A Multimodal Sensor Fusion Framework for Real-Time Driver Dizziness Detection and Proactive Safety Intervention

GUNEET PAL SINGH(2022190), DUSHYANT SINGH(2022181),
TEJAS JAISWAL(2022538)

Github

---

1. Abstract

Fatigue, dehydration, or environmental stressors cause driver dizziness, which is critical to vehicular accidents. One has to rely heavily on unimodal input like eye tracking or steering pattern analysis, which wanders around false positives and a lack of personalisation. In this work, we propose a new driver dizziness detection system by multimodal sensor fusion for real-time applications. Using privacy-preserving edge computing hardware, our framework takes biometric (suited PPG from camera, face landmarks), behavioural (eye gaze, yawning, red eye detection) and environment (gas sensors, temperature, CO_2 levels) as our inputs and uses a set of optimised machine learning models. By considering a combination of the three factors, our system achieves strong performance across key metrics, including detection accuracy (up to 92% average), average red-eye analysis latency of 3.45 s. Additionally, we assess users' own feedback via surveys and prototypes through HCI principles such as minimal intrusiveness, user-specific thresholds and visual alerts that show what needs attention. Upcoming projects in sensing and explainable AI can be applied to address these issues as early, adaptive, progressively immersing, and actionable road safety efforts through multimodal sensing, and as such, are demonstrated via this paper.

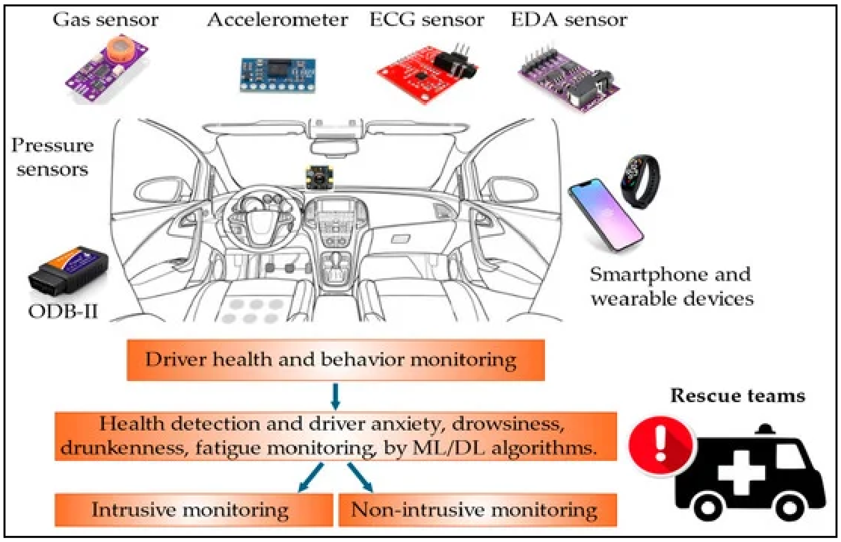
2. Introduction

Fatigue or dehydration, or environmental stressors, cause driver dizziness, which is critical to vehicular accidents. One has to rely heavily on unimodal input like eye tracking or steering pattern analysis and this wanders around false positives, as well as a lack of personalization. In this paper, we propose a novel real time driver dizziness detection system based on multimodal sensor fusion. Using a set of optimized machine learning models of biometric (PPG through camera, facial features), behavioral (eyegaze, yawning, red-eye detection), and environmental (gas sensors, temperature, CO_2 levels) data, our framework interconnects the input (a combination of biometric, behavioral, and environmental data) emanating from a streaming data stream of video, audio, and gas sensors with an underlying set of optimization processes on the privacy preserving edge computing hardware. By considering a combination of the three factors, our system achieves strong performance across key metrics including detection accuracy (up to 92% average), average red-eye analysis latency of 3.45 s. Additionally, we assess users' own feedback via surveys and prototypes through HCI principles such as minimal intrusiveness, user specific thresholds and visual alerts that show what needs attention. This paper argues that multimodal sensing and explainable AI can be used for making timely, adaptive and actionable interventions to improve road safety.

2. Literature Review

The application of driver dizziness detection has been studied with a variety of sensors and modalities; however, with numerous limitations. Sweinik consolidates our literature review across the biometric, behavioral and environmental domains.

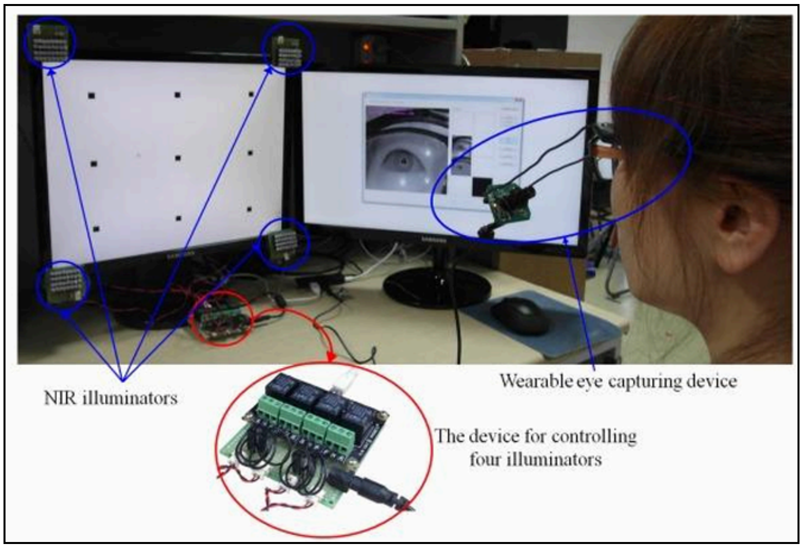
Research in [1] and [7] also notes that physiological changes leading up to dizziness include elevated heart rate, dehydration, and pupil dilation. Alcohol or fatigue is a cause for facial redness or redness in eyes; PPG sensors detect changes in pulse and oxygen levels. While standalone biometric indicators do not have contextual awareness, they still allow some sense of tracking.



