Task 3: Subject-Independent Activity Classification

LOSO vs 10-Fold CV, Model Comparison, and Feature Selection (Manual, PCA, RFE)

Luisa Faust

2025

Abstract

This report evaluates subject-independent activity recognition using tri-axial mobile sensor features. I (1) perform **Leave-One-Subject-Out** (LOSO) cross-validation, (2) compare several classifiers, (3) contrast LOSO with **10-Fold** CV, and (4) study feature selection via manual domain choice, **PCA**, and **RFE**. The code supports two data sources (EdgeML API or a local CSV fallback); all figures and tables here interpret the fallback run for clarity and reproducibility.

1 Data and Setup

I use Prof. Riedel's dataset from KIT[1] because my Task 1–2 dataset is too small to yield stable LOSO vs. 10-fold results.

Activities: *running (18)*, *sitting (41)*, *standing (29)*, *walking (56)* samples. Subjects/groups: 8 distinct IDs. I reuse the engineered features from Task 2 (windowed statistics, magnitudes, jerk, angular acceleration, etc.). Evaluation metrics: macro *Accuracy*, *Precision*, *Recall*, and *F1*. Splits: **LOSO** to assess subject generalization; **10-fold CV** to estimate in-distribution performance.

2 Why these design choices

- **LOSO** (Leave-One-Subject-Out) measures subject-independent generalization: each fold leaves one person completely unseen during training.
- **10-fold CV** estimates in-distribution performance: data are split into 10 folds and each fold is tested once (train on the other 9). Without grouping by subject, samples from the same person can appear in train *and* test (*subject leakage*), which inflates scores. We therefore report 10-fold as an *upper bound*, while LOSO is the honest subject-generalization benchmark.

[1] https://gitlab.kit.edu/kit/tm/pcs/teaching/css/exercise_jupyter/-/blob/master/data_snapshot/project_css25.pkl?ref_type=heads

