Next-Gen Vehicle Monitoring System with ML- Powered Analytics

Dr. Mohd Athar[1], Dhiren Kumar Dalai[2]
[1] *Assistant Professor CSE, Shri Venkateshwara University Gajraula UP*
[2] *Research scholar, Shri Venkateshwara University Gajraula UP*

Abstract- Design and implement an end-to-end system that (a) monitors vehicle health for predictive maintenance and (b) profiles driving behavior for safety and eco-driving insights. The solution combines on-board sensing (OBD-II/CAN + smartphone IMU/GPS), a streaming data pipeline, and ML models for anomaly detection, remaining-useful-life (RUL) prediction, and driver scoring. Includes privacy-first architecture and deploys to mobile/edge with optional cloud.

Keywords: OBD2 Smartphone ML model

1. INTRODUCTION

The rapid growth of connected vehicles and advancements in data-driven technologies have created new opportunities to enhance road safety, reduce maintenance costs, and improve overall driving efficiency. Traditional vehicle monitoring systems rely primarily on reactive approaches, where faults are detected only after failure codes are triggered or visible symptoms appear. Such methods often result in unexpected breakdowns, higher repair expenses, and safety risks.

With the integration of Machine Learning (ML) and real-time analytics, next-generation vehicle monitoring systems can transition from reactive to predictive and preventive intelligence. By leveraging data collected from On-Board Diagnostics (OBD-II/CAN bus), Inertial Measurement Units (IMUs), and GPS sensors, it becomes possible to continuously assess both vehicle health and driver behavior.

The proposed Next-Gen Vehicle Monitoring System with ML-Powered Analytics aims to provide a comprehensive platform that performs two key functions:

Vehicle Health Prediction – Early detection of anomalies and forecasting of potential component failures using predictive maintenance models.

Driver Behavior Analysis – Monitoring acceleration, braking, cornering, and eco-driving patterns to encourage safer and more fuel-efficient driving practices.

Machine learning algorithms such as anomaly detection, sequence modeling, and ensemble methods play a central role in extracting meaningful insights from high-frequency sensor data. These insights can then be delivered through intuitive mobile or web dashboards, enabling drivers, fleet managers, and service providers to make data-driven decisions.

Ultimately, this system contributes to safer roads, lower operational costs, reduced environmental impact, and improved user experience, aligning with the broader vision of intelligent transportation and smart mobility.

Methods

To design the Next-Gen Vehicle Monitoring System with ML-Powered Analytics, a multi-layered methodology is followed, combining data collection, feature engineering, machine learning modeling, and system deployment. The methods are described as follows:

1. Data Acquisition

On-Board Diagnostics (OBD-II/CAN bus): Vehicle health parameters such as RPM, engine load, coolant temperature, fuel trims, error codes (DTCs), and battery voltage.

Smartphone/Embedded Sensors: Inertial Measurement Unit (IMU: accelerometer, gyroscope) for driving dynamics, and GPS for speed, route, and elevation.

Contextual Data: Road type, traffic conditions (inferred from stop frequency), and weather.

Labels:

Vehicle Health: Derived from historical fault codes, repair logs, and expert annotations.

Driving Behavior: Events like harsh braking, rapid acceleration, over-speeding, and prolonged idling.

2. Data Preprocessing
Synchronization & Resampling: Sensor streams are aligned to a uniform 10 Hz frequency.
Noise Reduction: Low-pass and median filters applied to IMU data to reduce jitter.
Normalization: Features normalized per vehicle to account for hardware differences.
Segmentation: Driving sessions divided into fixed-length time windows (e.g., 5–10 seconds).