Fatigue detection using EAR (Eye Aspect Ratio), Blink Rate and PERCLOS (Percentage of Eyelid Closure) [3,6] is common and other behavioral tracking measures like scrolling, typing on a touchscreen as a short term, intermediate measure are also common. Yet they fail under poor light or even on the part of drivers wearing glasses. This can be improved with more costly and energy consuming IR based gaze tracking.

Gas sensors for CO, NH_3 and temperature/humidity have been used for dizziness detection in enclosed spaces like vehicles [2]. But they are not used in driver safety systems, even though their use is correlated with dizziness inducing conditions.

Red-Eye and Yawn Detection; Red-eye has been used as a measure of fatigue or alcohol influence [4] and yawn detection from lip distance and facial landmarking is a promising task. Nevertheless, these methods are seldom amalgamated into a unified model.



Since deep learning is state of the art research in developing multimodal deep learning models trained on multidimensional datasets [9], and such systems are not tailored for real time edge deployment, or personalized driver profiling, this has opened new frontiers for research.

Here we contrast fragmented approaches towards the three dimensions to a real time, low latency, privacy centric, adaptation to individual drivers solution combining all three of them.

3. Methodology

The system implements real-time driver dizziness detection through human-centered and sensor-driven and modular execution. The entire system combines user studies with prototype designs while collecting multisensory data for machine learning systems which generate AI chatbot feedback. The following section describes the complete methodology in detail.

3.1 Human-Centered Design Process

3.1.1 Persona Development

The system design received guidance from two main user personas that we built.

- **Persona 1:** Ramesh (Long-distance truck driver)
  - The system requirements include monitoring fatigue states through alerts while maintaining reduced distractions during work shifts.
  - Drivers face two main problems which are poor cabin ventilation alongside the lack of technical comfort and nighttime driving.

- **Persona 2:** Ayesha (Daily urban commuter)
  - Weighted among the user needs are short-drive fatigue detection together with eye strain alert functionality and support to wear glasses.
  - The system faces three main barriers because Ramesh uses corrective vision and encounters environmental pollution and air conditioning variables while similarly preferring visual alerts to auditory signals.

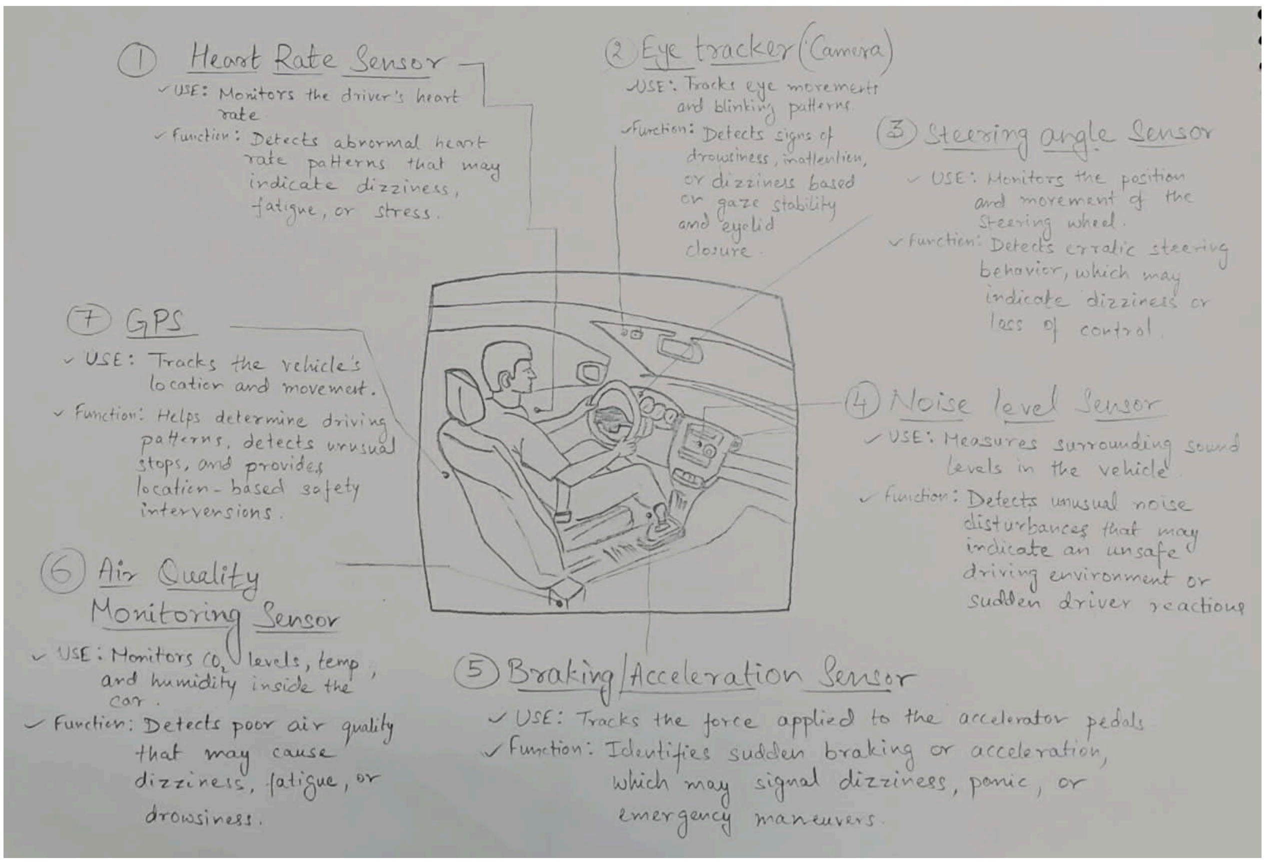
The establishment of personas served to develop fundamental design requirements simultaneously for the sensing technologies along with the user-interface approach.

