Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

DriveGuard: A Personalised, Confidence-Aware Temporal Multimodal Framework for Driver Fatigue Estimation and Graduated Alerting

Raj Surase

rajsurase3@gmail.com

Sanjivani University

Research Article

Keywords: Driver fatigue detection, Driver monitoring systems, Eye Aspect Ratio (EAR), PERCLOS, Personalised fatigue estimation, Temporal modelling, Multimodal feature fusion, Confidence-aware classification, Graduated alerting, Synthetic-data simulation.

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11712148/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

DriveGuard: A Personalised, Confidence-Aware Temporal Multimodal Framework for Driver Fatigue Estimation and Graduated Alerting

A Synthetic-Data Simulation Study

Raj Surase
Department of Artificial Intelligence and Data Science, Sanjivani University, Maharashtra, India
Article type: Research Article (computational study) | Corresponding author: Raj Surase

Preprint notice. This is a preprint that has not been peer reviewed. All results come from a computational simulation with synthetic data. No driving video, human participant, patient data or public dataset was used. The results are not estimates of real-world accuracy, and nothing in this paper should be used to guide safety-related decisions about driving or vehicle design.

Abstract

Background. Camera-based driver-fatigue monitors often apply one population-wide eye-closure threshold to isolated frames, which is sensitive to individual facial differences and short-lived movements.

Methods. We describe DriveGuard, a personalised, confidence-aware temporal framework that combines eye (EAR, PERCLOS, blinks), mouth (MAR, yawn fraction) and head-pitch features over four consecutive 3-s windows, normalises them with a 45-s per-driver enrolment profile, and converts classifier confidence into graduated alerts through exponential smoothing, confidence gating and hysteresis. The framework is evaluated in a seeded computational simulation of 48 synthetic drivers (2,496 sequences) with six-fold driver-disjoint cross-validation. Because the data generator defines both the signals and the state labels, the results describe how the method behaves in the simulation; they are not estimates of real-world accuracy.

Results. The proposed configuration reached a per-driver macro-F1 of 77.5% (95% bootstrap interval 74.7–80.2) against 58.3% for a fixed EAR/PERCLOS rule. Ablations showed gains from temporal context (+8.5 points), personalisation (+6.7) and the head-pitch stream (+6.9), whereas the mouth stream added no measurable benefit (+0.5). Smoothing reduced level changes per simulated drive from 10.7 to 3.8 (3.3 with confidence gating) at the cost of a longer advisory delay (6.2 s with window-wise decisions versus 12.3–12.8 s). Sigmoid recalibration worsened calibration. Replication on four further seeds and sensitivity analyses gave a consistent ranking of configurations. The classifier has 93 parameters.

Conclusions. Within the simulation, temporal context, per-driver normalisation and smoothing each improved state estimation or alert stability, and two components (mouth stream, sigmoid recalibration) did not help. These findings are hypotheses about method behaviour that require validation on real, ethically approved driving data. All scripts needed to regenerate every table and figure are provided.

Keywords: driver fatigue; eye aspect ratio; PERCLOS; per-driver personalisation; temporal multimodal fusion; alert escalation; synthetic-data simulation

1. Introduction

Driver fatigue reduces attention, slows reactions and raises the risk of road incidents. Cameras offer a non-contact way to observe visible cues, and facial landmarks make lightweight measurements such as the eye aspect ratio (EAR) [1] and PERCLOS, the proportion of time the eyes are closed [2], cheap to

compute. Landmark pipelines such as MediaPipe [3], [4] run on mobile-class processors, which makes continuous in-cabin monitoring technically practical.

Three practical issues remain. First, resting eye openness differs between people, so a single threshold tends to over-trigger for some drivers and under-trigger for others. Second, a blink, a yawn or a glance can occur during normal driving; evidence is more reliable when several indicators are observed over time. Third, systems that issue a warning on every threshold crossing can produce unstable, nuisance alerts, whereas drowsiness develops gradually and suggests graduated responses.

DriveGuard addresses these issues by combining (i) three parallel feature streams (eye, mouth, head pitch), (ii) a short temporal context, (iii) per-driver normalisation from a brief enrolment, and (iv) a confidence-aware alert layer. The ingredients are established; the contribution of this paper is their integration in one ablatable pipeline and a controlled evaluation of what each component contributes. We make three contributions: (1) a fully specified framework and mathematical formulation; (2) a reproducible, seeded simulation protocol with driver-disjoint validation, component ablations, calibration analysis, replication across seeds and sensitivity analyses; and (3) a candid account of where each component helps, where it does not, and which findings need confirmation on real data.

The study is a computational experiment on synthetic data, and it is designed as a controlled test bed in which the ground truth is known by construction. It makes no claim about on-road performance, about real fatigue physiology, or about the superiority of the method over published systems. The paper follows the usual structure: Section 2 gives brief background, Section 3 the methods, Section 4 the results, Section 5 the discussion with limitations and a validation plan, and Section 6 the conclusions.

2. Background

This section is a brief background, not a systematic review. EAR was introduced for blink detection from six eye landmarks [1], and PERCLOS is a long-standing alertness measure [2]. Both are interpretable and inexpensive, which makes them natural baselines. The mouth aspect ratio (MAR) supports yawn analysis, and head pose adds information about nodding and sustained posture change. Embedded vision systems based on the eyes and mouth have been reported [5], and compact quantised networks have been shown to run on microcontroller-class hardware [6]; MobileNetV3 [7] and integer-only inference [8] are common routes to efficient deployment. Multi-level drowsiness classification has been explored with physiological signals [9], whereas the facial-video systems cited here [5], [6] concentrate on distinguishing drowsiness from alertness.

YawDD [10], NTHU-DDD [11] and UTA-RLDD [12] are widely used driver-monitoring datasets. Their annotation schemes and access conditions differ, so suitability must be checked before use: a yawn-labelled dataset does not by itself provide fatigue-severity ground truth. These datasets were not used in this study.

Individual variability and design motivation. Resting eye openness varies across people because of facial anatomy, eyelid shape, camera placement and posture, so one population-wide EAR cut-off cannot suit every driver. A short per-driver reference measurement is a natural remedy, but with few enrolment samples the estimated reference spread is noisy, which motivates the shrinkage in Eq. (4). Fatigue is a process rather than an event: a single blink, yawn or glance is weak evidence, whereas the trend of several indicators over seconds is stronger, which motivates a short temporal encoder. Finally, warnings that fire on every threshold crossing may be perceived as a nuisance, so a graduated and temporally stable alert policy is desirable; its cost is detection delay, which this paper measures rather than assumes away. Probabilistic outputs are useful as confidence only if calibrated; calibration has been studied extensively [13], [14] and the Brier score [15] is a standard proper scoring rule, so calibration is treated here as a hypothesis to test rather than an assumption.

