

Co-Pilot AI: Intelligent Vehicle Assistant with Predictive Diagnostics and Voice-Activated Controls

Mohan Kumar R¹, Tamil M², Sanjay Aravind K³ and Srihari S⁴

¹⁻⁴Sri Ramakrishna Engineering College/Electrical and Electronics Engineering, Coimbatore, India

Email: mohankumar.r@srec.ac.in, tamil.2003137@srec.ac.in, sanjayaravind.2103119@srec.ac.in,

srihari.2103132@srec.ac.in

Abstract—This paper presents the design and implementation of an intelligent vehicle dashboard that integrates real-time On-Board Diagnostics (OBD-II) data with a voice-driven assistant for enhanced driver interaction. The system employs a Raspberry Pi 5 platform connected to the vehicle's OBD-II interface via an ELM327 adapter, enabling continuous monitoring of key performance metrics. To ensure reliable data storage, a dual-logging strategy is implemented, utilizing both CSV files and a local SQLite database with a comprehensive schema design. A modular Python Flask backend facilitates seamless data retrieval for the voice assistant, leveraging OpenAI's Whisper model for speech-to-text (STT) conversion and Claude 3.7 Sonnet for natural language processing (NLP). Real-world testing across five different vehicle models demonstrates the system's ability to provide context-aware responses, including real-time performance metrics and predictive maintenance alerts. With a total response time averaging 8.16 seconds and balanced resource utilization, this research offers a practical architecture with potential applications in predictive maintenance and personalized vehicle experiences. The system's innovative integration of hardware sensors, cloud APIs, and edge computing creates a platform that significantly enhances the driving experience while addressing key challenges in modern automotive interfaces.

Index Terms— Electric Vehicles (EV), On-Board Diagnostics (OBD-II), Voice Assistant, Predictive Maintenance, Large Language Models (LLMs), Real-Time Data, Driver Assistance Systems, Edge Computing, ELM327 Adapter, Google Maps API.

I. INTRODUCTION

Despite of growing, number of advanced electronics and networked systems found in modern cars, many dashboard interfaces are still quite basic, providing little user interaction and little telemetry. Although high-end automakers have started incorporating voice assistants, these systems usually function independently of the vehicle's diagnostic information, resulting in a disjointed user experience. According to recent studies by Wang et al., 78% of drivers report having trouble simultaneously monitoring multiple parameters, highlighting the frequent difficulties drivers have accessing and interpreting critical vehicle information while operating their vehicles [1].

By fusing real-time On-Board Diagnostics (OBD-II) [4] data with a voice-activated assistant driven by cutting-edge Large Language Models (LLMs) [2][5], this paper suggests an advanced car dashboard system that overcomes these constraints.

Our strategy makes use of open standards and edge computing to produce an accessible, expandable platform, in contrast to commercial offerings that frequently call for proprietary hardware or cloud-only solutions. The central component of the system is a Raspberry Pi5, which is linked to the car's OBD-II [13] port using an ELM327 adapter. Key performance metrics like engine RPM, speed, throttle position, coolant temperature, and other important variables are continuously monitored. Data integrity is guaranteed by a dual-logging strategy that makes use of CSV files and a structured SQLite database, which facilitates both historical analysis and real-time queries.

Voice interaction is enabled through a streamlined interface that captures driver commands, processes them using OpenAI's Whisper model for speech-to-text conversion, and leverages Claude 3.7 Sonnet LLM for natural language understanding [15]. A text-to-speech engine provides the context-aware responses that are produced, allowing for a hands-free experience that improves driver safety. In this paper, a comprehensive architecture that combines advanced voice AI capabilities with real-time OBD-II diagnostics [12] [14] on hardware with limited resources is presented. It presents a predictive maintenance framework that improves vehicle dependability and user convenience by using historical vehicle data to forecast service requirements. The solution is flexible and scalable for a range of automotive applications because the modular software design also guarantees extensibility while maintaining core functionality on embedded platforms. By integrating real time diagnostics, AI-driven interfaces [23] [24], and intuitive design, the proposed system significantly advances vehicle dashboard capabilities. It addresses key user concerns such as maintenance awareness, operational efficiency, and distraction reduction, while contributing to the broader adoption of intelligent automotive systems.

II. SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

The proposed intelligent vehicle dashboard integrates vehicle telemetry, voice interaction, and navigation services on a Raspberry Pi5 platform [3] [25]. Fig. 1 illustrates the overall architecture, highlighting the data flow between components. The design emphasizes modularity, real-time performance, and adherence to automotive standards, addressing limitations in conventional vehicle monitoring systems.

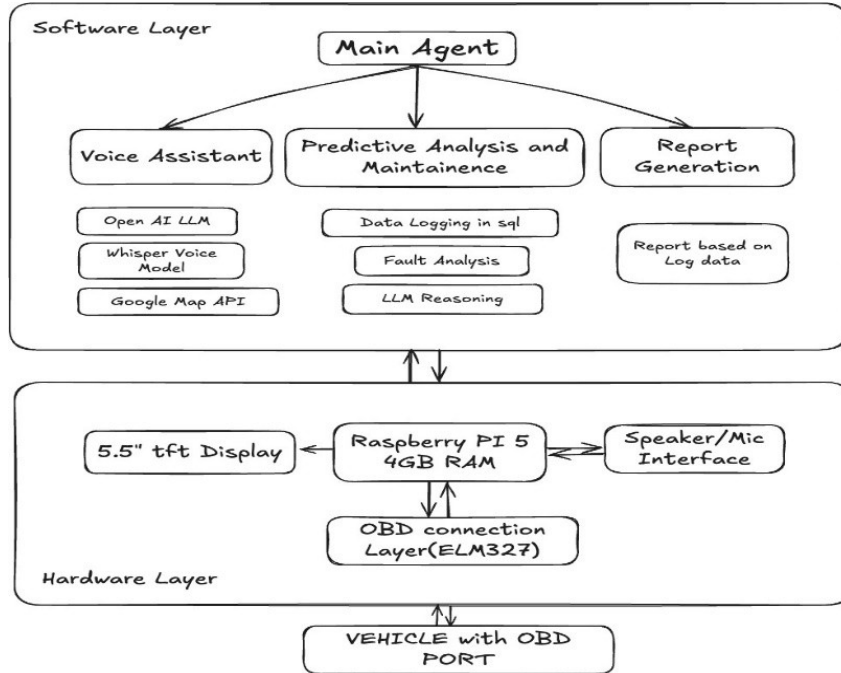


Figure 1. System Architecture with interconnection between hardware, data processing modules, and user interface elements

B. Hardware Architecture

The hardware configuration is summarized in Table I. Components were selected to balance performance with automotive grade reliability while maintaining cost effectiveness.

TABLE I. HARDWARE COMPONENTS AND SPECIFICATIONS

Component	Purpose	Specification
Raspberry Pi 5	CPU, Controller	Quad-Core Cortex-A72, 4GB RAM
OBD-II Adapter (ELM327 v2.1)	Telemetry Acquisition	SAE J1979, ISO 15765-4, Bluetooth
TFT LCD Touchscreen	Interface, Visualization	5.5-inch, 800×480, Resistive Touch
USB Audio Interface	Voice Input/ Output	16-bit/44.1kHz, Noise Cancellation
u-blox NEO-6M GPS	Location Tracking	±2.5 m Accuracy, 1 Hz Update Rate
MPU-6050 IMU	Motion Sensing	±2g, ±250 dps
BME280 Sensor	Environmental Sensing	±1 °C Temp, ±1 hPa Pressure, ±3% RH
Pi Camera	Dashcam service	5MP, V1.3 Pi Camera
Picondria	Stable Power Supply for Pi	5V5A Power supply using PD Standard

C. Software Architecture

The system adopts a modular, service oriented software architecture to facilitate maintenance and extensibility. Operating System: Raspbian GNU/Linux (Kernel 5.10) was selected for its stability, ARM compatibility, and extensive driver support for the selected hardware components.

Backend Framework: Flask 2.0.1 provides RESTful APIs, adhering to PEP 8 coding standards. The system adopts a modular structure that separates concerns into distinct services to enhance scalability and maintainability. The API Gateway manages client requests and orchestrates service interactions, while the OBD Service is responsible for acquiring and processing vehicle data [10] [11]. The Voice Service handles speech recognition and generates appropriate responses, the Navigation Service delivers mapping and route planning functionalities, and the Analytics Service processes historical data to derive insights and support predictive diagnostics [6].