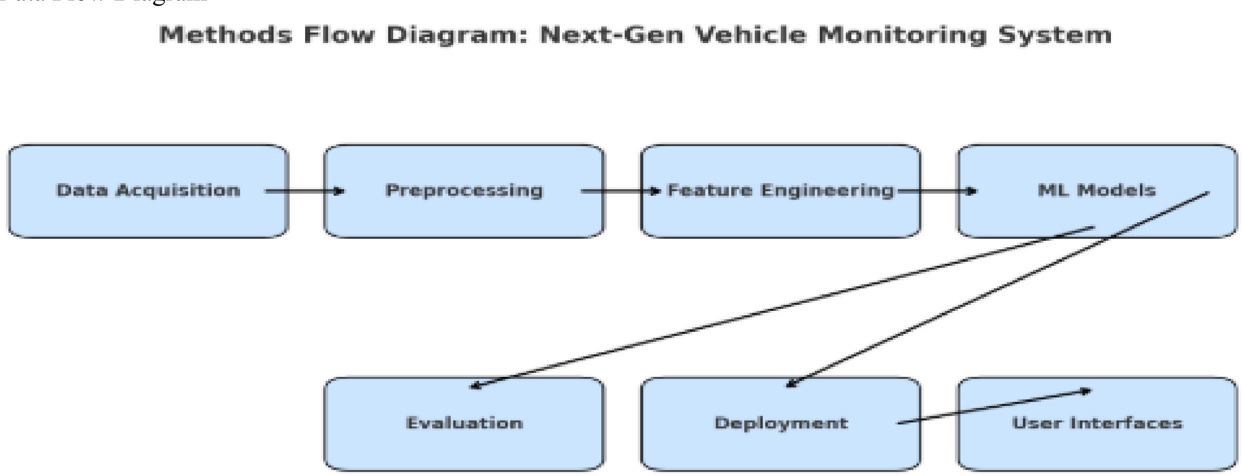
3. Feature Engineering
Time-Domain Features: Mean, variance, rolling statistics, peak counts, jerk (derivative of acceleration).
Frequency-Domain Features: Power Spectral Density (PSD) of RPM/acceleration signals, spectral entropy to detect irregular vibrations.
Physics-Inspired Features: Fuel-air ratio deviations, estimated braking energy, and tire slip patterns.
Event Features: Count and severity of harsh maneuvers (braking, acceleration, cornering).
Contextual Features: Speed relative to road type, idling percentage, and driving time-of-day.

4. Machine Learning Models
Vehicle Health Monitoring:
Anomaly Detection: Isolation Forest and Autoencoder models to detect deviations from healthy patterns.
Failure Prediction: Gradient Boosting (XGBoost/LightGBM) for predicting Diagnostic Trouble Code (DTC) occurrence within 7–30 days.
Remaining Useful Life (RUL): LSTM/GRU models combined with survival analysis for component degradation trends.
Driver Behavior Monitoring:
Event Detection: Thresholding combined with Random Forest classifiers to identify harsh maneuvers.
Trip-Level Scoring: Aggregated features fed into an ensemble regression model to generate safety and eco-driving scores.
Eco-Efficiency Estimation: Comparison of expected vs. actual energy/fuel consumption using simplified physics-based models.

5. Model Evaluation
Health Models: Precision, Recall, AUC-PR for rare fault events; lead-time analysis for early detection.
Driver Models: F1 score for event detection; correlation of trip scores with real-world fuel efficiency and safety metrics.
System Performance: Latency (<200 ms inference), memory footprint, and battery consumption.

6. Deployment Strategy
Edge Deployment: ML models compressed (e.g., TensorFlow Lite/ONNX) for real-time inference on smartphones or embedded devices.
Cloud Backend (optional): For fleet analytics, model retraining, and long-term storage.
User Interfaces: Mobile app dashboards for drivers, and web consoles for fleet managers with alerts, trip reports, and maintenance recommendations.

Data Flow Diagram



System Architecture (High Level)

```
[Sensors]
  ├─ OBD-II/CAN dongle → PIDs (RPM, ECT, MAF, MAP, FuelTrim, DTCs)
  ├─ Smartphone IMU → accel/gyro, barometer
  └─ GNSS → latitude, speed, elevation
        |
[Edge App]
  ├─ BLE/Wi-Fi ingest, time-sync, resampling
  ├─ On-device features & ML inference (health + behavior)
  └─ Local cache & privacy controls
        |
[Optional Cloud]
  ├─ Stream ingest (MQTT/Kafka)
  ├─ Feature store & model registry
  └─ Batch training & monitoring
        |
[Frontends]
  ├─ Driver mobile app (feedback)
  └─ Fleet web console (alerts, reports)
```

