Received: 30 August 2025 | Revised: 15 December 2025 | Accepted: 3 January 2026 | Published online: 20 January 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–9
DOI: 10.47852/bonviewAIA62027474

BON VIEW PUBLISHING

Python-Programmed Embedded ECU for Drowsiness-Responsive Dynamic Braking in Electric Vehicles

Kalaivanan Kumaran[1], Muhammad Nadzmi Razlan[1], Ahmad Safwan Abd Shukor[1] and Mohamad Tarmizi Abu Seman[1,*]

[1] *School of Electrical and Electronics Engineering, Universiti Sains Malaysia, Malaysia*

Abstract: Electric vehicles (EVs) are increasingly central to sustainable mobility, but braking control remains a safety-critical challenge. EVs must balance regenerative braking, which recovers kinetic energy, with dynamic braking, which ensures rapid deceleration in emergency situations. At the same time, driver drowsiness contributes to nearly one-fifth of serious accidents worldwide, underscoring the importance of systems that respond to external conditions and human states. This study presents a Python-programmed electronic control unit (ECU) on a Raspberry Pi Pico, integrating real-time driver drowsiness detection with adaptive braking control. Inputs from ultrasonic sensors, wheel encoders, and a camera-based drowsiness detection module are transmitted via the Message Queuing Telemetry Transport (MQTT) protocol. At the same time, a fuzzy inference engine processes driver condition, vehicle speed, and obstacle distance to generate proportional pulse width modulation (PWM) signals for motor braking. Experimental validation using a laboratory-scale prototype demonstrated distinct braking profiles under three conditions: slightly drowsy states produced proportional speed reductions as early warnings, drowsy states resulted in smooth full stops with consistent deceleration between 0.042 and 0.050 m/s², and emergency braking delivered rapid stops with shorter distances of 0.116 to 0.204 meters. While a 1-second latency was observed in some slightly drowsy runs, the system consistently adapted braking behavior and restored regular operation when drowsiness signals ceased. These findings validate that a Micro-Python-based ECU can reliably integrate behavioral monitoring with adaptive braking, offering a low-cost, scalable solution for future EV safety systems.

Keywords: electric vehicles, driver drowsiness detection, fuzzy braking, Raspberry Pi Pico, Micro-Python, embedded ECU

1. Introduction

Driver drowsiness remains one of the most critical human factors contributing to severe road accidents worldwide. Reduced alertness affects reaction time, decision-making, and vehicle control, often leading to delayed braking or loss of lane discipline. Recent studies have demonstrated that vision-based driver monitoring systems can effectively detect fatigue indicators, including prolonged eye closure, variations in head posture, and facial movement patterns, in real time [1, 2]. These systems provide a valuable foundation for enhancing road safety, particularly when implemented on embedded platforms that enable continuous monitoring during vehicle operation.

Most existing driver drowsiness detection systems primarily focus on warning mechanisms, including visual alarms and audible alerts [3]. While such approaches can increase driver awareness, they rely heavily on the assumption that the driver is capable of responding appropriately. In situations involving extreme fatigue or microsleep events, warning-based systems alone may be insufficient. This limitation has motivated recent research toward more proactive safety interventions, where vehicle control actions are automatically triggered when the driver's state deteriorates beyond a safe threshold [4].

Electric vehicles offer a suitable environment for such integration due to their electronically controlled drivetrains and flexible software architecture. Modern electric vehicles use multiple electronic control units to precisely manage propulsion, energy flow, and braking functions [5]. Regenerative braking systems, in particular, enable controlled deceleration through motor torque modulation while simultaneously recovering energy and enhancing overall efficiency [6]. Advanced control strategies have demonstrated that intelligent braking algorithms can enhance both vehicle stability and energy recovery under varying driving conditions [7].

Despite these advancements, limited research has addressed the direct coupling of driver state assessment with active braking control in electric vehicles. Existing regenerative braking studies generally focus on energy optimization, friction management, or battery longevity, without considering the driver's physiological condition as a control input [8]. Conversely, driver monitoring research often stops at detection and alert generation without extending into vehicle actuation. This separation highlights a significant gap in current intelligent vehicle safety frameworks.

To address this gap, embedded control solutions capable of real-time sensing, decision-making, and actuation are required. Recent work has demonstrated the feasibility of deploying complex perception and control algorithms on low-cost embedded platforms such as Raspberry Pi-based systems [9]. Python-based control frameworks further enable rapid development, flexibility, and integration of sensor data with motor control logic, making them suitable for experimental and prototype-level electronic control unit design.

In this context, the present study investigates a Python-programmed embedded electronic control unit that integrates real-time driver drowsiness detection with dynamic braking control in an electric

***Corresponding author:** Mohamad Tarmizi Abu Seman, School of Electrical and Electronics Engineering, Universiti Sains Malaysia, Malaysia. Email: mohdtarmizi@usm.my

vehicle platform. By linking vision-based fatigue assessment directly to motor braking commands, the proposed system aims to enhance road safety through timely and autonomous intervention. The approach builds upon recent developments in driver monitoring, embedded intelligence, and regenerative braking control, contributing toward safer and more adaptive electric vehicle systems [10]. A preliminary version of this work was presented at ROVISP 2025 under the title "Design and Implementation of a Drowsiness Aware Braking System Using an Embedded EV ECU." This manuscript represents a substantially extended version, incorporating in-depth methodological analysis, comprehensive literature integration, and a more thorough evaluation of system performance.