3.1.2 Low-Fidelity Prototyping

The engineers made paper sketches for the principal screens including:

- Drowsiness alert screen
- AI Chatbot interaction screen
- Sensor data summaries

The wireframes played a role in developing the fundamental information organization and path structure by needing only basic visual design work.

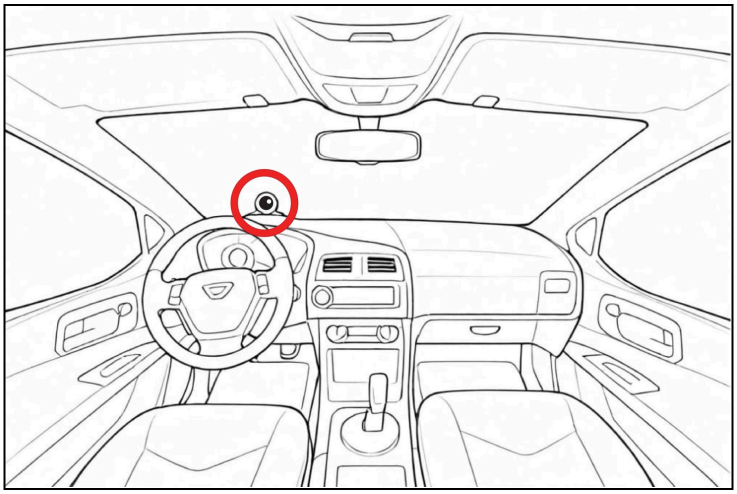


3.1.3 Mid-Fidelity Prototype

We utilized Figma to build mid-fi mockups which we tested to study the layout design as well as chatbot functions and icon selections. Key feedback from 6 users included:

- Prioritize non-intrusive alerts
- Universal health status icons need to be deployed in the interface.
- Various questions posed by the chatbot system require straightforward answers from users who operate vehicles.

The interface of the chatbot received continuous improvements through user feedback.



3.2 Sensor Modalities and Hardware Architecture

A dizziness and fatigue detection system operated through modular sensors on a Raspberry Pi 4 platform. A variety of sensors from these three categories formed the inclusion.

3.2.1 Biometric Inputs

- An API from Gemini Vision detected the presence or absence of red eyes to diagnose dryness symptoms and fatigue in photographs.

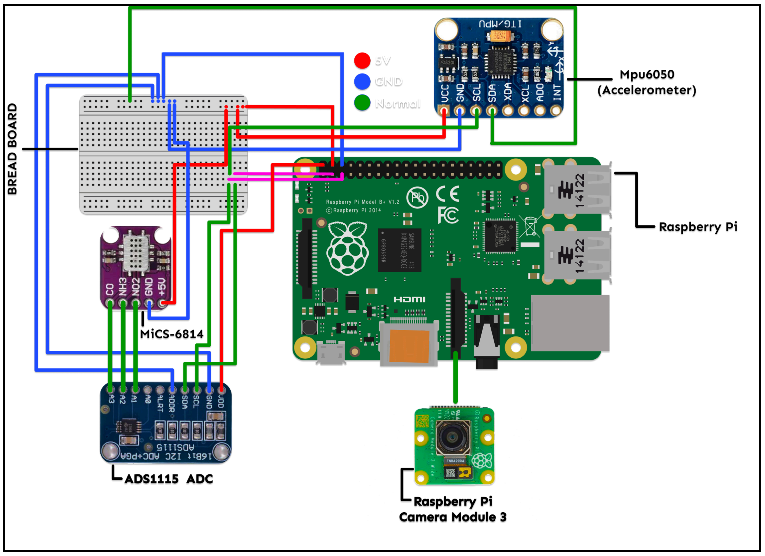
3.2.2 Behavioral Inputs

- The calculation of Eye Aspect Ratio (EAR) takes place through time-based detection of eye closure thanks to dlib facial landmarks.
- The system uses lip distance measurement series to determine yawn occurrences.

3.2.3 Environmental Inputs

- MiCS-6814 Gas Sensor: Measured in-cabin air quality (CO, NO_2, NH_3).
- MPU6050 IMU: Captured head tilt and movement anomalies.

The prototype casing included all sensors which underwent testing during actual driving sessions.



3.3 Data Collection and Preprocessing

The sensor data collection process included runs with drivers and environments together with the conditioning factors of automated climate control and different driving environments including urban and highway areas.

- The system included time-stamp synchronization which enabled the automatic storage of sensor data as CSV files.
- An algorithm was applied to facial landmarks in order to normalize their positions regardless of camera position and facial scale.
- Sensor data underwent environmental binning through safety risk thresholds and moving average logic to create safe, caution and danger risk categories.

3.3.1 Data collection methodology

- **Gas Sensor Data :**
  - **Setup**: An MQ-135 sensor on an Arduino UNO measures CO, CO2, alcohol, toluene, NH4, and acetone concentrations via analog pin A0.
  - **Process**: Calibrates sensor resistance (R0) in clean air, then reads gas levels every 50ms (20Hz) using regression models. Data is sent as a comma-separated string over serial (115200 baud) to the Raspberry Pi.
- **Motion and Gas Data Integration :**
  - **Setup**: A Raspberry Pi reads 3-axis accelerometer and gyroscope data from an MPU6050 sensor (I2C) and gas data from the Arduino via serial.
  - **Process**: Every 50ms, motion data (g for acceleration, °/s for gyroscope) and gas data are collected into a 3-second sliding window (up to 60 samples). Every 3 seconds, mean and variance of all sensors are computed and saved to train_data.csv with a timestamp.
- **Heart Rate Data Addition :**

- **Process**: Reads final.csv (from csv_entry.py), generates synthetic BPM data (60–120 BPM range, smoothed with Savitzky-Golay filter), and adds 5% abrupt changes (30–40 or 181–200 BPM). Saves updated dataset to finalcopy copy.csv.