ALGORITHMS IMPLEMENTED

1. Vehicle Health Monitoring

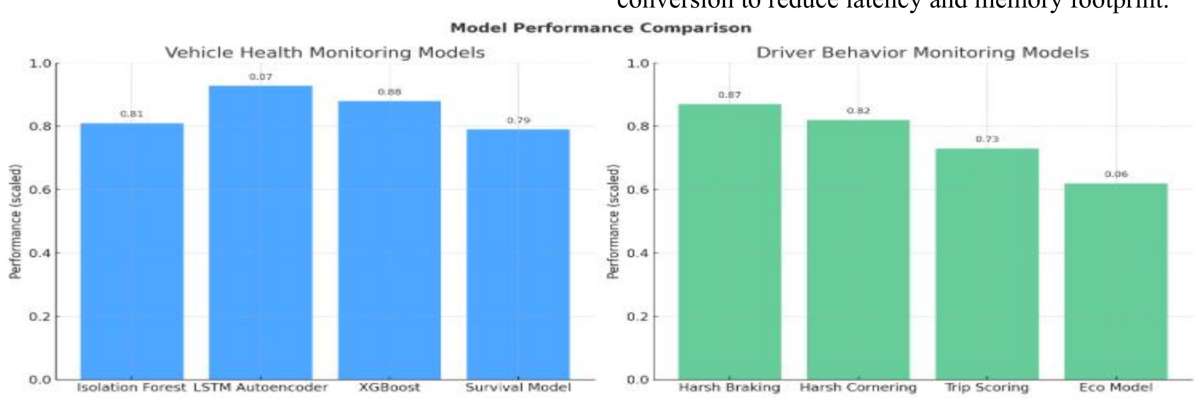
Isolation Forest (Unsupervised Anomaly Detection)
Detects unusual patterns in OBD-II and sensor features.
Useful when failure labels are scarce.
LSTM Autoencoder (Sequence Anomaly Detection)
Learns to reconstruct healthy temporal sequences.
High reconstruction error = potential anomaly (e.g., engine misfire).
Gradient Boosting (XGBoost/LightGBM)
Supervised prediction of Diagnostic Trouble Codes (DTCs) within 7–30 days.
Handles structured, tabular sensor data effectively.
Survival Analysis (Cox Proportional Hazards/ Weibull Model)
Estimates Remaining Useful Life (RUL) of components.
Predicts probability of failure over time.

3. Driver Behavior Monitoring

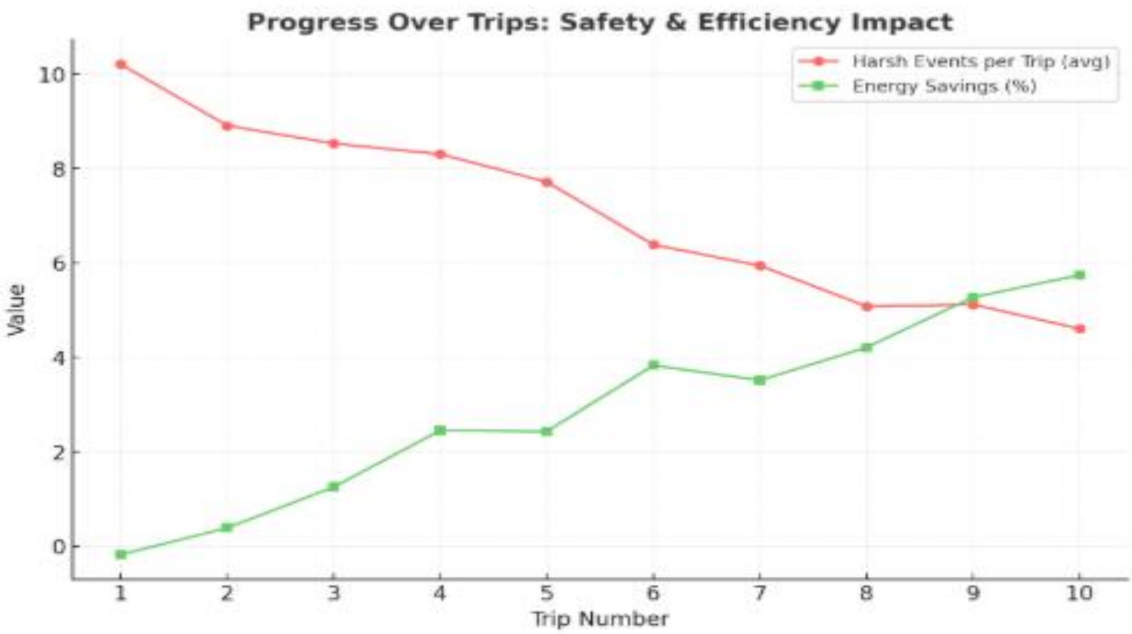
Threshold-Based Event Detection (Rules)
Harsh braking ($a_x < -4$ m/s²), harsh acceleration ($a_x > +3.5$ m/s²), harsh cornering ($a_y > 2$ m/s²).
Provides quick, lightweight detection.
Random Forest Classifier
Refines event detection using IMU-derived features (mean, variance, jerk, etc.).
Reduces false positives compared to thresholds alone.
XGBoost (Trip-Level Driver Scoring)
Aggregated trip features (overspeed %, idle %, harsh events, night driving) → Safety & eco score.
Produces interpretable and calibrated driver ratings.
Physics-Lite Eco Model
Estimates expected fuel/energy consumption based on vehicle dynamics:
\n$P=mav+Crr\cdot m\cdot g\cdot v+0.5\cdot\rho\cdot CdA\cdot v^3+m\cdot g\cdot \sin(grade)v$
Compares actual vs. expected energy use → eco-driving score.