2. Literature Review

Electric vehicles are increasingly relying on intelligent braking systems to enhance both energy efficiency and operational safety. Regenerative braking has been widely studied as an effective approach to recover kinetic energy during deceleration and convert it into electrical energy for battery charging. Recent studies have demonstrated that adaptive braking control strategies can significantly enhance energy recovery while maintaining vehicle stability and braking comfort [7, 10, 6, 8]. Advanced controllers, such as fuzzy logic-based and model-driven approaches, have demonstrated superior performance in optimizing braking torque distribution under varying speed and load conditions, resulting in improved state of charge retention and extended driving range [11, 12]. These findings confirm that regenerative braking plays a crucial role in enhancing the efficiency of electric vehicles, particularly when combined with intelligent control logic.

Parallel to energy optimization research, significant attention has been given to driver drowsiness detection as a major safety concern in intelligent transportation systems. Vision-based monitoring approaches dominate recent literature due to their nonintrusive nature and compatibility with real-time implementation. Techniques based on facial landmarks, eye closure duration, head posture, and yawning patterns have demonstrated high accuracy in identifying driver fatigue [13, 14, 1, 15]. Deep learning models, particularly convolutional neural networks, have further improved detection robustness under varying lighting and driver conditions [16, 17]. Several studies highlight that fatigue detection systems deployed on embedded platforms can operate with low latency while maintaining reliable performance, making them suitable for in-vehicle applications [2].

Despite the progress in detection accuracy, most drowsiness monitoring systems remain limited to warning-based interventions such as visual or auditory alerts. This approach assumes driver responsiveness, which may not hold during severe fatigue or microsleep events. Recent works have begun exploring proactive safety responses, where vehicle behavior is modified based on driver condition [18]. However, such implementations are still scarce and often lack integration with braking systems that can actively reduce vehicle speed in critical scenarios.

Embedded electronic control units form the backbone of modern electric vehicle architectures, enabling real-time coordination among sensing, decision-making, and actuation. Low-cost embedded platforms, such as Raspberry Pi-based systems, have been widely adopted for prototyping intelligent vehicle functions due to their flexibility and processing capability [19, 20]. Research indicates that Python-based control environments facilitate rapid algorithm development while ensuring sufficient performance for real-time applications. These embedded systems have successfully hosted perception algorithms, control logic, and actuator commands within a unified framework [9].

Communication protocols play a crucial role in enabling the reliable exchange of data between sensors, controllers, and monitoring interfaces. Lightweight publish-subscribe protocols such as MQTT have gained popularity in embedded automotive and industrial applications due to their low overhead and scalability [5, 21]. Studies demonstrate that MQTT-based communication supports real-time data transmission with minimal latency, making it suitable for safety-critical systems where a timely response is essential. Integration of embedded controllers with MQTT enables efficient coordination between driver monitoring modules and braking control units, facilitating closed-loop safety responses.

Although substantial research exists independently on regenerative braking, driver drowsiness detection, embedded control systems, and communication frameworks, there is limited work that addresses their unified integration within a single embedded control architecture. Existing studies often treat driver monitoring and braking control as separate subsystems, resulting in fragmented safety solutions [22]. This gap underscores the need for an integrated approach in which the driver's physiological state directly influences braking behavior through an embedded electronic control unit, supported by reliable communication and real-time decision-making.

2.1. Theoretical framework

This study is founded on three complementary theoretical pillars. First, adaptive regenerative braking control research demonstrates that fuzzy logic-based controllers can effectively map uncertain inputs into smooth braking torque adjustments, thereby improving stability under varying operating conditions [7, 12, 8]. Incorporating braking intention recognition further enables differentiated responses for normal and emergency scenarios [11], supporting fuzzy control as an effective strategy for adaptive braking.

Second, behavioral driver monitoring studies have established that visual cues, such as eyelid closure, blink rate, and facial expression, are reliable indicators of drowsiness [23]. Temporal behavior analysis and individual driving patterns further enhance driver state estimation accuracy [15]. These findings justify the use of camera-based drowsiness detection as supervisory input for braking decisions when driver responsiveness declines.

Third, embedded system and communication studies confirm that lightweight platforms, such as the Raspberry Pi Pico, can support sensing, control, and actuation tasks for experimental electric vehicle systems [9]. MQTT has been shown to provide low latency and reliable communication in constrained environments, enabling distributed control architectures with minimal computational overhead [24].

Together, these foundations define a human-centered adaptive control framework in which driver state and vehicle dynamics jointly influence braking behavior through fuzzy decision logic and embedded MQTT-based coordination.

2.2. Contributions of this study

Within the context defined above, this study makes four main contributions. First, it integrates a vision-based drowsiness detection module with a regenerative braking controller, allowing braking intensity to be adjusted according to both driver alertness and distance to potential obstacles. Existing works often consider these elements separately, whereas the present study treats driver state as a direct input to the braking decision. Second, it demonstrates that a Python-programmed Raspberry Pi Pico-based controller can manage sensor acquisition, fuzzy rule evaluation, and PWM generation within the response times required for electric vehicle braking, extending earlier feasibility studies on embedded Python control [5, 22]. Third, the study defines a fuzzy rule base and membership structure that reflects braking intention and drowsiness level, building on prior regenerative braking and optimization work while adapting it to a human-monitored context [25, 11]. Fourth, it uses MQTT as the communication layer between the vision module and the ECU, leveraging its low overhead and scalability to ensure the timely delivery of drowsiness events on a resource-constrained platform [20, 26]. Together, these contributions offer a practical pathway toward affordable, intelligent braking systems that enhance electric vehicle safety without relying on high-end automotive hardware.

3. Research Methodology

The proposed system was implemented as a compact, embedded architecture that integrates driver state monitoring, decision logic, and braking actuation. The overall framework comprises a vision-based drowsiness detection module, an embedded electronic control unit, and a motor braking interface, all of which are coordinated through lightweight communication. This structure enables real-time assessment of driver alertness and adaptive braking response under different operating conditions.

