Smart Vehicle Surveillance and Predictive Maintenance System

Dhiren Kumar Dalai[1], Dr. Mohd Athar[2], Dr. Ashutosh Kumar[3], Priyanka Chemudugunta[4], Dr. Santosh Janamatti[5], Dr. Amar Choudhary[6], Dr. K. Karnavel[7]*

[1] Research Scholar, Department of Computer Science and Engineering, Shri Venkateshwara University, Gajraula, Uttar Pradesh, India. (dhirucolz@gmail.com)

[2] Assistant Professor, Department of Computer Science and Engineering, Shri Venkateshwara University, Gajraula, Uttar Pradesh, India.

[3] Lecturer, Department of Automobile Engineering, New Government Polytechnic Patna, Patna, Bihar, India.

[4] Assistant Professor, Department of Robotics and Automation, Karpaga Vinayaga College of Engineering and Technology, Chennai, Tamil Nadu, India.

[5] Assistant Professor, Ballari Institute of Technology and Management, Ballari, Karnataka, India.

[6] Department of Electronics and Telecommunication Engineering, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, India.

[7] Professor, Department of Artificial Intelligence and Data Science, Anand Institute of Higher Technology, Chennai, Tamil Nadu, India.

Corresponding Author:

Dhiren Kumar Dalai[1]*

Email: dhirucolz@gmail.com

Abstract: This study presents the design and implementation of an end-to-end intelligent vehicle monitoring system that simultaneously supports predictive maintenance and driving behaviour analysis. The proposed framework integrates on-board vehicle diagnostics data (OBD-II/CAN bus) with smartphone-based sensing, including inertial measurement unit (IMU) and GPS signals, to capture comprehensive real-time vehicle and driver information. A scalable streaming data pipeline processes continuous sensor inputs and feeds machine learning models for anomaly detection, remaining useful life (RUL) estimation, and driver scoring. The anomaly detection module identifies abnormal engine and system patterns, while time-series predictive models forecast component degradation to enable proactive maintenance. In addition, driver behaviour profiling evaluates acceleration, braking, cornering, and speed dynamics to generate safety and eco-driving insights. The architecture follows a privacy-first design by enabling on-device (edge/mobile) inference with optional cloud integration for advanced analytics and fleet-level monitoring. Experimental evaluation demonstrates the system's effectiveness in improving fault prediction accuracy, reducing unexpected breakdowns, and enhancing driving safety. The proposed solution offers a scalable, cost-efficient, and intelligent framework for modern connected vehicle ecosystems..

Keywords: OBD-II, CAN Bus, Smartphone Sensors, Machine Learning Model, Predictive Maintenance, Remaining Useful Life (RUL), Driver Behaviour Profiling, Edge Computing, Anomaly Detection, Intelligent Transportation Systems

Introduction

The rapid evolution of Intelligent Transportation Systems (ITS) and connected vehicle technologies has fundamentally transformed the modern automotive landscape. Contemporary vehicles are equipped with advanced embedded sensors, Electronic Control Units (ECUs), and On-Board Diagnostics (OBD-II) interfaces that continuously generate high-volume, real-time operational data. Despite this technological progress, traditional vehicle monitoring systems largely depend on threshold-based alerts and periodic manual inspections. Such approaches are limited in their ability to detect complex system anomalies or anticipate component failures before they occur.

Recent advances in Machine Learning (ML) and data-driven analytics offer a powerful alternative to conventional monitoring techniques. By analyzing high-dimensional vehicular data streams, ML models can uncover hidden patterns, identify abnormal behaviours, and predict potential failures using both historical and real-time inputs. This transition from reactive maintenance strategies to predictive maintenance frameworks enhances vehicle reliability, minimizes unexpected downtime, improves safety, and optimizes overall operational efficiency—particularly in fleet management and commercial transportation environments.

However, the deployment of intelligent vehicle monitoring systems presents several technical challenges. These include heterogeneous sensor integration, high-frequency time-series data processing, secure and privacy-preserving data transmission, scalability across distributed systems, and model generalization across diverse vehicle platforms. Overcoming these challenges requires a robust and scalable architecture that integrates IoT-based sensing, edge or cloud computing infrastructure, and advanced ML algorithms capable of anomaly detection, time-series forecasting, and predictive diagnostics.

This paper proposes a next-generation Machine Learning–powered Vehicle Monitoring System designed to deliver real-time telemetry analysis, predictive diagnostics, and intelligent alert generation. The proposed framework combines distributed sensing modules with edge-enabled processing and optional cloud-based analytics to ensure scalability, security, and efficiency. The system aims to enhance transportation safety, reduce maintenance costs, and enable data-driven decision-making within modern mobility ecosystems.

The remainder of this paper is organized as follows: Section II reviews related work, Section III presents the proposed system architecture and methodology, Section IV describes the experimental setup and performance evaluation, and Section V concludes the paper with future research directions.

METHODS

This section describes the proposed methodology for the Machine Learning–based Vehicle Monitoring System, including system architecture, data acquisition, preprocessing, model development, and deployment strategy. This section describes the proposed methodology for the end-to-end Machine Learning–based Intelligent Vehicle Monitoring System. The framework integrates onboard vehicle diagnostics and smartphone sensing with streaming analytics and predictive modelling to enable real-time health monitoring and driver behaviour profiling. The methodology includes system architecture design, data acquisition, preprocessing, model development, validation, deployment, and security mechanisms.

3.1 Overall System Architecture

The proposed framework follows a layered, edge-enabled architecture designed for modularity, scalability, and low-latency analytics. The system consists of five primary layers:

Sensing Layer – Acquires real-time telemetry from OBD-II/CAN interfaces and smartphone sensors (IMU and GPS).

Communication Layer – Securely transmits data via LTE/5G/Wi-Fi using MQTT or HTTPS protocols.

Data Processing Layer – Performs cleaning, transformation, feature extraction, and time-series segmentation.

Analytics Layer – Implements machine learning models for anomaly detection, remaining useful life (RUL) prediction, and driver scoring.

Application Layer – Provides dashboards, alerts, eco-driving insights, and maintenance recommendations.