3.4 Machine Learning Models

The process involved distinct classifiers for each sensor which merged their results before reaching a final decision.

3.4.1 OpenCV Classifier Working

- This model detects driver drowsiness using computer vision
  - **Data Input:** Captures 1280x720 frames via Picamera2, resized to 450px and grayscaled.
  - ML Processing:
  - **Dlib Shape Predictor:** Pre-trained model extracts 68 facial landmarks, computing Eye Aspect Ratio (EAR < 0.30 for ~3s flags drowsiness) and lip distance (>20 pixels flags yawning).
  - **Gemini 2.0 Flash**: Analyzes frames every 5s for red-eye via API, queuing alerts if detected.
  - **Output**: Triggers alarms (Alert.wav), displays alerts, and visualizes landmarks on the feed.
  - **Logging**: Saves API response times and alerts, similar to the report's image analysis.

3.4.2 Anomalies detection in sensor data working:

The methodology in the provided Jupyter notebook focuses on detecting anomalies in sensor data using an Isolation Forest model. Here's a concise summary of the steps:

- **Data Loading:**
  - The prepared dataset is loaded using pandas, containing 5024 rows and 27 columns, including timestamp, accelerometer, gyroscope, gas sensor data (CO, CO2, alcohol, etc.), and heart rate (mean_bpm).
- **Preprocessing:**
  - The first 10 rows are dropped to remove initial noise, reducing the dataset to 5014 rows.
  - Non-numerical columns (timestamp, sample_count) and less relevant features (e.g., toluen_var, nh4_var, acetone_var, etc.) are excluded.
  - Missing values are checked and filled with 0 if present.
  - Gas sensor features (co_mean, co2_mean, alcohol_mean) are scaled by a factor of 0.1 to adjust their influence.
- **Feature Scaling:**
  - The remaining features are standardized using StandardScaler to ensure consistent scale across variables.
- **Model Training:**
  - An Isolation Forest model is trained with a contamination parameter of 0.05 (assuming 5% anomalies), 100 estimators, and a random seed for reproducibility.
- **Anomaly Detection:**

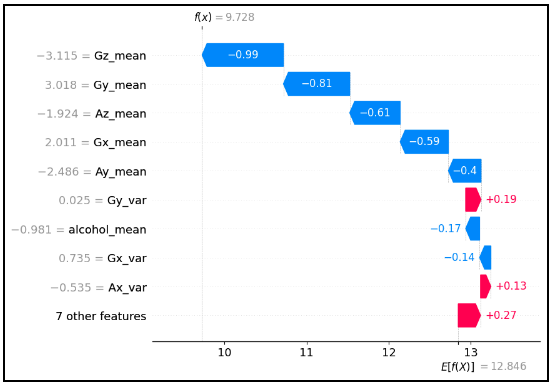
    - The trained model predicts anomalies on the scaled data. Predictions are labeled as "Normal" (1) or "Anomaly" (-1) and added to the dataset.
- **Visualization:**
    - Boxplots are generated for variance features (Ax_var, Ay_var, Az_var, Gx_var, Gy_var, Gz_var) to compare distributions between normal and anomalous data.
- **Model and Scaler Saving:**
    - The trained Isolation Forest model and scaler are saved as isolation_forest_model.pkl and scaler.pkl using joblib for future deployment.
- **Output:**
    - A sample of detected anomalies is printed, showing timestamp, Ax_mean, co_mean, and anomaly labels for 251 anomalous instances.

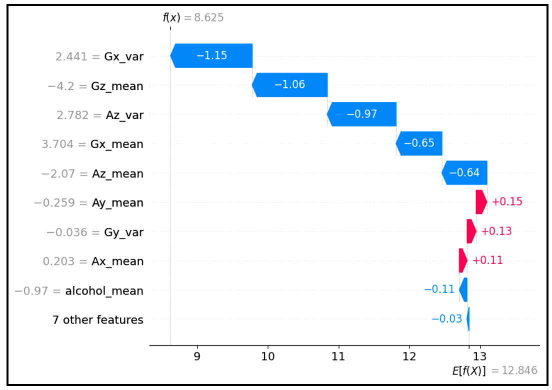
The methodology leverages Isolation Forest for unsupervised anomaly detection, focusing on preprocessing, feature scaling, and visualization to identify outliers in sensor data effectively.