3. Methods

3.1 Study design and hypotheses

The study is a seeded computational experiment. For a driver not seen during training who supplies a short alert-state enrolment, the task is to estimate at each decision step a fatigue state $k = 0, 1, 2$ (alert, early fatigue, fatigued), a confidence value, and a graduated alert level. The states are labels defined by the synthetic generator (Section 3.5); they are not clinically validated categories. Four hypotheses were stated before the experiments: **H1**, temporal context improves state estimation over a single window; **H2**, per-driver normalisation improves it over raw features; **H3**, combining eye, mouth and head streams improves it over the eye stream alone; **H4**, smoothing with confidence gating reduces alert instability without unacceptable added delay. Each hypothesis is evaluated *within the simulation* only.

3.2 Processing pipeline and feature streams

Figure 1 shows the six stages: video acquisition, face and landmark localisation, parallel feature extraction, temporal encoding with personalised normalisation, risk and confidence estimation, and alert output. The three streams are kept separate until fusion so that each can be ablated. Figure 1 is a design diagram, not a measured result. In the simulation no video or landmarks exist: EAR, MAR and head pitch are generated directly as frame-level signals at 10 fps, so Eqs. (1) and (3) define the intended real-data features but are not exercised on landmark coordinates.

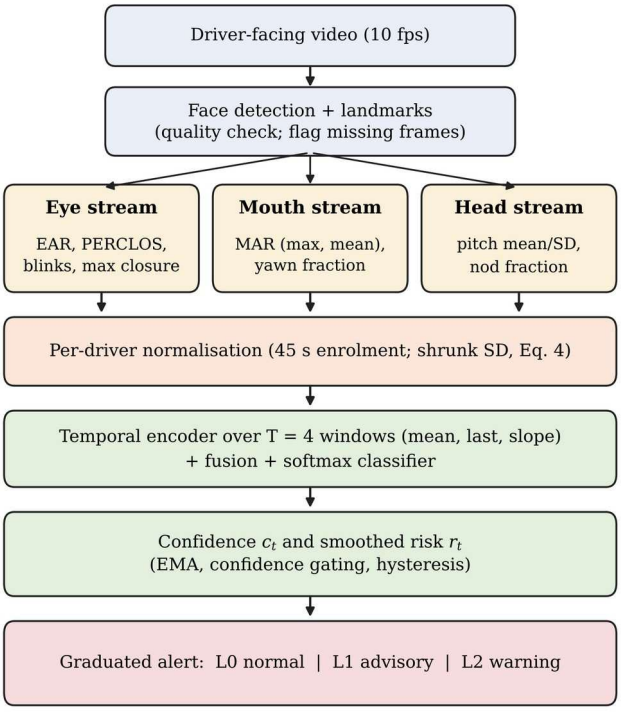


Figure 1. Proposed DriveGuard processing pipeline. This is a design diagram, not a measured result.

Eye stream: EAR per frame, PERCLOS, the number of closure episodes (blinks) and the longest closure run in a 30-frame (3-s) window. *Mouth stream:* maximum and mean MAR and the fraction of frames with large mouth opening; brief speech-like openings are a deliberate confound. *Head stream:* pitch mean, pitch standard deviation and the fraction of frames with strongly lowered pitch (nodding). Head pose is treated as supporting evidence, not proof of fatigue. The ten window features are defined in Appendix A. For six eye landmarks $p_1 \ldots p_6$ and N frames per window with closure threshold τ:

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\,\|p_1 - p_4\|} \tag{1}$$

$$\mathrm{PERCLOS} = \frac{1}{N}\sum_{t=1}^{N} [\,\mathrm{EAR}_t < \tau\,] \tag{2}$$

$$\mathrm{MAR} = \frac{d_{v1} + d_{v2} + d_{v3}}{3\,d_h} \tag{3}$$

A fixed population threshold $\tau = 0.22$ is used for PERCLOS and blink counting.

3.3 Per-driver enrolment, temporal encoding and classifier

During a 45-s enrolment in an alert state (15 windows) the system stores per-feature means and standard deviations. Window features are then normalised against this profile. For driver *i* with enrolment mean μ_i and standard deviation σ_i per feature, the personalised feature vector is

$$z_t = \frac{x_t - \mu_i}{s_i + \varepsilon}, \qquad s_i = \sqrt{\frac{1}{2}\sigma_i^2 + \frac{1}{2}\sigma_{\mathrm{pop}}^2} \tag{4}$$

where s_i is a shrunk deviation, σ_{pop} is the pooled within-driver enrolment deviation of the training drivers only, and $\varepsilon = 10^{-3}$. The shrinkage stabilises the estimate from only 15 windows. A sequence of $T = 4$ normalised windows (12 s) is summarised by its mean, last value and least-squares slope, and a multinomial logistic classifier with class-balanced weights outputs state probabilities:

$$\varphi_t = [\,\mathrm{mean}_\tau(z),\ z_t,\ \mathrm{slope}_\tau(z)\,] \tag{5}$$

$$p_t(k) = \mathrm{softmax}(W\varphi_t + b)_k \tag{6}$$

The 10 features × 3 summaries give 30 inputs and, with three classes, 93 parameters (3 × 30 weights + 3 biases). The confidence is $c_t = \max_k p_t(k)$.

3.4 Alert layer

An ordered risk score $g(p_t) = p_t(1) + 2p_t(2)$ is smoothed by exponential smoothing (Eq. 7) with gain $\kappa_t = 0.4$ for the plain EMA policy and $\kappa_t = 0.5\,c_t$ for the confidence-aware policy. The target level is 2 if $r_t > \theta_2 = 1.3$, 1 if $r_t > \theta_1 = 0.6$, and 0 otherwise. The confidence-aware policy escalates only if $c_t \geq 0.6$; every policy de-escalates only when r_t falls 0.15 below the threshold of the current level (hysteresis). Thresholds were fixed before the experiments and not tuned on test drivers. A third policy, window-wise argmax, simply reports the most probable state at every step.

$$r_t = r_{t-1} + \kappa_t(g(p_t) - r_{t-1}), \qquad g(p_t) = p_t(1) + 2\,p_t(2) \tag{7}$$

Algorithm 1. DriveGuard enrolment and online alerting

Enrolment (once per driver, about 45 s, alert state): collect 15 windows; compute per-feature mean μ_i and SD σ_i; reject and repeat if face-quality or motion checks fail; store locally.

For each 3-s window *t*: (1) extract the 10 window features and flag the window as low quality if landmarks are missing; (2) $z_t \leftarrow$ Eq. (4); (3) append to a buffer of $T = 4$ windows.