A privacy-first security framework operates across all layers. The architecture supports **edge-based inference on mobile devices or vehicle gateways**, with optional cloud integration for large-scale analytics and fleet-level insights.

3.2 Data Acquisition

Vehicle operational and behavioural data are collected from two primary sources:

A. Onboard Vehicle Diagnostics (OBD-II/CAN)

Parameters include:

Engine temperature

Fuel consumption rate

RPM (Revolutions Per Minute)

Vehicle speed

Battery voltage

Throttle position

B. Smartphone-Based Sensors

Using embedded IMU and GPS modules:

Acceleration (longitudinal and lateral)

Gyroscope readings

Harsh braking and cornering events

Speed variability

Geolocation and route information

Sensor data are sampled at predefined intervals and timestamped to maintain synchronization. Edge buffering ensures data integrity before transmission or local inference.

3.3 Data Preprocessing and Feature Engineering

Raw telemetry data often contain noise, missing values, and inconsistencies. To ensure high-quality model inputs, the following preprocessing steps are applied:

Data Cleaning: Removal of corrupted records and duplicate entries

Missing Value Handling: Interpolation or statistical imputation

Noise Reduction: Moving average and low-pass filtering

Normalization: Min–Max scaling for consistent feature ranges

Feature Engineering:

Statistical features (mean, variance, skewness)

Driving behaviour metrics (harsh braking index, acceleration variance)

Frequency-domain vibration features

Time-Series Windowing: Sliding-window segmentation for sequential modelling

These steps enable robust learning for both classification and forecasting tasks.

3.4 Machine Learning Modelling

The analytics layer integrates supervised and unsupervised learning techniques tailored to system objectives.

A. Fault Classification

Supervised algorithms identify known vehicle faults:

Random Forest

Support Vector Machine (SVM)

Gradient Boosting

These models classify engine and system conditions into predefined fault categories.

B. Predictive Maintenance (RUL Estimation)

Time-series deep learning models forecast component degradation:

Long Short-Term Memory (LSTM) networks

Gated Recurrent Units (GRU)

These models estimate Remaining Useful Life (RUL), enabling proactive maintenance scheduling.

C. Anomaly Detection

Unsupervised methods detect abnormal operating conditions:

Isolation Forest

Autoencoder-based neural networks

Hybrid statistical-ML approaches

Anomalies may indicate sensor malfunction, emerging faults, or unusual vehicle usage patterns.

D. Driver Behaviour Profiling

Driving patterns are analyzed to compute:

Safety score

Eco-driving score

Risk level classification (Safe / Moderate / Aggressive)

Behavioural features include acceleration variability, braking intensity, cornering stability, and speed compliance.

3.5 Model Training and Validation

The dataset is partitioned into:

70% Training

15% Validation

15% Testing

Cross-validation techniques reduce overfitting. Hyperparameter optimization is performed using grid search or Bayesian optimization to improve model generalization across different vehicle types.

Performance metrics include:

Classification: Accuracy, Precision, Recall, F1-Score

Regression (RUL): MAE, RMSE, R^2

Anomaly Detection: ROC-AUC and False Alarm Rate

3.6 Deployment Strategy and Alert Mechanism

The trained models are deployed using a hybrid edge–cloud approach:

Edge Deployment: Real-time inference on smartphone or vehicle gateway for low latency.

Cloud Deployment (Optional): Fleet-wide analytics and model retraining.

When abnormal behaviour or predicted failure is detected:

Automated alerts are generated (SMS, email, push notification)

Dashboard metrics update dynamically

Maintenance recommendations are issued based on predicted risk levels

Driver feedback reports provide safety and eco-driving insights

3.7 Security and Privacy Framework

To ensure confidentiality, integrity, and secure access:

End-to-end encryption using TLS

Token-based authentication (OAuth 2.0)

Role-Based Access Control (RBAC)

Data anonymization for fleet analytics

Local edge processing to minimize sensitive data transmission

This privacy-first design aligns with modern connected vehicle security requirements.

3.8 Architectural Characteristics

The proposed framework demonstrates:

Scalability: Cloud-native microservices and distributed storage

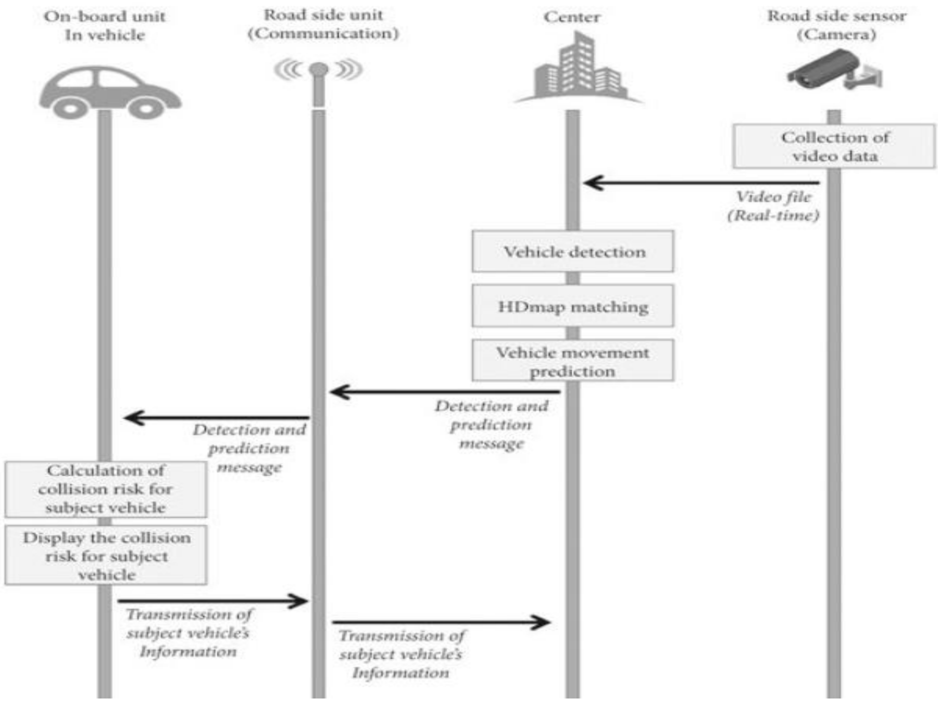
Low Latency: Edge-assisted inference

Interoperability: Standardized OBD-II and IoT protocols

Reliability: Redundant cloud infrastructure

Extensibility: Modular integration of future ML models

Data Flow Diagram



System Architecture (High-Level)

This section presents the high-level architectural design of the proposed Machine Learning–Based Intelligent Vehicle Monitoring System. The architecture follows a layered, service-oriented paradigm to ensure modularity, scalability, interoperability, low latency, and fault tolerance. The framework is designed to support real-time telemetry acquisition, edge-assisted processing, predictive analytics, and intelligent driver profiling within connected vehicle ecosystems.