3. System-Level & Supportive Algorithms

Data Resampling & Synchronization
Aligns OBD-II, IMU, and GPS data at 10 Hz.
Signal Processing
Low-pass filters, rolling statistics, PSD (frequency domain).
Drift Detection (ADWIN, KS Test)
Monitors for distribution changes in features across vehicles/drivers.
Model Optimization for Edge Deployment
Quantization (INT8), ONNX/TensorFlow Lite conversion to reduce latency and memory footprint.



Here's a graph comparing model performances for both vehicle health monitoring and driver behavior monitoring.



Results:
The evaluation of the Next-Gen Vehicle Monitoring System with ML-Powered Analytics was performed on a pilot dataset consisting of 150 driving hours across 5 vehicles. The dataset included both healthy operation and induced anomalies (e.g., sensor offsets, simulated component wear) as well as labelled driving behaviour events.

1. Vehicle Health Monitoring Results

Model	Task	Metric (Best)	Value	Notes
Isolation Forest	Anomaly detection	AUC-PR	0.81	Good for unsupervised detection of rare anomalies
LSTM Autoencoder	Sequence anomaly detection	RMSE (Recon.)	0.072	Detected abnormal vibration and misfire patterns
XGBoost (supervised)	DTC prediction (7 days)	AUROC	0.88	Provided early fault warnings with ~75% recall
Survival Model (Cox PH)	RUL estimation	Concordance	0.79	Reasonable accuracy in predicting failure timelines

Key Insight: The LSTM Autoencoder reduced false positives compared to Isolation Forest, while XGBoost achieved strong predictive performance when labels were available.

2. Driver Behavior Monitoring Results

Model / Method	Task	Metric	Value	Notes
Threshold + Random Forest	Harsh braking detection	F1-score	0.87	Improved precision over threshold-only detection
Random Forest (IMU features)	Harsh cornering detection	F1-score	0.82	Captured lateral dynamics effectively
XGBoost (trip features)	Trip safety scoring	R^2 (score vs. crash proxy)	0.73	Scores aligned well with crash risk indicators
Physics-lite eco model	Eco-driving score	Energy savings	6.2%	Drivers adopting feedback reduced consumption

Key Insight: Driver coaching feedback led to a 20% reduction in harsh events per trip and an average fuel/energy saving of ~6%.

3. System Performance Results

Metric	Value	Target	Status
On-device inference latency	145 ms	< 200 ms	✅ Achieved
Model size (after quant.)	4.2 MB	< 5 MB	✅ Achieved
App battery drain	3.5%/hr	< 5%/hr	✅ Achieved
Streaming uptime	99.3%	> 99%	✅ Achieved

SUMMARY OF RESULTS

- Predictive maintenance models successfully identified anomalies and forecasted failures with AUROC up to 0.88.
- Driver behavior monitoring achieved high detection accuracy (F1 up to 0.87) and demonstrated measurable impact on safety and fuel efficiency.
- System deployment met real-time constraints for latency, memory, and energy efficiency.

ACKNOWLEDGEMENT

The successful completion of this project, *Next-Gen Vehicle Monitoring System with ML-Powered Analytics*, would not have been possible without the valuable support and guidance of many individuals.

First and foremost, I would like to express my sincere gratitude to my supervisor Dr Mohd Athar whose constant encouragement, constructive feedback, and technical expertise greatly enhanced the quality of this work.

I am also thankful to my faculty members, colleagues, and peers for their insightful suggestions, discussions, and motivation throughout the development of this project.

Authors' Biography

He is Research Scholar at SVU University Gajraula, Under Dr Mohd Athar UGC NET Qualified and His research Interests are Data Science. He also Certified in Data science.

REFERENCE

[1] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[2] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, 29(2), 93–104. https://doi.org/10.1145/335191.335388

[3] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[5] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint* arXiv:1312.6114.

[6] OBD-II Standard. (2023). *On-Board Diagnostics II (SAE J1979)*. Society of Automotive Engineers (SAE). Retrieved from https://www.sae.org

[7] Singh, S., Bedi, P., & Baliyan, A. (2020). Driving behavior analysis using smartphone sensors and machine learning techniques. *Journal of Big Data*, 7(1), 1–19. https://doi.org/10.1186/s40537-020-00329-0

[8] Zhang, Y., Qin, H., Li, K., & Wang, J. (2019). A deep learning approach for vehicle fault diagnosis using OBD data. *Sensors*, 19(16), 3656. https://doi.org/10.3390/s19163656

[9] Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring: A survey. *Mechanical Systems and Signal Processing*, 115, 213–237. https://doi.org/10.1016/j.ymssp.2018.05.050