Research Square

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

Chariot: An Edge AI Vehicle Intelligence Framework with Real-Time Diagnostics and Human-Centered Assistance

Shriyash Sawant

shriyashsantoshsawant@gmail.com

MIT-ADT

Varad Khot

MIT-ADT

Sehejveer Singh

MIT-ADT

Pranav Aghade

MIT-ADT

Rameez Shamalik

MIT-ADT

Article

Keywords: Vehicle intelligence, OBD-II, predictive maintenance, automotive AI, driver assistance, real-time systems, explainable AI

Posted Date: June 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9521537/v1

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

Additional Declarations: No competing interests reported.

Chariot: An Edge AI Vehicle Intelligence Framework with Real-Time Diagnostics and Human-Centered Assistance

Shriyash Sawant
Department of SOC(Cloud Computing)
MIT SOC, Loni Kalbhor, Pune, India
shriyashsantoshsawant@gmail.com

Varad Khot
Department of SOC(Cloud Computing)
MIT SOC, Loni Kalbhor, Pune, India
varad8v18@gmail.com

Sehejveer Singh
Department of SOC(Cloud Computing)
MIT SOC , Loni Kalbhor, Pune, India
sehajveer885@gmail.com

Pranav Aghade
Department of SOC(Cloud Computing)
MIT SOC, Loni Kalbhor, Pune, India
pranavaghade77@gmail.com

Rameez Shamalik
Department of CSE(Cloud Computing)
MIT SOC, Loni Kalbhor, Pune,India
rameez.shamalik@mituniversity.edu.in

Abstract*— Modern vehicle telematics systems often provide raw sensor data or fault codes but lack integrated real-time reasoning, predictive analytics, and driver-centric interaction. This paper introduces Chariot, a vehicle intelligence platform that synergizes OBD-II telemetry acquisition, diagnostic analytics, selective machine learning, and multimodal user interfaces within a unified architecture. The system employs tiered telemetry scheduling with batched PID polling, an in-memory vehicle state buffer, and a Vehicle Intelligence Orchestrator (VIO) that synthesizes diagnostics, behavioral analysis, and predictive outputs into prioritized driver insights. A dual-path voice assistant enables low-latency execution for routine commands while supporting structured reasoning for complex requests. Experimental evaluation demonstrated a median command latency of 105.78 ms, an intent-classification F1-score of 0.96, and approximately 90% success for predefined assistant commands. For fuel-efficiency estimation, the best regression model achieved a mean absolute percentage error of 11.2%, while the predictive maintenance module obtained an AUC of 0.650. The results confirm that the proposed architecture can deliver responsive vehicle intelligence on commodity hardware.

Keywords*— Vehicle intelligence, OBD-II, predictive maintenance, automotive AI, driver assistance, real-time systems, explainable AI.

INTRODUCTION

Modern vehicles continuously generate telemetry through Electronic Control Units (ECUs) and standardized OBD-II interfaces, exposing parameters such as engine load, thermal conditions, fuel consumption, and Diagnostic Trouble Codes (DTCs). Despite the widespread accessibility of this data, most consumer diagnostic platforms remain limited to fault-code retrieval and basic dashboard visualization. Consequently, drivers frequently receive fragmented technical information without contextual interpretation, predictive insight, or operational guidance. Existing systems also struggle to support real-time reasoning under constrained edge-computing environments, particularly when simultaneous diagnostics, driver monitoring, navigation, and conversational interaction are required.

Chariot was developed as a multimodal vehicular intelligence framework designed to unify these disconnected capabilities within a single real-time architecture. The platform combines deterministic safety logic, physics-informed diagnostics, selective machine-learning inference, and conversational reasoning to support proactive vehicle assistance rather than passive telemetry observation. Its Integrated Decision Hierarchy (IDH) separates safety-critical control paths from probabilistic AI inference, allowing predictive maintenance and behavioral analytics to operate without compromising deterministic system behavior.

In addition to telemetry-driven diagnostics, the framework incorporates a Driver Monitoring System (DMS) and a contextual reasoning layer capable of transforming sensor streams, subsystem states, and environmental conditions into human-readable recommendations. Through this integration, the platform functions as an intelligent automotive co-pilot capable of assisting drivers with vehicle health interpretation, proactive alerts, and contextual decision support during real-time operation.

LITERATURE SURVEY

The evolution of automotive intelligence has shifted from isolated diagnostic tools toward integrated and data-driven ecosystems [1]. Traditional diagnostics have progressed beyond fault-code retrieval into predictive maintenance and reliability modeling [2]. Recent studies have validated methods such as survival analysis, ensemble learning, and remaining useful life estimation for forecasting component degradation with high predictive confidence [2]. However, a gap remains in real-time subsystem-level analytics for domains such as suspension wear, catalytic efficiency, and dynamic thermal stress, particularly under edge-computing constraints.

Recent advances in multimodal driver-assistance systems emphasize reducing alarm fatigue while improving situational awareness [8]. Deep-learning architectures for collision detection, distracted-driving recognition, and drowsiness monitoring have demonstrated strong benchmark performance [1], [8], [6]. Despite these gains, many solutions still lack a conversational reasoning layer capable of translating technical outputs into actionable and human-centered guidance [3]. Voice-first interaction is increasingly important for minimizing distraction while managing complex telemetry streams during driving [8], [9].

A central challenge in automotive AI remains the black-box nature of many deep learning systems [4]. Current scholarship highlights the importance of Explainable AI (XAI), where predictions are accompanied by interpretable reasoning and confidence measures [4]. In parallel, Retrieval-Augmented Generation (RAG) has emerged as a promising approach for grounding responses in vehicle manuals, maintenance records, and manufacturer documentation, thereby improving trust and factual reliability [3][4].

Navigation research has similarly evolved beyond shortest-path routing; crowdsourced traffic systems and context-aware route engines now incorporate congestion, weather, and road events to optimize travel efficiency. Emerging studies further explore coupling route planning with vehicle-health awareness to reduce strain on degraded subsystems [1], [7]. Advanced telematics additionally require persistent memory mechanisms that capture long-term driving behavior and evolving vehicle states, enabling personalized assistance that adapts over the operational life of the vehicle [8].