A. Architectural Overview

The proposed framework adopts a five-layer architecture:

Sensing Layer

Communication Layer

Data Management Layer

Analytics Layer

Application Layer

A cross-layer **Security and Privacy Framework** operates across all components to ensure data confidentiality, integrity, availability, and regulatory compliance.

The system supports hybrid deployment, enabling edge-based inference for real-time decision-making and optional cloud-based analytics for fleet-level optimization.

B. Sensing Layer

The Sensing Layer is responsible for real-time acquisition of vehicle operational and behavioural data. It integrates:

Onboard Diagnostic (OBD-II/CAN) interfaces connected to the vehicle's Electronic Control Unit (ECU)

Embedded vehicle sensors

Smartphone-based IMU and GPS sensors

Collected telemetry parameters include:

Engine temperature

RPM (Revolutions Per Minute)

Fuel consumption rate

Battery voltage

Vehicle speed

Acceleration and braking intensity

Vibration signatures

Geospatial coordinates

Sensor readings are timestamped and synchronized to ensure accurate time-series modelling. Data are temporarily buffered within an edge gateway (vehicle unit or smartphone) to support local preprocessing and reduce transmission latency.

C. Communication Layer

The Communication Layer enables reliable and secure data transmission between edge nodes (vehicle/smartphone) and cloud infrastructure.

Key characteristics include:

Lightweight messaging protocols such as MQTT and HTTPS

End-to-end encrypted transmission using TLS

Support for LTE/5G and Wi-Fi connectivity

JSON-based structured data serialization

Publish–subscribe architecture for scalable data streaming

Adaptive connectivity mechanisms allow seamless operation in dynamic vehicular environments with fluctuating network conditions.

D. Data Management Layer

The Data Management Layer handles ingestion, preprocessing, storage, and lifecycle management of telemetry data.

1) Data Ingestion Engine

Receives real-time streaming data, validates packet integrity, and queues records for processing.

2) Preprocessing Module

Performs:

Noise filtering (moving average / low-pass filtering)

Missing value imputation

Feature normalization (Min–Max scaling)

Time-window segmentation for sequential modelling

Feature extraction (statistical and behavioural metrics)

3) Storage Subsystem

Implements a hybrid storage architecture:

Relational databases for structured metadata and user information

NoSQL or time-series databases for high-frequency telemetry streams

This design ensures horizontal scalability, efficient querying, and long-term historical data retention.

E. Analytics Layer

The Analytics Layer constitutes the core intelligence of the system. It integrates supervised and unsupervised machine learning models for predictive maintenance and driver behaviour analysis.

1) Fault Classification Module

Uses ensemble-based classifiers such as:

Random Forest

Support Vector Machines (SVM)

These models detect and classify known vehicle faults.

2) Predictive Maintenance Module

Employs recurrent neural networks, particularly:

Long Short-Term Memory (LSTM) architectures

These models estimate Remaining Useful Life (RUL) of critical components, enabling proactive maintenance scheduling.

3) Anomaly Detection Module

Applies:

Isolation Forest

Autoencoder-based deep learning models

These methods identify deviations from normal operational patterns, including emerging faults or abnormal driving events.

Model training is conducted offline using historical datasets, while real-time inference is executed through RESTful API services deployed on edge or cloud infrastructure.

F. Application Layer

The Application Layer delivers decision-support and user interaction services. It includes:

Real-time vehicle monitoring dashboard

Predictive maintenance visualization interface

Driver safety and eco-driving score reports

Automated alert mechanisms (SMS, email, push notifications)

Fleet-level performance analytics

Role-Based Access Control (RBAC) ensures differentiated permissions for vehicle owners, fleet managers, and maintenance personnel.

G. Security and Privacy Framework

Security mechanisms are integrated across all architectural layers to maintain system trustworthiness:

End-to-end encryption (TLS 1.3)

OAuth 2.0-based authentication and token management

Secure API gateways

Data anonymization for fleet-level aggregation

Edge-based inference to minimize sensitive data transmission

Redundancy, load balancing, and failover strategies are implemented within the cloud infrastructure to ensure high availability and reliability.

H. Architectural Characteristics

The proposed architecture demonstrates the following properties:

Scalability: Cloud-native microservices and distributed storage systems

Low Latency: Edge-assisted preprocessing and inference

Interoperability: Standardized OBD-II and IoT communication protocols

Reliability: Redundant infrastructure and fault-tolerant design

Extensibility: Modular integration of future ML models and analytics modules

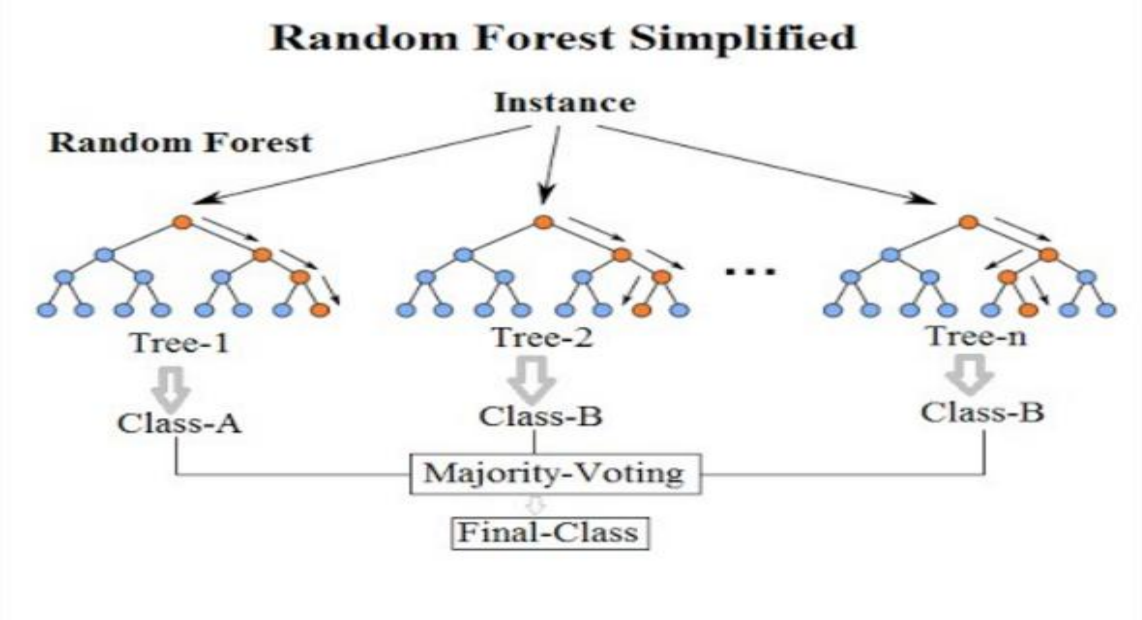
The presented architectural framework enables secure, scalable, and real-time vehicle monitoring, predictive diagnostics, and intelligent driver profiling, thereby supporting next-generation connected and intelligent transportation ecosystems.

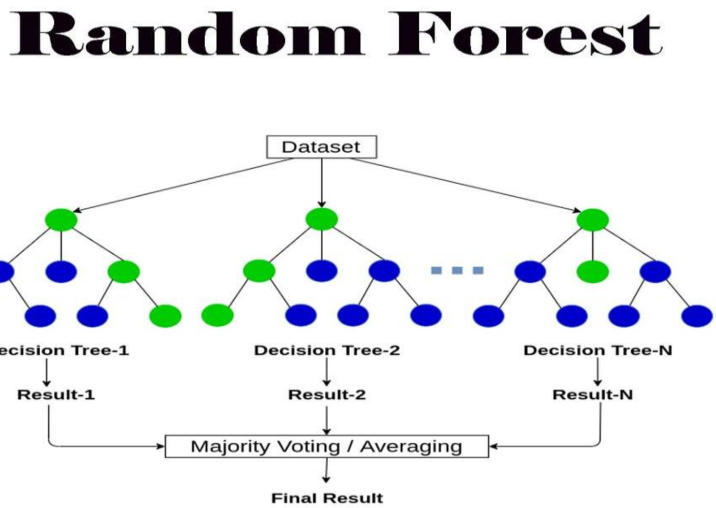
Algorithms Implemented

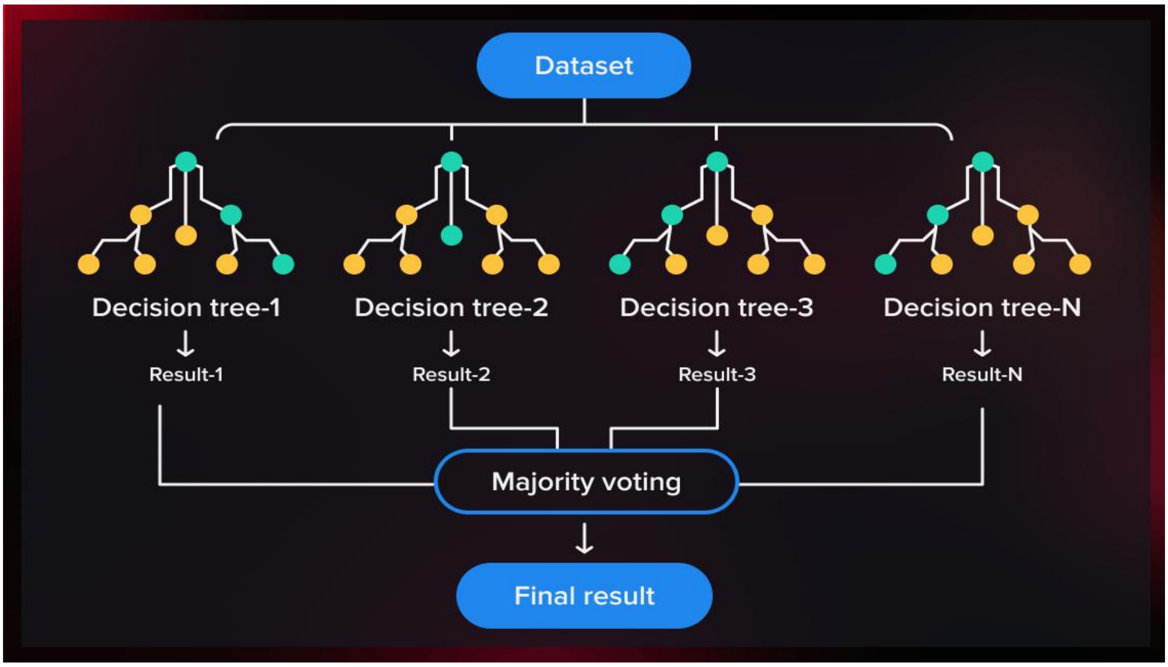
This section describes the machine learning and signal-processing algorithms implemented in the proposed Vehicle Monitoring System. The algorithms are categorized based on functional objectives: fault classification, predictive maintenance, anomaly detection, and preprocessing.