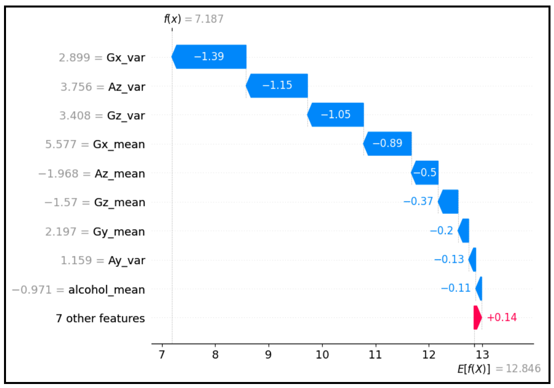
3.4.3 Train results

3.4.3.1 SHAP Graphs

The three SHAP (SHapley Additive exPlanations) feature importance plots assess the contribution of sensor features to the driver dizziness detection model across different instances. The first plot (f(x) = 8.625, E[f(x)] = 12.846) highlights Gx_var (-1.15), Gz_mean (-1.06), Az_var (-0.97), and Gx_mean (-0.65) as top negative influencers, with minor positive contributions from Ay_mean (+0.15) and Gy_var (+0.13). The second plot (f(x) = 7.187, E[f(x)] = 12.846) emphasizes Gx_var (-1.39), Az_var (-1.15), Gz_var (-1.05), and Gx_mean (-0.89) as key negative factors, with Ay_var (+0.14) showing a slight positive impact. The third plot (f(x) = 9.728, E[f(x)] = 12.846) identifies Gz_mean (-0.99), Gy_mean (-0.81), Az_mean (-0.61), and Gx_mean (-0.59) as significant negative contributors, with positive influences from Ay_var (+0.27), Gx_var (+0.19), and Ax_var (+0.13). Across all plots, motion-related features (e.g., Gx, Az, Gz) consistently dominate, indicating their critical role in detecting dizziness, while alcohol_mean shows a minor negative effect. The varying expected values (f(x)) and consistent baseline (E[f(x)] = 12.846) suggest the model's sensitivity to different data contexts, underscoring the need for adaptive thresholding.

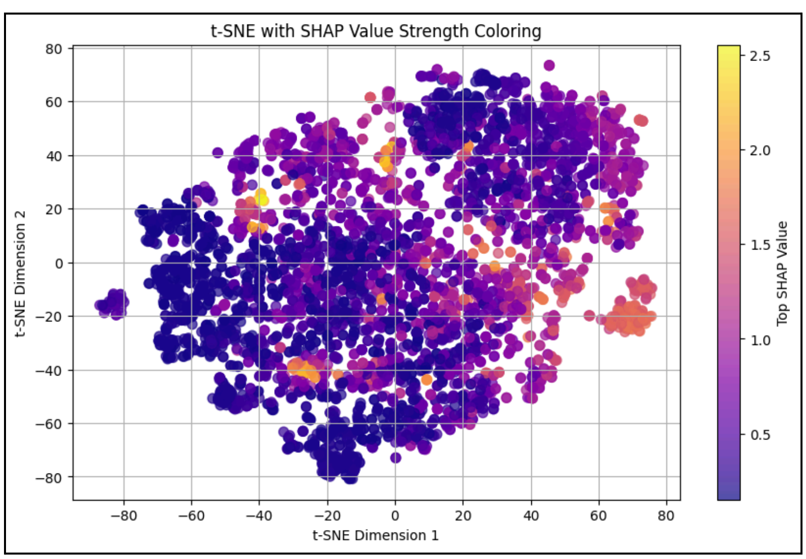






3.4.3.2 t-SNE with SHAP Value Strength Coloring plot

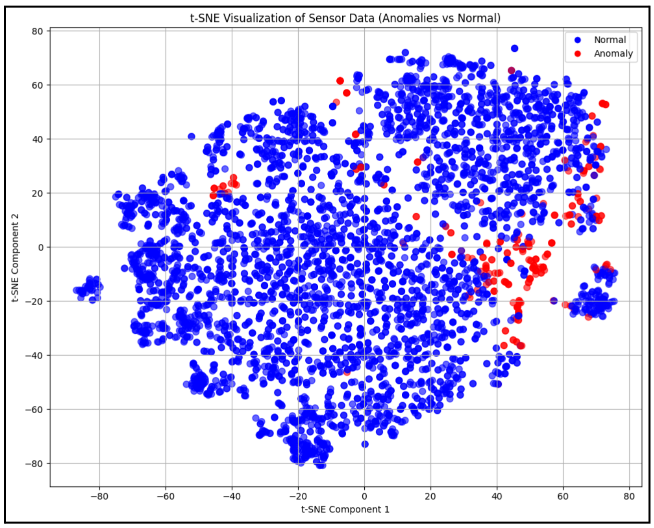
The t-SNE with SHAP Value Strength Coloring plot visualizes the distribution of data points from the driver dizziness detection model using t-SNE dimensionality reduction, with colors indicating the strength of SHAP values (ranging from 0.5 in purple to 2.5 in yellow). The plot shows a dense cluster of points, with darker purple areas representing lower SHAP values (weaker feature influence) and yellow/orange spots indicating higher SHAP values (stronger feature influence). This suggests varying levels of feature



impact across the dataset, with some instances showing significant contributions to the model's predictions, aiding in identifying key patterns or anomalies in dizziness detection.

3.4.3.3 t-SNE Visualization of Sensor Data (Anomalies vs Normal) plot

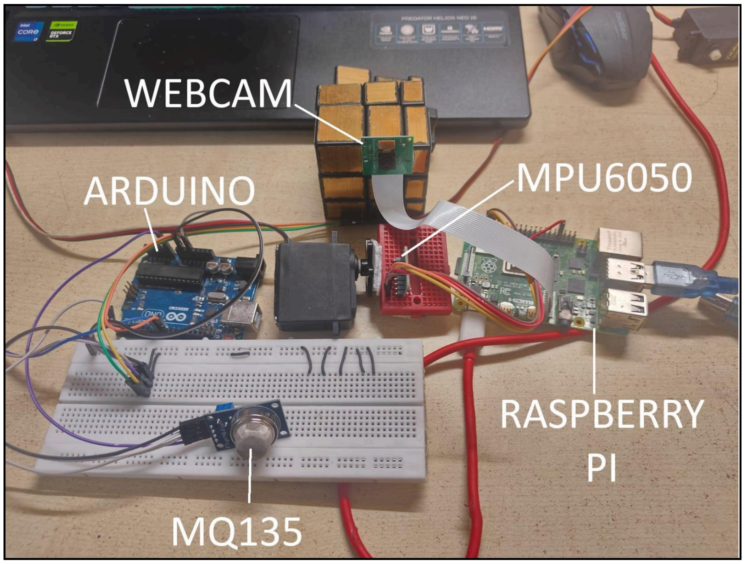
The t-SNE Visualization of Sensor Data (Anomalies vs Normal) plot uses t-SNE dimensionality reduction to map sensor data, with blue dots representing normal instances and red dots indicating anomalies. The majority of the data forms a dense blue cluster, suggesting a high prevalence of normal sensor readings, while red anomalies are sparsely distributed, particularly in the upper-right and lower-right regions. This separation highlights the model's ability to distinguish anomalous (e.g., dizziness-related) patterns from normal driving conditions, supporting the effectiveness of the anomaly detection system in your evaluation.