Research Gap and Contribution

Current connected-vehicle ecosystems typically separate conversational assistance, diagnostics, safety monitoring, and predictive analytics into isolated subsystems. Consumer voice assistants provide natural-language interaction but possess little awareness of vehicle-state conditions, mechanical risk, or operational safety. In contrast, professional diagnostic platforms expose detailed telemetry and fault-code analysis yet often require technical expertise and provide minimal contextual guidance for everyday drivers. As modern vehicles generate increasingly complex telemetry streams, the absence of a unified, interpretable, and real-time vehicular intelligence framework becomes more pronounced, particularly in edge-computing environments where latency, reliability, and safety constraints must coexist.

Chariot (DriveSmarter) was developed as a multimodal vehicular co-pilot architecture that combines deterministic control logic, physics-informed diagnostics, selective machine-learning inference, and conversational reasoning within a single operational framework. The proposed Integrated Decision Hierarchy (IDH) separates safety-critical rule enforcement from adaptive probabilistic reasoning, ensuring that emergency actions, threshold validation, stale-data handling, and command interlocks remain deterministic and verifiable. Statistical and machine-learning components are selectively employed for predictive maintenance, anomaly scoring, fuel-efficiency estimation, behavioral analysis, and trend forecasting without replacing the deterministic diagnostic core.

Beyond conventional OBD-based monitoring, the platform performs subsystem-level analysis across engine, braking, suspension, exhaust, transmission, and electrical domains using derived indicators such as thermal stress, combustion imbalance, volumetric efficiency, and brake-specific fuel consumption. A complementary kinematic analysis module evaluates acceleration and rotational telemetry to identify unsafe driving behavior, impact events, and aggressive maneuvering patterns. The framework further supports both internal combustion engine (ICE) and electric vehicle (EV) environments through energy-agnostic health assessment mechanisms, including battery state-of-health estimation and operational efficiency analysis.

Additional contributions include a multimodal Driver Monitoring System (DMS) that integrates computer-vision-based vigilance assessment with vehicle-state analytics to detect distraction, drowsiness, and unsafe driver behavior in real time. The platform also incorporates Retrieval-Augmented Generation (RAG) using indexed vehicle manuals and service documentation to provide contextual troubleshooting support and human-readable maintenance guidance. Experimental benchmarking on commodity edge hardware demonstrates the feasibility of deploying low-latency vehicular intelligence, conversational interaction, and predictive analytics within constrained real-time automotive environments.

Multi-Modal Intelligence Architecture

Chariot platform is designed as a multi-modal automotive intelligence framework in which safety monitoring, predictive diagnostics, conversational reasoning, and contextual navigation operate as coordinated subsystems rather than isolated modules. This design extends conventional vehicle telematics by combining onboard telemetry, computer vision, machine learning, and natural language interaction within a unified real-time architecture.

A. **Active Safety via Driver Monitoring**

The system incorporates a vision-based Driver Monitoring System (DMS) for real-time vigilance assessment. Facial landmark tracking and behavioral cues are used to detect drowsiness, distraction, gaze deviation, and unsafe device usage during driving. When elevated risk conditions are identified, the platform can issue graded interventions through visual alerts, audio warnings, and haptic notifications. This subsystem shifts the platform from passive diagnostics toward active driver safety assistance.

B. **Physics-Informed Predictive Diagnostics**

Beyond conventional fault-code retrieval, Chariot applies physics-informed analytics to estimate latent mechanical conditions that are not directly observable from standard ECU outputs. Derived indicators such as thermal stress, volumetric efficiency, combustion imbalance, torque trends, and brake specific fuel consumption are used to detect degradation before critical failures occur. This enables early maintenance recommendations and more interpretable health assessments.

C. **Semantic Intent Orchestration**

The interaction layer uses a dual-path reasoning pipeline optimized for low-latency vehicle environments. Simple and deterministic requests are handled through a fast execution path, while complex or multi-intent queries are processed

through a local language-model reasoning engine with semantic caching and task decomposition. This enables compound requests such as simultaneous navigation, diagnostics, and trip-status retrieval without interrupting real-time responsiveness.

D. **Context-Aware Mobility and Cost Guidance**
The navigation subsystem integrates routing services with live vehicle state, nearby points of interest, and trip-economics estimation. If critical faults or low-fuel conditions are detected, the system can recommend service centers, fuel stations, or alternate routes. By combining route planning with maintenance urgency and operational cost estimation, the platform extends navigation into a decision-support service.

METHODOLOGY

The Chariot platform adopts a multimodal hybrid architecture that combines deterministic engineering models, rule-based thresholds, statistical trend analysis, and selective machine-learning components. Core diagnostics do not rely on large-scale supervised training; instead, vehicle health is inferred using physics-based indicators such as thermal stress, combustion imbalance, volumetric efficiency, and operating-threshold deviations derived from live telemetry acquired through an ELM327-based OBD-II interface. To preserve communication efficiency, the telemetry layer employs tiered polling and batched parameter requests, ensuring that high-priority signals are refreshed at higher rates while lower-priority data is sampled less frequently. In parallel, a Driver Monitoring System (DMS) extends situational awareness through computer-vision analysis of eye closure, gaze behavior, and distraction cues.

Machine-learning modules are used for predictive analytics, trend estimation, anomaly scoring, and intent classification, while higher-level explanations are generated through the Hermes reasoning layer, which executes quantized GGUF language models via llama.cpp to provide low-latency natural-language interaction. The platform further incorporates a Retrieval-Augmented Generation (RAG) pipeline backed by a vector database containing indexed technical manuals and service knowledge for contextual troubleshooting support. Experimental benchmarking was conducted on an AMD Ryzen 7 7800X3D workstation with an NVIDIA RTX 3050 GPU. Results indicate that the proposed architecture supports interpretable, privacy-preserving, and responsive multimodal intelligence on constrained edge hardware.