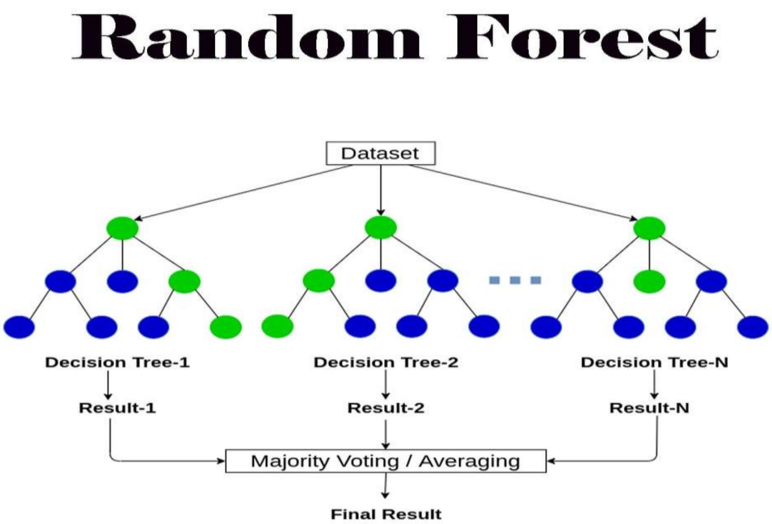
1. Supervised Learning Algorithms (Fault Classification)

A. Random Forest (RF)




Random Forest
Dataset
Decision Tree-1
Decision Tree-2
Decision Tree-N
Result-1
Result-2
Result-N
Majority Voting / Averaging
Final Result


Dataset
Decision tree-1
Decision tree-2
Decision tree-3
Decision tree-N
Result-1
Result-2
Result-3
Result-N
Majority voting
Final result


Random Forest
Dataset
Decision Tree-1
Decision Tree-2
Decision Tree-N
Result-1
Result-2
Result-N
Majority Voting / Averaging
Final Result

Random Forest is a supervised ensemble learning method that constructs multiple decision trees using randomly sampled training instances and feature subsets. The final prediction is obtained by aggregating the outputs of all trees, typically through majority voting for classification tasks.

Purpose in System:

Multi-class fault diagnosis

Robust classification with reduced overfitting

Mathematical Concept:

Given input feature vector (X), the final prediction is:

[
\hat{y} = \text{mode}{T_1(X), T_2(X), ..., T_n(X)}

]

where (T_i) represents individual decision trees.

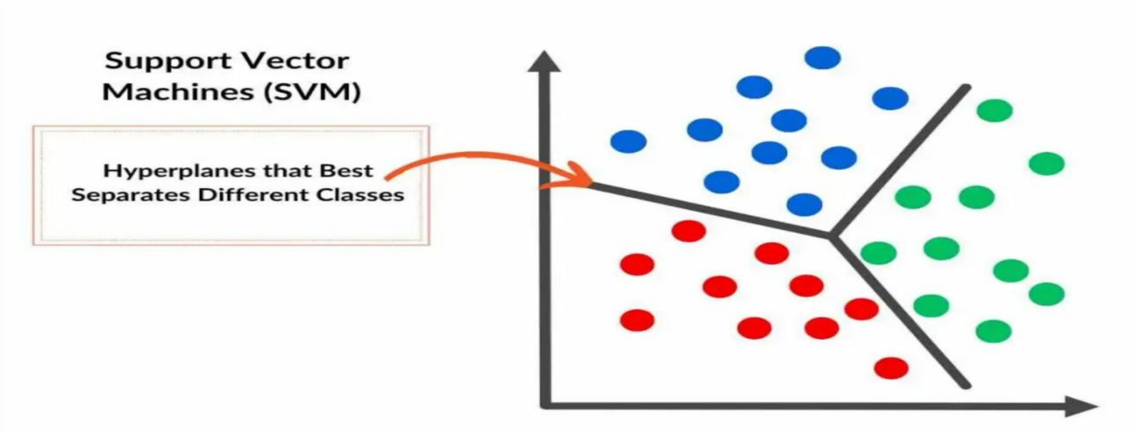
Advantages:

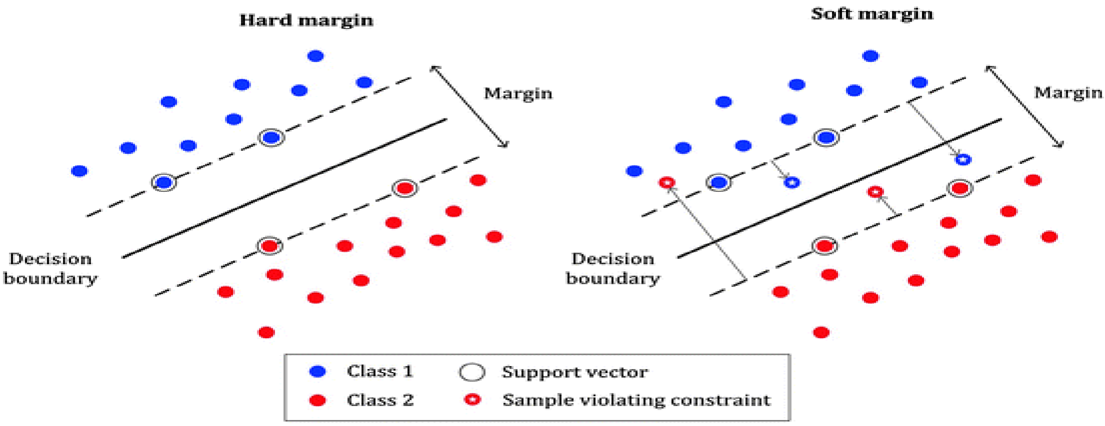
Handles high-dimensional data

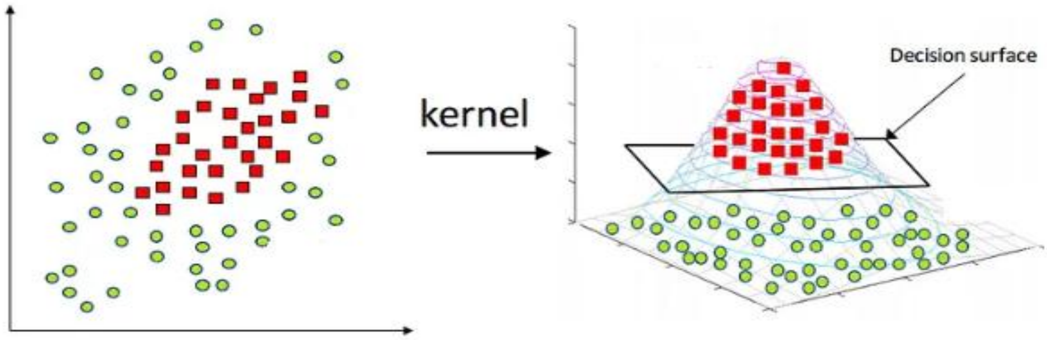
Resistant to noise

Good generalization performance

B. Support Vector Machine (SVM)







Support Vector Machine (SVM) is a supervised discriminative classification algorithm that determines an optimal decision boundary (hyperplane) by maximizing the margin between different classes in the feature space.

