Hypo/Hyperglycemia Detection from ECG: Summary Report

Daniele Uras
Matricola: 11094104

Filippo Saccomano
Matricola: 10771424

Gabriele Carta
Matricola: 11097253

Abstract

This report replicates the ECG-based pipeline proposed in the reference paper for non-invasive detection of glycemic excursions using wearable data. ECG signals are synchronized with CGM measurements, cleaned, segmented into beats, and used to extract morphology, RR, HRV, and circadian features. Features are aggregated over non-overlapping 1-minute windows, while CGM values are used only for labeling hypoglycemic (< 70 mg/dL) and hyperglycemic (> 180 mg/dL) events.

Following the reference study, labels are assigned via forward alignment of ECG to CGM, quality control is applied using HRConfidence filtering, and Random Forest models are trained under patient-wise temporal validation. In addition to single-threshold baseline models, we implement the multi-threshold morphology strategy described in the paper, where beat-level classifiers trained at different glucose cutoffs are fused at the interval level.

Under this setup, the fusion model achieves the best overall performance, with AUC 0.813 ± 0.155 for hypoglycemia and 0.755 ± 0.156 for hyperglycemia. We further analyze data quality, class imbalance, circadian effects, and feature importance, and we discuss deviations from the reference results in light of differences in subject selection, event distribution, and temporal split implementation.

1 Introduction

This work reproduces, end to end, the ECG-based approach proposed in the reference paper for detecting glycemic excursions using wearable data. As in the paper, the pipeline links Zephyr ECG (250 Hz) with Dexcom CGM (5-min sampling), uses the ECG stream only to extract predictors, and uses CGM only to assign labels. In our implementation (`SignalLab.ipynb`), the full workflow is automated: session indexing, ECG cleaning, beat detection, feature extraction, 1-minute aggregation, quality control, and model evaluation.

The reference problem is formulated as two binary tasks, exactly as in the paper: (i) hypoglycemia detection (CGM < 70 mg/dL) and (ii) hyperglycemia detection (CGM > 180 mg/dL), each versus its corresponding negative class. In code, labels are assigned by aligning each beat timestamp to the next available CGM reading (forward merge) with a maximum tolerance of 15 minutes, matching the design choice described in the reference study. All interval-level metrics are computed on non-overlapping 1-minute windows, consistent with the paper's evaluation unit.

A key motivation of the paper is that ECG changes are not guaranteed to appear only exactly at the clinical cutoffs (70 and 180 mg/dL). Instead, morphology patterns can emerge in a neighborhood around those thresholds and can vary across patients. For this reason, the reference paper proposes a multi-threshold fusion strategy: several beat-level morphology models are trained at multiple glucose cutoffs, and their posterior probabilities are summarized into interval-level descriptors that are then fused with HRV/time features for the final decision. Our code implements the same idea: (1) train multiple beat-level Random Forest models at different thresholds, (2) aggregate their probabilities within 1-minute windows into probability-distribution features, and (3) combine them with HRV and circadian encodings to obtain the final fusion model. We therefore report results by comparing both the baseline models (single-threshold morphology and HRV variants) and the fusion model, in the same spirit as the reference.

2 Methods

This section describes, step by step, the pipeline from raw wearable ECG and CGM files to interval-level models for hypo- and hyperglycemia detection. For each step, we explicitly state (i) what the reference paper does and (ii) what our code implements.

2.1 Configuration and thresholds

The main parameters used throughout the pipeline are reported in Table 1. All values match the reference paper's acquisition/setup and the choices implemented in our script.

Parameter	**Value**
FS	250 Hz
Window length	1 min (non-overlapping)
RR bounds	$[300, 2000]$ ms
$\mathrm{HRC}_{\min}$	90
Hypoglycemia threshold	70 mg/dL
Hyperglycemia threshold	180 mg/dL
CGM alignment tolerance	15 min (forward)

Table 1: Main pipeline parameters.

Sampling frequency ($FS = 250\,\mathrm{Hz}$).

The ECG sampling rate is fixed by the wearable device and defines the time resolution for peak detection and fiducial-point delineation. *Paper vs. us:* the reference paper uses Zephyr ECG at 250 Hz; our code sets `FS=250` and builds the sample-level timeline accordingly.

Window length (1 min).

We aggregate beat-level features into non-overlapping 1-minute windows, which are the evaluation unit of the reference study. This provides a practical compromise: short enough for timely detection, but long enough to stabilize morphology/HRV estimates within each interval. *Paper vs. us:* the paper evaluates all non beat-level models at 1-minute intervals; our script performs the same 1-minute aggregation for training and metrics.

RR bounds (300–2000 ms).

RR intervals outside $[300, 2000]$ ms (about 200–30 bpm) are treated as implausible and typically reflect motion artifacts, missed peaks, or false detections. *Paper vs. us:* the paper applies physiological sanity checks during preprocessing; our code enforces the same bounds to reduce artifact-driven HR/HRV distortion.

Quality threshold ($\mathrm{HRC}_{\min} = 90$).

HRC (HRConfidence) is a device-provided signal-quality/confidence score (0–100) indicating the reliability of heart-rate tracking. We keep only beats occurring during high-confidence segments (HRC > 90), to prevent poor-quality ECG from propagating into delineation and features. *Paper vs. us:* the paper explicitly filters beats using HRC > 90 after visual inspection of different cutoffs; our code uses `HRC_THRESHOLD = 90.0` and removes low-quality portions before fiducials/features.

Glucose thresholds (70 and 180 mg/dL).

We define hypoglycemia as glucose < 70 mg/dL and hyperglycemia as glucose > 180 mg/dL, producing two binary tasks. *Paper vs. us:* these are the same clinical cutoffs used in the reference; our code applies them after CGM alignment to create the two label sets.

CGM alignment tolerance (15 min, forward).