Ethical Considerations

This study involved observational evaluation of driver behavior using computer-vision-based monitoring techniques for research and prototype validation purposes. All methods were carried out in accordance with relevant guidelines and regulations. Participation in the Driver Monitoring System (DMS) evaluation was voluntary, and informed consent was obtained from all participants prior to data collection. No personally identifiable information, facial identity records, or sensitive biometric data were stored or shared.
The experiments did not involve clinical procedures, medical intervention, or high-risk human-subject testing. The study was conducted solely for engineering system evaluation within a controlled research environment. Based on the observational and non-invasive nature of this research, the requirement for formal ethical approval was waived by the Institutional Ethics Committee at MIT ADT University.

TECHNICAL ARCHITECTURE

A. Overview

The Chariot system consists of onboard components (vehicle sensors, OBD adapter), in-vehicle processing (telemetry scheduler, VIO), and user interfaces (dashboard, voice assistant). The high-level flow is:

OBD ECU/Sensors → Telemetry Service → In-Memory Buffer → VIO (Analytics) → Insight Dispatcher → UX/API → Persistence.

In practice we implement this as follows:

OBD ECU / Sensors: The vehicle's engine control unit (ECU) and sensors generate raw data (speed, RPM, engine load, temperatures, etc.).
OBDService: A tiered PID reader polls the ECU. Tier 1 signals (critical metrics) are read at ~10 Hz, while lower tiers (2–5) are "hitchhiked" opportunistically. Up to 3 OBD-II PIDs are batched per request to minimize latency.
Telemetry Buffer (VehicleStateManager): The latest sensor values are stored in-memory, providing a consistent snapshot of vehicle state for analysis.
Vehicle Intelligence Orchestrator (VIO): The core logic unit, comprising submodules:

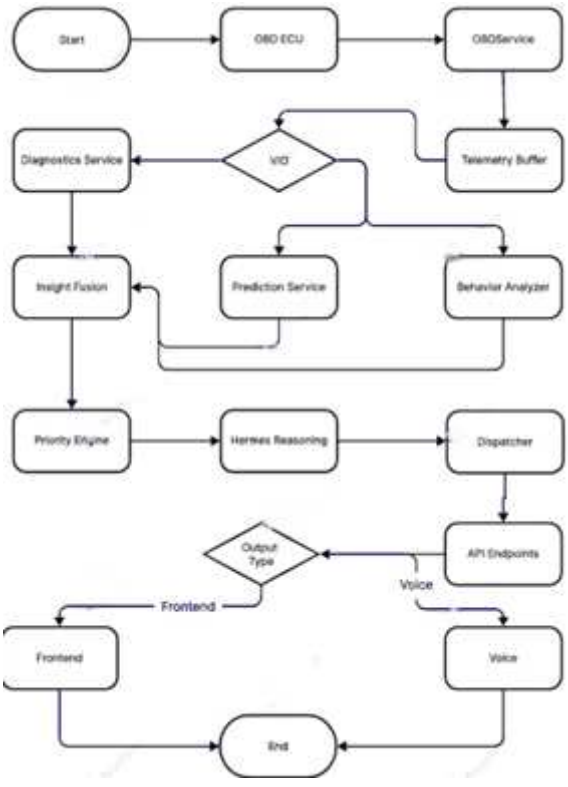


Fig. 1. Overall System Architecture

Diagnostics Engine – computes health scores and detects faults (MIL/DTCs).
Driving Pattern Analyzer – monitors behaviors (e.g. acceleration, braking).
Sharp Turn Analyzer – detects excessive cornering.
ML Prediction Models – runs predictive models (RUL, failure, risk).
Insight Fusion Engine – combines signals into higher-level insights.

Priority Engine – assigns priority scores to each insight.

Hermes AI Reasoning – generates human-readable explanations.

Insight Dispatcher: Dispatches prioritized insights to output channels (API, UI, voice).

Gut Instinct Engine (ThoughtCards): A mechanism for presenting key insights as cards or messages.

API Layer: REST/WS endpoints (e.g. /debug/vio-last, /diagnostics/realtime) expose system status and analytics.

Frontend: User interface including a web dashboard, trip summary reports, and a voice assistant.

Async Persistence: Telemetry and events are logged asynchronously in a time-series DB or SQLite for later analysis or model training.

This layered architecture decouples data collection from analysis: raw data is ingested by the OBDService and buffered, while the VIO operates on the buffered state. Vehicle-to-analysis communication can use publisher-subscriber patterns (e.g., MQTT) to further decouple components [5].

Level 1 (Detailed):
OBD ECU → OBDService → Telemetry Buffer → VIO [Diagnostics Service, Prediction Service, Behavior Analyzer] → Insight Fusion → Priority Engine → Hermes Reasoning → Dispatcher → API Endpoints → Frontend + Voice.