Driver drowsiness detection was performed using a camera-based vision module that continuously monitored facial indicators associated with fatigue. The captured visual data were processed to extract features related to eye closure, blink duration, and head posture. These indicators were mapped to discrete driver states representing alert, mildly drowsy, sustained drowsy, and emergency conditions. The resulting driver state information served as supervisory input to the braking controller.

The electronic control unit was implemented using a Raspberry Pi Pico platform programmed in Python. This embedded controller received driver state data and vehicle-related inputs, including speed and distance information, and executed decision logic to determine appropriate braking actions. A fuzzy logic-based control strategy was employed to translate driver state and vehicle context into graded braking commands. This approach allowed smooth modulation of motor braking torque while avoiding abrupt transitions that could compromise stability or passenger comfort.

Communication between the vision module and the electronic control unit was established using the MQTT protocol. The driver monitoring module published drowsiness state updates, while the ECU subscribed to these messages and responded accordingly. MQTT was selected due to its low overhead and suitability for constrained embedded environments, enabling timely and reliable data exchange without excessive computational load.

Braking actuation was achieved through motor speed control, where braking intensity was adjusted according to the output of the fuzzy controller. Under mild drowsiness, gradual speed reduction was applied as an early intervention mechanism. Sustained drowsiness triggered controlled deceleration, while emergency conditions resulted in rapid braking to minimize stopping distance. The entire system was evaluated on a prototype setup to verify real-time operation, response consistency, and the feasibility of low-cost embedded deployment.

4. Result and Discussion

4.1 Prototype implementation

A prototype vehicle was developed to validate the integration of the driver drowsiness detection system with the ECU-controlled dynamic braking mechanism. The physical assembly, illustrated in Figures 1–3, demonstrates the hardware implementation based on the schematic design.

From the top view in Figure 1, the key components include the Raspberry Pi Pico 2W microcontroller, the L298N motor driver, the XL4015 buck converter, LEDs, and a piezoelectric buzzer, all of which are centrally positioned to optimize wiring efficiency. The bottom view, shown in Figure 2, displays the dual DC motors equipped with built-in encoders, which are used to capture real-time wheel speed. The front view, shown in Figure 3, highlights the placement of HC-

Figure 1
Top view of the ECU prototype showing the Raspberry Pi Pico 2W, L298N motor driver, XL4015 buck converter, LEDs, and piezo buzzer

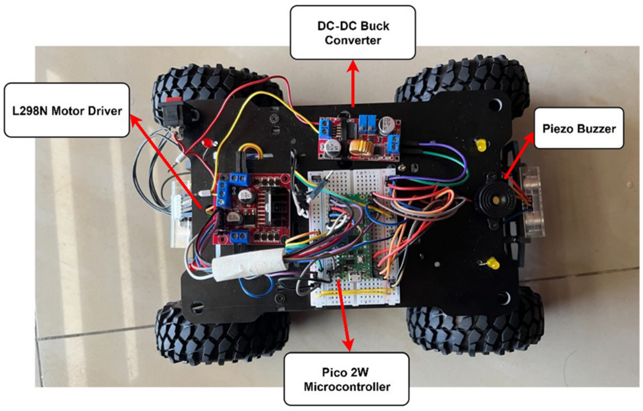


Figure 2
Bottom view of the ECU prototype highlighting dual DC motors with integrated encoders for wheel speed

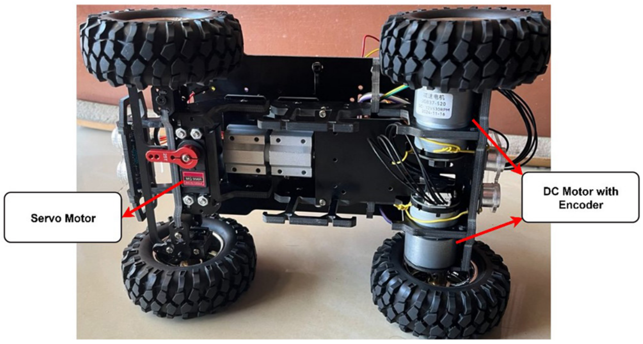
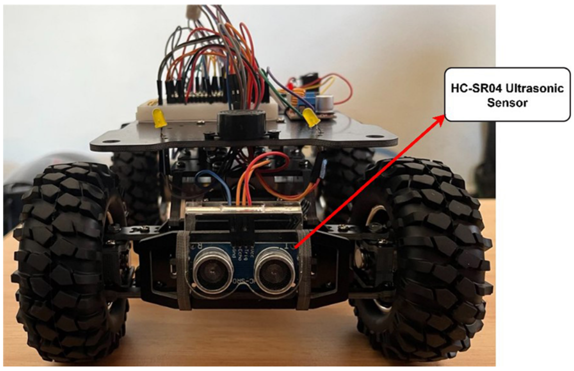


Figure 3
Front view of the ECU prototype displaying HC-SR04 ultrasonic sensors for obstacle detection and vehicle proximity sensing



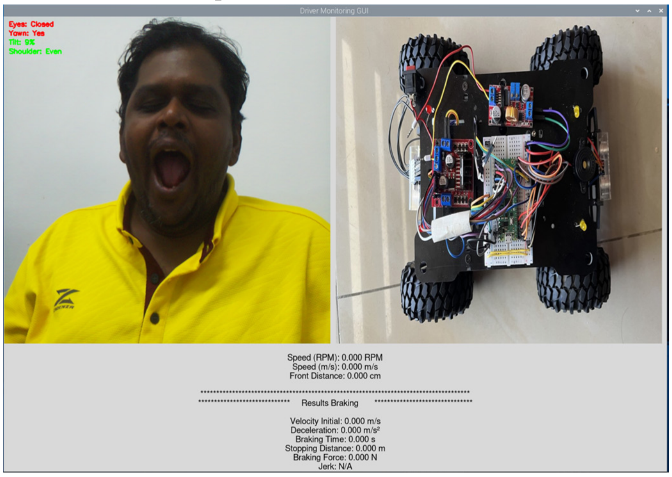
SR04 ultrasonic sensors for obstacle detection, replicating the front and rear vehicle proximity sensing commonly found in real-world driving scenarios.