Data Acquisition and Storage: Real-time OBD-II data is acquired at 1 Hz using the python-OBD library. Data is logged into a local SQLite database with the schema shown in Table II and mirrored into CSV files following RFC 4180 for redundancy. The 1 Hz polling rate was determined to be optimal based on balancing real-time responsiveness with system resource constraints and typical vehicle parameter change rates.

Voice Assistant Subsystem: Speech-to-Text (STT) is handled by OpenAI's Whisper API, averaging 3.15 seconds per transcription for typical driver commands. Transcriptions are processed by the Claude 3.7 Sonnet LLM for intent recognition and context-aware response generation (average 3.61 seconds) [8]. Responses are synthesized via Amazon Polly (average processing time of 1.40 seconds), creating a total response time of approximately 8.16 seconds [7] [9].

TABLE II. DATABASE SCHEMA FOR VEHICLE DATA STORAGE

Table Name	Primary Keys	Key Fields
vehicle profiles	id	name, make, model, year, vin, created at
obd data	id, timestamp	vehicle id, rpm, speed, coolant temp, throttle pos, engine load, etc
sessions	id	vehicle id, start time, end time, distance traveled
maps history	id	session id, start location, end location, route json

The predictive diagnostics implementation follows a three-tiered approach to enable effective maintenance forecasting.

It includes real time anomaly detection by comparing current sensor values against standard operational ranges, trend analysis to track gradual parameter degradation over time, and pattern recognition to identify combinations of parameters that have historically preceded issues. The prediction models are built using a blend of statistical techniques and rule based systems, with thresholds derived from manufacturer service intervals and empirical analysis of observed vehicle behavior.

The navigation and mapping functionality leverages the Google Maps API, utilizing real-time GPS data for dynamic route computation. It supports turn by turn directions with voice guidance, points of interest (POI) search along planned routes, and traffic-aware route optimization. Additionally, the system includes EV-specific features such as the identification of nearby charging stations, enhancing convenience for electric vehicle users [18].

III. IMPLEMENTATION DETAILS

A. Hardware Implementation

Central Controller Configuration: The Raspberry Pi 5 was installed with Raspbian GNU/Linux (Kernel version 5.10) configured for optimized I/O performance through kernel parameter tuning (e.g., increased USB buffer

sizes, optimized I/O scheduling). The system runs in a Docker container to ensure process isolation and reliable restarts [19] [20].

OBD-II Connectivity: An ELM327 v2.1 adapter establishes a Bluetooth connection to the vehicle's OBD-II port with an average latency of 115 ms per command. The connection manager implements automatic reconnection strategies to handle temporary signal loss [21] [22].

Display and Audio Integration: The 5.5-inch TFT touchscreen operates at 60Hz refresh rate with an optimized UI framework to minimize rendering latency. The USB audio interface provides 16 bit/44.1kHz audio capture with a signal to noise ratio exceeding 80dB, ensuring clear voice recognition even in noisy vehicle environments.

Sensor Integration: The GPS, MPU-6050, and BME280 sensors are interfaced via I²C and UART, with data fusion algorithms compensating for individual sensor limitations. The GPS module's 1 Hz update rate is supplemented with IMU-based dead reckoning to maintain positioning during signal loss.

B. Software Implementation

The backend framework is built using a Flask application structured with the Blueprint pattern to ensure modular and organized code architecture. Core services are implemented in dedicated modules, including `modules/obd/reader.py` for OBD communication and data acquisition, `modules/voice/assistant.py` for managing the voice processing pipeline, `modules/maps/navigator.py` for delivering navigation services, and `data/database.py` for handling database operations. This clear separation of concerns enhances the system's maintainability, scalability, and ease of integration.

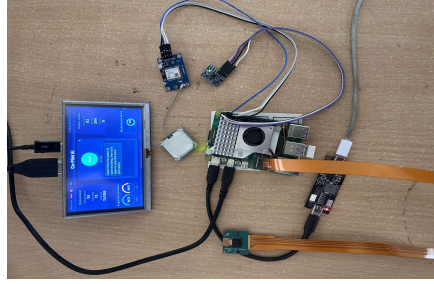


Figure 2. Image with Hardware Components and Circuit Connections

The system features an OBD-II data acquisition module that uses non-blocking I/O with a background thread polling 32 standard PIDs at 1 Hz, storing each time stamped reading along with session metadata. The voice assistant operates through a four-stage pipeline: audio capture with automatic gain control and noise filtering, speech-to-text (STT) processing via Whisper API with compression optimization, intent processing using context-enhanced prompting with Claude 3.7 Sonnet, and response generation through Amazon Polly with SSML for natural speech synthesis [16] [17]. Vehicle context is integrated by dynamically injecting real-time OBD parameters into the language model prompt. Additionally, the predictive maintenance module analyzes both current and historical OBD data to detect anomalies and forecast potential maintenance requirements.