Because CGM is sampled at a much slower rate than ECG, each beat is matched to the nearest CGM reading in the *forward* direction, with a maximum tolerance of 15 minutes. This avoids assigning a beat to a CGM point that is too far away in time. *Paper vs. us:* the paper states "nearest CGM reading in the forward direction"; our code implements this with an as-of forward merge and `tol = 15 min`.

2.2 Dataset indexing and loading

We access the dataset by specifying a root folder and the sub-paths for ECG waveforms, sensor summaries, and CGM files. Subject/session identifiers are parsed from the folder structure (3-digit subject ID, plus session folders with a `YYYY_MM_DD` pattern).

We first build a subject-to-CGM map by scanning for `glucose.csv` files. Then we scan ECG sessions by searching for `*_ECG.csv` files and keep only sessions for which (i) the corresponding

`*_Summary.csv` exists and (ii) a CGM file is available for the same subject. This yields a session index (`df_index`) storing subject/session identifiers and the resolved file paths.

CGM is loaded by parsing `date` and `time` into a timestamp and converting glucose to numeric. When CGM is provided in mmol/L, we convert to mg/dL with the factor 18.0182. For each session, the session start time is taken from the first timestamp in the Summary file. The ECG waveform is read from the `EcgWaveform` column and we generate a sample-level time axis using $FS = 250$ Hz. The Summary file is also used to extract the per-second HRC signal and the device-provided HR at 1 Hz, later aligned to beats by mapping R-peak sample indices to seconds. *Paper vs. us:* the paper uses Zephyr summary signals (HR and HRC at 1 Hz) and Dexcom CGM (5-min); our indexing/loading mirrors this structure and uses the same signals for subsequent filtering and HR features.

2.3 ECG preprocessing and beat detection

Each ECG recording is cleaned using NeuroKit2 (`nk.ecg_clean(..., method="neurokit")`). R-peaks are then detected on the cleaned signal with `nk.ecg_peaks`. Given detected sample indices $\{r_k\}$, RR intervals are computed as

$$\mathrm{RR}_k\,[\mathrm{ms}] = \frac{(r_{k+1} - r_k)}{FS} \cdot 1000.$$

Paper vs. us: the reference pipeline uses NeuroKit2 for R-peak detection; our code applies the same library and ordering (clean → peaks → RR).

2.4 Beat-level quality filtering

To limit the impact of poor signal quality, we filter using the device HRC metric. We retain only beats occurring when HRC > 90, and discard sessions with insufficient reliable coverage or too few valid beats after filtering. This step is critical: errors at this stage propagate to fiducial delineation and morphology computation. *Paper vs. us:* the paper motivates HRC-based filtering (HRC > 90) to avoid noisy beats contaminating morphology features; our code implements the same cutoff and discards low-coverage sessions.

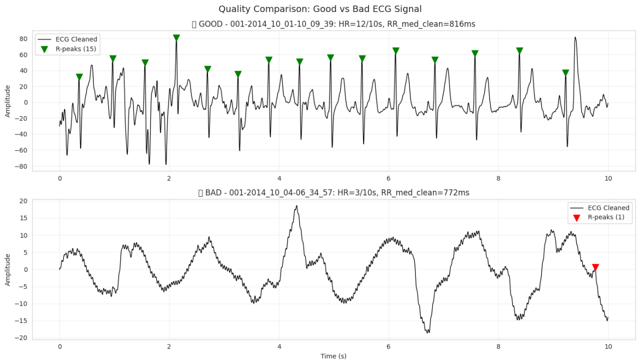


Figure 1: ECG quality and motivation for filtering. **What we see:** the top panel (GOOD) has stable, sharp QRS complexes and consistent R-peak detections (markers align with true peaks), leading to reliable RR and fiducial extraction. The bottom panel (BAD) is dominated by baseline wander/noise and distorted morphology; R-peaks can be missed or falsely detected, producing implausible RR and unreliable fiducials. **Paper comparison:** this is the same qualitative argument used in the reference paper's preprocessing section: HRC filtering is introduced to prevent noisy segments from biasing beat delineation and downstream features.

2.5 Fiducial-point delineation and beat-level feature extraction

For each retained beat, we delineate fiducial points (P, Q, R, S, and T peaks) using NeuroKit2. Beats where NeuroKit2 fails to detect any of P/Q/S/T are discarded. This matches the reference pipeline, which excludes beats with missing fiducials to avoid undefined morphology features.

From valid fiducials we compute morphology descriptors in the same families used by the paper: (i) fiducial amplitudes, (ii) temporal intervals (ms) between fiducials (e.g., PR, QT, QRS-related timings), (iii) distance-like features combining amplitude and timing differences, and (iv) slope features defined as Δamplitude / Δtime for selected fiducial pairs. In addition, we retain rhythm-related quantities (RR and device HR) as inputs for HR/HRV characterization. *Paper vs. us:* the paper reports 35 morphology features and 18 HRV time-domain features; our implementation follows the same feature families and the same "discard if missing fiducials" rule (minor naming/count differences can still exist and are treated as replication differences).