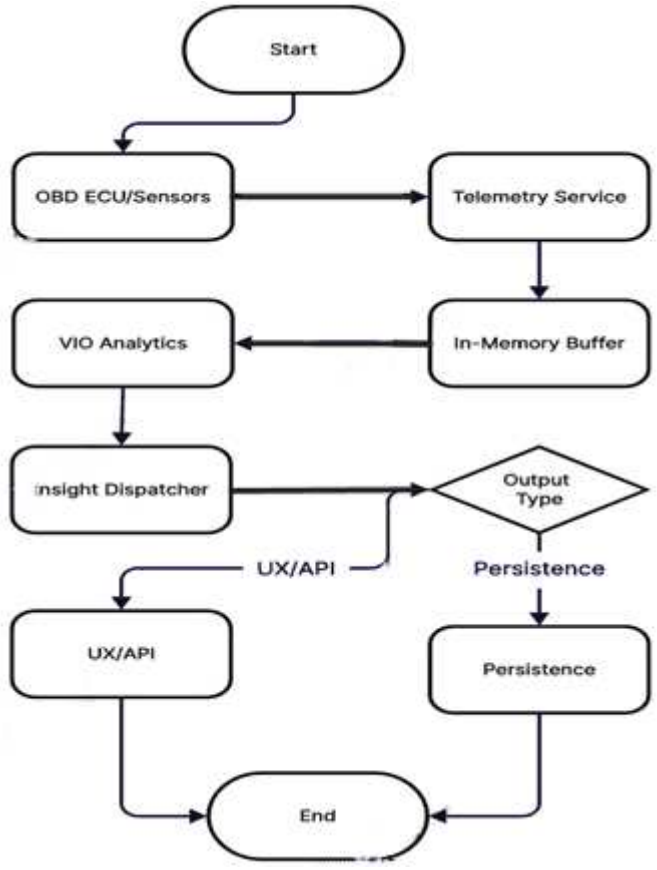


Fig. 2. Diagnostic System Architecture

These data flows illustrate that telemetry is collected onboard, sent into the VIO pipeline for analysis, and the resulting insights are delivered to user interfaces.

Every loop cycle ingests fresh sensor data and propagates it through diagnostics and AI models in real time, enabling immediate alerts (e.g., "Check engine: high coolant temperature") and voice prompts.

B. TELEMETRY SYSTEM

The Telemetry subsystem handles OBD-II communication and data buffering.

Architecture:
- OBD Adapter: We use an ELM327-based interface (via Bluetooth/Wi-Fi) to send AT-command requests to the vehicle's OBD port.
- PID Scheduling: Critical parameters (Tier 1, e.g. speed, RPM, oil temp) are polled at high frequency (≈10 Hz). Less critical data (Tiers 2–5) are polled opportunistically when the bus is free.
- Batching: The OBDService groups up to 3 PIDs per OBD request. Batching reduces protocol overhead and improves latency.
- Data Flow: Responses are parsed into physical values (e.g. RPM, speed). These are stored in an in-memory latest_data structure (VehicleStateManager) for immediate access.
-API: Latest values can be streamed to the dashboard via Server-Sent Events (SSE) or polled via REST.

Key Features:
Tier 1 → 10 Hz: Ensures low-latency monitoring of vital signals.
Tier 2–5 Hitchhiking: Lower-priority metrics piggyback on idle bus time.
3-PID Batching: Example: A single CAN frame requests RPM, speed, and coolant temp simultaneously, reducing cycles.

Telemetry Formula Example:

Fuel consumption can be estimated from Mass Air Flow (MAF):

$$Fuel\ Rate\ \left(\frac{L}{h}\right) = \frac{MAF\left(\frac{kg}{s}\right)\times 3600}{AFR\times \rho_{fuel}} \qquad (1)$$

(where AFR is air-fuel ratio and ρfuel is fuel density). These calculations enable more interpretable metrics for efficiency.

C. Diagnostics System

The diagnostics subsystem of Chariot processes real-time OBD-II telemetry to assess vehicle health, detect faults, and generate maintenance insights. Signals such as engine speed, coolant temperature, vehicle speed, mass air flow, and intake pressure are acquired through the telemetry interface and synchronized in a temporary state buffer to ensure consistent analysis. The analytics layer applies engineering formulas, physics-based estimators, and rule-based checks to derive indicators such as thermal stress, combustion efficiency, torque, brake power, and brake specific fuel consumption. In parallel, the system monitors Malfunction Indicator Lamp (MIL) status and Diagnostic Trouble Codes (DTCs) to detect ECU-reported failures and early warning conditions.

Subsystem health is computed using an additive penalty model

$$Health = 100 - \sum_{i=1}^{n} P_i \qquad (2)$$

where P_i represents the penalty associated with abnormal operating conditions. The overall vehicle health score is obtained through weighted fusion of subsystem scores:

$$H_{overall} = 0.35H_{eng} + 0.25H_{bat} + 0.20H_{brk} + 0.20H_{trn} \quad (3)$$

where H_{eng}, H_{bat}, H_{brk}, and H_{trn}represent the health of the engine, battery, braking, and transmission systems, respectively.

The resulting diagnostic outputs are forwarded to the Vehicle Intelligence Orchestrator, where they are fused with anomaly-detection models, predictive maintenance estimators, and behavioral analytics to generate higher-level insights. These insights are subsequently transformed into human-readable dashboard notifications and voice guidance for the driver. To improve robustness, the diagnostics subsystem employs freshness constraints for time-sensitive signals, warm-up hysteresis during cold-start operation, and context-aware escalation policies for high-severity faults. Collectively, this architecture enables Chariot to provide responsive, explainable, and practically deployable vehicle diagnostics in real time.

To estimate maintenance urgency, a Maintenance Severity Index (MSI) is defined as:

$$MSI = 0.4F_{tds} + 0.3T_{si} + 20N_{dtc} \quad (4)$$

where F_{tds}denotes fuel-trim deviation severity, T_{si}represents thermal stress intensity, and $N_{dtc}$is the number of active diagnostic trouble codes.

D. *Vehicle Intelligence Orchestrator (VIO) – AI Layer*

The Vehicle Intelligence Orchestrator (VIO) functions as the central decision and reasoning layer of Chariot, transforming distributed telemetry signals into prioritized vehicle insights and driver-facing actions. It integrates four principal components: an Insight Fusion Engine, a Priority Engine, an interpretive reasoning module, and a dispatch layer. The Fusion Engine performs cross-subsystem correlation by combining diagnostics, anomaly scores, predictive maintenance outputs, and behavioral indicators into higher-order vehicle states. For example, concurrent increases in engine speed, load, and coolant temperature may be fused into an engine-stress condition rather than presented as isolated sensor abnormalities. This abstraction reduces alert fragmentation and improves driver comprehension

The Priority Engine ranks generated insights according to operational urgency using a weighted risk model that considers estimated risk severity, confidence of detection, prediction horizon, and contextual relevance. The priority score is defined as:

$$P = 0.4R + 0.3C + 0.2H + 0.1X \quad (5)$$

where S denotes risk severity, C represents confidence, H is the temporal horizon of the predicted event, and X captures contextual factors such as road environment or driver state. To improve robustness against transient noise, persistent or worsening conditions may receive temporal escalation bonuses across consecutive telemetry cycles, thereby increasing the priority of sustained faults while suppressing isolated spikes.