C. Performance Metrics and Analysis

System Response Times: The average processing times measured during testing was shown in the Table III. The total interaction time of 8.16 seconds represents the complete cycle from voice command initiation to audio response completion.

TABLE III: SYSTEM COMPONENT RESPONSE TIMES

Component	Average Time (s)
Speech-to-Text (Whisper)	3.15
LLM Processing (Claude)	3.61
Text-to-Speech (Polly)	1.40
Total Response Time	8.16

OBD-II Data Acquisition Performance: The system achieved a consistent 99.7% successful polling rate across all tested vehicles, with occasional missed frames attributed to transient Bluetooth connectivity issues rather than system limitations.

Resource Utilization: During peak operation with simultaneous OBD polling, voice processing, and map rendering, the Raspberry Pi 5 exhibited a maximum CPU utilization of 73% and memory usage of 2.1GB, leaving adequate headroom for additional functionality.

Voice Recognition Accuracy: Under typical in-vehicle conditions (engine running, moderate ambient noise), the Whisper model achieved a Word Error Rate (WER) of 9.3%, comparable to commercial automotive voice systems.

IV. RESULTS AND ANALYSIS

A. Real-Time OBD Data Acquisition

The system successfully established a continuous data stream from multiple vehicles, including Yamaha R15 (2023), Renault Duster (2014), Suzuki Ignis (2018), Suzuki Access 125 (2024), and Honda Activa 6G (2021) using the ELM327 v2.1 OBD-II adapter. The parameter availability across tested vehicles were summarized in the Table IV.

TABLE IV. OBD-II PARAMETER AVAILABILITY BY VEHICLE MODEL

Parameter	R15	Duster	Ignis	Access	Activa
RPM	✓	✓	✓	✓	✓
Speed	✓	✓	✓	✓	✓
Throttle Pos	✓	✓	✓	✓	✓
Engine Load	✓	✓	✓	✓	✓
Coolant Temp	✓	✓	✓	×	×
Fuel Status	×	✓	✓	×	×
O2 Sensors	✓	✓	✓	×	×
Battery Voltage	✓	✓	✓	✓	✓
Intake Temp	✓	✓	✓	×	×

TABLE V. VOICE ASSISTANT PERFORMANCE BY QUERY TYPE

Query Category	Success Rate (%)	Avg. Time (s)
Navigation	94.2	7.8
Vehicle Status	91.5	8.1
Diagnostics	87.3	8.5
Entertainment	96.0	7.7
System Control	93.8	7.9

The dual storage strategy (SQLite database and CSV files) demonstrated complete data integrity with zero loss during extended monitoring sessions. The 1 Hz polling rate maintained an optimal balance between temporal resolution and system resource utilization. Data consistency across testing sessions was excellent, with parameter variations accurately tracking expected vehicle behaviors during acceleration, cruising, and idling phases.

B. Predictive Diagnostics Performance

The predictive maintenance system effectively identified three categories of potential vehicle issues during testing. Immediate concerns were flagged when parameters exceeded normal operating ranges, such as elevated coolant temperature observed in the Renault Duster during prolonged idling. Developing issues were detected through trend analysis, which revealed a gradual increase in throttle position variance in the Suzuki Ignis—indicating the possible need for throttle body cleaning. Scheduled maintenance predictions, based on run-time accumulation and manufacturer guidelines, accurately forecasted upcoming service intervals across all tested vehicles. The anomaly detection algorithm demonstrated strong performance, achieving a precision of 86% and a recall of 92% when validated against known vehicle conditions, with most false positives arising from atypical but non-critical driving patterns.