3 Results

3.1 LOSO CV across 9 classifiers

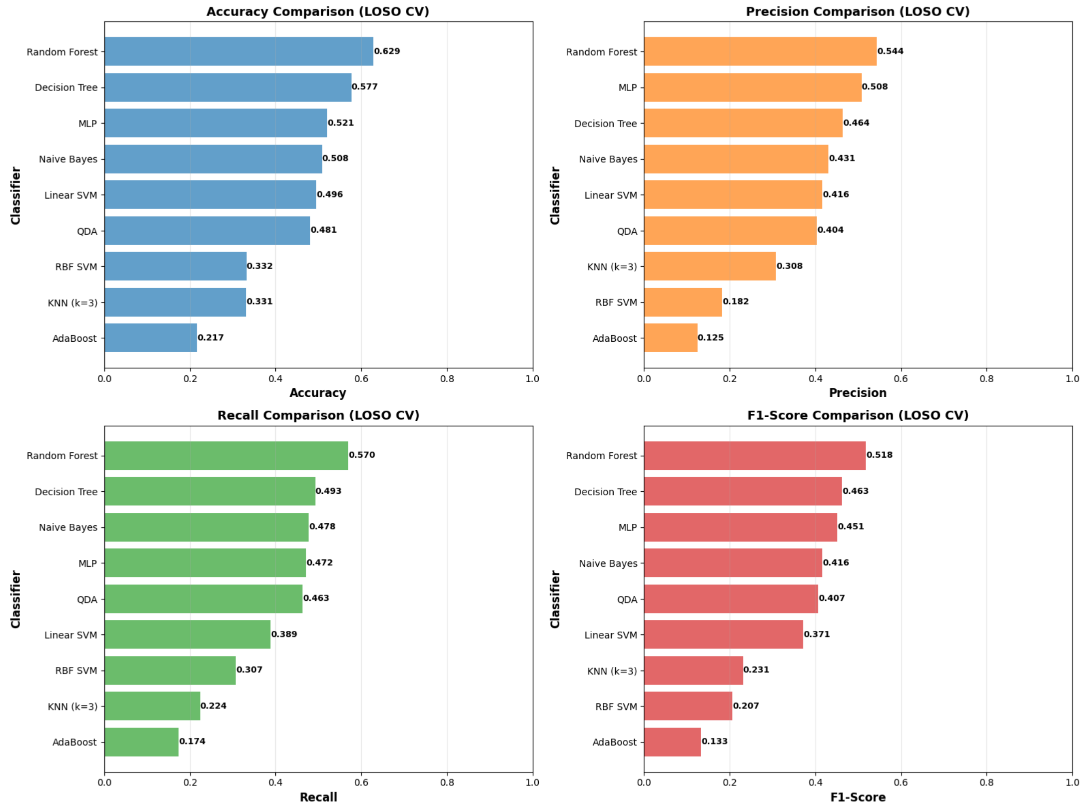

Figure 1: Classifier comparison (LOSO): bar plots of Accuracy, Precision, Recall, F1.

Table 1: LOSO CV over all subjects (higher is better).

Classifier	Accuracy	Precision	Recall	F1	Acc Std	F1 Std
Random Forest	0.629	0.544	0.570	0.518	0.201	0.273
Decision Tree	0.577	0.464	0.493	0.463	0.271	0.321
MLP	0.521	0.508	0.472	0.451	0.288	0.284
Naive Bayes	0.508	0.431	0.478	0.416	0.335	0.354
QDA	0.481	0.404	0.463	0.407	0.363	0.376
Linear SVM	0.496	0.416	0.389	0.371	0.227	0.251
KNN (k=3)	0.331	0.308	0.224	0.231	0.138	0.116
RBF SVM	0.332	0.182	0.307	0.207	0.208	0.137
AdaBoost	0.217	0.125	0.174	0.133	0.177	0.120

Takeaway. **Random Forest** leads in macro-F1 under LOSO, with Decision Tree second. Variability is sizable (stds), which is expected with small, imbalanced per-subject splits (high variation across subjects)

3.2 LOSO vs 10-Fold (top-3 models)

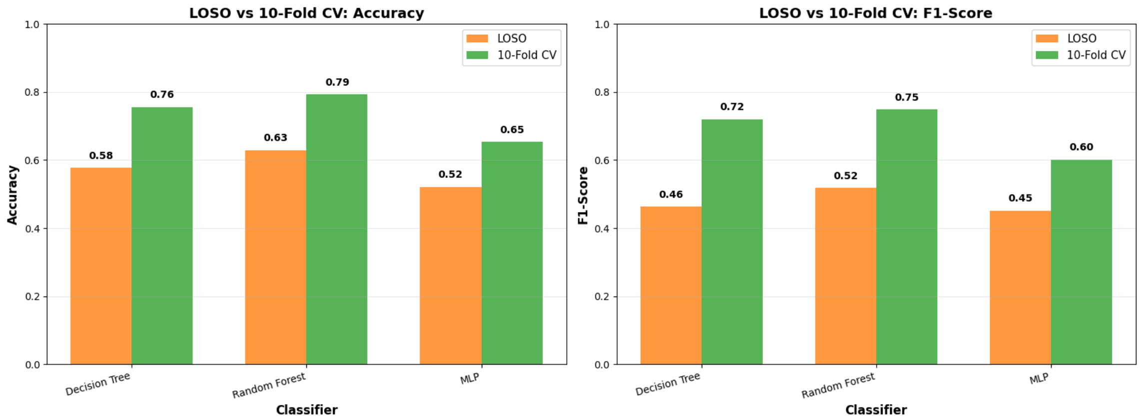

Figure 2: Accuracy and F1: LOSO (orange) vs 10-Fold CV (green).

Table 2: Top-3 models: LOSO vs 10-Fold CV.

	LOSO				10-Fold			
Classifier	Acc	F1	Acc Std	F1 Std	Acc	F1	Acc Std	F1 Std
Decision Tree	0.577	0.463	0.271	0.321	0.756	0.720	0.166	0.189
Random Forest	0.629	0.518	0.201	0.273	0.793	0.748	0.095	0.148
MLP	0.521	0.451	0.288	0.284	0.653	0.602	0.155	0.175

Interpretation. 10-Fold substantially improves Acc and F1 for all three (roughly +25–56% relative), since subjects also appear in training folds; LOSO remains the honest generalization test for new users.
Note: This does not mean the LOSO models are inferior; 10-fold benefits from subject overlap (in-distribution evaluation), whereas LOSO measures strict out-of-subject generalization.