This layout ensures functional consistency, ease of debugging, and reliable interaction between all components, forming the experimental basis for evaluating braking behavior under different driver states.

4.2. Braking performance evaluation

The ECU's performance was evaluated through controlled experiments designed to simulate three distinct driver conditions: slightly drowsy, fully drowsy, and emergency braking. Tests were conducted in a laboratory environment over a 10-meter track, with the prototype transmitting data to a fixed Raspberry Pi 4 unit that hosted the graphical user interface (GUI) for parameter monitoring, as shown in Figure 4.

Figure 4
Graphical user interface (GUI) for driver monitoring and braking parameter visualization



4.2.1. Slightly drowsy driving condition

Under slightly drowsy conditions, braking forces of 30%–60% were applied. The PWM duty cycle was reduced proportionally, with speed decrements calculated as:

$$Decrement = 10000 + (20000 - 10000) * (Fb - 30) / (60 - 30) \quad (1)$$

The new motor speed was then set as:

$$New\ speed = 50000 - Decrement \quad (2)$$

where *Fb* denotes the braking force percentage and New speed represents the updated motor speed.

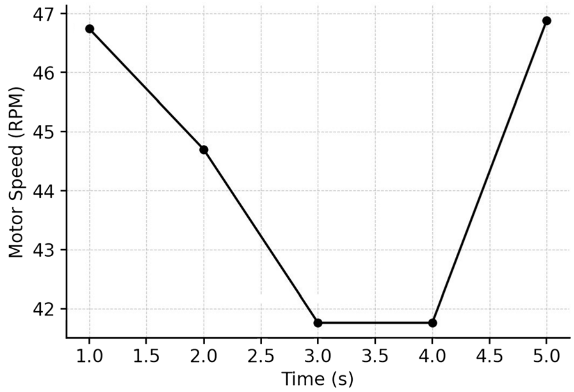
Results from three runs confirmed a consistent system response. A representative dataset for run 1 is presented in Table 1, while the corresponding speed-time profile is shown in Figure 5.

Across all runs, the system successfully reduced motor speed when drowsiness was detected and restored once normal conditions resumed. In some cases, a 1-second response delay was observed likely due to algorithm buffering or mechanical inertia. While this confirms the system's adaptability, it also indicates scope for improving real-time responsiveness.

Table 1
Test run 1 for slightly drowsy state

Time (s)	Brake force	Speed decrement (16-bit resolution)	Speed (16-bit resolution)	Motor speed (RPM)
1	0	0	50000	46.74
2	45.95	15317	50000	44.70
3	33.28	11093	34684	41.76
4	41.25	13750	38907	41.76
5	0	0	36250	46.88

Figure 5
Motor speed vs. time for run 1 under slightly drowsy conditions



4.2.2. Fully drowsy braking condition

For sustained drowsiness, braking forces of 60%–100% were applied, triggering a smooth deceleration until the vehicle came to a complete stop. The PWM decrement formula was adjusted accordingly:

$$Decrement = 8000 + (15000 - 8000) * (Fb - 61) / (100 - 61) \quad (3)$$

$$New\ speed = Current\ speed - Decrement \quad (4)$$

where *Current speed* represents the instantaneous motor speed prior to braking.

The algorithm iterated every 0.1 second until the motor speed reached zero. Table 2 summarizes the results of five runs under drowsy conditions.

The ECU consistently achieved controlled braking, with stopping times ranging from 3.2 to 4.3 seconds and distances of 0.26 to 0.39 meters. Deceleration remained moderate (0.042–0.050 m/s²), confirming the design's intention to prioritize stability and gradual halting.

4.2.3. Emergency braking condition

Emergency braking was triggered when the front ultrasonic sensor detected an obstacle within 30 centimeters, overriding fuzzy logic decisions. The ECU immediately cut PWM to zero, producing rapid deceleration. Results for five runs are presented in Table 3.

Emergency braking consistently produced higher deceleration values (0.098–0.151 m/s²), reducing stopping distance to 0.12–0.20 meters and braking time to 1.6–3.2 seconds. This demonstrates the system's capacity to prioritize collision avoidance over comfort in critical situations.

Table 2
Braking performance under drowsy conditions

Run	Brake force	Initial velocity (m/s)	Braking time (s)	Stopping distance (m)	Deceleration (m/s²)
1	99.992	0.161	3.213	0.259	0.050
2	100	0.188	3.753	0.353	0.050
3	99.987	0.192	3.885	0.373	0.049
4	99.990	0.183	4.335	0.396	0.042
5	99.986	0.195	3.907	0.380	0.050

Table 3
Braking performance under emergency conditions

Run	Front distance (m)	Initial velocity (m/s)	Braking time (s)	Stopping distance (m)	Deceleration (m/s²)
1	25.5	0.072	3.20	0.116	0.023
2	4.06	0.171	1.68	0.143	0.102
3	14.43	0.167	1.70	0.142	0.098
4	18.59	0.248	1.64	0.204	0.151
5	14.88	0.178	1.79	0.160	0.099