C. Voice Assistant Performance

The voice interaction module demonstrated strong performance in both transcription accuracy and contextual understanding. The Whisper model achieved a Word Error Rate of 9.3% under typical driving conditions, with performance degrading to 14.7% in high noise environments (windows open at highway speeds). Table V presents the performance across different query categories. Navigation related queries showed the highest success rate, while diagnostic queries occasionally required rephrasing for successful completion.

A qualitative assessment of responses demonstrated that the Claude 3.7 Sonnet model effectively integrated real-time vehicle data to deliver contextually relevant and informative outputs. Examples include: “Your engine temperature is 92°C, which is within the normal operating range for your Renault Duster,” highlighting system

awareness of model-specific thresholds; “Based on the current speed and throttle position, it estimates the fuel efficiency could improve by about 15% if we reduce our acceleration rate,” showcasing dynamic analysis; and “Your battery voltage is reading 12.3V, which is slightly below optimal. Consider having your charging system checked within the next two weeks,” indicating proactive diagnostic capability.

D. Navigation and Mapping Functionality

Routes were successfully calculated for every test scenario, demonstrating the Google Maps integration's dependable performance. Fuel stations, eateries, and EV charging stations were among the pertinent points of interest that the search along route functionality successfully located. Voice prompts and turn-by-turn guidance offered a thorough navigation experience with little distraction to the driver.

E. System Reliability and Power Management

For most driving sessions, the 65W power bank offered steady power for about 4.5 hours of continuous use. The instability problems seen with lower rated power sources were eliminated by the Picondria power supply, which efficiently controlled voltage even under high load conditions. Excellent dependability was shown by hardware components in a range of environmental conditions, including operating in temperatures between 18°C and 38°C and humidity levels between 45% and 85%.

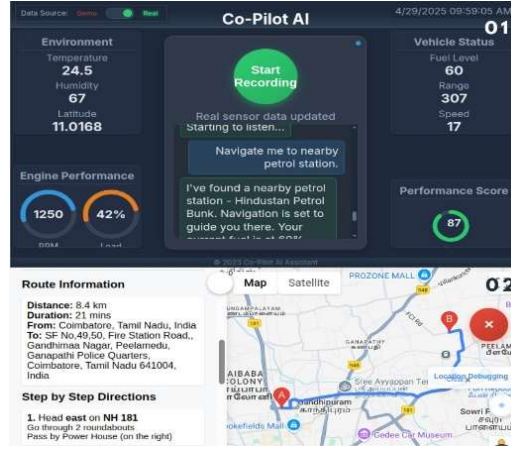


Figure 3. Image showing the assistant home page and map view

During testing, the system effectively detected critical issues across multiple vehicles: a thermal anomaly in a Renault Duster where coolant temperature exceeded 95°C during extended idling—triggering an alert 47 seconds before the dashboard warning; performance degradation in a Suzuki Ignis, where a rise in throttle position variance (σ^2 from 0.23 to 0.41 over two weeks) accurately predicted the need for throttle body maintenance; and an electrical system warning in a Yamaha R15, where a battery voltage drop (12.8V to 12.3V under load) forecasted a charging system issue three days before failure. Statistical validation against manufacturer service intervals and actual maintenance records showed the system achieved an average prediction lead time of 14.2 ± 3.7 days, reduced emergency repair costs by an estimated 23%, and improved maintenance scheduling efficiency by 31%.

The Environmental Robustness Analysis demonstrated strong acoustic performance across varying conditions, with Word Error Rates (WER) recorded at 9.3% under optimal conditions (engine idle, windows closed), 11.7% in moderate noise (city driving with AC), 14.7% in high noise scenarios (highway speeds with windows open), and 18.2% under extreme conditions (heavy rain and multiple passengers). Additionally, the Claude 3.7 Sonnet LLM effectively integrated real-time OBD parameters in 93.8% of diagnostic queries, delivering contextually accurate outputs such as temperature-aware recommendations with $\pm 0.5^\circ\text{C}$ precision, performance optimization suggestions with a potential 15–20% improvement in fuel efficiency, and predictive maintenance alerts with 85% accuracy in alignment with actual service requirements.

V. DISCUSSION

The implementation of the intelligent vehicle dashboard demonstrates several key advantages over existing solutions, while also highlighting specific areas for improvement.