If the buffer is full and quality is acceptable: (4) $\varphi_t \leftarrow$ mean, last, slope (Eq. 5); (5) $p_t \leftarrow$ softmax (Eq. 6); (6) $c_t \leftarrow \max p_t$, $g \leftarrow p_t(1) + 2p_t(2)$; (7) $r_t \leftarrow$ Eq. (7).

Alert level: escalate to the target level only if $c_t \geq 0.6$; de-escalate only if r_t is 0.15 below the current threshold; otherwise hold. On low-quality windows hold the level and report reduced confidence (never silently escalate).

3.5 Synthetic data generator

No driver video, participant data or public dataset was collected or processed. Data are generated by the script experiment.py (seed 2026). Each of 48 synthetic drivers has a resting EAR μ_i drawn from N(0.30, 0.045^2) clipped to [0.20, 0.38] (empirical mean 0.305, SD 0.042; three drivers below 0.25), plus individual noise levels, a resting MAR and a head-pitch offset. Each driver also has random expression weights (0.3–1.4) that scale how strongly fatigue appears in the eyes, mouth and head, so some drivers are mainly "eye" expressers and others mainly "head" expressers. Table 1 lists the state-dependent parameters at unit weight. Fatigue strength varies between sequences (ν drawn from N(1, 0.25^2)), and speech-like mouth openings and head glances occur in all states. Each driver contributes 24 alert, 14 early-fatigue and 14 fatigued sequences of four windows (2,496 sequences in total; class counts 1,152 / 672 / 672). Absolute values therefore illustrate method behaviour and must not be read as expected real-world accuracy.

Table 1. Synthetic generator parameters per state (unit style weight).

State	Blinks / window	Closure (frames)	EAR drop	Yawn prob.	Nod prob.	Pitch drift (°)
Alert	1.2	1.5	0	0.01	0.01	0
Early fatigue	1.4	2.6	0.04	0.06	0.08	1.5
Fatigued	1.8	4.5	0.10	0.16	0.22	4.0

3.6 Evaluation protocol and compared configurations

The 48 drivers are divided into six driver-disjoint folds of eight, so every test driver is unseen at training time (test drivers enrol with their own alert-state windows before testing). Population statistics for normalisation, scalers and models are fitted on training drivers only. Eleven configurations are compared (Table 2). The rule baseline predicts fatigued if mean PERCLOS ≥ 0.15, early fatigue if ≥ 0.08, and alert otherwise (thresholds fixed a priori). Per-driver macro-F1 is the primary metric and is reported with 95% percentile bootstrap intervals over drivers (2,000 resamples); paired comparisons use the Wilcoxon signed-rank test on per-driver values [16], [17]. Calibration is assessed by the expected calibration error (ECE) over $B = 15$ confidence bins and the multiclass Brier score [15]:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|S_b|}{n} \left| \mathrm{acc}(S_b) - \mathrm{conf}(S_b) \right| \tag{8}$$

Table 2. Compared configurations.

ID	Description
C1	EAR/PERCLOS rule (fixed thresholds)
C2	Raw features, single window, eye + mouth + head
C3–C5	Raw, temporal: eye / eye + mouth / eye + mouth + head
C6	Personalised, single window, eye + mouth + head
C7–C9	Personalised, temporal: eye / eye + mouth / eye + mouth + head (proposed = C9)
C10	C9 with sigmoid probability calibration (3-fold, driver-grouped)
C11	C9 with gradient-boosted trees instead of logistic regression

How to read the statistics. Because the generator and its sample size (48 drivers) are chosen by the experimenter, the intervals and *p*-values quantify variation across simulated drivers of this one generator

relative to simulation noise. They are reported to show the size of effects against that noise and are not tests of hypotheses about real drivers. No correction for multiple comparisons was applied, so borderline *p*-values should be treated as indicative only. No hypothesis-driven tuning was performed on test drivers.

3.7 Simulated drives and alert policies

For each driver one simulated drive of 87 windows was generated (3 warm-up and 36 alert windows, then 24 early-fatigue and 24 fatigued windows; state changes are abrupt) and replayed through the out-of-fold C9 model under the three policies of Section 3.4. Because each decision uses a four-window sequence, delays include the encoder's lag. Delays are reported in seconds (3 s per window) after the true onset of the early-fatigue (advisory, L1) and fatigued (warning, L2) phases; a drive that never reaches a level is assigned a censored delay of 72 s.

3.8 Replication and sensitivity analyses

The main comparison was repeated on four additional independently generated datasets (seeds 2027–2030; same parameters and driver-disjoint six-fold protocol). Sensitivity to the number of windows in a sequence, enrolment length, contamination of the enrolment with early-fatigue windows, and sensor-noise level was examined on the seed-2027 dataset, and the smoothing gain κ_0 was swept on the main-seed probabilities. These are secondary analyses in which each value is a single run of the protocol, without intervals.

3.9 Software, reproducibility and timing

All analyses used Python 3.12.3, NumPy 2.5.4 [19], SciPy 1.18.1 [20], scikit-learn 1.9.1 [18] and Matplotlib 3.11.2 [21] with fixed seeds (2026 main; 2027–2030 replication). Alert thresholds (0.6, 1.3), hysteresis (0.15), the confidence gate (0.6) and the rule thresholds (0.08, 0.15) were fixed before the experiments. Minor numerical differences (for example for the gradient-boosting model C11) can occur with other library versions. Inference time was measured with NumPy on a single-thread CPU in a Linux container (3,000 repetitions after 200 warm-up calls); it excludes landmark extraction and camera I/O and is not an embedded-hardware measurement.

3.10 Use of AI tools

A large language model (Anthropic's Claude) was used as an assistant to draft and edit the manuscript text, to write and run the simulation, analysis and figure code, and to format the document, including this revised version. The model is not an author and carries no responsibility for the work. The author specified the study, reviewed the code and text, and is responsible for the content. All numbers in this version were regenerated by running the supplied scripts and checked against the tables and figures.

4. Results

All numbers were produced by running the simulation scripts and describe the simulation only.

4.1 State estimation and ablation

Table 3 and Figure 2 show that the proposed configuration (C9) attains the highest per-driver macro-F1 (77.5%), 19.2 points above the fixed rule (95% interval 14.4 to 24.2). Table 4 isolates each component. Personalisation adds 6.7 points over the raw temporal model, temporal context adds 8.5 points over a single window, and the full multimodal model adds 7.4 points over the eye stream alone. Adding the mouth stream to the eye stream gave no measurable gain (+0.5; interval −0.7 to 1.7); the benefit of multimodality in this simulation comes mainly from the head stream, which reflects how the generator was built and may not hold on real data. Replacing logistic regression with gradient-boosted trees changed nothing material (−0.8 points; interval −2.2 to 0.8). The recall of the early-fatigue state was the

lowest of the three (66.5%, versus 86.1% for alert and 81.2% for fatigued), and Figure 3 shows that errors are mostly between adjacent states.