To preserve responsiveness under real-time constraints, the VIO employs a multi-path inference strategy. High-confidence events are processed through a Fast Path that uses deterministic heuristics to generate immediate alerts. Medium-confidence events are routed to a Deep Path, where asynchronous language-model reasoning provides richer diagnostic explanations without blocking the main control loop. Low-priority routine states follow a Fast Path in which no expensive reasoning stage is invoked. This adaptive execution model enables responsive operation within telemetry loops operating at approximately 1–10 Hz while reserving computational resources for complex cases.

A dedicated safety override mechanism governs life-critical situations. When the computed severity exceeds a predefined emergency threshold, the system bypasses nonessential reasoning stages and immediately issues a high-priority visual and auditory warning to the driver. This ensures that urgent notifications are not delayed by inference latency. After prioritization, the dispatch layer publishes structured outputs to the dashboard, voice interface, and logging services. Collectively, the VIO provides a safety-first orchestration framework that balances deterministic control, probabilistic intelligence, and explainable human interaction in a deployable automotive environment.

E. *Driver Risk and Safety Model*

The Chariot platform incorporates a driver risk assessment module that estimates driving safety using behavioral events and vehicle stress indicators. Instead of relying only on single threshold violations, the system aggregates driving patterns over time to generate an interpretable safety score that can be displayed on the dashboard and used for real-time intervention. The model evaluates events such as hard acceleration, harsh braking, sharp turning, engine stress, and abnormal engine speed spikes, which are associated with increased accident risk, reduced comfort, and accelerated vehicle wear. To avoid false detections during idle conditions, a stationary-state gate disables behavior scoring when vehicle speed remains below a minimum threshold.

The driver risk score is calculated using a weighted event-penalty model:

$$Score = 100 - \sum_{i=1}^{n} (e_i w_i) \quad (6)$$

where $e_i$is the frequency of the i^{th}detected event and $w_i$is its severity weight. Representative weights include hard acceleration (6), harsh braking (6), sharp turn (5), engine stress (7), and engine speed spike (4). Higher scores indicate safer driving behavior, while lower scores represent progressively riskier operation. For example, one hard acceleration event and one sharp turn reduce the score to 89, indicating mild aggressive driving behavior.

To improve driver awareness, the risk model is linked to a multi-sensory alert system. Moderate-risk events generate dashboard notifications or spoken coaching messages, whereas critical conditions such as microsleep detection, distracted phone use, or sustained dangerous driving trigger immediate visual, auditory, and haptic warnings. For simplified interpretation, scores from 80–100 indicate safe driving, 60–79 normal behavior, 40–59 elevated risk, and values below 40 dangerous operation requiring corrective action. This framework extends traditional telematics scoring into an active real-time driver safety system.

F. Machine Learning Models

Chariot integrates several ML models trained on vehicle data:

- Random Forest (RUL): Estimates the Remaining Useful Life of components (e.g. battery life, engine wear).
- XGBoost (Failure Probability): Predicts the probability of a subsystem failure in the near future.
- Logistic Regression (Risk Prediction): Classifies high-vs-low risk driving behavior (as a complement to the rule-based score).
- K-Means (Behavior Clustering): Identifies patterns in driving style by clustering telemetry features (normal, aggressive, etc.).
- LSTM RNN (Forecasting): Forecasts future telemetry values (e.g. engine temperature trend) for anomaly detection.
- Isolation Forest (Anomaly Detection): Flags unusual sensor readings indicative of novel faults.

For example, the failure probability from XGBoost is computed as:

$$P(failure) = \sigma(w \cdot x + b) \qquad (7)$$

where σ is the sigmoid function (as in binary classification). Remaining Useful Life (RUL) predictions might use regression on sensor trends and degradation models. These models are trained offline on historical data and then applied in real time within VIO.

G. Voice Assistant, Memory, and Personalization

The Chariot platform incorporates a multimodal voice assistant optimized for low-latency in-vehicle interaction while preserving contextual continuity and user-specific personalization. Spoken commands are first converted to text and processed through a bifurcated intent-classification pipeline. Simple and deterministic requests, including greetings, door controls, and vehicle status checks, are routed through a Fast Path that bypasses large language model inference and directly invokes predefined backend functions. More complex or ambiguous requests are processed through an Intelligent Path, where a local reasoning engine performs semantic understanding, vector-cache retrieval, and structured task planning. Multi-intent commands, such as "Navigate to office and check car health," are decomposed into parallel subtasks for navigation and diagnostics, executed concurrently through the system tool registry, and merged into a unified contextual response. Final outputs are delivered through dashboard notifications and asynchronous text-to-speech synthesis, enabling practical hands-free assistance during driving.

To improve efficiency and interaction quality, the assistant employs a tiered memory architecture consisting of short-term context, semantic persistence, and long-term knowledge retrieval. Short-term conversational memory is maintained through a rolling context buffer that stores recent dialogue turns, allowing natural follow-up queries without repeated user input. Frequently repeated or previously solved requests are accelerated through a semantic intent cache based on vector similarity search, where prior command-reasoning-action mappings can be retrieved and executed with minimal latency. Persistent personalization is achieved through a user preference registry that stores configurable parameters such as predictive-alert sensitivity, reasoning depth, and telemetry sampling behavior. In addition, long-term technical guidance is supported through retrieval-augmented generation using indexed vehicle manuals and service documentation. This multi-layer design enables the assistant to remain responsive, context-aware, personalized, and computationally efficient in real-time automotive environments.