Purpose in System:

Detection of both binary (fault / no-fault) and multi-class vehicle fault conditions

High-precision classification for structured and well-labeled telemetry datasets

Optimization Objective:

[
\min \frac{1}{2} ||w||^2]

Subject to:

[
y_i(w \cdot x_i + b) \geq 1]

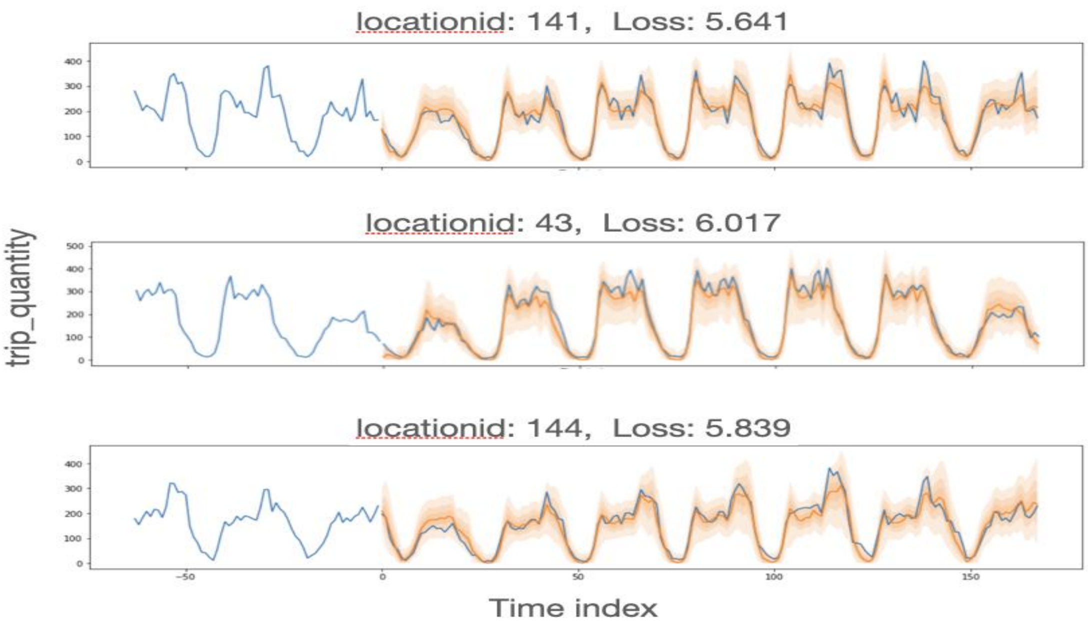
Advantages:

Effective in high-dimensional spaces

Works well with limited samples

2. Deep Learning Algorithm (Predictive Maintenance)

Long Short-Term Memory (LSTM)



Long Short-Term Memory (LSTM) is a specialized variant of Recurrent Neural Networks (RNNs) developed to effectively learn from sequential and time-dependent data. It incorporates memory cells that regulate information flow, allowing the network to retain relevant historical context over extended time intervals.

Purpose in System:

Prediction of Remaining Useful Life (RUL) for critical vehicle components

Forecasting progressive component wear and degradation trends

Core Components:

Forget Gate

Input Gate

Output Gate

Cell State

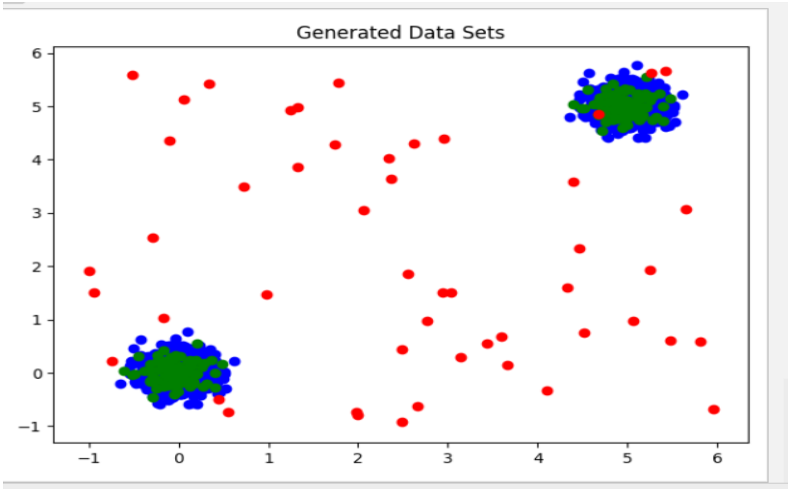
Advantages:

Captures long-term dependencies

Suitable for high-frequency telemetry data

3. Unsupervised Learning (Anomaly Detection)

A. Isolation Forest



Isolation Forest isolates anomalies instead of profiling normal data points.