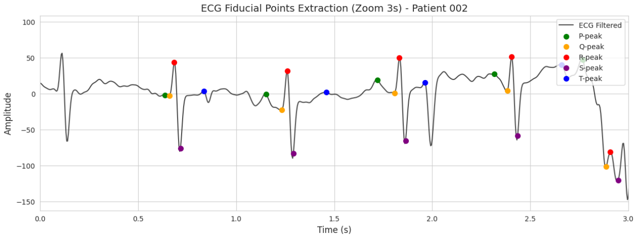
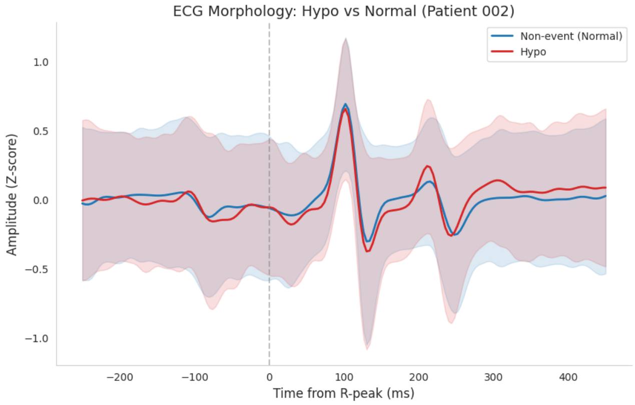
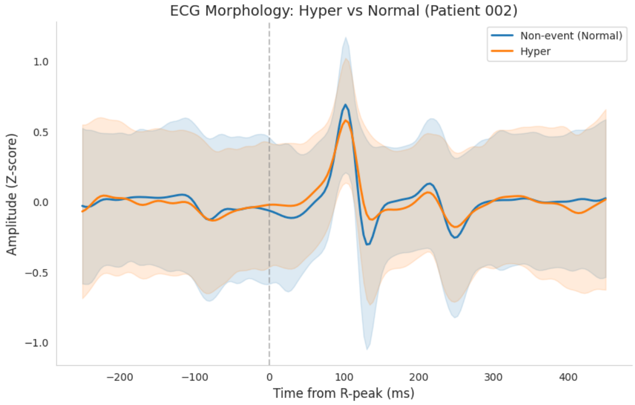


Figure 2: ECG cleaning and fiducial-point delineation (example subject). **What we see:** after cleaning, R-peaks (anchor) are correctly localized on the QRS complexes, and P/Q/S/T points are placed consistently around each beat. This enables stable computation of intervals (e.g., PR, QT, QRS width) and amplitude-based features. Beats where one of these markers is missing are excluded (otherwise intervals/slopes would be undefined). **Paper comparison:** the reference paper follows the same NeuroKit2-based fiducial extraction and explicitly discards beats with missing P/Q/S/T peaks.



(a) Hypoglycemia example.



(b) Hyperglycemia example.

Figure 3: Morphology examples under different glycemic states (same subject).

What the figure shows: average beat morphologies (z-scored) for event vs. non-event windows, with shaded variability. Differences, when present, tend to be subtle and can appear around QRS amplitude/width and ST/T-wave regions; however, overlap between bands is large. **Interpretation:**this plot is descriptive, not a universal "signature": morphology shifts are subject- and context-dependent.**Paper comparison:** this matches the message of the reference paper, which uses morphology examples to illustrate the type of timing/amplitude descriptors extracted from fiducials, while emphasizing strong inter-subject variability.

2.6 Glucose alignment (labeling)

To assign labels, we align ECG-derived beat timestamps to CGM measurements using a forward-direction as-of merge, which associates each beat with the next available CGM sample. A maximum tolerance of 15 minutes is enforced to avoid spurious matches to temporally distant glucose readings. Beat-level labels are computed for both binary tasks according to the aligned CGM value: hypo if glucose < 70 mg/dL, and hyper if glucose > 180 mg/dL. Interval-level labels are then defined on non-overlapping 1-minute windows, consistent with the evaluation unit. *Paper vs. us:* the paper states forward CGM assignment and 1-minute interval evaluation; our code implements the same forward alignment rule and tolerance, then aggregates to 1-minute windows for model training and scoring.

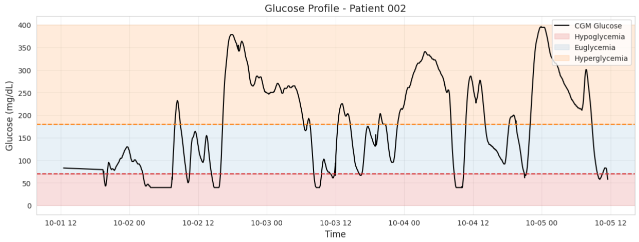


Figure 4: CGM glucose profile (example subject). **What we see:** CGM samples are sparse compared to ECG (5-min points vs. beat-level ECG), with time periods falling into hypo (< 70), euglycemic (70–180), and hyper (> 180) ranges. This clarifies why a forward match with a tolerance is needed: without it, many beats would be paired with stale CGM values, or not paired at all. **Paper comparison:** this is consistent with the paper's labeling definition ("nearest forward CGM reading") and illustrates the same ECG/CGM rate mismatch that motivates the alignment strategy.

2.7 Interval aggregation and HRV computation

Beat-level features are aggregated into fixed, non-overlapping 1-minute windows, which is the evaluation unit used in the reference paper. This reduces single-beat noise and yields more stable predictors at the interval scale.

Paper vs. our code. The paper computes interval-level features from beats and evaluates models per 1-minute interval. Our script implements the same structure by assigning each beat to `interval_id = floor(beat_time, 1min)` and then aggregating within each interval.

Morphology aggregation.

Within each minute, morphology features (amplitudes, intervals, distances, slopes) are summarized using window-level means. When available, device HR is also included as an additional per-beat predictor and is aggregated in the same way. This matches the paper's idea of moving from beat descriptors to robust interval descriptors.

RR cleaning and HRV features.

RR outliers are removed using the same physiological bounds used elsewhere in the pipeline (300–2000 ms). Time-domain HRV metrics are computed on the cleaned RR sequence when enough beats are available in the window. In our code, HRV is computed only if the interval contains at least 20 cleaned RR values, to avoid unstable estimates on short sequences. Computed metrics follow the paper's time-domain family: MeanNN, SDNN, RMSSD, pNN50, Min/Max, percentiles (20/80), and IQR.

Replication note. The paper reports 18 HRV time-domain features; our implementation follows the same family, but the exact feature count can differ slightly depending on (i) the minimum-beat rule per window and (ii) which percentiles are included.

Circadian encoding.

Time-of-day is encoded using sine/cosine transforms of the hour (24h periodicity). This follows the paper's motivation to capture circadian effects without introducing discontinuities at midnight.

2.8 Quality control and dataset rebuilding

Before training, we run a dedicated quality-control (QC) stage that checks (i) RR plausibility, (ii) session reliability, and (iii) class coverage (hypo/normal/hyper). After discarding low-quality sessions, we rebuild the cleaned beat-level and interval-level datasets used for modeling.