4.3. Comparative study with the existing system

The braking performance of the proposed ECU was tested under both drowsy and emergency conditions, and the outcomes were compared across three critical areas: braking time, stopping distance, and deceleration. As shown in Figure 6, emergency braking consistently achieved faster responses, requiring only 1.64 to 3.20 seconds to stop the vehicle, compared to 3.2 to 4.3 seconds under drowsy braking. This difference reflects the controller's ability to apply gentle deceleration during periods of drowsiness while prioritizing rapid stopping in emergency situations. Similarly, Figure 7 shows that stopping distances were much shorter in emergency conditions, ranging from 0.116 to 0.204 meters, compared to drowsy conditions, where they ranged from 0.259 to 0.396 meters. Finally, Figure 8 highlights contrasting deceleration profiles: drowsy braking produced moderate and stable values between 0.042 and 0.050 meters per second squared, consistent with the design goal of passenger comfort, whereas emergency braking generated higher and more

Figure 7
Stopping distance comparison between drowsy and emergency conditions

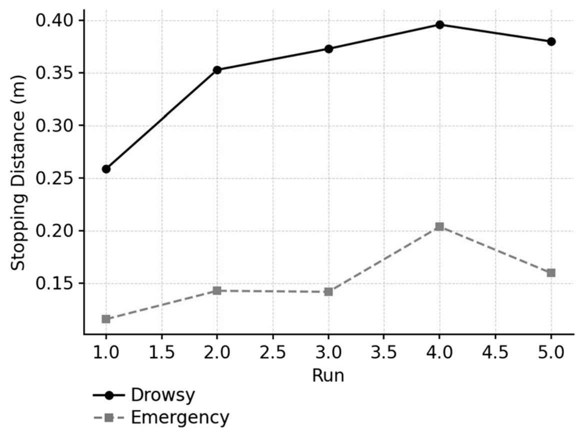


Figure 6
Braking time comparison between drowsy and emergency conditions

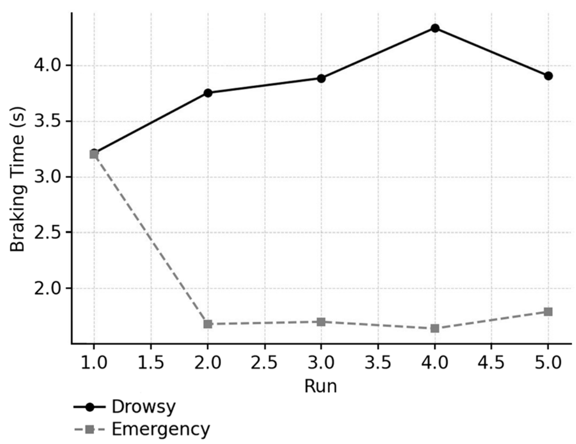
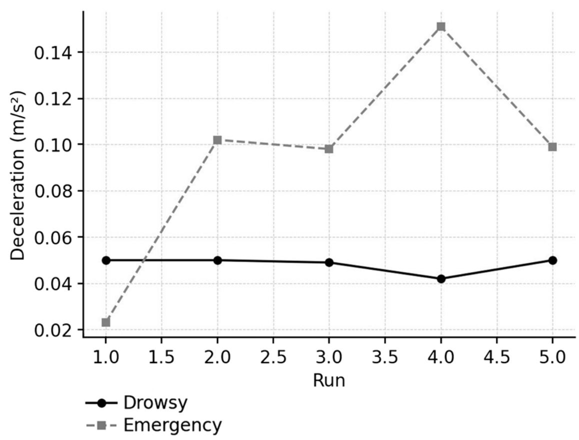


Figure 8
Deceleration comparison between drowsy and emergency conditions



variable values between 0.098 and 0.151 meters per second squared, aligning with the principle of collision avoidance.

These results are consistent with established findings in the literature. Previous research on fuzzy braking controllers demonstrated that adaptive inference approaches make smoother braking responses compared to binary schemes [27, 28]. In contrast, autonomous emergency braking systems were designed to minimize stopping distance even at the cost of ride comfort [29]. The present ECU unites these two approaches by incorporating real-time driver drowsiness as a control input, thereby extending conventional frameworks that rely solely on vehicle-state measurements [30, 31]. This human-aware dimension represents a significant advancement toward personalized safety mechanisms.

From an implementation standpoint, most earlier systems relied on high-end ECUs or computationally intensive software platforms. By contrast, the proposed ECU demonstrates that a Raspberry Pi Pico running Micro-Python can achieve deterministic real-time performance in braking applications [32, 33]. Furthermore, the adoption of MQTT-based communication between the driver monitoring system and the ECU supports rapid and reliable data exchange, with latency values well within the millisecond range as reported in multiple evaluations [34–37]. This modular communication layer not only ensures timely actuation but also enhances scalability for integration with other safety subsystems.

Overall, the comparative study emphasizes three distinctive advances of the proposed design: first, the integration of human-state

Table 4
Comparison of existing studies with the proposed drowsiness-aware braking ECU