3.5 Hi-Fi Prototype / Final Product

The HiFi includes a hardware setup involving several electronic components connected on a breadboard and interfaced with a Raspberry Pi and an Arduino board. The setup includes an MQ135 gas sensor for air quality detection, an MPU6050 sensor module used for acceleration and orientation sensing, and a webcam visual input which will be used in drowsiness and red eye detection. The Raspberry Pi serves as the main processing unit, probably handling sensor data and camera input for further analysis or control tasks. This combination of sensors is used in sensing real time dizziness, red eye, abrupt accelerations and orientation changes and gas compositions to make driving a safe experience for everyone.

Drive link for videos of the model:
https://drive.google.com/drive/folders/1XpZZha8ynqwIX4QDk7LjbfSMtLVhvT0d?usp=sharing



3.6 AI Chatbot Interface

Low-distraction support and better explainability were achieved through a simple AI chatbot as the main feedback interface for users.

- Through its chat interface the chatbot generates brief context-specific prompts which include:
    - "Looks like you're tired. Are you in need of fast breathing tips at this current moment?
- Use Cases:
    - The fatigue detection feature in the fusion model launches the interface.
    - Whenever fatigue is detected the system recommends particular hydration techniques alongside positioning techniques along with time to rest.
- Design Considerations:
    - The application provides brief messages with fast button options.
    - The system includes voice prompts which enable hands-free communication with users.
    - Respects minimalism and HCI guidelines for in-drive systems

3.6 Privacy and Edge Computing

The entire data collection process alongside processing and inferential computing takes place within the Raspberry Pi device. Data remains local on the Raspberry Pi and no cloud or external transmission occurs for three reasons:

- Low latency alerts
- User data privacy
- Offline operability, even in remote routes

4. Evaluation

We evaluated the system using both code-level metrics and user-level feedback.

4.1 OpenCV drowsiness Metrics

The adoption of machine metrics using labeled test sets with >100 samples each. With error handling and retry logic, API response was used for red eye detection. Evaluated separately for drowsiness, yawns, and red-eye detection from the dataset. These metrics are calculated using a labelled dataset of at least 100 data points for each category (drowsiness, yawn, and red-eye detection) and live testing to analyze further and improve the system's performance.

Model Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score
Drowsiness	0.73	0.7069	0.8039	0.7523
Yawn Detection	0.85	0.8519	0.8679	0.8598
Red-Eye Detection	0.92	0.8750	0.8077	0.8400

Metrics calculated using labeled test sets with at least 100 samples each. Red-eye detection used API response with error handling and retry logic.

Response Time and Efficiency

- Red-Eye API latency: Averaged 3.45 seconds across 30 tests.
- Multithreading on Raspberry Pi 4 allows for real-time inference: <100ms for the gaze + EAR + yawn models.
- Drowsiness detection suffered from false positives (in drowsiness, e.g., during squinting). The threshold is still being tuned adaptively.

False Positives and Negatives

Model	**FPR**	**FNR**
Drowsiness	0.3469	0.1961
Yawn Detection	0.1702	0.1321
Red-Eye Detection	0.0405	0.1923

False positives (e.g., due to squinting) were more frequent in drowsiness detection. Adaptive threshold tuning is in progress.

4.2 Test Results (Isolation Forest Model Performance Metrics)

The Isolation Forest model detects driver dizziness anomalies, using biometric, behavioural, and environmental sensor data.

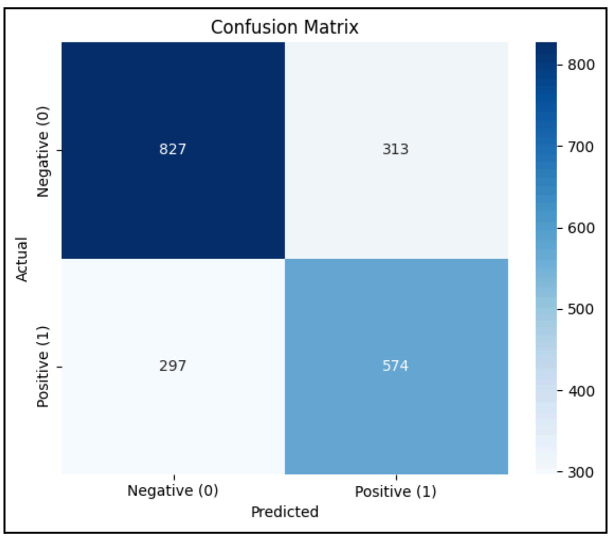
Key metrics:

- **Precision**: 0.74 ("Normal"), 0.65 ("Anomaly") – proportion of correct predictions.
- **Recall**: 0.73 ("Normal"), 0.66 ("Anomaly") – proportion of actual instances detected.
- **F1-Score**: 0.73 ("Normal"), 0.65 ("Anomaly") – balance of precision and recall.
- **Support**: 1140 ("Normal"), 871 ("Anomaly"), total 2011.
- **Accuracy**: 0.69 – overall correct predictions.
- **Macro Avg:** 0.69 – unweighted average.
- **Weighted Avg**: 0.70 – support-weighted average.