Paper vs. our code. The reference paper motivates strict filtering (especially HRC-based) to prevent corrupted ECG from biasing delineation and features. Our QC reproduces that intent and makes it explicit with summary tables and plots; in addition, we apply session-level discard rules to remove sessions with unreliable RR statistics.

Session discard criteria (implemented).

For each subject-session we compute: (i) RR coverage = fraction of beats whose RR survives the [300,2000] ms filter, (ii) median clean RR (physiologic plausibility), (iii) basic duration/beat-count sanity checks. A session is discarded if RR coverage is below 0.8 or if the median clean RR falls outside [350,1200] ms. These thresholds are not meant as clinical rules; they are practical guards against sessions dominated by artifacts.

RR interval filtering (beat-level)	
Raw RR beats	950,668
Filtered beats	81,160 (8.54%)
Clean RR beats	869,508 (91.46%)
Raw median RR	740.0 ms (81.1 bpm)
Clean median RR	716.0 ms (83.8 bpm)
Glucose conditions (interval-level)	
HYPO intervals	1,165 (8.0%), $\mu = 53.7\,\mathrm{mg/dL}$
Normal intervals	9,189 (63.0%), $\mu = 124.6\,\mathrm{mg/dL}$
HYPER intervals	4,227 (29.0%), $\mu = 240.0\,\mathrm{mg/dL}$

Table 2: Summary statistics for RR filtering and CGM-derived condition prevalence. RR filtering removes non-physiological beats (likely artifacts) while preserving the global RR distribution. The interval-level glucose breakdown highlights class imbalance, especially for hypoglycemia.

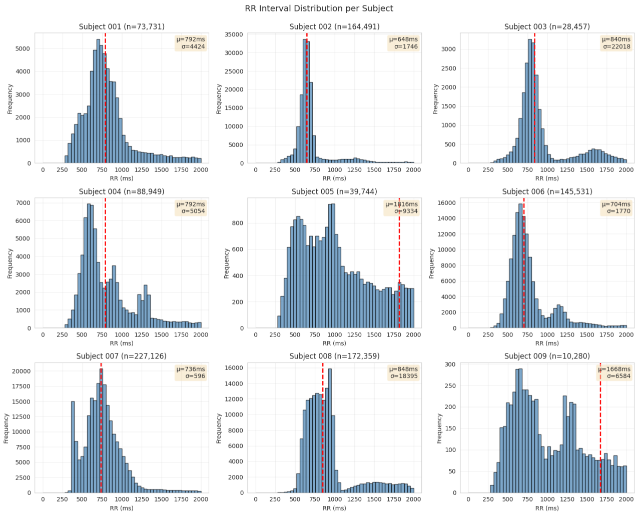


Figure 5: RR interval distributions across subjects (beat-level). **What we see:** subjects have markedly different RR medians and spreads, with some showing broader or multi-modal distributions. **What it implies:** strong inter-subject physiology and heterogeneous signal quality, which can cause fold-to-fold performance variability under subject-wise evaluation. **Paper comparison:** this supports the same conclusion discussed in the reference paper: morphology/HRV baselines are subject-dependent, so conservative temporal validation is needed.

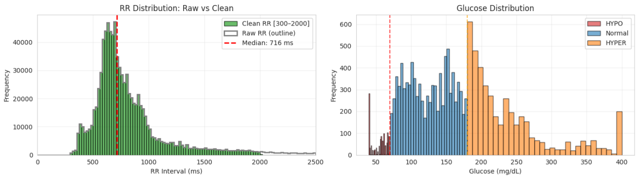


Figure 6: Glucose and RR distributions before vs. after QC. **Left (RR):** raw RR (outline) vs cleaned RR (filled) shows that out-of-range beats are removed while the physiological mass (around the median) stays consistent. **Right (glucose):** hypo/normal/hyper histograms show the intrinsic class imbalance and verify that QC does not introduce a large shift in glucose coverage. **Paper comparison:** the reference workflow emphasizes filtering without biasing coverage; this plot is our explicit check for that requirement.

2.9 Model training and evaluation

All classifiers are implemented as `RandomForestClassifier`. We evaluate two tasks: hypoglycemia (CGM < 70 mg/dL) and hyperglycemia (CGM > 180 mg/dL).

Paper vs. our code. The reference paper trains Random Forest baselines and a fusion model, and it uses block-based temporal splitting to reduce leakage. Our code mirrors that setup: training is done per subject, with 1-hour time blocks split into 5 folds.

Implemented model variants.

We reproduce the model family used in the paper:

- ***M_Beat***: beat-level Random Forest using morphology features and device HR (when present). Output is beat-level probability.
- ***M_MV***: interval score from *M_Beat* via majority voting within the minute (fraction of beats above a learned threshold). The beat threshold is optimized on the training blocks by maximizing F1 on the precision-recall curve.
- ***M_Morph***: interval-level Random Forest trained on aggregated morphology means (and mean device HR when present).
- ***M_HRV***: interval-level Random Forest trained on HRV time-domain features plus circadian encoding (hour sine/cosine).
- ***M_Morph_HRV***: interval-level Random Forest combining aggregated morphology with HRV and circadian features.
- ***MF_Fusion***: multi-threshold fusion. Several beat-level models are trained at different glucose cutoffs; their predicted probability distributions are summarized per minute (e.g., mean probability and binned probability mass) and fused with HRV/circadian features at interval level.

Multi-threshold fusion cutoffs.

Following the reference paper, we train weak beat-level models at multiple glucose thresholds and fuse them at the interval level.
In our implementation, the cutoffs are: (hypo side) 55, 60, 65, 70, 75, 80, 85, 90 mg/dL and (hyper side) 150, 165, 180, 200, 225, 250 mg/dL. Final evaluation remains anchored to the clinical thresholds (70 and 180 mg/dL).

Validation protocol (comparability note).

