precision	Recall	F1	ROC-AUC	Time (s)
kNN	0.914	0.875	0.966	0.918	0.924	3.431
Decision Tree	0.828	0.839	0.810	0.825	0.828	0.110
MLP	0.922	0.889	0.966	0.926	0.968	10.856
Naive Bayes	0.707	0.707	0.707	0.707	0.765	0.051

Table 2: Model Performance Summary – Tight vs. Loose Grip

Model	Accuracy	Precision	Recall	F1	ROC-AUC	Time (s)
kNN	0.957	0.949	0.966	0.957	0.957	1.659
Decision Tree	0.991	0.983	1.000	0.991	0.991	0.051
MLP	0.974	0.966	0.983	0.974	0.982	3.887
Naive Bayes	0.957	0.949	0.966	0.957	0.958	0.032

- All models achieve high accuracy and overall performance for both tasks; however, the screwing task produces lower results for all models.
- MLP achieves the highest accuracy and ROC-AUC for the screwing task, but is more computationally expensive. kNN provides a good trade-off between accuracy and computational cost.
- Decision Tree classifier achieves a really impressive 0.991 accuracy for the grip task, with a recall of 1.0, we will discuss this result later.

To further analyze class separation and model performance, we performed dimensionality reduction using both PCA and t-SNE. (Plots are in the appendix).

Observations from the figure:

- PCA provides a general view of the feature space but does not correctly separate the classes.
- t-SNE, being a non-linear method, slighyly better separates the classes and helps visually understand classifier performance.
- the clusters for both methods are more consistent in the second grip task than in the screwing one.
- Confusion matrices indicate very few misclassifications for all models. Most of the misclassifications are observed for Naive Bayes in the screwing task. But they do not follow any clear trend.

4.6 Feature-Group Importance

Feature importance was analyzed by grouping features by axis and domain: x_time, y_time, z_time, abs_time, x_freq, y_freq, z_freq, abs_freq. Decision Tree used built-in feature importance, while kNN, MLP, and Naive Bayes used permutation importance.

Figure 6: Feature importance by model and feature group. Top: Screwing vs. Unscrewing. Bottom: Tight vs. Loose Grip. Color intensity represents relative importance.

Key insights:

- Apart from Decision Trees, the models do not rely on a single feature group for prediction, which demonstrates flexibility and highlights the relevance of both time-domain and frequency-domain features.
- In Decision Trees, especially for the grip task, the frequency-domain z features are crucial for prediction, likely reflecting the exaggerated rotational movements between grip types.

5 Conclusions

Through this project, I learned how to build an end-to-end machine learning pipeline: collecting raw data, preprocessing it, extracting relevant features, fine-tuning hyperparameters, training and comparing multiple models, and extracting insights from their performance.

One key takeaway is the crucial importance of data collection. For the grip recognition task, I deliberately exaggerated the rotational movement between grips because I anticipated it might be difficult to separate the classes otherwise. This design choice led to a situation where a simple Decision Tree model could rely primarily on a single z-axis frequency feature to almost perfectly predict all cases.

I also observed that more complex models, such as MLP, tend to distribute importance across multiple features, which is beneficial for generalization.

Finally, even though classification accuracies are very high, this does not necessarily imply that the classes form clearly separated clusters when visualized using dimensionality reduction methods such as PCA or t-SNE. This highlights the difference between predictive performance and intrinsic data structure.

References

[1] Task Description Image: Pervasive Computing JKU Lectures.

