

Heart Rate Estimation from Wrist PPG + Accelerometer (PPG-DaLiA)

Chouaib (github.com/ChouaibB)

January 23, 2026

Contents

1. Data source: PPG-DaLiA (Zenodo benchmark: PPGDalia)	2
2. Load data (TRAIN/TEST) + quick inspection	2
3. Signal sanity-check and channel naming	3
4. Feature extraction	4
4.1. PPG periodicity under low vs high motion	5
4.2. Motion vs PPG periodicity (binned view)	6
5. Modeling	6
6. Results and error analysis	7
6.1 True vs predicted heart rate	8
6.2 Error vs motion intensity	8
6.3 Error distribution	9
6.4 Summary and takeaways	9

1. Data source: PPG-DaLiA (Zenodo benchmark: PPGDalia)

Dataset: *PPGDalia* (time-series regression benchmark derived from the public **PPG-DaLiA** dataset)

- **Zenodo record (download + description):**
 - <https://doi.org/10.5281/zenodo.3902728>
- **Original dataset page (background / context):**
 - <https://archive.ics.uci.edu/ml/datasets/PPG-DaLiA>
- **Local path (expected):** download + unzip into `./data/PPGDalia/`
- **License:** Creative Commons Attribution 4.0 International (**CC BY 4.0**)

PPG-DaLiA is a public wearable physiology dataset for **wrist PPG-based heart rate estimation under motion**. It includes recordings from **15 subjects** performing a range of real-life activities, with **ECG-derived heart rate** as ground truth. The Zenodo **PPGDalia** variant used here is a **benchmark-ready export**: it is **pre-segmented into windows** and distributed in a standardized time-series regression format (the `.ts` format). Signals in this benchmark version are provided as ML-ready time series (segmented and typically normalized/standardized for benchmarking), and the `.ts` files do not preserve physical sensor units.

File format (Zenodo benchmark): two `.ts` files with a predefined split:

- `PPGDalia_TRAIN.ts`
- `PPGDalia_TEST.ts`

Each sample is **one window** (one multivariate time series) with:

- **4 dimensions:** `dim_0`, `dim_1`, `dim_2`, `dim_3`
- **1 target label** per window: **heart rate (bpm)**

Channel meaning (from dataset description):

- `dim_0`: wrist **PPG**
- `dim_1-dim_3`: wrist **accelerometer (x, y, z)**

(The `.ts` header does not explicitly name channels; we treat the 4 dimensions as $1 \times \text{PPG} + 3 \times \text{ACC}$ as described.)

What we use in this notebook: the provided **TRAIN/TEST split**, using **wrist PPG + wrist ACC** as inputs and **HR (bpm)** as the regression target. We keep preprocessing minimal and focus on **motion-aware / reliability features** and **robust evaluation**.

2. Load data (TRAIN/TEST) + quick inspection

We load the pre-defined `PPGDalia_TRAIN.ts` / `PPGDalia_TEST.ts` files from `./data/PPGDalia/`, convert the HR labels to numeric, and run a few quick sanity checks (shapes, value ranges, and a single-sample peek).

Loaded:

```
X_train: (43215, 4) | y_train: (43215,)
X_test : (21482, 4) | y_test : (21482,)
```