To reduce leakage from correlated time-series samples, the reference paper uses 1-hour temporal blocks distributed across 5 folds. Our code implements the same idea inside each subject: beats and intervals are assigned to `hour_block = floor(time, 1H)`, hour blocks are shuffled, then split into five folds. Train/test masks are built from hour blocks, and interval masks use the hour of `interval_start`. This keeps evaluation temporally separated while staying patient-wise.

Evaluation metrics.

We report AUC to match the reference paper. However, AUC alone does not describe alert burden under class imbalance (e.g., hypo is only ∼8% at interval level in Table 2). For practical interpretation, it is also necessary to report thresholded metrics (precision/PPV, sensitivity, specificity) and alert-style indicators (e.g., false alarms per day), which we discuss later.

3 Results

This section reports interval-level performance for hypoglycemia and hyperglycemia detection. We compare baseline aggregation strategies and the multi-threshold fusion model, mirroring the reference paper's experimental logic (single-family baselines → combined models → fusion).

Paper vs. our code (important for comparability). The reference paper evaluates at 1-minute resolution and uses conservative temporal splitting to reduce leakage. Our implementation also evaluates on non-overlapping 1-minute windows and uses 5-fold temporal validation based on 1-hour blocks within each subject (hour-block folds), so training and test sets are separated in time while remaining patient-wise.

3.1 Performance overview across models

Table 3 summarizes patient-level AUC (mean ± std across subjects) for each model. Across both tasks, the fusion strategy is the best or tied-best on average: for hypoglycemia, `MF_Fusion` reaches AUC 0.813 ± 0.155; for hyperglycemia, `MF_Fusion` reaches AUC 0.755 ± 0.156. Figures 7–8 visualize the same comparison.

Paper comparison. This layout replicates the paper's summary figure(s) where models are compared by mean AUC with variability across subjects. The same qualitative message holds: fusion (and morphology+HRV combinations) are the most robust across patients, while single-family baselines can be competitive only for specific tasks.

Observed pattern across tasks.

Three consistent trends emerge (and they match the paper's interpretation): (i) HRV-only (`M_HRV`) is weaker for hyperglycemia than morphology-based aggregation, (ii) morphology aggregation (`M_Morph`) is strong for hyperglycemia, (iii) combining families (`M_Morph_HRV`) and fusion (`MF_Fusion`) yields the most stable performance across subjects.

Code check. These variants correspond exactly to the six model branches implemented in the script: beat-level RF (`M_Beat`), minute-level vote (`M_MV`), minute-level RF on aggregated morphology (`M_Morph`), minute-level RF on HRV+time (`M_HRV`), their concatenation (`M_Morph_HRV`), and multi-threshold probability-descriptor fusion (`MF_Fusion`).

Table 3: Patient-level AUC (mean ± std) for hyperglycemia (Hyper) and hypoglycemia (Hypo) detection, comparing our models with the reference paper.

Task	Model	Mean AUC	Std
Hyper **(our)**	**MF_Fusion**	**0.755**	**0.156**
	M_Beat	0.700	0.118
	M_HRV	0.660	0.118
	M_MV	0.716	0.157
	M_Morph	0.749	0.145
	M_Morph_HRV	0.755	0.147
Hyper **(paper)**	**MF_Fusion**	**0.782**	**0.10**
	M_Beat	0.721	0.10
	M_HRV	0.780	0.07
	M_MV	0.470	0.07
	M_Morph	0.719	0.07
	M_Morph_HRV	0.769	0.07

Task	Model	Mean AUC	Std
Hypo **(our)**	**MF_Fusion**	**0.813**	**0.155**
	M_Beat	0.710	0.207
	M_HRV	0.801	0.145
	M_MV	0.744	0.213
	M_Morph	0.763	0.189
	M_Morph_HRV	0.777	0.188
Hypo **(paper)**	**MF_Fusion**	**0.749**	**0.10**
	M_Beat	0.676	0.10
	M_HRV	0.629	0.08
	M_MV	0.250	0.09
	M_Morph	0.684	0.07
	M_Morph_HRV	0.718	0.07

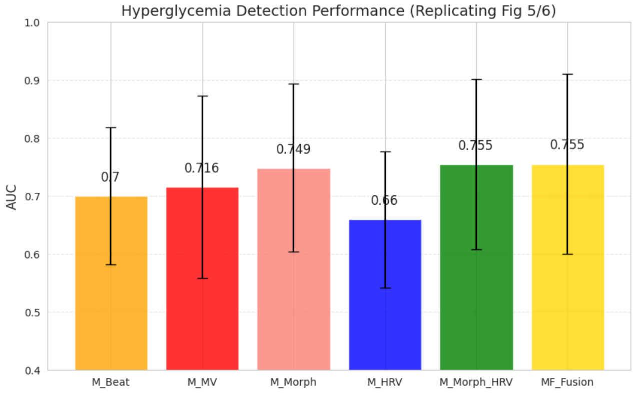


Figure 7: Hyperglycemia detection performance (AUC) across models. **What we see:** morphology aggregation (`M_Morph`, 0.749) clearly outperforms HRV-only (`M_HRV`, 0.660); combined/fusion models reach the best mean AUC (0.755). Error bars (std) indicate strong subject-to-subject variability, consistent with heterogeneous physiology and event density. **Paper comparison:** this mirrors the paper's barplot comparison (same model ordering and interpretation), where morphology-driven models are typically stronger for hyperglycemia and fusion is the most robust.