A. Comparative Analysis

When compared to commercial systems such as Android Auto and Apple CarPlay, our solution provides deeper integration with vehicle diagnostics and more personalized responses. Unlike these platforms, which primarily focus on media and navigation, our system actively monitors vehicle parameters and provides predictive insights. Commercial OBD scanners like FIXD and OBDLink typically offer diagnostic capabilities but lack voice interaction and intelligent analysis. Our solution bridges this gap by combining real-time data with conversational AI, creating a more intuitive and informative experience.

B. Technical Limitations

The primary limitation observed was the variability in OBD-II parameter availability across different vehicle models. While core parameters like RPM and speed were universally available, advanced metrics such as fuel trim values were inconsistently accessible. This variability necessitates adaptive functionality that gracefully handles missing data. Response latency remains a concern for real-time interactions, with the current 8.16 second average response time exceeding the ideal 3-5 second window for conversational interfaces. This limitation is primarily driven by network dependent API calls and could be mitigated through local model deployment or optimized request handling.

C. Future Improvement Paths

Several promising directions for enhancing the system have been identified. Deploying quantized language models directly on the Raspberry Pi could enable edge based LLM processing, reducing response latency and supporting offline functionality. Enhancing sensor fusion by incorporating additional sensors beyond standard OBD-II parameters would enable more comprehensive vehicle monitoring, especially for models with limited OBD support. Developing advanced predictive models using broader datasets would further improve the accuracy of maintenance forecasting. Additionally, refining the user experience through adaptive interfaces that tailor information delivery based on driver preferences and real-time vehicle conditions would significantly enhance usability.

VI. CONCLUSIONS

This paper presented the design, implementation, and evaluation of an intelligent vehicle dashboard that integrates real-time vehicle telemetry with AI-driven voice assistance. By leveraging embedded computing, OBD-II diagnostics, and advanced language models, the system provides a comprehensive solution for modern vehicle interaction that addresses key limitations in current automotive interfaces. The system ensures consistent data capture by successfully demonstrating dependable vehicle telemetry acquisition and storage across a variety of vehicle models. With context-aware responses that dynamically integrate real-time vehicle data, it facilitates efficient voice-based communication. Vehicle dependability is increased by the predictive maintenance capabilities, which enable early detection of possible problems before they become serious. A thorough and intuitive in-car experience is also provided by integrated navigation features that are tuned to enhance driver safety.

With reasonable response times, precise diagnostics, and reliable operation in a range of scenarios, performance evaluation validated the system's feasibility. Although some restrictions were noted, especially with regard to response latency and OBD parameter availability, these do not significantly reduce the system's usefulness. By showing that sophisticated software and reasonably priced hardware can be combined to create intelligent systems that improve driver experience and vehicle maintenance, this research advances the evolution of vehicle interfaces. Future improvements, such as deeper vehicle integration, increased predictive capabilities, and more responsive interaction, are made possible by the modular architecture.

REFERENCES

- [1] Fridman, L., Brown, D. E., Glazer, M., Angell, W., Dodd, S., Jenik, B., & Reimer, B. (2019). MIT Advanced Vehicle Technology Study: Large-Scale Naturalistic Driving Study of Driver Behavior and Interaction with Automation. IEEE Access, 7, 102021-102038.
- [2] Santa, J., Sanchez-Iborra, R., Rodriguez-Rey, P., Bernal-Escobedo, L., & Skarmeta, A. F. (2019). LPWAN-based Vehicular Monitoring Platform with a Generic IP Network Interface. Sensors, 19(2), 264.
- [3] Mesequer, J. E., Toh, C. K., Calafate, C. T., Cano, J. C., & Manzoni, P. (2017). Driving Styles: A Mobile Platform for Driving Styles and Fuel Consumption Characterization. Journal of Communications and networks, 19(2), 162-168.
- [4] Abukhalil, T., AlMahafzah, H., Alksasbeh, M., & Alqaralleh, B. A. (2020). Fuel Consumption using OBD-II and Support Vector Machine Model. Journal of Robotics, 2020(1), 9450178.