Category	Reference	Method/platform	Output/focus	Limitation	Difference from proposed work
Regenerative braking	[11]	Dual fuzzy braking with force distribution	Smooth braking, intention-based control	Depends on pedal behavior	Uses the drowsiness state instead of pedal intention
	[7]	ECE-compliant braking control	Regulatory-compliant stability	No human state input	Includes behavioral trigger
	[12]	Multi-mode regenerative braking	Energy recovery optimization	Fixed braking modes	Selects braking intensity dynamically based on fatigue
	[8]	Fuzzy neural network braking	Adaptive torque control	High computational load	Uses lightweight Python fuzzy logic suitable for microcontrollers
Drowsiness detection	[13]	Edge-based facial fatigue detection	On-device visual detection	No actuation	Links detection to immediate braking
	[15]	EEG and facial hybrid detection	High sensitivity	Requires contact sensors	Uses noncontact camera-based sensing
	[17]	CNN-based real-time detection	High accuracy	No ECU integration	Forms a complete sensing-to-actuation pipeline
	[9]	IoT-assisted deep learning model	Remote fatigue monitoring	Not vehicle focused	Provides local real-time braking response
Embedded ECU	[5]	Embedded vision-based device	Mechanical task automation	Not vehicle related	Targets electric vehicle safety
	[21]	RV IoT multi-architecture board	Edge computing platform	General-purpose hardware	Designed for safety-critical braking
	[22]	Micro-Python extended ESP32 design	Rapid prototyping	No braking pathway	Unifies sensing, inference, and actuation in one ECU
MQTT communication	[18]	MQTT for home automation	Reliable low-overhead messaging	Not safety critical	Enables real-time brake triggering
	[38]	MQTT for environmental sensing	Low-latency data transfer	Monitoring only	Used for actuation rather than monitoring
	[39]	MQTT in incubator control	Stable communication	Not vehicular	First application of MQTT in drowsiness-driven braking
Proposed system	–	Raspberry Pi Pico, camera, MQTT	Real-time drowsiness detection and adaptive braking	–	First unified system combining drowsiness sensing, braking control, and MQTT communication

monitoring into braking decisions [31, 32]; second, the feasibility of low-cost Micro-Python ECU deployment [32, 33]; and third, the successful use of MQTT messaging for real-time distributed control [34–37]. Together, these features confirm the novelty and practical value of the system as a cost-effective, human-aware safety solution for intelligent electric vehicles.

4.4. Extended comparison with recent literature

To further contextualize system performance, the proposed ECU was compared with existing work across regenerative braking, drowsiness detection, embedded control, and MQTT communication. Table 4 provides a structured comparison.

Across these domains, regenerative braking studies improved energy recovery and braking smoothness [11, 7, 12, 8], drowsiness detection studies enhanced recognition accuracy [13, 15, 17, 9], embedded systems research advanced hardware efficiency [5, 21, 22], and MQTT works demonstrated reliable low-latency communication [18, 38, 39]. However, none connected behavioral state detection to immediate braking actuation within a unified ECU. The proposed design, therefore, establishes the first integrated pathway that links human state sensing, embedded decision-making, and real-time motor braking.

4.5. Novelty of this study

Prior studies have independently improved regenerative braking performance, drowsiness detection accuracy, embedded system efficiency, and MQTT-based communication reliability [5, 7, 8, 9, 11–13, 15, 17, 18, 21, 22, 38, 39]. However, existing works do not integrate behavioral state detection with immediate braking actuation within a unified embedded control unit, as systematically compared in Table 4. This study addresses that gap by establishing a direct sensing-to-actuation pathway linking driver state and real-time motor braking.

The primary contribution lies in introducing a human-centered braking trigger in which detected drowsiness initiates adaptive regenerative braking rather than relying solely on pedal input or vehicle dynamics. A second contribution is the demonstration that real-time visual inference and braking control can be achieved using low-cost Raspberry Pi-based embedded processing, eliminating dependence on high-end automotive ECUs. A third contribution is the application of MQTT as an event-driven communication layer between detection and braking modules, enabling timely and reliable actuation within a safety-critical context.

Together, these contributions define a unified embedded architecture that integrates behavioral sensing, decision logic, and adaptive braking, providing a scalable foundation for human-aware safety systems in electric vehicles.

5. Conclusion

This research presented a Python-programmed embedded ECU on a Raspberry Pi Pico for drowsiness-responsive dynamic braking in electric vehicles. The system integrated real-time sensor data with driver monitoring inputs and applied fuzzy logic to adjust braking force across varying levels of drowsiness, including slightly drowsy, fully drowsy, and emergency conditions. Prototype validation confirmed reliable performance, where subtle speed reductions provided early warnings, controlled full stops ensured safety under sustained drowsiness, and rapid braking minimized stopping distances in emergencies. These results demonstrate that Python-based embedded control can meet safety-critical requirements on low-cost microcontrollers, offering a practical, intelligent, and scalable solution for EV safety systems.

Recommendations

Future work will expand this prototype toward larger-scale platforms and real vehicle testing. Integration with full-scale EV hardware, including hydraulic braking systems and regenerative subsystems, will be essential to validate the ECU in realistic driving conditions. At the same time, driver monitoring improvements will require deploying more advanced AI models capable of maintaining robust performance under diverse lighting and environmental situations, such as night driving. Performance optimization will also be a priority. Micro-Python execution can be enhanced through hybrid coding strategies, such as C extensions or embedded coprocessors, which reduce latency in safety-critical scenarios. Finally, compliance with automotive communication standards, such as the CAN bus protocol, will be pursued to ensure seamless integration with existing vehicle architectures. These directions will strengthen the system's reliability, scalability, and industry relevance, bringing it closer to real-world electric vehicle applications.

Acknowledgement

The authors gratefully acknowledge the support provided by the Higher Learning Provider (HLP) and the Ministry of Higher Education Malaysia (KPT), whose contributions were instrumental in enabling the successful completion of this research.

Ethical Statement

The authors declare that this study did not require formal ethical approval because it involves nonmedical engineering research focused on system development and prototype validation. Under the research governance practices of Universiti Sains Malaysia, Institutional Review Board (IRB) or ethics committee approval is not required for this type of nonmedical, nonclinical engineering research.

All facial images included in this manuscript were used with explicit permission from the participant, who is also one of the authors. The images were captured solely for the purpose of demonstrating the proposed system's functionality, and authorization for their use in academic publications has been obtained.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Kalaivanan Kumaran: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing. **Muhammad Nadzmi Razlan:** Software, Validation, Investigation, Data curation, Writing – review & editing. **Ahmad Safwan Abd**

Shukor: Software, Validation, Investigation, Data curation, Writing – review & editing. **Mohamad Tarmizi Abu Seman:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Project administration.