Channels (dims): 4

HR (bpm) train: min/median/max = 41.9 / 85.8 / 187.0

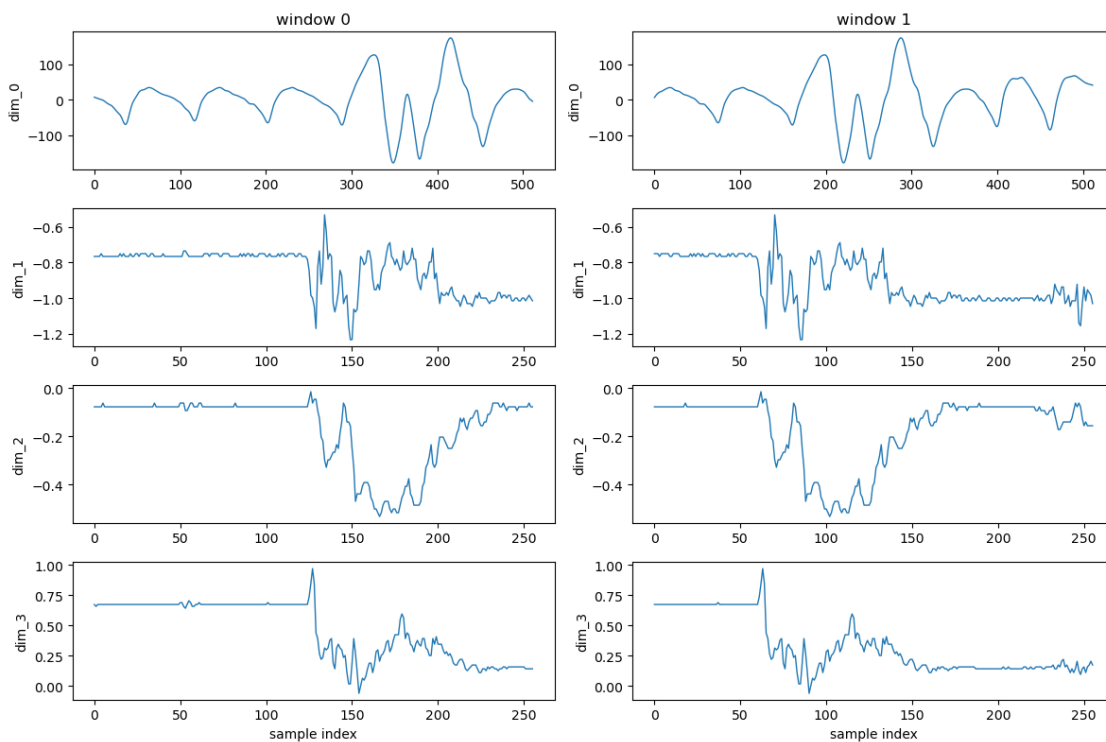
HR (bpm) test : min/median/max = 41.7 / 83.8 / 171.1

Example window index 0 (train): lengths per dim = [512, 256, 256, 256]

First 5 HR labels (train): [49.61136908 50.32399248 52.70833578 55.64079409 57.65840574]

3. Signal sanity-check and channel naming

We visually inspect a couple of example windows to confirm which channel corresponds to PPG versus motion.



- The quasi-periodic waveform of dim_0 confirms it to be the ppg channel.
- The shorter length, step changes and bursts observed at dim_1 to dim_3 confirms them to be our acc_x, acc_y, acc_z channels.

Info X_train:

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 43215 entries, 0 to 43214  
Data columns (total 4 columns):  
#   Column  Non-Null Count  Dtype  
#
```

```

---  -----  -----  -----
0   ppg      43215 non-null  object
1   acc_x    43215 non-null  object
2   acc_y    43215 non-null  object
3   acc_z    43215 non-null  object
dtypes: object(4)
memory usage: 1.3+ MB
None