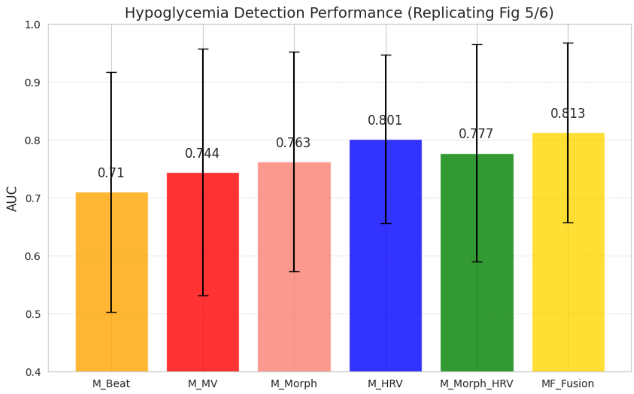


Figure 8: Hypoglycemia detection performance (AUC) across models. **What we see:** HRV-only is already strong (0.801), morphology-only is competitive (0.763), and fusion achieves the best mean AUC (0.813). Variability is larger than for hyperglycemia (notably for beat-level and MV), consistent with the lower prevalence and higher noise around the hypo threshold. **Paper comparison:** the paper reports the same qualitative message: HRV features are particularly informative for hypoglycemia, and fusion benefits from complementary information across feature families.

3.2 Excursion-threshold coverage across patients

Fusion is built on the idea that useful ECG changes may appear around (not only beyond) clinical cutoffs. Figure 9 quantifies how the fraction of beats labeled as "excursion" changes as the glucose threshold is moved.

What we observe. For hypoglycemia, as the threshold decreases (e.g., from 90 → 55 mg/dL), the fraction of beats labeled as hypo drops for all subjects, but the slope differs strongly by subject. For hyperglycemia, as the threshold increases (150 → 250 mg/dL), event fraction drops as expected, again with strong patient variability.

Paper comparison. This figure is directly aligned with the motivation plot in the paper: it shows why training multiple weak classifiers at several thresholds can enrich the representation near the clinical decision point, where the single-threshold problem is most brittle.

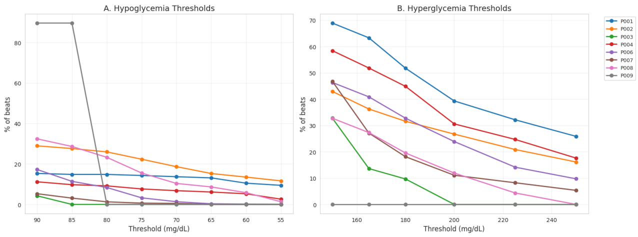


Figure 9: Percentage of beats labeled as excursion as a function of the threshold, shown per patient.

What we see: large inter-subject differences in how quickly event density grows when moving toward milder thresholds (especially near the clinical cutoffs 70 and 180 mg/dL). **Why it matters:** multi-threshold fusion can exploit graded information around the cutoff, rather than relying on a single hard boundary. **Paper comparison:** same rationale and same type of plot used to justify fusion in the reference study.

3.3 Detection rate by severity

AUC summarizes ranking performance but does not show where errors concentrate along the glucose axis. We therefore stratify events by severity and report, for the best-performing setup, the fraction of correctly detected events (TP) versus missed events (FN) in each glucose band (Figure 10).

What we observe. For both tasks, near-threshold bands are harder than extreme bands. This is expected: ECG correlates of glucose are weaker and noisier near the clinical threshold, while severe excursions are more separable.

Paper comparison. The paper discusses the same clinical/physiological intuition: the "borderline" region drives most ambiguity, which is one reason fusion (using neighborhood thresholds) is beneficial.

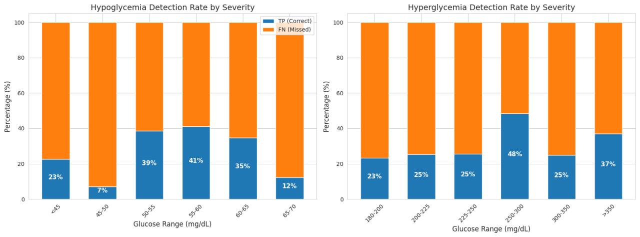


Figure 10: Detection rate by severity for the best-performing setup. **What we see:** TP rates increase in more extreme glucose ranges, while the band close to the clinical cutoff shows the largest FN share. **Interpretation:** this is consistent with harder separability near threshold and supports the multi-threshold fusion idea. **Paper comparison:** same qualitative trend is reported in the reference workflow when stratifying performance by severity.

In hypoglycemia, near-threshold bands (e.g., 65–70 mg/dL) show lower TP compared to more severe ranges. In hyperglycemia, very high glucose ranges show higher TP compared to the 180–200 mg/dL band.

3.4 Circadian distribution of hypoglycemia events

Figure 11 shows the distribution of hypoglycemia events across the 24-hour day. Events are not uniformly spread: they cluster in specific hours.

What we learn from the plot. The peak concentration suggests structured risk patterns (sleep/meal/insulin routines, activity, etc.) rather than random occurrence. This justifies including time-of-day features, because ECG-only descriptors do not encode these external rhythms.

Paper vs. our code. The reference paper encodes circadian time using cyclical features (sine/cosine) and reports that time can carry predictive value. Our implementation matches this: for each 1-minute interval we compute hour-of-day and encode it as $\sin(2\pi h/24)$ and $\cos(2\pi h/24)$, then include it in `M_HRV`, `M_Morph_HRV`, and `MF_Fusion`. So Figure 11 is not just descriptive: it directly motivates a feature block that is actually used by the models.

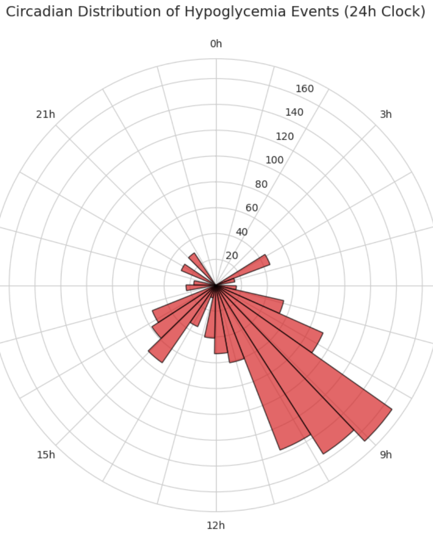


Figure 11: Circadian distribution of hypoglycemia events over a 24-hour clock. **What we see:** event counts peak in a limited set of hours rather than being flat. **Why it matters:** this supports cyclical time-of-day encoding as an additional signal beyond ECG-derived features. **Paper comparison:** same motivation and same modeling choice (cyclical encoding) as in the reference workflow.