3.3 Feature selection: Manual vs PCA vs RFE

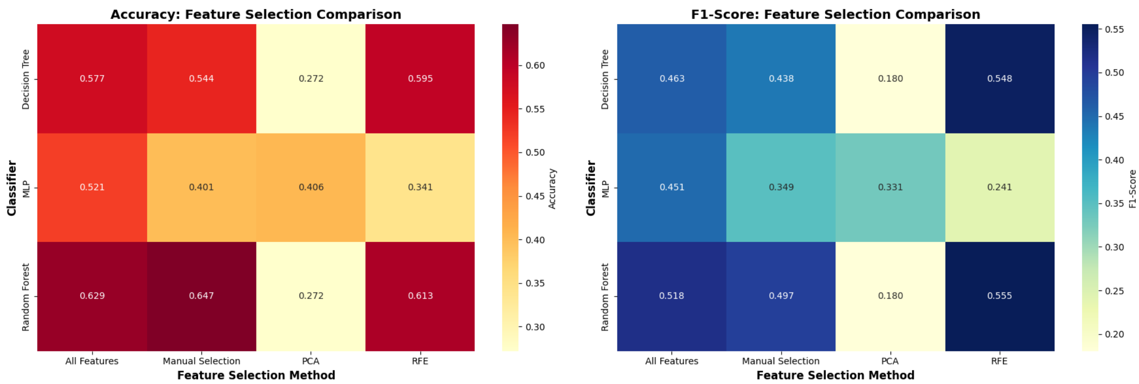

Figure 3: Heatmaps of Accuracy and F1 for each classifier × feature set (LOSO).

Table 3: Feature-selection comparison (LOSO, top-3 models).

Classifier	**Feature Set**	Accuracy	F1	Acc Std	F1 Std
Decision Tree	All Features	0.577	0.463	0.271	0.321
Decision Tree	Manual Selection	0.544	0.438	0.290	0.321
Decision Tree	PCA (95% var)	0.272	0.180	0.179	0.118
Decision Tree	RFE (14 feats)	**0.595**	**0.548**	0.381	0.397
Random Forest	All Features	0.629	0.518	0.201	0.273
Random Forest	Manual Selection	**0.647**	0.497	0.190	0.269
Random Forest	PCA (95% var)	0.272	0.180	0.179	0.118
Random Forest	RFE (14 feats)	0.613	**0.555**	0.366	0.382
MLP	All Features	**0.521**	**0.451**	0.288	0.284
MLP	Manual Selection	0.401	0.349	0.283	0.267
MLP	PCA (95% var)	0.406	0.331	0.326	0.328
MLP	RFE (14 feats)	0.341	0.241	0.283	0.217

What this says.

- **Best combo**: Random Forest + **RFE** gives the top macro-F1 among tested pairs; Random Forest + Manual also lifts accuracy. PCA collapsed the 74 features to *one* component (99.3% variance), which harmed tree models—likely because a single projection erased class-separating structure present outside the dominant variance direction.
- **RFE features**: 14 selected (e.g., `accX_mean`, `accY_mean`, `accZ_std`, `gamma_std`, min/max of accel channels), i.e., steady-state level + variability + extremes—intuitively linked to gait/onset differences.

4 Discussion (how to read/justify the results)

- **LOSO vs 10-Fold**. Use LOSO to claim subject independence; report 10-Fold to show potential if you can calibrate per user.
- **Model choice**. RF/DT perform well with heterogeneous, non-linear signals and mixed-scale features; SVMs underperform likely due to limited tuning and class imbalance.
- **Feature selection**. Wrapper methods (RFE) can beat PCA when variance does not align with discriminative directions. Manual subsets help interpretability; keep them as a baseline.
- **Uncertainty**. Report standard deviations. The spread across left-out subjects is meaningful variability, not noise.