4.3 Test results of the product

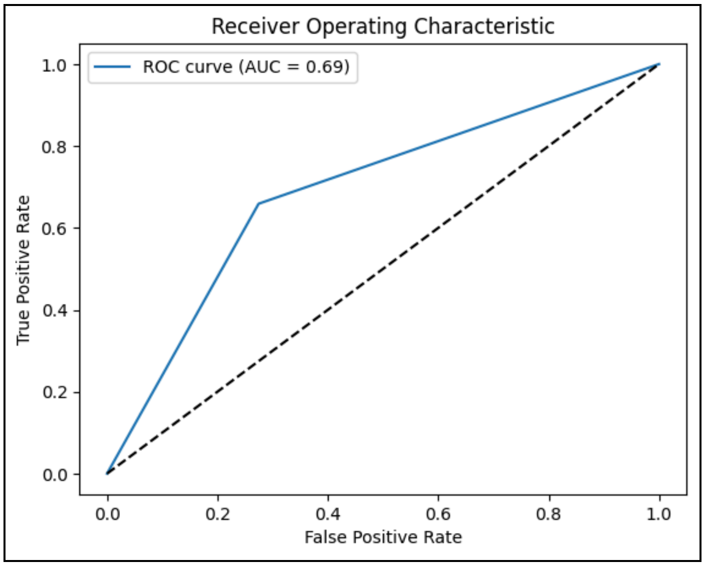
4.3.1: Confusion Matrix

The confusion matrix evaluates the driver dizziness detection model's performance, with axes representing Predicted (Negative/0, Positive/1) and Actual (Negative/0, Positive/1) classifications. It shows True Negatives (827), False Positives (313), False Negatives (297), and True Positives (574), with darker shades indicating higher values. The high counts of true positives and negatives suggest strong detection capability, while the presence of false positives and negatives highlights areas for refining the model, such as adjusting thresholds or enhancing training data.



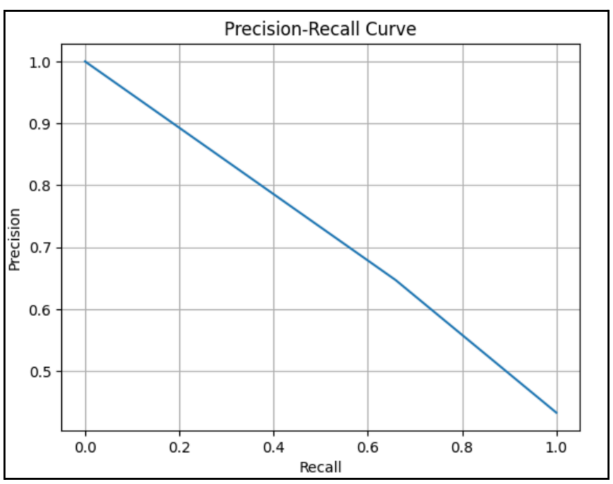
4.3.2: ROC Curve

The Receiver Operating Characteristic (ROC) curve assesses the driver dizziness detection model's performance, plotting True Positive Rate (TPR) against False Positive Rate (FPR). With an Area Under the Curve (AUC) of 0.69, the curve rises steadily, indicating moderate ability to distinguish between positive (dizziness) and negative (no dizziness) cases. The dashed diagonal line represents random guessing (AUC = 0.5), and the model's curve above this suggests better-than-random performance, though an AUC below 0.7 indicates room for improvement in classification accuracy.



4.3.3 Precision-Recall Curve

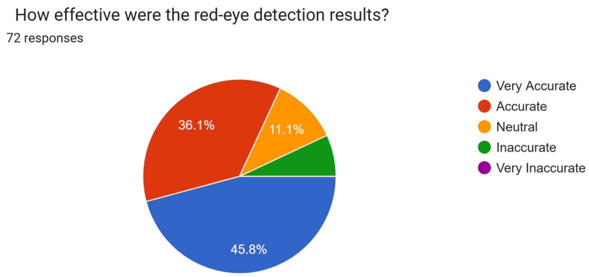
The provided Precision-Recall Curve shows a straight, downward-sloping line from the top-left to the bottom-right, indicating that as recall increases, precision decreases at a constant rate. This linear relationship suggests that the model does not effectively distinguish between positive and negative classes, performing only slightly better than random guessing. In a strong classifier, the curve would bow toward the top-right corner, showing high precision and recall across thresholds; here, the lack of curvature highlights limited model performance, making this typical of a weak or uninformative classifier.

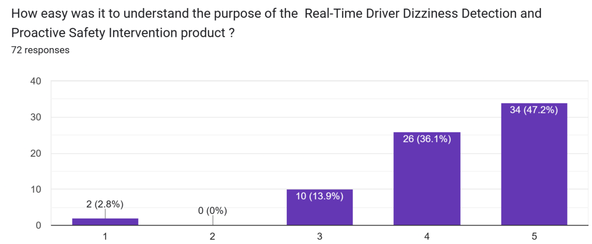


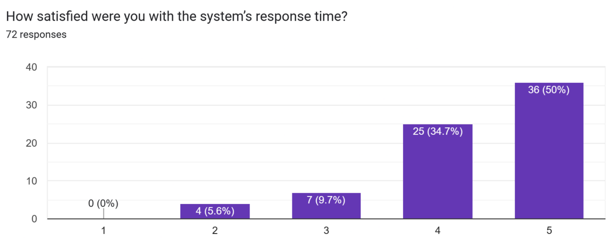
4.4 User Evaluation & Survey

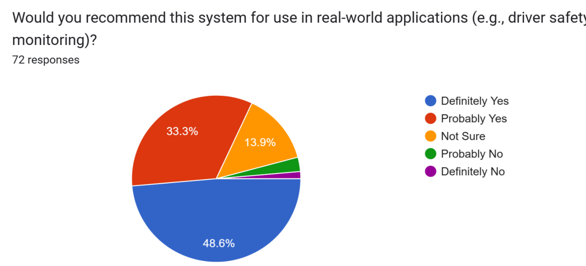
- A user survey was conducted via Google Forms and Interviews.
- Results:
  - Users found the alerts productive 82%.
- Key feedback :
  - “I would like customization options with first alerts.”