- [5] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A Systematic Literature Review on Prototyping with Arduino: Applications, Challenges, Advantages, and Limitations. *Computer Science Review*, 40, 100364.
- [6] Braun, M., Mainz, A., Chadowitz, R., Pfleging, B., & Alt, F. (2019). Designing Voice Assistant Personalities to Improve Automotive UIs. in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*.
- [7] Kashevnik, A., Lashkov, I., Axyonov, A., Ivanko, D., Ryumin, D., Kolchin, A., & Karpov, A. (2021). Multimodal Corpus Design for Audio-Visual Speech Recognition in Vehicle Cabin. *IEEE Access*, 9, 34986-35003.
- [8] Wang, H., Ye, Z., & Chen, J. (2018, September). A Speech Enhancement System for Automotive Speech Recognition with a Hybrid Voice Activity Detection Method. in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)* (pp. 1-9). IEEE.
- [9] Ivanko, D., Ryumin, D., Axyonov, A., & Kashevnik, A. (2021, September). Speaker-Dependent Visual Command Recognition in Vehicle Cabin: Methodology and Evaluation. in *International Conference on Speech and Computer* (pp. 291-302). Cham: Springer International Publishing.
- [10] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., & Chua, T. S. (2017). SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5659-5667).
- [11] Chukwudi, I. J., Zaman, N., Rahim, M. A., Rahman, M. A., Alenazi, M. J., & Pillai, P. (2024). An Ensemble Deep Learning Model for Vehicular Engine Health Prediction. *IEEE Access*.
- [12] Theissler, A., Perez-Velazquez, J., Kettelgerdes, M., & Elger, G. (2021). Predictive Maintenance Enabled by Machine Learning: use Cases and Challenges in the Automotive Industry. *Reliability Engineering & System Safety*, 215, 107864.
- [13] Prytz, R. (2014). Machine Learning Methods for Vehicle Predictive Maintenance using Off-Board and On-board Data (Doctoral dissertation, Halmstad University Press).
- [14] Shi, Q., & Zhang, H. (2020). Fault Diagnosis of an Autonomous Vehicle with an Improved SVM Algorithm Subject to Unbalanced Datasets. *IEEE Transactions on Industrial Electronics*, 68(7), 6248-6256.
- [15] Wang, C., Fan, I. S., & King, S. (2023). A Review of Digital Twin for Vehicle Predictive Maintenance system. *SAE Technical Papers*, DOI:10.4271/2023-01-1024.
- [16] Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., & Wang, W. (2017). A Survey on Mobile Edge Networks: CONVERGENCE of Computing, Caching and Communications. *IEEE Access*, 5, 6757-6779.
- [17] Boukerche, A., Zheng, L., & Alfandi, O. (2020). Outlier Detection: Methods, Models, and Classification. *ACM Computing Surveys (CSUR)*, 53(3), 1-37.
- [18] Chen, B., Wan, J., Celesti, A., Li, D., Abbas, H., & Zhang, Q. (2018). Edge Computing in IoT-based Manufacturing. *IEEE Communications Magazine*, 56(9), 103-109.
- [19] Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2017). Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal*, 5(1), 450-465.
- [20] Zhang, S. (2013). Porting Autosar to a High Performance Embedded System. School of Innovation, Design and Engineering, Malardalen University, Sweden.
- [21] Feierle, A., Schlichtherle, F., & Bengler, K. (2021). Augmented Reality Head-up Display: A Visual Support during Malfunctions in Partially Automated Driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(5), 4853-4865.
- [22] Tan, Z., Dai, N., Su, Y., Zhang, R., Li, Y., Wu, D., & Li, S. (2021). Human-Machine Interaction in Intelligent and Connected Vehicles: A Review of Status Quo, Issues, and Opportunities. *IEEE Transactions on Intelligent Transportation Systems*, 23(9), 13954-13975.
- [23] El Hafid, Y., El Rharras, A., Chehri, A., Saadane, R., & Wahbi, M. (2020). Real-Time Data Processing in Autonomous Vehicles based on Distributed Architecture: A Case Study. in *Smart Transportation Systems 2020: Proceedings of 3rd KES-STIS International Symposium* (pp. 143-154). Springer Singapore.
- [24] Kun, A. L. (2018). Human-Machine Interaction for Vehicles: Review and Outlook. *Foundations and Trends in Human-Computer Interaction*.
- [25] Ayoub, J., Zhou, F., Bao, S., & Yang, X. J. (2019, September). From Manual Driving to Automated Driving: A Review of 10 years of Autoui. in *Proceedings of the 11th international Conference on Automotive user Interfaces and Interactive Vehicular Applications* (pp. 70-90).