3.5 Physiological link: glucose and HRV

We inspected the relationship between glucose and HRV as a sanity check and to interpret why HRV-only models can work well for some subjects but not for all. Figure 12 plots RMSSD (log scale) versus glucose, with the hypo/hyper cutoffs marked.

What we learn from the plot. The scatter shows a broad spread of RMSSD values at similar glucose levels and a non-linear structure. This indicates that autonomic response is not a simple one-to-one function of glucose and is strongly subject/context dependent. So HRV contains information, but it cannot fully determine glycemic state by itself, especially near thresholds.

Paper vs. our code. The reference paper uses time-domain HRV features (including RMSSD) and argues that HRV is informative but benefits from combination with morphology and/or fusion descriptors. Our code follows the same logic: RMSSD is computed per 1-minute window (on cleaned RR) and enters `M_HRV`, `M_Morph_HRV`, and `MF_Fusion`. The "wide spread" visible in Figure 12 is consistent with why `M_HRV` is competitive for hypoglycemia but not dominant for hyperglycemia in our results.

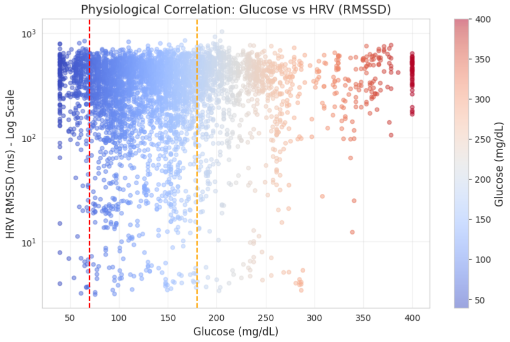


Figure 12: RMSSD (log scale) vs glucose; dashed lines mark hypo/hyper thresholds. **What we see:** large dispersion of RMSSD at fixed glucose and no clean separability at the thresholds. **Interpretation:** HRV is informative but not determinative, motivating combined and fusion models. **Paper comparison:** consistent with the reference paper's rationale for hybrid/fusion approaches.

3.6 Feature importance

To interpret the trained Random Forest models, we inspect feature-importance rankings for: (i) the best-performing single-cutoff interval-level model, and (ii) the multi-threshold fusion model. Because Random Forest importance can vary across folds and patients, we report mean importance across patients and summarize features by families (HRV, time, morphology/probability descriptors).

Paper vs. our code. The reference paper provides the same kind of analysis: it checks whether performance is driven mainly by HRV, by morphology, or by the fusion descriptors built from multiple thresholds. Our code replicates this by extracting `feature_importances_` from the fitted forests, aggregating importances across subjects (and folds), and plotting relative contributions per feature group.

Interval-level importance.

Figure 13 shows group-level importance for the best interval-level model. For both tasks, the dominant contribution comes from aggregated morphology-derived statistics (in this plot, grouped as “Aggregated probability features”), while HRV and time-of-day contribute less but non-zero.

How to read it. If one group accounts for most importance, it means the forest mostly splits on features from that group. Here the plot suggests that, under our split and labeling, morphology-driven descriptors explain most of the predictive signal at interval level, with HRV acting as complementary context.

Paper comparison. This matches the reference paper’s finding that morphology-derived information can dominate for hyperglycemia and that HRV/time can add smaller but useful gains depending on the task and patient.

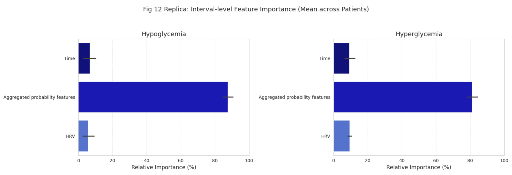


Figure 13: Interval-level feature importance for the best-performing interval-level model (mean across patients). **What we see:** aggregated morphology/probability descriptors dominate, while HRV and time-of-day add smaller contributions. **Interpretation:** interval models rely primarily on beat-derived information under the current split/labels. **Paper comparison:** consistent with the reference paper’s group-level interpretation for interval models.

Fusion-level importance.

Figure 14 reports feature importance for `MF_Fusion`, grouping descriptors into: time-of-day, HRV, probability features at the clinical cutoff (70 for hypo, 180 for hyper), and probability features from *other* thresholds.

What we learn from the plot. For both tasks, probability features from thresholds *other than* the clinical cutoff are the main driver (about 71% for hypo and about 56% for hyper in this run). This directly supports the fusion rationale: the model benefits from neighborhood thresholds, not only from the single cutoff used for evaluation.

The secondary contribution comes from the clinical-threshold block (features at 70/180), while HRV and time-of-day remain smaller but present. So fusion is not just “HRV + one morphology model”: it is mainly leveraging the multi-threshold probability landscape.

Paper comparison. This is the key check against the paper: the reference study argues that multi-threshold fusion adds information around the cutoff; our importance plot confirms that those off-cutoff descriptors receive most weight. If, instead, the 70/180 block dominated, fusion would reduce to a near single-threshold approach.

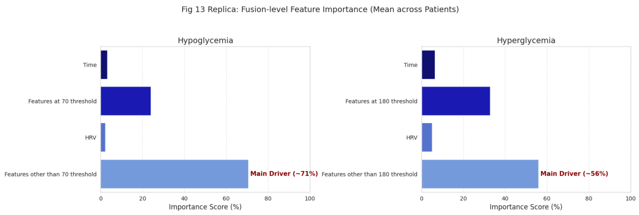


Figure 14: Fusion-level feature importance for `MF_Fusion` (mean across patients). **What we see:** features from thresholds other than the clinical cutoff dominate (main driver), while the clinical-threshold block is secondary; HRV and time contribute marginally. **Interpretation:** fusion is using neighborhood-threshold information as intended. **Paper comparison:** this mirrors the paper’s conclusion that multi-threshold probability descriptors add information beyond single-cutoff modeling.