[2] Staudenmaier, G., et al. *Phyphox: Smartphone Experiments for Physics Education.* Available at: https://phyphox.org

[3] Code and Data Repository: Available at: https://github.com/Victorbs18/Activity-Recognition

Appendix

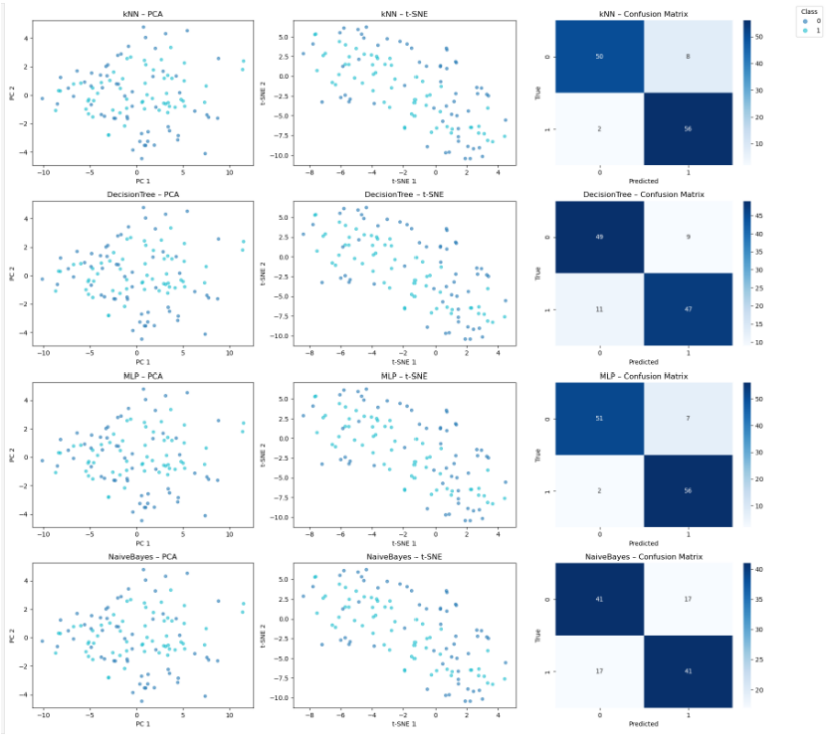

Figure 7: Dimensionality reduction and classifier performance for the Screwing task. Each row corresponds to a model (kNN, Decision Tree, MLP, Naive Bayes). Columns: PCA scatter plots, t-SNE scatter plots, and confusion matrices.

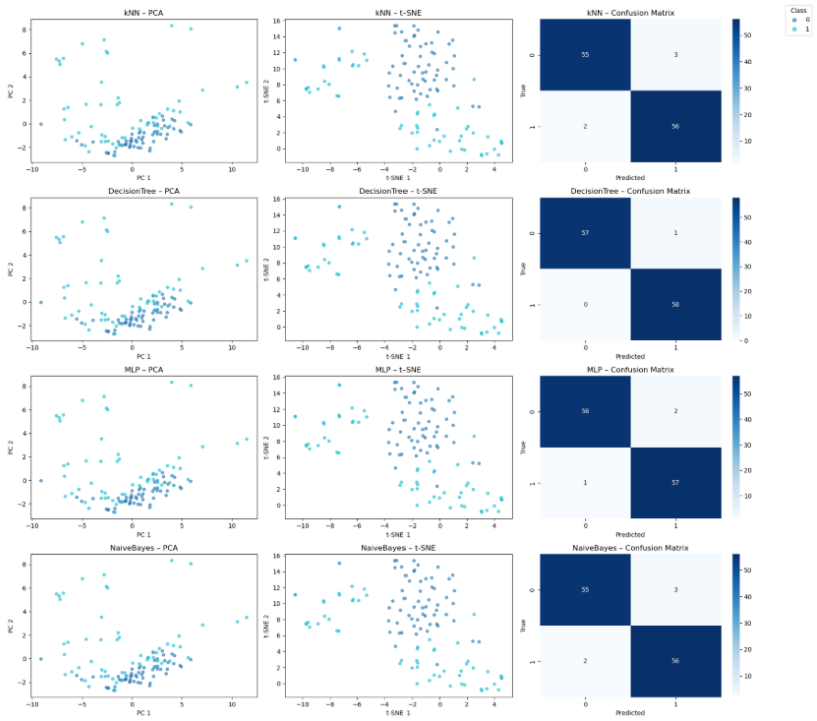

Figure 8: Same for Grip task