References

[1] Sneha, C. I., Rohini, M., Manoj, S. O., Giri, J., Odeh, M., & Sunheriya, N. (2024). Real-time driver monitoring system using computer vision to detect drowsiness for enhanced traffic safety. In *International Conference on Emerging Trends in Engineering and Medical Sciences*, 807–812. https://doi.org/10.1109/ICETEMS64039.2024.10965037

[2] Murugan, S., Abiishek, G., Kumar, P., & Rubini, B. (2024). Next–Gen driver monitoring system. In *Global Power, Energy and Communication Conference*, 795–799. https://doi.org/10.1109/GPECOM61896.2024.10582567

[3] Usharani, P., Devan, P. K., Saravanan, S., Hariharan, R., & Karkuzhali, S. (2024). Sensor based identification of driver behaviour. *International Journal of Vehicle Structures and Systems*, *16*(5), 686–689. https://doi.org/10.4273/ijvss.16.5.06

[4] Singh, P., Behera, P. K., & Pattnaik, M. (2024). An interleaved DC-DC converter controlled BLDC motor drive for electric vehicle application. In *International Conference on Smart Power Control and Renewable Energy*, 1–6. https://doi.org/10.1109/ICSPCRE62303.2024.10675304

[5] Reddi, C., & Ghosh, A. (2024). Bidirectional DC converter using Li-Ion battery for EV applications. In *International Conference on Communication Systems and Network Technologies*, 1270–1275. https://doi.org/10.1109/CSNT60213.2024.10546150

[6] Prasad, S. L., & Gudipalli, A. (2024). Optimal energy consumption control strategy for electric vehicle. *IEEE Access*, *12*, 197734–197747. https://doi.org/10.1109/ACCESS.2024.3512188

[7] Ashok, B., Kannan, C., Deepak, C., Ramesh, R., Narendhra, T. M., Farrag, M. E., … & Kavitha, C. (2024). Model based integrated control strategy for effective brake energy recovery to extend battery longevity in electric two wheelers. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *238*(9), 2843–2864. https://doi.org/10.1177/09544070231167616

[8] Gupta, G., Sudeep, R., Ashok, B., Vignesh, R., Kannan, C., Kavitha, C., … & Emara, A. (2024). Intelligent regenerative braking control with novel friction coefficient estimation strategy for improving the performance characteristics of hybrid electric vehicle. *IEEE Access*, *12*, 110361–110384. https://doi.org/10.1109/ACCESS.2024.3440210

[9] Maheswari, P., Gowriswari, S., Balasubramani, S., Babu, A. R., NK, J., & Murugan, S. (2024). Intelligent headlights for adapting beam patterns with Raspberry Pi and convolutional neural networks. In *International Conference on Device Intelligence, Computing and Communication Technologies*, 182–187. https://doi.org/10.1109/DICCT61038.2024.10533159

[10] Munaaf, S. A., Vasudevan, B., & Routray, S. (2024). Regenerative braking solutions for electric vehicles: Advancing efficiency and energy recapture. In *International Conference on Communication and Electronics Systems*, 331–337. https://doi.org/10.1109/ICCES63552.2024.10860067

[11] Qin, Y., Zheng, Z. A., & Chen, J. (2024). Dual-fuzzy regenerative braking control strategy based on braking intention recognition. *World Electric Vehicle Journal*, *15*(11), 524. https://doi.org/10.3390/wevj15110524

[12] Liu, B., Li, G., & Wang, S. (2024). Research on multi-mode braking energy recovery control strategy for battery electric vehicles. *Applied Sciences*, *14*(15), 6505. https://doi.org/10.3390/app14156505

[13] Chen, M., He, P., & Wang, Y. (2024). The driver drowsiness detection system based on human-computer interaction. In *AIP Conference Proceedings*, *3194*(1), 040003. https://doi.org/10.1063/5.0222836

[14] Salem, D., & Waleed, M. (2024). Drowsiness detection in real-time via convolutional neural networks and transfer learning. *Journal of Engineering and Applied Science*, *71*(1), 122. https://doi.org/10.1186/s44147-024-00457-z

[15] Sun, Y., Wang, R., Zhang, H., Ding, N., Ferreira, S., & Shi, X. (2024). Driving fingerprinting enhances drowsy driving detection: Tailoring to individual driver characteristics. *Accident Analysis & Prevention*, *208*, 107812. https://doi.org/10.1016/j.aap.2024.107812

[16] Trigka, M., Dritsas, E., & Mylonas, P. (2024). Driver sleepiness detection using machine learning models on EEG data. In *Hellenic Conference on Artificial Intelligence*, 1–4. https://doi.org/10.1145/3688671.3688780

[17] Liu, K., & Guo, P. (2024). Design and implementation of fatigue driving detection system based on OpenCV. In *International Conference on Computer Information Science and Artificial Intelligence*, 166–172. https://doi.org/10.1145/3703187.3703215

[18] Chen, C. Y., Wu, S. H., Huang, B. W., Huang, C. H., & Yang, C. F. (2024). Web-based Internet of Things on environmental and lighting control and monitoring system using node-RED, MQTT and Modbus communications within embedded Linux platform. *Internet of Things*, *27*, 101305. https://doi.org/10.1016/j.iot.2024.101305

[19] He, W., Baig, M. J. A., & Iqbal, M. T. (2024). An open-source supervisory control and data acquisition architecture for photovoltaic system monitoring using ESP32, Banana Pi M4, and Node-RED. *Energies*, *17*(10), 2295. https://doi.org/10.3390/en17102295