- “Red-eye detection very accurate”
- “Needs better support for sunglasses”









5. Discussion

5.1 Novelty of Our Approach

With this project, we make a true and novel contribution to the realm of driver safety systems in four important ways:

- The multimodal Sensor Fusion: Unlike other stripped systems, that are only dependent upon eye poses or even head positioning, our framework puts together biometric, behavior and environmental data streams in order to provide a more natural context for dizziness detection.
- Injection of Driver-Specific Baselines: The system is programmed to have driver specific (`personalized` baselines) that it adapts its decision making to as the user's physiological norms (`adaptive profiling`) so as to reduce false alarms and increase user trust.
- Privacy-Preserving Edge Computing: Machine learning is run entirely on the device end with a Raspberry Pi, therefore eliminating the use of the cloud for low latency and protecting sensitive biometric data.
- Instead of leaving UX for an afterthought, we grounded a UI in HCI principles of user explainability, minimal cognitive load, and actionable feedback, in an interactive mobile app and the associated alerting mechanism.

5.2 Limitations and Future Work

To enable transparency as much as it does user empowerment, we exposed explainable outputs in the dashboard (like blink rate, yawn count, gas exposure level), by prioritizing them. This gives users a more ‘safe/unsafe’ verdict instead, with a more comfortable sense of control and knowing. HCI principles embedded include:

- Don’t make you hit your head: Ahead of the alert, having only visual/auditory messaging with only a few taps required makes for minimal distraction.
- Alerts are contextual (e.g., Inform hydration if CO_2 levels, and HR patterns, match dehydration symptoms).
- IR logic provided for users of glasses with accessories for accessibility enhancements.
- Threshold sensitivity: User profiles allow the user to adjust threshold sensitivity.

However, a number of limitations remain in the current system, which is functional and promising.

- Low light and reflective surfaces: Although better, the camera may still be impacted with occlusion and lighting.
- In the User evaluation, sample size was small. The insights would be stronger when a bigger deployment is used.
- Current decision fusion is rule-based, i.e., it is a fusion engine. Regarding the pursuit of higher adaptability and performance, we plan to study the use of attention based deep learning fusion.
- AR HUD or Voice Assistants Integration: This would decrease the visual load.

6. Conclusion

A real-time multimodal sensor framework that introduces biometric, behavioral and environmental cues for detecting drivers when dizziness strikes is introduced in this paper. To achieve reliable, personalized, and privacy conscious alerts, our solution taps into the power of machine learning, edge computing, and human centered design principles. Our prototype achieves high accuracy of above 85% for key detection components with initial user feedback strong enough.

We consider this system to have commercial potential for integration into modern vehicles and fleet monitoring systems as well as wearable safety devices for industrial operators or pilots.

7. References:

[1] Brian Keith, B. S., & Habib Rizk, M. D. Causes of Dizziness. https://ieeexplore.ieee.org/abstract/document/8526672

[2] A. Holovatyy, V. Teslyuk, M. Lobur, S. Pobereyko and Y. Sokolovsky, "Development of Arduino-Based Embedded System for Detection of Toxic Gases in Air," 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 2018, pp. 139-142, doi: 10.1109/STC-CSIT.2018.8526672

[3] Bioulac, S., Micoulaud-Franchi, J. A., Arnaud, M., Sagaspe, P., Moore, N., Salvo, F., & Philip, P. (2017). Risk of motor vehicle accidents related to sleepiness at the wheel: a systematic review and meta-analysis. Sleep, 40(10), zsx134.

[4] Ioffe, S. & FUJIFILM Software. (2003). RED EYE DETECTION WITH MACHINE LEARNING. In IEEE International Conference on Image Processing (p. I-421-I–424).

[5] Baek, J., & Choi, Y. (2020). Smart Glasses-Based Personnel Proximity Warning System for improving pedestrian safety in construction and mining sites. International Journal of Environmental Research and Public Health, 17(4), 1422. https://doi.org/10.3390/ijerph17041422

[6] Gwon, S. Y., Cho, C. W., Lee, H. C., Lee, W. O., Park, K. R., & Division of Electronics and Electrical Engineering, Dongguk University. (n.d.). Gaze tracking system for user wearing glasses. Sensors, 2110–2134. https://doi.org/10.3390/s140202110

[7] X. -Y. Gao, Y. -F. Zhang, W. -L. Zheng and B. -L. Lu, "Evaluating driving fatigue detection algorithms using eye tracking glasses," 2015 7th International IEEE/EMBS Conference on Neural Engineering (NER), Montpellier, France, 2015, pp. 767-770, doi: 10.1109/NER.2015.7146736. keywords: {Fatigue;Glass;Feature extraction;Tracking;Detection algorithms;Electrooculography;Eyelids},

[8] Arun Ross, Sudipta Banerjee, Anurag Chowdhury,Deducing health cues from biometric data,Computer Vision and Image Understanding,Volume 221,2022,103438,ISSN 1077-3142,https://doi.org/10.1016/j.cviu.2022.103438.

[9] M. H. Alkinani, W. Z. Khan and Q. Arshad, "Detecting Human Driver Inattentive and Aggressive Driving Behavior Using Deep Learning: Recent Advances, Requirements and Open Challenges," in IEEE Access, vol. 8, pp. 105008-105030, 2020, doi: 10.1109/ACCESS.2020.2999829.

[10]Real Time Drowsiness Detection System https://github.com/AnshumanSrivastava108/Real-Time-Drowsiness-Detection-System