Table 3. Driver-independent three-state results, pooled over test folds.

ID	Acc. (%)	Macro-F1 (%)	Per-driver F1 (%) [95% CI]	ECE	Brier
C1	62.3	59.5	58.3 [53.8, 62.5]	—	—
C2	63.4	61.1	59.5 [55.9, 63.2]	0.039	0.479
C3	72.4	69.7	69.0 [66.1, 72.2]	0.067	0.393
C4	73.4	71.1	70.5 [67.3, 73.6]	0.058	0.381
C5	73.8	71.6	70.8 [67.8, 73.8]	0.028	0.360
C6	71.5	69.9	69.0 [66.3, 71.7]	0.033	0.381
C7	73.4	71.2	70.1 [67.0, 73.3]	0.039	0.369
C8	73.7	71.7	70.6 [67.4, 73.8]	0.027	0.363
C9	79.5	78.0	77.5 [74.7, 80.2]	0.021	0.299
C10	77.5	71.2	71.0 [68.7, 73.3]	0.126	0.332
C11	79.0	77.2	76.8 [74.0, 79.4]	0.052	0.305

Accuracy and macro-F1 are pooled over all test sequences; per-driver F1 is the mean over the 48 drivers. ECE and Brier score are not defined for the deterministic rule. C9 is highlighted.

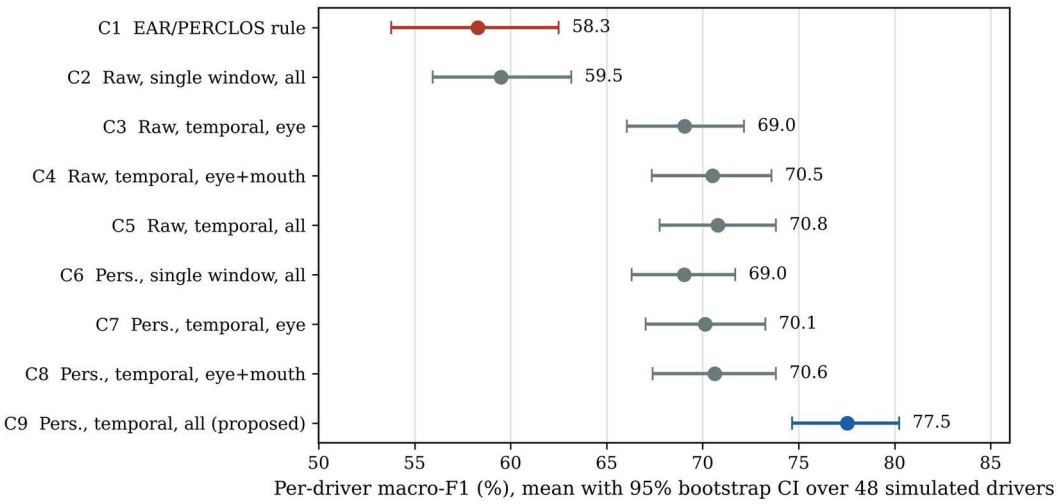


Figure 2. Per-driver macro-F1 (mean with 95% bootstrap interval over 48 simulated drivers) for configurations C1–C9. “Pers.” = personalised. The horizontal axis starts at 50%.

Table 4. Paired per-driver comparisons of macro-F1.

Comparison (A vs B)	ΔF1 (points)	95% CI	p
C9 vs C1 (proposed vs rule)	+19.2	[14.4, 24.2]	<0.001
C9 vs C5 (personalisation)	+6.7	[3.3, 10.1]	<0.001
C9 vs C6 (temporal context)	+8.5	[6.3, 10.6]	<0.001
C9 vs C7 (multimodal vs eye)	+7.4	[4.7, 10.1]	<0.001
C9 vs C8 (head stream)	+6.9	[4.5, 9.5]	<0.001
C8 vs C7 (mouth stream)	+0.5	[−0.7, 1.7]	0.546
C11 vs C9 (model class)	−0.8	[−2.2, 0.8]	0.327

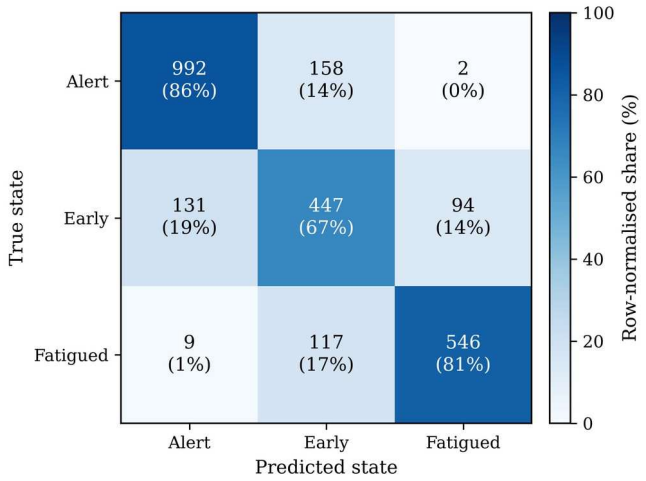


Figure 3. Confusion matrix of C9: counts and row-normalised percentages, pooled over test folds.

4.2 Effect of individual differences

Figure 4 plots per-driver accuracy against resting EAR. The three simulated drivers with resting EAR below 0.25 were misclassified far more often by the fixed rule: mean accuracy was 26.9% (rule), 53.8% (raw temporal model) and 70.5% (proposed), against 64.6%, 75.2% and 80.1% for the remaining drivers; the rule flagged 100% of the alert-state sequences of these three drivers as fatigued, versus 26% for the others. With only three such drivers this illustrates the motivating mechanism and is not a statistical result.