[20] Salama, M., & Raslen, B. (2024). MQTT in action: Building reliable and scalable home automation systems. *Sakarya University Journal of Computer and Information Sciences*, *7*(3), 494–509. https://doi.org/10.35377/saucis...1504759

[21] Pk, A. V., & Shah, V. A. (2024). Realization of ultracapacitor as sole energy storage device in induction motor drive electric vehicle with modified state timing based field weakening control algorithm. *Journal of Energy Storage*, *81*, 110352. https://doi.org/10.1016/j.est.2023.110352

[22] Oubelaid, A., Belkhier, Y., Ahmed, A. A., Avikal, S., & Bajaj, M. (2024). Fuzzy power management for low carbon footprint vehicles. In *E3S Web of Conferences*, *564*, 02002. https://doi.org/10.1051/e3sconf/202456402002

[23] Alguindigue, J., Singh, A., Narayan, A., & Samuel, S. (2024). Biosignals monitoring for driver drowsiness detection using deep neural networks. *IEEE Access*, *12*, 93075–93086. https://doi.org/10.1109/ACCESS.2024.3423723

[24] Patti, G., Leonardi, L., Testa, G., & Bello, L. L. (2024). PrioMQTT: A prioritized version of the MQTT protocol. *Computer Communications*, *220*, 43–51. https://doi.org/10.1016/j.comcom.2024.03.018

[25] Ning, X., Wang, Z., Lin, Y., Yin, Y., Wang, J., & Hong, Y. (2024). Optimization of regenerative braking control strategy

in single-pedal mode based on electro-mechanical braking. *IEEE Access*, *12*, 170994–171014. https://doi.org/10.1109/ACCESS.2024.3485198

[26] Innamorati, C., Boiano, A., Redondi, A. E., & Cesana, M. (2025). The pulse of MQTT in the wild: A large-scale traffic analysis. *Computer Networks*, 111845. https://doi.org/10.1016/j.comnet.2025.111845

[27] Yin, Z., Ma, X., Su, R., Huang, Z., & Zhang, C. (2023). Regenerative braking of electric vehicles based on fuzzy control strategy. *Processes*, *11*(10), 2985. https://doi.org/10.3390/pr11102985

[28] Chaoyi, W., Hongtao, S., Guangwu, D., & Yuan, F. (2024). Optimized fuzzy regenerative braking strategy for electric vehicles based on sparrow search algorithm. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 09544070251361973. https://doi.org/10.1177/09544070251361973

[29] Rajendar, S., Rathinasamy, D., Pavithra, R., Kaliappan, V. K., & Gnanamurthy, S. (2022). Prediction of stopping distance for autonomous emergency braking using stereo camera pedestrian detection. *Materials Today: Proceedings*, *51*, 1224–1228. https://doi.org/10.1016/j.matpr.2021.07.211

[30] Arakawa, T. (2021). Trends and future prospects of the drowsiness detection and estimation technology. *Sensors*, *21*(23), 7921. https://doi.org/10.3390/s21237921

[31] Perrotte, G., Bougard, C., Portron, A., & Vercher, J. L. (2024). Monitoring driver drowsiness in partially automated vehicles: Added value from combining postural and physiological indicators. *Transportation research part F: traffic psychology and behaviour*, *100*, 458–474. https://doi.org/10.1016/j.trf.2023.12.010

[32] Cho, S. Y., Delgado, R., & Choi, B. W. (2023). Feasibility study for a Python-based embedded real-time control system. *Electronics*, *12*(6), 1426. https://doi.org/10.3390/electronics12061426

[33] Davidson, Z. C., Dang, S., & Vasilakos, X. (2024). Blended laboratory design using Raspberry Pi Pico for digital circuits and systems. *IEEE Transactions on Learning Technologies*, *17*, 1170–1183. https://doi.org/10.1109/TLT.2024.3363230

[34] Gentile, A. F., Macrì, D., Carnì, D. L., Greco, E., & Lamonaca, F. (2024). A network performance analysis of MQTT security protocols with constrained hardware in the dark net for DMS. *Applied Sciences*, *14*(18), 8501. https://doi.org/10.3390/app14188501

[35] Silva, D., Carvalho, L. I., Soares, J., & Sofia, R. C. (2021). A performance analysis of internet of things networking protocols: Evaluating MQTT, CoAP, OPC UA. *Applied Sciences*, *11*(11), 4879. https://doi.org/10.3390/app11114879

[36] Chang, Y. H., Wu, F. C., & Lin, H. W. (2025). Design and implementation of esp32-based edge computing for object detection. *Sensors*, *25*(6), 1656. https://doi.org/10.3390/s25061656

[37] Pawar, S., Panigrahi, N., Jyothi, A. P., Lokhande, M., Godse, D., & Jadhav, D. B. (2023). Evaluation of delay parameter of MQTT protocol. *International Journal of Engineering Trends and Technology*, *71*(3), 227–235. https://doi.org/10.14445/22315381/IJETT-V71I3P223

[38] Kumar, L. S., & Ayyarao, T. S. L. V. (2024). Real-time environmental monitoring: IoT integration and remote access via Raspberry Pi and VNC viewer. In *International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles*, 1–6. https://doi.org/10.1109/ICPEEV63032.2024.10931978

[39] Amelia, A., Fahmi, N., Pranoto, H., Sundawa, B. V., Hutauruk, I. S., & Arief, A. (2020). MQTT protocol implementation for monitoring of environmental based on IoT. In *International Conference on Applied Science and Technology*, 700–703. https://doi.org/10.1109/iCAST51016.2020.9557694

How to Cite: Kumaran, K., Razlan, M. N., Shukor, A. S. A., & Seman, M. T. A. (2026). Python-Programmed Embedded ECU for Drowsiness-Responsive Dynamic Braking in Electric Vehicles. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62027474