4 Discussion

Our results indicate that combining complementary ECG information improves discrimination of glycemic excursions. Across both tasks, the multi-threshold fusion strategy (`MF_Fusion`) is the most robust on average, consistent with the idea that relevant changes can appear in a neighborhood around the clinical cutoffs rather than only exactly at 70 and 180 mg/dL.

Paper vs. our code. This is the same core rationale of the reference paper: train multiple weak beat-level models at different glucose thresholds and summarize their outputs at interval level. Our feature-importance analysis also supports this: in `MF_Fusion`, probability descriptors from thresholds *other than* the clinical cutoff carry most of the weight (Figure 14), which is exactly what fusion is supposed to exploit.

4.1 Comparison with the reference paper

Overall, our pipeline matches the reference workflow in the main design choices: ECG cleaning and R-peak detection with NeuroKit2, HRC-based quality control (HRC > 90), forward ECG-to-CGM alignment with tolerance, 1-minute non-overlapping interval aggregation, and Random Forest baselines plus multi-threshold fusion.

At the same time, three factors can shift the reported AUC and the feature-importance profiles.

Temporal validation and leakage control.

The reference study uses 1-hour temporal blocks split into 5 folds to reduce leakage from correlated time series. Our code implements the same principle: each beat/interval is assigned to an hour block, blocks are shuffled, and folds are built from blocks within each subject. So the validation *type* is aligned with the paper.

Remaining comparability detail. Even when the unit is "1-hour blocks", small implementation choices matter: how blocks are defined (by absolute clock hour vs sliding windows), how incomplete hours are handled, and whether blocks are balanced by event density. These details can change fold difficulty and therefore AUC.

Subject/event filtering.

The reference paper may exclude subjects with too few excursions, to avoid unstable per-subject AUC estimates and unreliable training. In our implementation, we discard sessions based on RR coverage and plausibility, but we do not enforce a strict "minimum number of hypo/hyper events per subject" rule before evaluation. This can affect both mean AUC and standard deviation: including low-event subjects often increases variance and can either inflate or depress the mean depending on which subjects remain.

Feature implementation and delineation failures.

The paper specifies 35 morphology and 18 HRV time-domain features. Our code follows the same feature families and discards beats with missing fiducial points, but small differences can remain due to: (i) NeuroKit2 delineation failures on noisy beats, (ii) exact feature definitions (e.g., which fiducial pairs are used for slopes/distances), (iii) aggregation rules (mean vs median, and the minimum-beat constraint for HRV per minute). These differences can shift both performance and which feature groups dominate importance plots.

Quantitative comparison.

The reference paper reports fusion AUC values around ~ 0.75 for hypoglycemia and ~ 0.78 for hyperglycemia under their evaluation protocol. In our replication, `MF_Fusion` reaches 0.813 ± 0.155 for hypoglycemia and 0.755 ± 0.156 for hyperglycemia. Given the sensitivity to subject inclusion, event density, and fold construction, the fact that our values are in the same range (and show the same model ranking trends) supports the validity of the replication. The higher hypoglycemia AUC is consistent with our strong HRV contribution in hypo (Figure 8) and with the dominance of off-cutoff fusion descriptors (Figure 14); the slightly lower hyperglycemia AUC can be explained by subject-dependent morphology behavior and the higher variability across patients (Figure 7).

Paper-aligned sanity checks from our figures. Several plots support that the pipeline behaves as intended, in the same way the paper motivates each stage: (i) QC filtering removes corrupted RR without distorting glucose coverage (Figure 6), (ii) threshold coverage varies strongly across subjects, motivating fusion (Figure 9), (iii) near-threshold ranges are harder, while extreme excursions are easier (Figure 10), (iv) feature importance confirms that multi-threshold descriptors are the main fusion driver (Figure 14).

4.2 Practical meaning of interval-level performance

Interval-level AUC is a good summary to compare modeling strategies, but it does not directly reflect alert usability under class imbalance. In our cleaned interval dataset, hypoglycemia accounts for only about 8% (Table 2). So the operating threshold matters: a model can rank well (high AUC) but still generate many false alerts if tuned aggressively.

Paper vs. our code. The reference paper emphasizes evaluation beyond AUC when thinking about deployment. Our current report follows the paper in reporting AUC as the primary comparable metric, but the same deployment-oriented additions are needed here: precision (PPV), sensitivity/specificity at chosen thresholds, and alert-burden measures (e.g., false alarms/day). In addition, event-level scoring is more realistic: consecutive positive 1-minute windows should be merged into a single excursion event before counting detections and false alarms.

4.3 Limitations and next steps

This study is ECG-only, so it cannot represent noncardiac confounders that affect glucose dynamics (meals, insulin dosing, stress, activity, sleep). In addition, the number of subjects is limited and excursion events are relatively sparse, which increases fold-to-fold variability and widens confidence intervals.

The next step is directly tied to the paper's evaluation framing and to what our current code already supports: add alert-oriented metrics and event-level scoring to connect interval AUC to practical usability.

5 Conclusions

We implemented and evaluated a wearable ECG pipeline for hypoglycemia and hyperglycemia detection based on 1-minute interval aggregation and Random Forest modeling. Across both tasks, the multi-threshold fusion strategy (`MF_Fusion`) provides the best overall robustness across subjects, matching the reference paper's main conclusion that neighborhood-threshold information improves decisions beyond a single cutoff.

Our results also clarify the complementary role of feature families. HRV and circadian time features are particularly relevant for hypoglycemia, while aggregated morphology is stronger for hyperglycemia. Feature-importance analysis supports that fusion is not driven only by the clinical cutoffs: probability descriptors from multiple thresholds dominate model importance, consistent with the intended fusion mechanism.

Future work will focus on a tighter match to the reference split and filtering protocol, and on evaluation closer to deployment needs (PPV/sensitivity trade-offs, false alarms/day, and event-level scoring).