Info X_test:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21482 entries, 0 to 21481
Data columns (total 4 columns):
#   Column    Non-Null Count  Dtype
---  -----  -
0   ppg      21482 non-null  object
1   acc_x    21482 non-null  object
2   acc_y    21482 non-null  object
3   acc_z    21482 non-null  object
dtypes: object(4)
memory usage: 671.4+ KB
None

```

4. Feature extraction

We convert each window into a **small, interpretable feature vector** that captures two core concepts relevant to on-wrist heart rate estimation: **motion intensity** and **PPG periodicity**.

Together, these features allow the model to **account for motion artefacts at the modeling level**, rather than attempting to explicitly denoise or reconstruct the PPG signal.

Motion intensity

We use the accelerometer as a **proxy for motion artefacts** affecting the PPG signal. Higher and more variable wrist motion is typically associated with stronger corruption of the optical PPG waveform.

For each window, we compute:

- the **acceleration magnitude**: $|acc| = \sqrt{acc_x^2 + acc_y^2 + acc_z^2}$
- the **mean** of $|acc|$
- the **standard deviation** of $|acc|$

These statistics are widely used and sufficient to summarize overall motion level and variability within a window. Including them as model inputs enables **explicit, interpretable motion-aware compensation**.

PPG periodicity

We use a simple **frequency-domain proxy** to quantify how “heart-rate-like” the PPG waveform looks.

We focus on a **physiologically plausible heart-rate band** of **0.7–3.5 Hz** (**42–210 bpm**), which is commonly adopted in prior work on PPG-based HR estimation, including studies using the **PPG-DaLiA dataset**.¹

For each window, we compute:

- the **dominant frequency** (peak frequency) of the PPG power spectrum within **0.7–3.5 Hz**
- the **peak power ratio**: power at that dominant frequency divided by total power in the same band

Intuition: a clean PPG window tends to show a clear spectral peak in this band, while motion-corrupted windows often exhibit weaker or less stable periodic structure. Combined with motion features, this allows the model to adapt its predictions under varying signal reliability.

	acc_mag_mean	acc_mag_std	ppg_dom_freq_hz	ppg_peak_power_ratio
0	1.018543	0.071625	0.750	0.574718
1	1.018886	0.073404	0.750	0.285373
2	1.016308	0.069871	0.875	0.416325
3	1.018729	0.071553	0.875	0.686951
4	1.020364	0.126706	0.875	0.483571