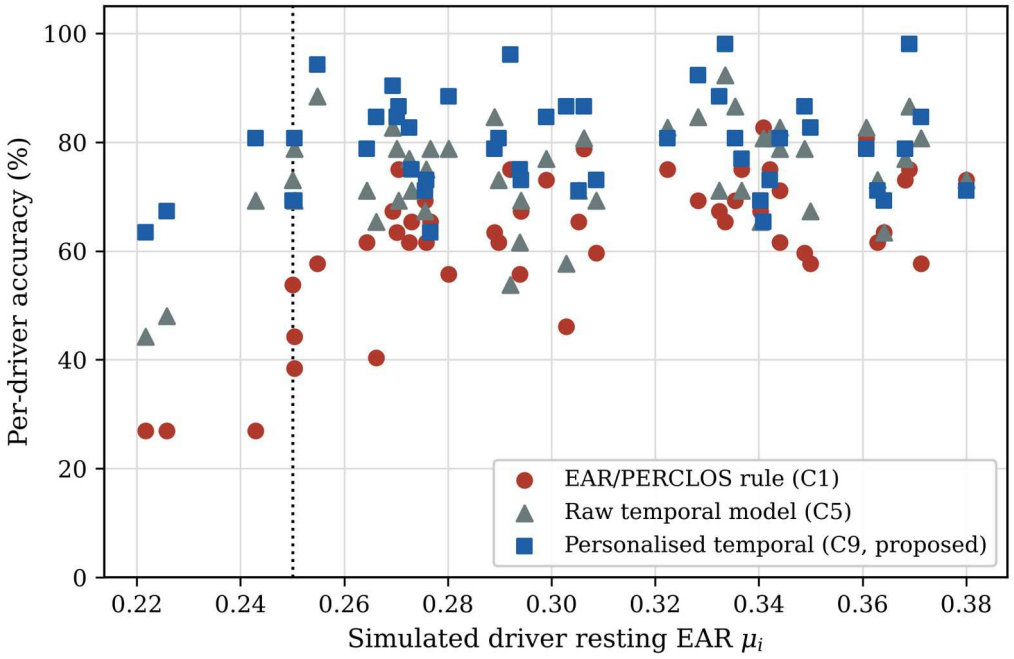


Figure 4. Per-driver accuracy versus resting EAR for the rule (C1), raw temporal model (C5) and proposed model (C9). The dotted vertical line marks resting EAR = 0.25.

4.3 Calibration and confidence

The uncalibrated class-balanced logistic model was already reasonably calibrated (ECE 0.021, Brier 0.299). Sigmoid recalibration (C10) made calibration worse (ECE 0.126, Brier 0.332) and lowered early-fatigue recall to 31.0%, probably because recalibration on class-imbalanced training data pushes predictions toward the majority (alert) class. The calibration step as specified is therefore not supported by this simulation, and C9 (uncalibrated) was used for the alert experiments. Figure 5 shows the reliability curves.

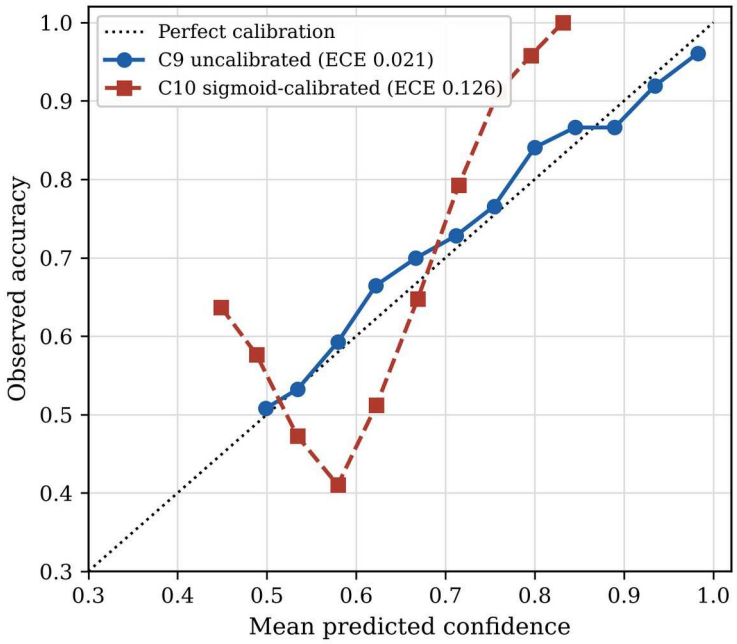


Figure 5. Reliability diagram for C9 (uncalibrated) and C10 (sigmoid-calibrated); the dotted line is perfect calibration. Bins with fewer than 10 sequences are omitted.

4.4 Graduated alerting in simulated drives

Smoothing sharply reduced alert instability (Table 5): level changes fell from 10.7 to 3.8 per drive with EMA and to 3.3 with confidence gating (both $p < 0.001$ against the preceding policy), and the any-alert rate in the alert phase fell from 14.8% to 11.5% and 10.6% (confidence-aware versus EMA, $p = 0.153$). No policy produced an L2 warning during the alert phase, so warning false alerts cannot be compared. The cost is delay: the advisory arrived after 6.2 s with window-wise decisions, 12.3 s with EMA and 12.8 s with confidence gating (confidence-aware versus EMA, $p = 0.011$); the warning delay was 8.7, 6.3 and 7.2 s, and window-wise decisions failed to reach L2 in 2% of drives (one driver). H4 is thus partly supported: smoothing clearly reduces instability, while the additional benefit of confidence gating is small and comes with a statistically detectable but modest increase in delay. Figure 6 gives an example.

Table 5. Simulated-drive alert policies (mean over 48 drives).

Metric	Window-wise	EMA + hysteresis	Confidence-aware
Any alert (≥ L1) in alert phase (%)	14.8	11.5	10.6
Warning (L2) in alert phase (%)	0.0	0.0	0.0
Level changes per drive	10.7	3.8	3.3
Delay to L1 after onset (s)	6.2	12.3	12.8
Delay to L2 after onset (s)	8.7	6.3	7.2
Drives never reaching L2 (%)	2.1	0.0	0.0
Level MAE (levels)	0.219	0.208	0.197

Delay for drives that never reach a level is censored at 72 s.