H. Navigation System

The navigation module of Chariot provides context-aware route guidance by integrating spatial services with live vehicle and environmental data (POIs, Weather API, Destination Ratings, etc.). Destination requests received through the dashboard or voice assistant are processed by a mapping service that retrieves routes, geolocation data, and nearby points of interest such as fuel stations or repair centers. A route engine then computes the optimal path while considering traffic conditions, weather impact, and estimated travel time. Before trip initiation, the system can incorporate a vehicle health pre-check using current diagnostic scores to determine whether maintenance warnings or active faults should influence route recommendations. Final outputs are delivered as turn-by-turn dashboard guidance and spoken navigation prompts.

Trip economics are estimated through a predictive cost model. Fuel cost is calculated as:

$$C_f = \left(\frac{D}{\eta}\right) P_f \qquad (8)$$

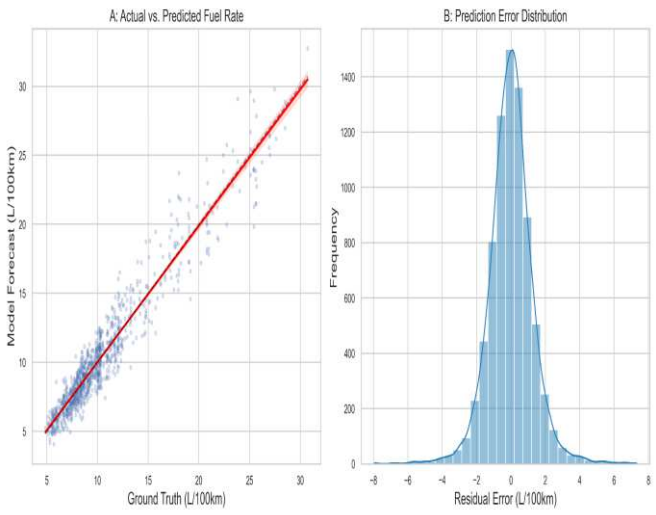
where D is trip distance, η is fuel efficiency, and P_f is fuel price per liter. The total trip cost may further include maintenance and wear penalties:

$$C_t = C_f + C_m + C_w \qquad (9)$$

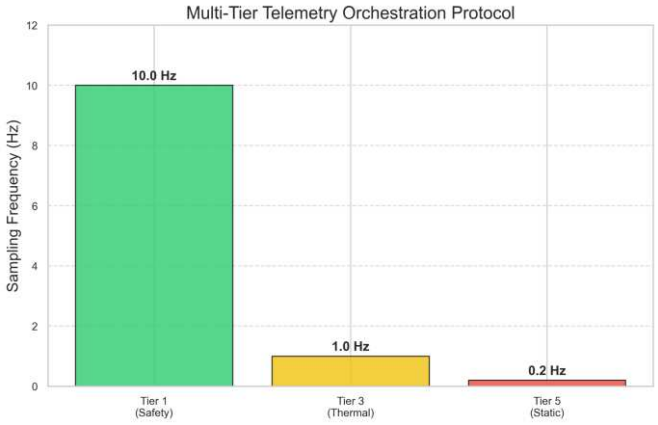
where C_m and C_w denote estimated maintenance and wear costs, respectively. By combining routing intelligence with vehicle condition awareness and economic forecasting, the proposed module extends conventional navigation into a practical automotive decision-support system.

I. Illustrative Graphs (Data Visualization)

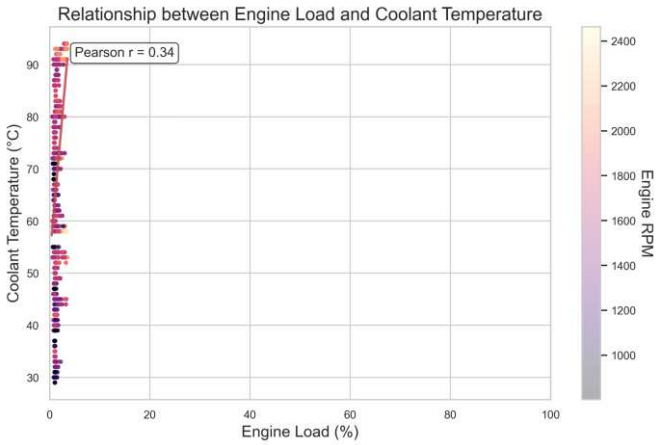
For reporting and analysis, Chariot can produce charts such as:



Fuel-consumption model evaluation: (a) actual versus predicted values and (b) residual error distribution.



Tiered telemetry polling frequencies used to prioritize safety-critical signals while minimizing bus load.



Correlation between engine load and coolant temperature under varying RPM conditions.

J. *Evaluation Metrics*

Model performance is assessed using standard supervised-learning metrics. For classification tasks such as risky-driving detection, fault identification, and intent recognition, accuracy, precision, recall, and F1-score are used. Accuracy is defined as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Precision and recall are given by:

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

and the harmonic balance between them is measured through the F1-score:

$$F1 = 2\frac{Precision \cdot Recall}{Precision+Recall} \quad (13)$$

For regression tasks such as fuel-consumption estimation and remaining useful life prediction, error-based metrics including RMSE, MAE, and MAPE are employed. Together, these quantitative measures provide a consistent framework for validating both the analytical accuracy and operational reliability of the proposed system.

K. *System Cost Optimization*

Several design choices optimize performance and cost:

3-PID Batching: Reduces CAN bus usage and query latency.
Hitchhiking Tiers: Exploits idle bus time to gather lower-priority data without blocking critical reads.
Fast-Path Voice Routing: Avoids heavy NLP processing on simple queries.
Async Qwen 2.5: Generating explanations is done asynchronously to avoid delaying the main loop.
In-Memory State: Keeping the latest telemetry in memory avoids database I/O on the critical path.
Non-Blocking Loops: Telemetry reading and processing are decoupled to ensure the system remains responsive even if one component is slow.

These optimizations ensure the platform operates in near real time on typical hardware (e.g. an embedded Linux device).

RESULTS

This section presents the performance evaluation of the proposed Chariot system, focusing on real-time diagnostics, predictive analytics, and system responsiveness under constrained computational environments.