Feature matrix shapes (train - test): (43215, 4) (21482, 4)

4.1. PPG periodicity under low vs high motion

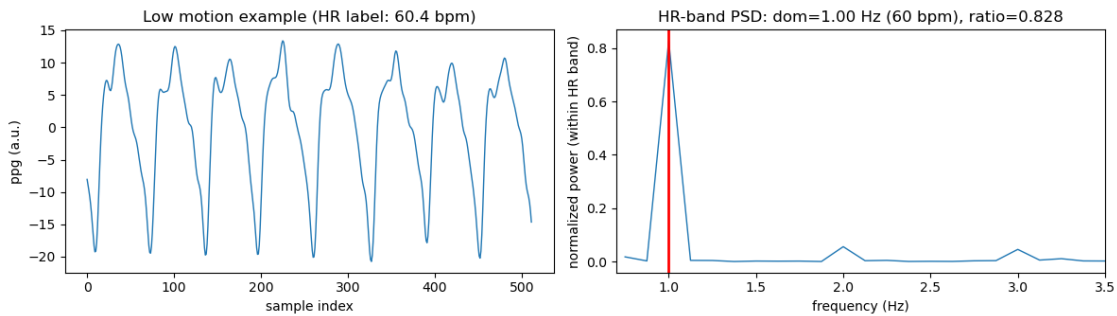
We visualize two example windows—one with **low motion** and one with **high motion**. For each example, we show:

- the **time-domain PPG**, and
- the **band-limited, band-normalized power spectrum** in the physiological HR band (**0.7–3.5 Hz**), with the **dominant frequency** highlighted.

The spectrum is **normalized within the HR band** and **cropped to that band** to visualize cardiac periodicity under motion.

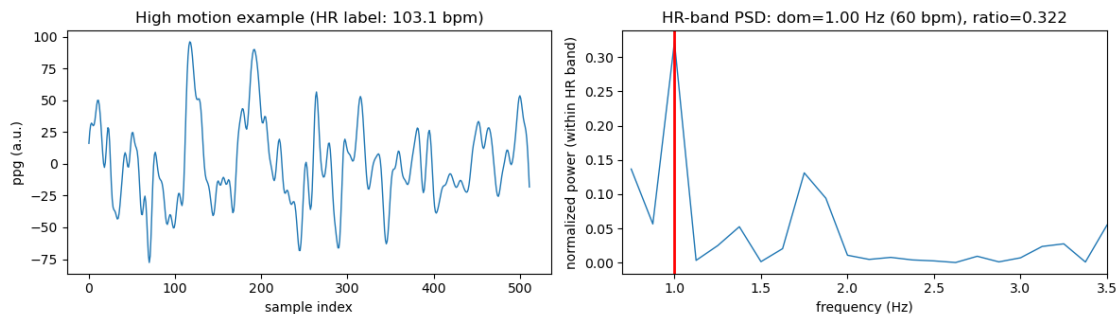
Low-motion example idx : 4738 | acc_mag_std: 0.0

High-motion example idx: 5706 | acc_mag_std: 0.7733447856251788



¹Reiss, A., Indlekofer, I., Schmidt, P., & Van Laerhoven, K. (2019).

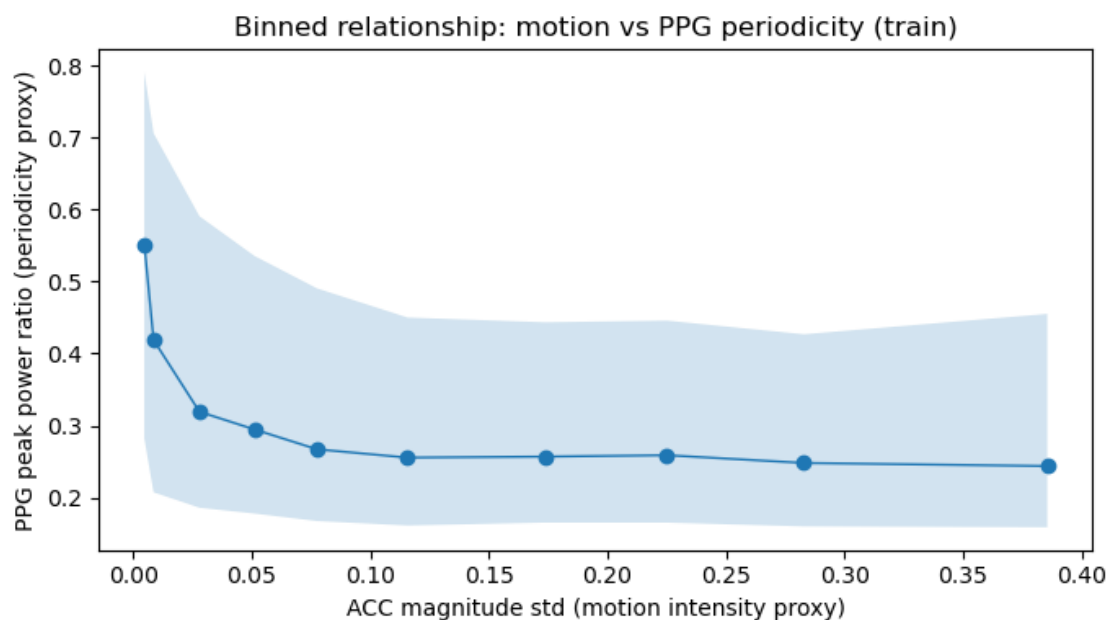
Deep PPG: Large-Scale Heart Rate Estimation with Convolutional Neural Networks. **Sensors**, 19(14), 3079.
<https://doi.org/10.3390/s19143079>



4.2. Motion vs PPG periodicity (binned view)

We summarize the relationship between **motion intensity** and **PPG periodicity** by binning windows according to accelerometer-based motion and examining the distribution of the **PPG peak power ratio** within each bin.

This view highlights a key wearable intuition: as motion increases, **high periodicity becomes less common**, even though low periodicity can occur at all motion levels.



5. Modeling

We train and compare a small set of **standard regression models** using the extracted features.

The goal is to establish **clean, well-understood baselines** rather than to optimize performance. All models are trained on the predefined training split, with samples **shuffled once before fitting** to avoid any ordering effects.

We **do not perform cross-validation** (e.g. k-fold or group-wise CV) or inner hyperparameter tuning:

- the dataset provides a **fixed train/test split**, and
- **subject identifiers are not available** in this benchmark format, making group-aware validation infeasible.

Models:

- **Ridge Regression**: a stable linear baseline suitable for correlated features.
- **Support Vector Regression (SVR)**: a commonly used nonlinear regressor in biosignal tasks.
- **HistGradientBoostingRegressor (HGBR)**: a strong tree-based nonlinear model with sensible defaults.

Metrics (evaluated on the test split):

- **MAE (bpm)** and **RMSE (bpm)** as core regression error metrics.