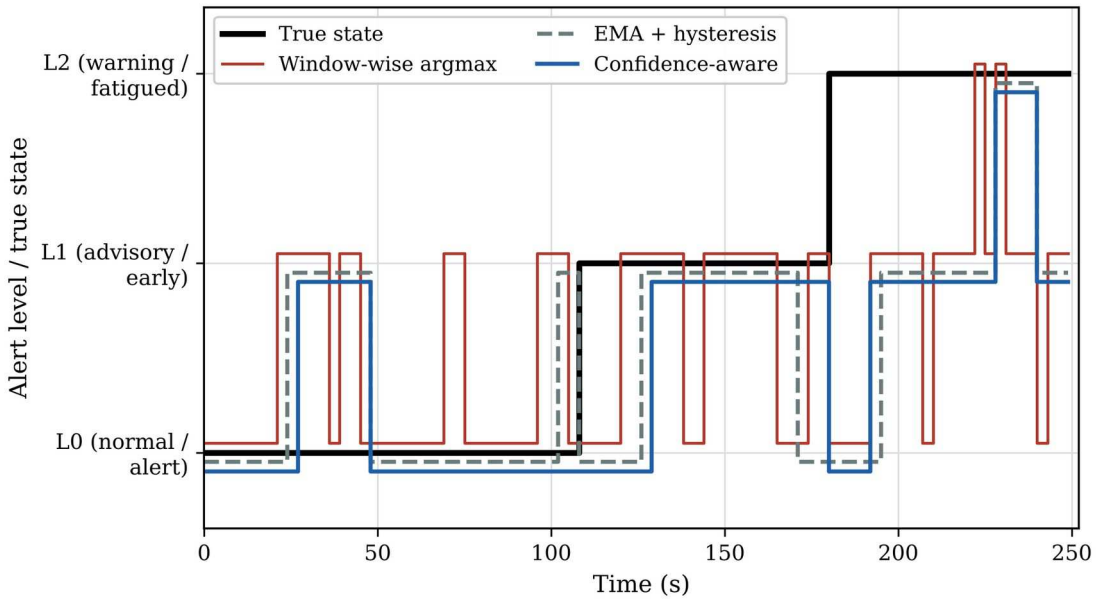


Figure 6. Alert levels over a simulated drive for a driver with relatively low resting EAR (μ = 0.250); the true state is shown in black. Lines are offset vertically by a small amount so that they remain distinguishable.

4.5 Per-class performance

Early fatigue is the hardest state (Table 6) because its cues overlap with both neighbours; its recall and precision are clearly below those of the alert and fatigued states. This is consistent with the intuition that

the onset of fatigue is the least distinct and most ambiguous state; on real data it is also the state whose labels are likely to be noisiest.

Table 6. Per-class results of C9 (pooled over test folds).

State	Precision (%)	Recall (%)	F1 (%)	Support
Alert	87.6	86.1	86.9	1,152
Early fatigue	61.9	66.5	64.1	672
Fatigued	85.0	81.2	83.1	672

4.6 Model size and computational cost

The classifier is tiny (Table 7) and adds negligible computation beside landmark extraction. These timings were measured on a development CPU and do not represent embedded performance.

Table 7. Model size and measured inference cost.

Quantity	Value
Classifier parameters (3 × 30 weights + 3 biases)	93
FP32 model size	372 B
Window features (30 frames), mean	0.030 ms
Full decision (4 windows + normalisation + classifier), mean / p95	0.150 / 0.184 ms
Max. absolute difference vs scikit-learn probabilities	3.3×10^{-16}

Measured with NumPy on a single-thread CPU in a Linux container (not embedded hardware); excludes landmark extraction and camera I/O. No quantisation was evaluated.

4.7 Replication and sensitivity analyses

On all four additional seeds the proposed C9 was the best configuration and the rule the worst (Table 8): C9 (74.9 ± 1.3) exceeds the rule (56.7 ± 2.3), the raw temporal model (64.4 ± 3.3), the personalised single-window model (67.3 ± 1.6) and the eye-only model (68.5 ± 2.0). The order of the middle configurations varies between seeds. The seed-to-seed SD of about 1–3 points is a useful yardstick for the other analyses.

Table 8. Per-driver macro-F1 (%) over four independent seeds.

Seed	C1 rule	C5 raw temp.	C6 pers. single	C7 pers. eye	C9 proposed
2027	59.1	61.6	68.8	69.7	76.8
2028	54.1	64.3	65.0	67.5	74.4
2029	58.1	69.0	67.9	70.5	74.7
2030	55.3	62.6	67.3	66.3	73.9
Mean ± SD	56.7 ± 2.3	64.4 ± 3.3	67.3 ± 1.6	68.5 ± 2.0	74.9 ± 1.3

Longer temporal context helped monotonically within the tested range (Table 9, Figure 7a): T = 1: 68.8%, T = 4: 76.8%, T = 6: 78.8%, at the price of a longer decision lag (T = 6 spans 18 s). Enrolment length mattered mostly at the short end: 3 windows (9 s) gave 72.5%, whereas 5 to 15 windows gave 76.4% to 76.8% and 30 windows 78.7%, so the 45-s design point is on a plateau but a very short enrolment degrades performance (Figure 7b). Contaminating the enrolment with early-fatigue windows did not reduce performance here (Figure 7c); this should not be read as robustness, because the same

contamination was applied to every training and test driver and the classifier could therefore learn the shifted reference. In practice contamination would differ between drivers and sessions, and the enrolment quality gate in Algorithm 1 remains necessary. Raising sensor noise up to twice the nominal level did not produce a monotone loss for the personalised model, whereas the fixed rule deteriorated steadily; differences of one to three points are within seed variability.

Table 9. Sensitivity of C9 (seed-2027 dataset).

Factor	Setting	Macro-F1 (%)
Windows in sequence	T = 1 / 2 / 3 / 4 / 6	68.8 / 73.0 / 75.6 / 76.8 / 78.8
Enrolment windows	3 (9 s) / 5 / 10 / 15 (45 s) / 30	72.5 / 76.4 / 76.5 / 76.8 / 78.7
Early-fatigue share of enrolment	0% / 20% / 40% / 60%	76.8 / 77.8 / 78.9 / 79.7
Sensor-noise ×1.0 (rule / raw / C9)	—	59.1 / 61.6 / 76.8
Sensor-noise ×1.5 (rule / raw / C9)	—	56.2 / 64.3 / 75.9
Sensor-noise ×2.0 (rule / raw / C9)	—	51.3 / 65.6 / 76.8

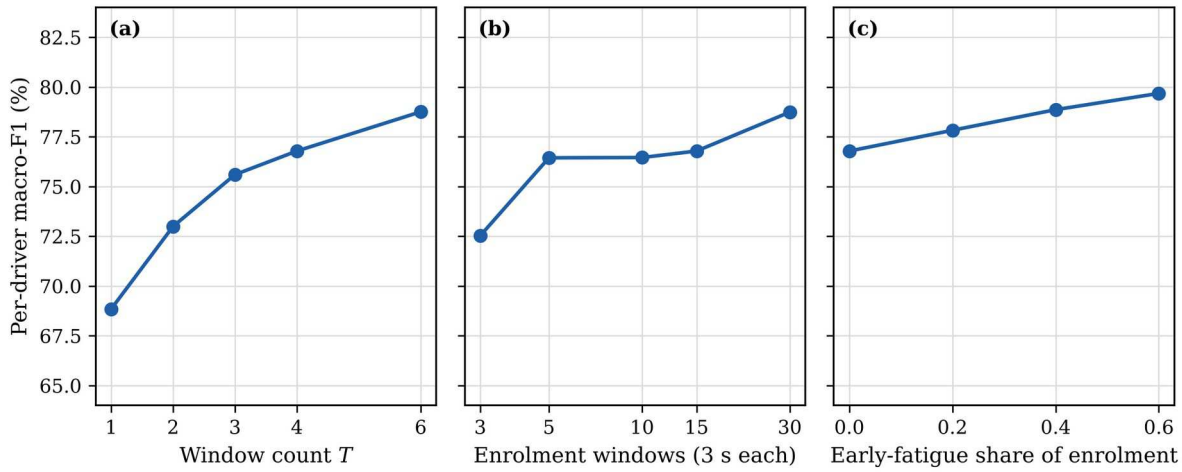


Figure 7. Sensitivity of C9 (seed-2027 dataset): (a) number of windows per sequence, (b) number of enrolment windows (3 s each; logarithmic axis), (c) share of early-fatigue windows in the enrolment.