Appendix: Classifier Cheat Sheet

Sources: scikit-learn User Guide [1], Breiman (Random Forest) [2], Quinlan (Decision Trees/C4.5) [3], Cover & Hart (kNN) [4], Cortes & Vapnik (SVM) [5], Schölkopf & Smola (Kernel methods) [6], Freund & Schapire (AdaBoost) [7], Hastie, Tibshirani & Friedman (QDA/Naive Bayes overview) [8], Goodfellow, Bengio & Courville (MLP/Deep Learning) [9], and lecture slides *Kontextsensitive Systeme* (Prof. Riedel, KIT) [10].

Table 4: Brief model overview, strengths, and caveats.

Model	**Short explanation / strengths & caveats**
Random Forest	Ensemble of decision trees (bagging). Handles non-linear, mixed-scale, and noisy features; robust and provides feature importances. Caveats: can prefer majority class if unbalanced; less smooth boundaries; needs trees/hyperparameters tuned.
Decision Tree	Greedy axis-aligned splits. Very interpretable and fast; captures interactions. Caveats: prone to overfitting and instability when used alone.
MLP (Neural Net)	Feed-forward network for non-linear decision surfaces. Captures complex interactions. Needs scaling and careful tuning; risk of overfitting on small data.
Naive Bayes	Generative model assuming conditional independence. Extremely fast, good baseline. Assumption often violated; probabilities simplistic; roughly linear boundaries in log-space.
QDA	Quadratic Discriminant Analysis (class-specific Gaussians with full covariance). Curved boundaries when class covariances differ. Needs enough data; covariance estimation can be unstable; sensitive to scaling/outliers.
Linear SVM	Max-margin linear classifier. Strong with high-dimensional data; robust with limited samples; needs scaling. Only linear boundary; may underfit non-linear structure.
RBF SVM	Kernel SVM with Gaussian RBF. Flexible non-linear boundaries; can be very accurate if C, γ well tuned. Sensitive to hyperparams; slower on larger sets; needs scaling.
KNN (k=3)	Instance-based voting by nearest neighbors. No training; captures local structure. Sensitive to scaling and k; suffers in high dimensions; slower at inference.
AdaBoost	Boosting of weak learners (often stumps). Focuses on hard examples; strong on clean signals. Sensitive to noise/outliers; tuning of estimators/learning rate matters.

Table 5: Metric definitions and interpretation (macro multi-class unless noted).

Metric	**Binary definition / formula**	**Multi-class (macro) & interpretation**
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Computed over all instances (global correctness). Can look optimistic under class imbalance because majority classes dominate.
Precision (PPV)	$\frac{TP}{TP+FP}$	Macro Precision: $\frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i+FP_i}$. "How pure are predicted positives?" Penalizes false positives.
Recall (Sensitivity)	$\frac{TP}{TP+FN}$	Macro Recall: $\frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i+FN_i}$. "How many actual positives found?" Penalizes misses/false negatives.
F1-score	$F1 = 2\,\frac{P\cdot R}{P+R}$	Macro F1: $\frac{1}{C}\sum_{i=1}^{C}F1_i$ with $F1_i = 2\,\frac{P_i R_i}{P_i+R_i}$. Balances Precision vs. Recall; *note:* macro F1 $\neq$ F1 of macro Precision/Recall.

Notes. Macro = unweighted mean over classes (treats all classes equally); Micro = pool TP/FP/FN across classes (weighted by support; dominated by frequent classes); Weighted macro = per-class metrics weighted by class support.

Sources: scikit-learn User Guide [1], Powers (evaluation of Precision/Recall/F1/Accuracy) [11], and lecture slides *Kontextsensitive Systeme* (Prof. Riedel, KIT) [10].

Appendix: Feature Selection (short)

Method	Short explanation
Manual (domain)	Hand-picked, physically meaningful features (e.g., Acc/Gyro magnitudes, per-window mean/variance, Jerk/AngAcc); interpretable baseline.
PCA	Unsupervised projection that preserves variance and reduces collinearity; helps simpler/linear models, but can drop low-variance yet discriminative cues.
RFE	Supervised wrapper that iteratively removes least useful features w.r.t. a base estimator; yields compact, task-specific subsets.

Sources: scikit-learn User Guide [1], Jolliffe & Cadima (PCA) [12], Guyon et al. (RFE) [13], and lecture slides *Kontextsensitive Systeme* (Prof. Riedel, KIT) [10].