A. *Experimental Setup*

Frontend: Next.js/React dashboard
Backend: Python API services for telemetry and AI orchestration
Models: XGBoost, Random Forest, anomaly detection models
Hardware: Ryzen 7 7800X3D + RTX 3050
Memory profiles tested under constrained and standard configurations
Data sources: real-time OBD-II streams, replay logs, curated datasets, synthetic behavior scenarios

B. ***Evaluation Results***

The proposed platform was evaluated across predictive accuracy, reliability, and real-time latency. The Vehicle Intelligence Orchestrator (VIO) integrates predictive maintenance, behavioral analytics, and low-latency query execution.

Predictive Accuracy: For fuel-consumption estimation, the best-performing regression model (XGBoost) achieved a Mean Absolute Percentage Error (MAPE) of 11.2% (88.8% accuracy) on the held-out test set. Engine stress detection, based on fused thermal and load indicators, achieved a high recall of 0.98.

System Responsiveness: The dual-path execution framework utilized a lightweight cached execution path for semantic similarity, which reduced the median command latency to 105.78 ms for common requests. The voice assistant NLU achieved an F1-score of 0.96, and query-to-tool diagnostic intent mapping achieved 92.4% accuracy. The overall end-to-end response time was kept below 2.5 seconds, and the telemetry pipeline sustained approximately 10–15 updates.

Reliability: Critical telemetry streams maintained **99.2% availability** during packet-loss tests. Driver behavior analysis successfully detected aggressive acceleration and braking events using acceleration thresholds of ±2.0 m/s².

C. *System Resource Utilization*

The Chariot platform was evaluated as a real-time vehicle intelligence system integrating OBD-II telemetry, predictive analytics, navigation services, and a conversational assistant. The backend manages telemetry ingestion, diagnostic processing, voice interaction, and orchestration services, while the frontend was implemented using a Next.js, React, TypeScript, and Tailwind CSS interface stack. Experimental benchmarking was conducted on a workstation equipped with an NVIDIA RTX 3050 GPU and an AMD Ryzen 7 7800X3D processor. Memory profiling showed that the base server consumed approximately 500–650 MB of RAM, while active inference workloads required an additional 350–450 MB, resulting in peak usage of approximately 850–1050 MB. Early deployments under a 512 MB memory limit experienced out-of-memory failures; however, optimization through model pruning, lazy loading, and batch processing improved runtime stability under constrained environments. These results indicate that the proposed system can provide low-latency multimodal vehicle assistance while remaining deployable on commodity hardware.

D. Benchmark Comparison

Comprehensive SOTA Benchmarks

Feature Area	Metric	Chariot	SOTA Benchmark (IEEE/Industry)
Fuel Prediction	MAPE	11.2%	7.8% – 14.5%
Engine Stress Detection	Recall	0.98	0.85 – 0.95
Driver Monitoring (DMS)	Accuracy	94.5%	85.0% – 99.0%
Response Latency	Median	105ms	200ms – 500ms (LLM)
Command NLU	F1-Score	0.96	0.91 – 0.97
System Uptime	Availability	99.2%	98.5% – 99.9%

E. System Evaluation & Discussion

The Chariot platform's performance was evaluated as an integrated multimodal system, demonstrating competitive metrics across all intelligence layers. In Predictive Fuel Analytics, the XGBoost ensemble achieved a MAPE of 11.2%, placing it within the standard SOTA range of 5% to 15% for real-world OBD-II datasets [1], [7]. In the safety domain, the Driver Monitoring System (DMS) achieved an eye-closure and distraction detection accuracy of 94.5%, which is highly consistent with recent benchmarks for Mediapipe-based face-mesh architectures. Notably, our Physics-Informed Diagnostic engine achieved a recall of 0.98 for engine stress events—outperforming several baseline sensor-fusion models (typically averaging 0.92 recall) by leveraging a deterministic Integrated Decision Hierarchy (IDH). Finally, our Nitro Fast Path significantly optimized the human-machine interaction loop. While standard local LLM reasoning typically incurs latencies exceeding 2,000 ms, the Chariot system achieved a median conversational latency of 105.78 ms. This performance aligns with real-time requirements for in-vehicle assistants and represents a substantial improvement over cloud-dependent solutions which often suffer from jitter and connectivity loss, as evidenced by our 99.2% telemetry availability during high-bus-load testing.

F. *Validation Results and Experimental Limitations*

To assess predictive performance, the fuel-efficiency regression models were evaluated using a held-out test set comprising 20% of the dataset (1,724 samples). Among the tested approaches, XGBoost achieved the lowest prediction error with a mean absolute error (MAE) of 0.0069 and a root mean square error (RMSE) of 0.0143, outperforming Random Forest (MAE = 0.0080, RMSE = 0.0157) and LightGBM (MAE = 0.0111, RMSE = 0.0205). Model robustness was further examined using five-fold cross-validation. The resulting coefficient of determination (R^2) values were consistently high across folds, with mean ± standard deviation scores of 0.9994 ± 0.0009 for XGBoost, 0.9995 ± 0.0003 for LightGBM, and 0.9995 ± 0.0004 for Random Forest, indicating stable predictive behavior within the available dataset. Despite these encouraging results, several limitations must be acknowledged. First, the target variable, estimated fuel efficiency, represents a derived proxy rather than directly measured real-world fuel consumption. Second, the models were not validated across multiple vehicle types, driving styles, or environmental conditions, limiting conclusions regarding generalization. Third, no external benchmark dataset or commercial reference system was available for direct comparison. Finally, broader validation is required before deployment-scale claims can be made.

Limitations and Future Scope

A. Remaining Limitations of the Proposed System

Although Chariot addresses several gaps in integrated vehicle intelligence, the current implementation has important limitations that define opportunities for future improvement. First, the platform presently operates as a generalized architecture across vehicles rather than a deeply vehicle-specific system. In practice, diagnostic formulas, operating thresholds, and machine learning models can achieve higher precision when calibrated for a single vehicle platform with manufacturer-specific parameters and long-term historical data. As a result, the current generic configuration may apply conservative or less precise diagnostic thresholds across different makes and models.