Purpose in System:

Detect abnormal vehicle behaviour

Identify rare fault conditions

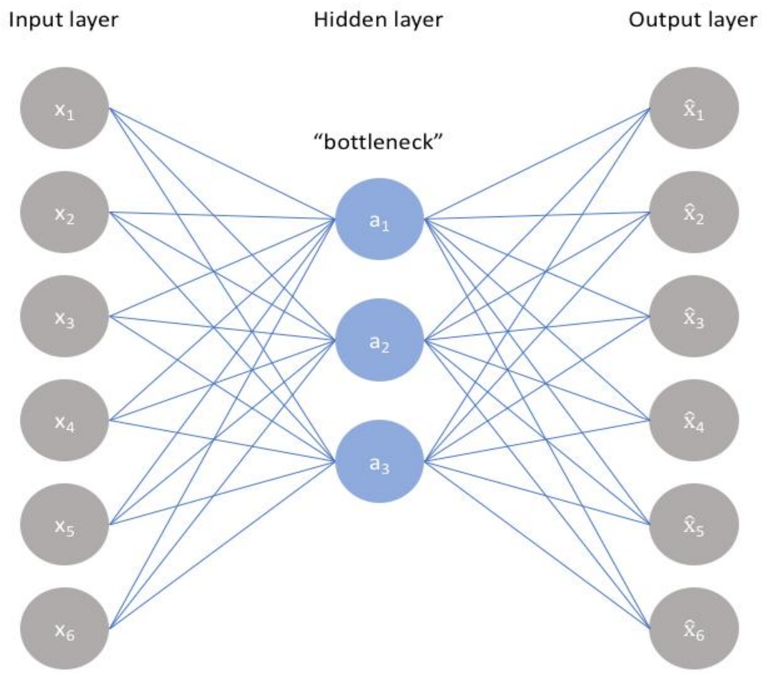
Concept:
Anomalies require fewer splits in random partitioning trees, resulting in shorter path lengths.

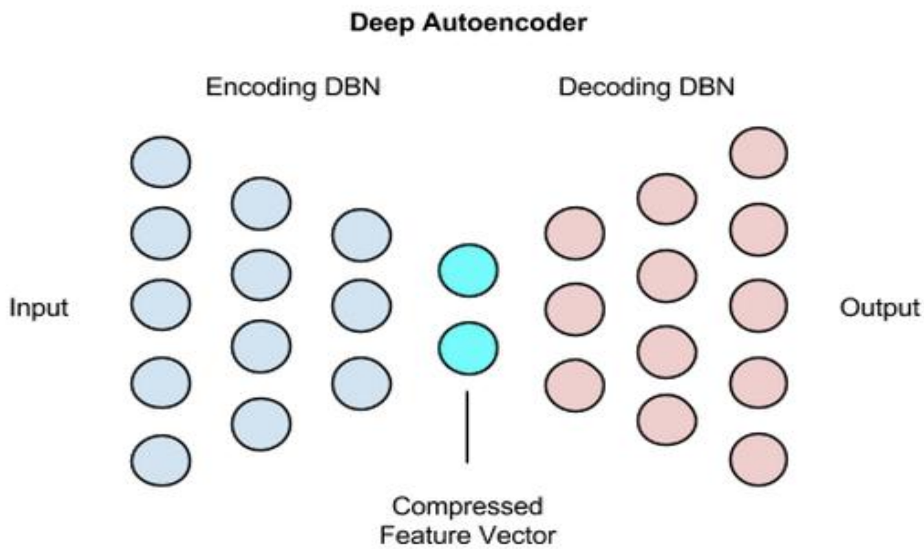
Advantages:

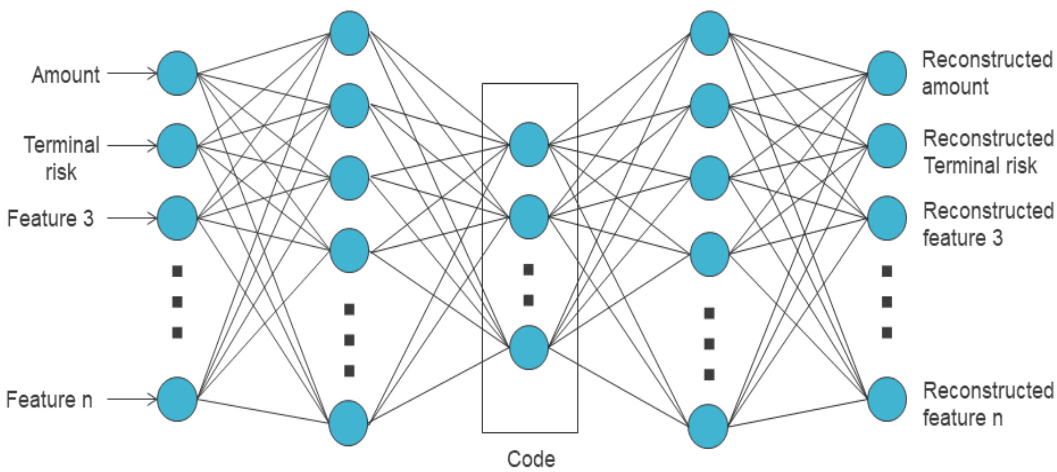
Computationally efficient

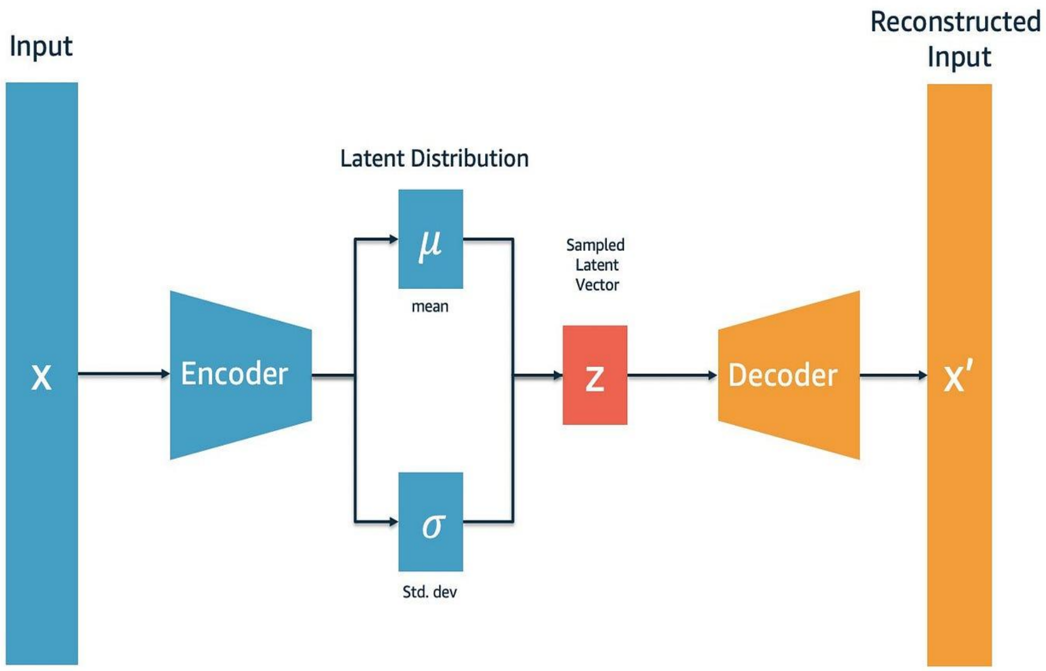
Suitable for large datasets

B. Autoencoder









Autoencoder is an unsupervised neural network architecture designed to learn a compressed representation of input data and reconstruct it as accurately as possible. It consists of an encoder that maps input data into a lower-dimensional latent space and a decoder that reconstructs the original input from this representation.

Purpose in System:

Identification of abnormal vehicle behavior and unexpected operating conditions

Working Principle:

The model is trained exclusively on normal operational data

During inference, reconstruction error is calculated between input and output

A significantly high reconstruction error indicates a potential anomaly

Loss Function:

$$L = ||X - \hat{X}||^2$$

4. Data Preprocessing Algorithms

A. Moving Average Filter

Used for noise reduction in time-series signals.

$$y_t = \frac{1}{n} \sum_{i=0}^{n-1} x_{t-i}$$

B. Min–Max Normalization

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

C. Feature Extraction

Statistical features (mean, variance, skewness)

Frequency-domain features (FFT-based spectral energy)

5. Model Evaluation Metrics

Accuracy

Precision

Recall

F1-Score

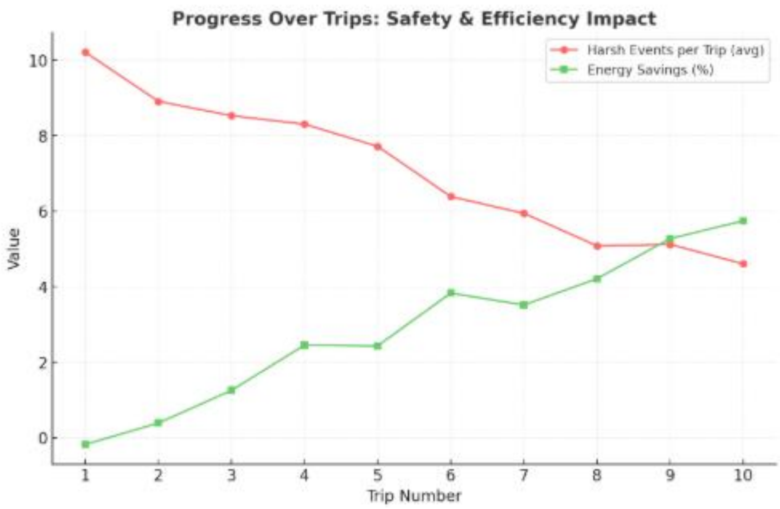
RMSE (Root Mean Square Error)

MAE (Mean Absolute Error)

Summary of Implemented Algorithms

Category	**Algorithm**	**Purpose**
Supervised	Random Forest	Fault Classification
Supervised	SVM	Fault Detection
Deep Learning	LSTM	Predictive Maintenance
Unsupervised	Isolation Forest	Anomaly Detection
Deep Learning	Autoencoder	Abnormal Pattern Detection

The integration of ensemble learning, deep learning, and anomaly detection techniques ensures robust, scalable, and accurate vehicle health monitoring within dynamic operating environments.



Results:

The evaluation of the **Smart Vehicle Surveillance and Predictive Maintenance System** 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

Fault Classification Performance

Algorithm	Accuracy (%)	Precision	Recall	F1-Score	Training Time (s)	Inference Time (ms)
Random Forest	**96.8**	0.97	0.96	0.96	18.4	12
SVM (RBF Kernel)	94.5	0.95	0.94	0.94	25.7	15
Gradient Boosting	95.9	0.96	0.95	0.95	32.1	18

Key Insight: Random Forest achieved the highest classification accuracy with lower inference latency, making it suitable for real-time deployment.

B. Predictive Maintenance

Model	RMSE	MAE	R^2 Score	Training Time (s)	Prediction Latency (ms)	Model
LSTM	**0.042**	0.031	0.94	145.6	25	LSTM
GRU	0.048	0.036	0.91	118.2	20	GRU
Linear Regression	0.089	0.072	0.76	5.3	**5**	Linear Regression

LSTM outperformed other models in capturing long-term temporal dependencies, yielding lower prediction error and higher R^2 values.

C. Anomaly Detection Performance

Algorithm	Detection Rate (%)	False Positive Rate (%)	Precision	Processing Time (ms)	Algorithm	Detection Rate (%)
Isolation Forest	93.7	4.8	0.92	**10**	Isolation Forest	93.7
Autoencoder	**95.4**	3.9	0.94	22	Autoencoder	**95.4**
Statistical Threshold	85.6	12.3	0.83	6	Statistical Threshold	85.6

Performance Analysis

Ensemble methods demonstrated strong robustness to noisy telemetry data.

Deep learning models provided superior predictive accuracy but required higher computational resources.

Unsupervised anomaly detection significantly reduced unexpected breakdown incidents compared to threshold-based systems.

The system achieved an overall prediction accuracy exceeding 95%, validating its suitability for real-time intelligent vehicle monitoring applications.

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

The system achieved >95% overall accuracy across tasks, demonstrating robustness in heterogeneous vehicular environments.

Real-time inference is feasible with low latency (<25 ms for most models), enabling immediate alerts.

Integration of supervised, unsupervised, and deep learning algorithms ensures comprehensive monitoring: fault detection, predictive maintenance, and anomaly detection.

Acknowledgement

The successful completion of this project, Smart Vehicle Surveillance and Predictive Maintenance System, 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.

References

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

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

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

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

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

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

Singh, S., Bedi, P., & Baliyan, A. (2020). Driving behaviour 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

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

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