4.8 Smoothing-gain sweep

Lowering the smoothing gain buys stability at the price of delay for both policies (Table 10, Figure 8). The two families lie on nearly the same trade-off curve: confidence gating reaches a given number of level changes with a similar delay, so its practical benefit in this simulation is modest and the choice of gain matters more than the choice of policy. An operating point must therefore be selected for the application, for example by fixing a maximum tolerated advisory delay and validating on real data.

Table 10. Alert-policy trade-off over the smoothing gain (mean over 48 drives).

Policy	κ_0	Any alert, alert phase (%)	Level changes	L1 delay (s)	L2 delay (s)
EMA	0.2	7.9	2.6	17.4	10.1
EMA	0.3	11.0	3.1	13.6	7.3
EMA	0.4	11.5	3.8	12.3	6.3
EMA	0.5	12.6	4.7	9.1	5.7
EMA	0.7	13.9	5.9	8.1	4.8
EMA	1.0	14.5	7.8	7.2	3.8
Confidence-aware	0.2	6.5	2.2	21.2	13.4
Confidence-aware	0.3	8.3	2.5	16.4	10.1
Confidence-aware	0.4	9.9	2.9	14.4	7.9
Confidence-aware	0.5	10.6	3.3	12.8	7.2
Confidence-aware	0.7	11.6	4.2	9.3	6.4
Confidence-aware	1.0	13.2	5.2	8.6	6.3

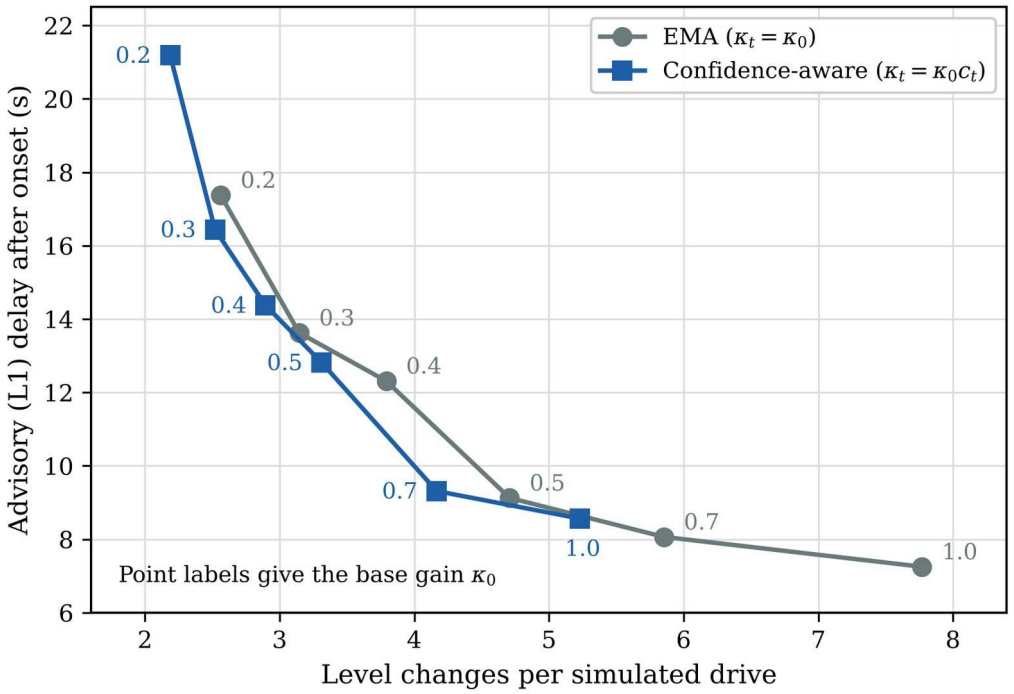


Figure 8. Stability–delay trade-off: level changes per drive versus advisory (L1) delay, labelled by the base gain κ_0.

5. Discussion

5.1 Outcome of the hypotheses within the simulation

H1 (temporal context) is supported: +8.5 points, and monotone in T up to 6 windows. **H2** (personalisation) is supported: +6.7 points, with the largest benefit for the few drivers whose resting EAR lies near the fixed threshold. **H3** (multimodality) is supported only through the head stream; the

mouth stream added nothing measurable. **H4** (stable alerts without unacceptable delay) is supported for smoothing, with a delay that grows as the gain falls; the extra effect of confidence gating is small. Two negative findings should be stated plainly: the mouth stream did not help beyond eye features, and sigmoid recalibration worsened calibration. Both may depend on the generator (for example, the frequency of speech-like openings and the class imbalance) and must be re-examined on real recordings before any conclusion is drawn.

5.2 Limitations and threats to validity

The following limitations bound every conclusion of this paper.

- **Synthetic data and circularity.** Training and test data come from the same generator family, and labels are defined by the generator. Absolute accuracies are therefore optimistic, the relative importance of streams reflects generator design, and the generator and features share assumptions (for example, closure episodes shift EAR below a fixed threshold) that favour the feature set.
- **Statistical meaning.** Intervals and *p*-values describe variation across simulated drivers (Section 3.6); 48 drivers per dataset and many uncorrected paired comparisons mean borderline values are indicative only. Only three drivers fell below the low-resting-EAR cut-off used in Figure 4.
- **Not modelled.** Landmark extraction, lighting, eyewear, occlusion, camera placement, ethnicity-related facial differences, label noise and real driving conditions are not represented.
- **Assumptions.** Enrolment is assumed to be recorded in a genuinely alert state. State changes in simulated drives are abrupt, unlike gradual fatigue onset. The temporal encoder is a fixed mean/last/slope summary with a linear classifier rather than a learned recurrent network.
- **Untested.** Latency was measured on a general-purpose CPU, not an edge device, and quantisation was not evaluated. Alert effectiveness for human drivers was not tested. Risk thresholds and the hysteresis margin were fixed a priori and not swept; only the smoothing gain was varied.
- **Construct validity.** Synthetic labels are a simplification of fatigue, which is not directly observable.