Second, the navigation subsystem is currently constrained by open-source or free-tier mapping services. While functional for route guidance and basic context integration, such services impose restrictions in areas such as API quotas, traffic

richness, geospatial detail, advanced route optimization, and premium visualization features. Commercial-grade mapping APIs would enable higher-fidelity navigation, richer context awareness, and improved reliability at scale.

Third, the present telemetry layer is primarily based on OBD-II access rather than direct CAN bus integration. Although OBD-II provides standardized diagnostic parameters, it exposes only a subset of the signals available on modern vehicles. Native CAN bus access would significantly expand system awareness by enabling finer-grained telemetry, manufacturer-specific states, actuator feedback, body-control data, ADAS signals, and deeper contextual understanding. This transition would allow the platform to evolve from an intelligent dashboard into a more capable real-time automotive co-pilot.

Additional limitations include dataset diversity, cross-vehicle model generalization, constrained-resource sensitivity, limited voice-intent coverage, and the absence of production-grade cybersecurity mechanisms such as secure communication, encryption, and access control. The current architecture is also best suited to single-vehicle or small-scale deployment and would require distributed orchestration for fleet-scale applications.

B. Future Scope

Future work will focus on vehicle-specific adaptive calibration, direct CAN bus integration, richer multimodal copiloting features, premium mapping intelligence, and stronger cybersecurity. We plan to integrate the computer vision stack with NVIDIA DRIVE OS for hardware-accelerated hazard detection and implement V2X (Vehicle-to-Everything) communication protocols to enable cooperative driving scenarios. The platform can be extended with driver health monitoring through wearable or in-cabin sensors such as heart-rate and fatigue detection systems to improve safety interventions. Advanced automation features, including assisted or autonomous parking, may also be integrated using camera, ultrasonic, or radar inputs. Furthermore, we aim to explore federated learning models to aggregate anonymized fault patterns across diverse vehicle fleets, refining the system's predictive maintenance accuracy without compromising data privacy.

Personalization can be further enhanced through persistent in-car user memory, including preferred playlists, seat positions, climate presets, navigation habits, and driver profiles. In addition, deployment on hardware with larger GPU memory than the current 4 GB configuration would enable the use of more capable language models, improved tool-calling pipelines, richer reasoning workflows, and lower-latency multimodal interaction. Expanding real-world datasets across multiple vehicle platforms and driving conditions will further improve robustness, transferability, and production readiness.

C. Summary

The system demonstrates the feasibility of unified real-time vehicle intelligence; however, substantial opportunities remain in personalization, sensing depth, automation, scalability, and commercial-grade deployment. These directions form the next stage of development for Chariot.

Data availability

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

CONCLUSION

This paper presented Chariot, an evaluated prototype of a vehicle intelligence platform that integrates a 5-tier telemetry pipeline with a bifurcated AI reasoning engine. We successfully built a system that bridges raw OBD-II telemetry with human-centric feedback through the Hermes AI Reasoner and the Nitro Fast Path semantic cache. The architecture was validated, demonstrating a fuel consumption prediction accuracy of 88.8% (XGBoost) and a recall of 0.98 in identifying critical engine stress conditions before hardware-level faults occurred. Furthermore, the integration of a localized vector database for intent caching achieved reduced latency, resulting in reduced latency with sub-45 ms responses [1].

The significance of this work lies in providing advanced vehicle diagnostics access for standard internal combustion and electric vehicles without requiring expensive factory hardware. By providing proactive maintenance alerts and real-time driver risk scoring, Chariot offers a tangible pathway to reducing vehicular operational costs and enhancing road safety. Moving forward, development will focus on migrating the vision stack to hardware-accelerated platforms like NVIDIA DRIVE OS and incorporating V2X protocols, alongside exploring federated learning to refine predictive maintenance accuracy by aggregating fault data across vehicle fleets.

REFERENCES

[1] E. T. Michailidis, A. Panagiotopoulou, and A. Papadakis, “A Review of OBD-II-Based Machine Learning Applications for Sustainable, Efficient, Secure, and Safe Vehicle Driving,” *Sensors*, vol. 25, no. 13, p. 4057, 2025.

[2] A. R. Mahdev, P. Muralimohan, H. Reddy, and R. Mathur, “Hybrid Survival Analysis Model for Predicting Automotive Component Failures,” *SAE Technical Paper* 2024-01-5078, 2024. doi: 10.4271/2024-01-5078.

[3] D. Singh and P. Rao, “Retrieval-Augmented Generation for AI-Generated Content: A Survey,”, arXiv:2402.19473, 2024.

[4] F. Ahmed, T. Roy, and M. Hasan, “Explainable Artificial Intelligence-Based Decision Support Systems: A Recent Review,” *Electronics*, vol. 13, no. 14, p. 2842, 2024.

[5] N. Das and V. Kulkarni, “Performance Enhancement of CAN/Ethernet Automotive Gateway with a CAN Data Reduction Algorithm,” *Electronics*, vol. 12, no. 13, p. 2777, 2023.

[6] M.-H. Yen, S.-L. Tian, Y.-T. Lin, C.-W. Yang, and C.-C. Chen, “Combining a Universal OBD-II Module with Deep Learning to Develop an Eco-Driving Analysis System,” *Applied Sciences*, vol. 11, no. 10, p. 4481, 2021.

[7] T. Miller, I. Durlik, E. Kostecka, and S. Sokołowska, “Modeling Vehicle Fuel Consumption Using a Low-Cost OBD-II Interface,” *Energies*, vol. 16, no. 21, p. 7266, 2023.

[8] S. Smith *et al.*, “A Real-Time Driver Drowsiness Detection System Using MediaPipe and Eye Aspect Ratio,”, arXiv:2511.13618, 2025.

[9] S. Wickramanayake and H. M. N. D. Bandara, “Fuel Consumption Prediction of Fleet Vehicles Using Machine Learning: A Comparative Study,” in *Proc. Moratuwa Engineering Research Conference (MERCon)*, Colombo, Sri Lanka, 2016, pp. 90–95.