- A simple **within-tolerance accuracy** to support intuition (e.g., % predictions within ± 5 bpm and ± 10 bpm of the true HR).

Model: Ridge: 100%|

| 1/1

[00:00<00:00, 238.65it/s, 0.0s]

Model: SVR: 100%|

| 1/1

[01:06<00:00, 66.41s/it, 66.4s]

Model: HGBR: 100%|

| 1/1

[00:00<00:00, 5.44it/s, 0.2s]

	model	MAE (bpm)	RMSE (bpm)	Accuracy within ± 5 bpm (%)	\
0	SVR	8.515425	12.951099	52.294945	
1	HGBR	9.199082	13.783455	49.213295	
2	Ridge	12.745528	16.196043	23.089098	

	Accuracy within ± 10 bpm (%)	Train+test time (s)
0	70.794153	66.407979
1	68.839028	0.182878
2	46.578531	0.003872

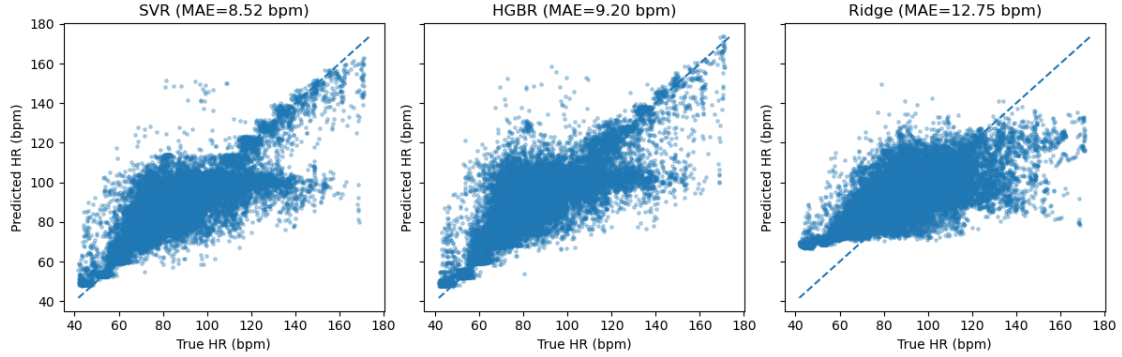
6. Results and error analysis

Across all metrics, **SVR** achieves the best overall performance on this benchmark, with the lowest **MAE** and **RMSE**, and the highest **within-tolerance accuracy**. This suggests that moderate nonlinear modeling is beneficial for HR estimation from compact, motion-aware features.

HGBR performs comparably but slightly worse across metrics, indicating that tree-based non-linearities do not offer a clear advantage over SVR in this low-dimensional feature setting. In contrast, **Ridge regression** underperforms substantially, highlighting the limitations of a purely linear model for this task.

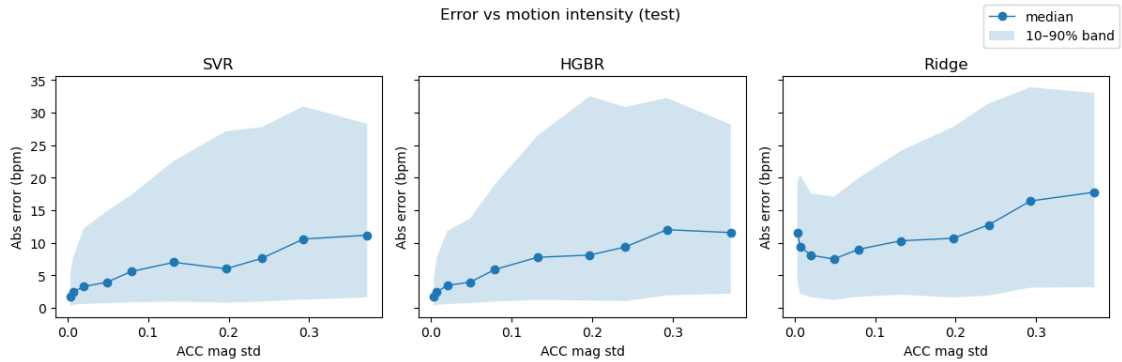
In terms of efficiency, training time varies widely: Ridge and HGBR train almost instantly, while SVR is significantly more expensive. This trade-off between accuracy and compute will be important when considering deployment or scaling.

6.1 True vs predicted heart rate



All models capture the overall HR trend, but clear differences emerge. **SVR** shows the tightest alignment with the identity line across the full HR range, with noticeably less spread at both low and high heart rates. **HGBR** follows closely but exhibits slightly larger variance and more underestimation at higher HRs. **Ridge** shows strong regression-to-the-mean behavior, compressing predictions toward the mid-range and failing to track extremes, which explains its substantially higher MAE.

6.2 Error vs motion intensity

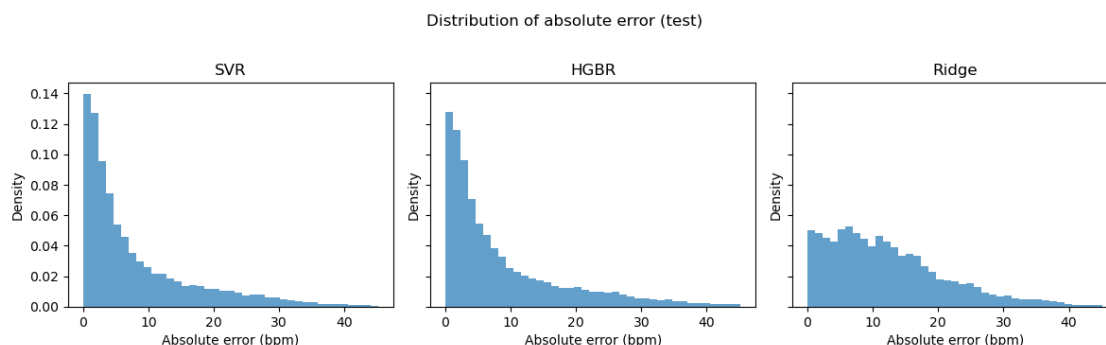


All three models show a clear increase in error as motion intensity rises, confirming the expected impact of motion artefacts on wrist PPG-based HR estimation.