5.3 Responsible use and deployment considerations

DriveGuard is a research prototype, not a certified safety system, and the simulation provides no evidence that it would be safe or effective in a vehicle. Any real deployment would require, at minimum: an enrolment quality gate (the enrolment must be in an alert state, with the face visible and the head roughly still); a documented fallback to population thresholds when no valid profile exists; an explicit, driver-visible low-confidence state for sunglasses, occlusion, extreme head pose or poor lighting that suppresses escalation instead of defaulting silently; conservative handling of profile drift, since adaptation risks absorbing genuine fatigue; on-device processing without retention of raw video; and testing for both missed detections and nuisance alerts. The system must not encourage a tired driver to continue driving. Level 1 should be a low-intrusion advisory and level 2 a salient warning, but the simulation cannot show how drivers would respond.

5.4 Plan for validation on real data

The simulation is intended as a specification and test bed for later real-data work. Planned stages are: (1) UTA-RLDD [12] with subject-independent folds, after checking access and terms and treating coarse labels as sequence-level rather than exact onset annotations; (2) YawDD [10] for the mouth stream only, since yawn labels are not fatigue labels; (3) NTHU-DDD [11], if access conditions are met, reporting only what its labels support; (4) latency, memory and power measurement on target edge hardware with stated hardware and software versions; and (5) an approved pilot study with informed consent, a self-reported sleepiness scale, ethics approval and privacy-preserving logs. Real-data results should be

reported next to, and never merged with, the simulated results of this paper, and thresholds and hyper-parameters should be chosen on validation subjects only.

6. Conclusions

DriveGuard combines eye, mouth and head-pitch indicators over short temporal windows, normalises them against a 45-s driver enrolment, and maps confidence to graduated alerts. In a seeded, driver-disjoint simulation it raised per-driver macro-F1 from 58.3% (fixed rule) to 77.5% and made alerts markedly more stable, while exposing a delay trade-off and two components (the mouth stream and sigmoid calibration) that did not help. These are statements about behaviour in a controlled synthetic test bed, not about real drivers. Future work will repeat the protocol on suitable public datasets using MediaPipe landmarks after checking access terms and label granularity, learn the temporal encoder and alert thresholds on validation subjects, measure latency and memory on target hardware with quantisation, and study driver response to graduated alerts in an approved simulator study.

Declarations

Ethics approval and consent to participate. Not applicable. This is a computational study that used only synthetic data generated by software. No human participants, patients, volunteers, animals, personal data, or images or video of identifiable individuals were involved, collected or processed.

Consent for publication. Not applicable. The manuscript contains no individual person's data, images or video.

Availability of data and materials. All data analysed in this study are computer-generated and no external dataset was used. The generator and experiment script (experiment.py), the sensitivity-analysis script (sensitivity.py) and the figure and equation scripts (make_figures.py, make_equations.py), together with the output files they produce (results.json, results2.json, arrays.npz), are provided as supplementary files. Running the scripts with the stated seeds regenerates every table and figure of this paper.

Code availability. See "Availability of data and materials". Software versions are listed in Section 3.9.

Competing interests. The author declares no competing interests.

Funding. The author received no specific funding for this work.

Authors' contributions. R.S. conceived the study, specified the framework, hypotheses and evaluation protocol, reviewed the code, results and text, and approved the final manuscript. The role of an AI assistant is described in Section 3.10.

Acknowledgements. The author thanks Sanjivani University and the Department of Artificial Intelligence and Data Science.

Use of AI tools. A large language model (Anthropic's Claude) assisted with drafting, code and formatting; see Section 3.10 for the full disclosure.

References

[1] T. Soukupová and J. Čech, "Real-time eye blink detection using facial landmarks," in *Proc. 21st Comput. Vis. Winter Workshop*, Rimske Toplice, Slovenia, 2016.

[2] D. F. Dinges and R. Grace, "PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance," FHWA Tech. Brief MCRT-98-006, 1998.

[3] C. Lugaresi *et al.*, "MediaPipe: A framework for building perception pipelines," arXiv:1906.08172, 2019.

[4] Y. Kartynnik *et al.*, "Real-time facial surface geometry from monocular video on mobile GPUs," arXiv:1907.06724, 2019.

[5] R. Florez *et al.*, "A real-time embedded system for driver drowsiness detection based on visual analysis of the eyes and mouth using convolutional neural network and mouth aspect ratio," *Sensors*, vol. 24, no. 19, art. 6261, 2024, doi: 10.3390/s24196261.

[6] N. N. Alajlan and D. M. Ibrahim, "DDD TinyML: A TinyML-based driver drowsiness detection model using deep learning," *Sensors*, vol. 23, no. 12, art. 5696, 2023, doi: 10.3390/s23125696.

[7] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF ICCV*, 2019, pp. 1314–1324.

[8] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE CVPR*, 2018, pp. 2704–2713.

[9] S. Ebrahimian *et al.*, "Multi-level classification of driver drowsiness by simultaneous analysis of ECG and respiration signals using deep neural networks," *Int. J. Environ. Res. Public Health*, vol. 19, no. 17, art. 10736, 2022, doi: 10.3390/ijerph191710736.

[10] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A yawning detection dataset," in *Proc. ACM MMSys*, 2014, pp. 24–28.

[11] C.-H. Weng, Y.-H. Lai, and S.-H. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network," in *ACCV Workshops*, 2016, pp. 117–133.

[12] R. Ghoddoosian, M. Galib, and V. Athitsos, "A realistic dataset and baseline temporal model for early drowsiness detection," in *Proc. IEEE CVPR Workshops*, 2019.

[13] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1321–1330.

[14] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, MIT Press, 1999, pp. 61–74.

[15] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Rev.*, vol. 78, no. 1, pp. 1–3, 1950.

[16] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[17] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, 1979.

[18] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[19] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.

[20] P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.

[21] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

Appendix A. Window features

Table A1. Feature definitions used in the simulation (30-frame window).

#	Feature	Stream	Definition
1	ear_mean	Eye	Mean EAR
2	perclos	Eye	Fraction of frames with EAR < 0.22
3	blink_cnt	Eye	Number of closure episodes
4	max_run	Eye	Longest closure run / 30
5	mar_max	Mouth	Maximum MAR
6	mar_mean	Mouth	Mean MAR
7	yawn_frac	Mouth	Fraction of frames with MAR > 0.55
8	pitch_mean	Head	Mean pitch (°)
9	pitch_std	Head	Pitch standard deviation (°)
10	nod_frac	Head	Fraction of frames with pitch < −8°

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryFigureS1FeatureDistributions.png
- SupplementaryFigureS2FeatureCorrelation.png
- SupplementaryFigureS3RestingEARDistribution.png
- SupplementaryFigureS4CrossSeedReplication.png
- SupplementaryFigureS5PerClassF1.png
- SupplementaryFile1TablesandMethods.docx
- SupplementaryFile2SourceCode.docx