Across the full motion range, **SVR** maintains the lowest median error and a relatively tighter error band, indicating better robustness to motion. **HGBR** follows a similar trend but with consistently higher median errors and a wider upper tail under high motion. **Ridge** exhibits the strongest degradation, with higher errors even at low motion and the widest spread overall, reflecting its limited capacity to model nonlinear motion effects.

These results reinforce that **nonlinear models handle motion-related variability more effectively**, but no model is fully immune to high-motion conditions.

6.3 Error distribution



The error distributions further clarify the differences between models. **SVR** concentrates most of its mass at low errors, with a steep drop-off and a relatively short tail, indicating more consistently accurate predictions. **HGBR** shows a similar shape but with a heavier tail, reflecting a higher frequency of larger errors. **Ridge** exhibits a much flatter distribution with substantial mass at higher errors, confirming its tendency toward large deviations and poor handling of difficult windows.

6.4 Summary and takeaways

This notebook demonstrates a compact, reliability-aware workflow for wrist-based heart rate estimation from PPG and accelerometer data.

Across models, **SVR consistently performs best**, achieving the lowest average error, the highest within-tolerance accuracy, and the most stable behavior under motion. **HGBR** offers competitive performance but shows greater sensitivity to high-motion conditions, while **Ridge regression** struggles to model nonlinear motion effects.

Motion artefact compensation is handled here in a **simple and interpretable way**: by explicitly incorporating accelerometer-derived motion features into the regression model. This allows the model to adapt its predictions under varying motion levels without attempting to explicitly denoise or reconstruct the PPG signal.

The error analyses confirm a key wearable insight: **motion does not uniformly degrade predictions**, but it increases both median error and the likelihood of large failures. Even with lightweight features, motion-aware modeling already provides substantial robustness gains.

Overall, the results highlight the importance of:

- **nonlinear models** for PPG-based HR estimation,
- **explicit motion awareness** as a practical form of artefact compensation,
- **interpretable, reliability-oriented features** as a strong baseline before moving to more complex approaches such as end-to-end deep models (e.g., DeepPPG).