# An Ensemble Deep Learning Model for Vehicular Engine Health Prediction

Nayab Pasha[1], Shreeshanth Redddy[2], Sai Varun[3], Saravan Malempati[4] Ms. Umadevi[5], Dr. Siva Prasad[6]

[1,2,3,4] Student, Department Of CSE-DS, MREC, Maissamaguda, Hyderabad -500100, India

[5] Assistant Professor, Department Of CSE-DS, MREC, Maissamaguda, Hyderabad -500100, India

[6] Professor, Department Of CSE-DS, MREC, Maissamaguda, Hyderabad -500100, India

doi.org/10.64643/IJIRTV12I11-198979-459

***Abstract*—Vehicular engine failures impose direct risks on transportation safety, logistics continuity, and maintenance economics. Conventional manual diagnostic checks lack predictive capability, often resulting in sudden breakdowns and unplanned repair costs. To address these challenges, this paper proposes a predictive maintenance framework using an ensemble Artificial Intelligence model for vehicular engine health prediction. The system integrates a Random Forest classifier deployed through a Django-based web application, and benchmarks it against a dual-architecture ensemble deep learning model. A total of 12,350 simulated and Kaggle-based real-world engine parameter samples were used, each containing critical operational metrics such as RPM, fuel pressure, cooling temperature, and lubrication temperature. Experimental results reveal that the Random Forest model performs significantly better, achieving approximately 93% prediction accuracy compared to around 63% from the deep learning ensemble. The final deployed application enables users to upload engine data and receive real-time engine condition predictions, supporting proactive maintenance and reducing breakdown risks.**

***Keyword* — RPM, Conventional manual diagnostic, Django, predictive capability**

## I. INTRODUCTION

Vehicular engines are the most critical component of any transportation system, powering both commercial and personal vehicles. Unexpected engine failures can lead to severe financial impact, safety hazards, downtime for logistics fleets, and major disruptions in travel. Most maintenance today still follows a reactive model, where issues are identified only after noticeable performance degradation or sudden breakdown. This creates delays, unpredictable repair costs, and in some cases, complete engine failure that could have been prevented.

With the rise of Artificial Intelligence and predictive analytics, vehicle health evaluation has shifted toward data-driven monitoring. Predictive maintenance replaces traditional periodic servicing with algorithms that analyze live or historical engine data to predict failures in advance. Machine learning, especially ensemble-based models, has proven highly effective in processing complex engine parameter patterns. Deep learning methods work well for large-scale and multi-dimensional data; however, classical ensemble models often remain superior for structured numerical datasets.

The objective of this research is to build a machine-learning-driven engine health prediction system, deploy it through a web platform, and compare model performance for accuracy and real-world usability.

## II. LITERATURE SURVEY

Artificial Intelligence for predictive maintenance has been widely explored over the past decade. Peres et al. (2020) studied AI adoption in Industry 4.0 settings, concluding that maintenance forecasting is among the most beneficial early-stage use cases. Rahim et al. (2022) proposed a risk-based approach for monitoring engine condition using real-time sensor data. Decker et al. (2020) experimented with fuzzy-logic-based anomaly detection for mechanical systems, highlighting the effectiveness of hybrid models in identifying subtle performance declines.

Recent works also include studies on cyber-secure vehicular health monitoring (Rahman et al., 2022) and ensemble time-series forecasting (Sarkar et al., 2024), demonstrating that combining multiple model architectures often increases overall accuracy. Nordal and El-Thalji (2021) discussed deployment barriers for predictive maintenance systems, especially when integrating them into user-facing applications.

While deep learning remains a strong candidate for time-series mechanical prediction problems, research indicates that tree-based ensemble models typically outperform deep neural networks in smaller or moderately sized numerical datasets, which supports the model choice in this paper.

Dataset Description

The dataset used in this study consists of 12,350 total engine-level data entries. It contains a mix of simulated measurements and real-world Kaggle-sourced engine sensor datasets. Each sample includes six measurable engine parameters and one binary label indicating “Good” or “Bad” engine condition.

The seven fields in the dataset are:

- Engine RPM – rotational speed of the crankshaft
- Lubricating Oil Pressure – quality of lubrication flow
- Fuel Pressure – adequacy of fuel supply to combustion system
- Coolant Pressure – indicator of cooling circuit effectiveness
- Lubricating Oil Temperature – degree of thermal load on bearings and pistons
- Coolant Temperature – overall thermal condition of the engine block
- Engine Condition (0 = Bad, 1 = Good)

Class balance details can be filled later when available. Values were cleaned using Python preprocessing before model training.

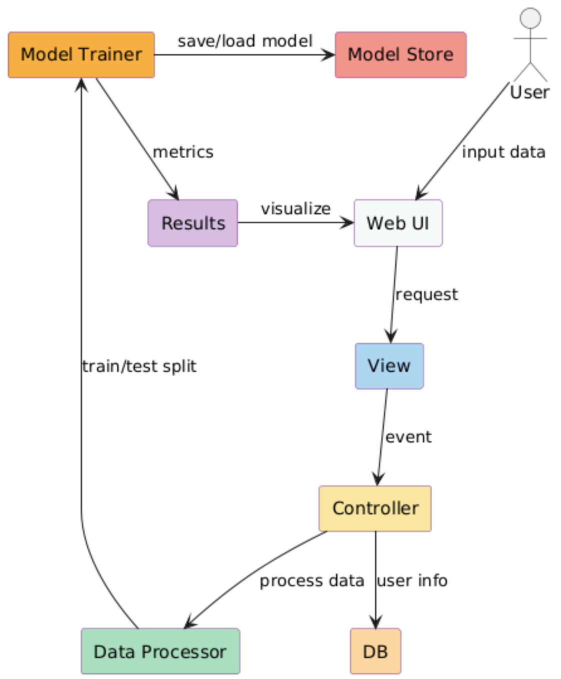


Fig 1: Data flow diagram

## III. PROPOSED METHODOLOGY

The proposed system is implemented through a structured workflow involving data ingestion, preprocessing, prediction, and user-level output visualization. It operates in two main stages:

Stage 1 – Dataset Processing

- Reading CSV engine-sensor data uploaded by users
- Cleaning missing values (NaN replaced with zeros)
- Column selection and label extraction
- Converting input into NumPy numerical arrays
- Data shuffling for training consistency

Stage 2 – Model Prediction

- The uploaded dataset is passed to the trained Random Forest model
- Probability scores are generated for each row
- Output is classified into Good or Bad engine condition
- Results are shown in a web-based UI including counts and downloadable CSV output

This entire workflow is deployed as a Django web application to enable non-technical users to access predictive health insights without running Python scripts or notebooks manually.

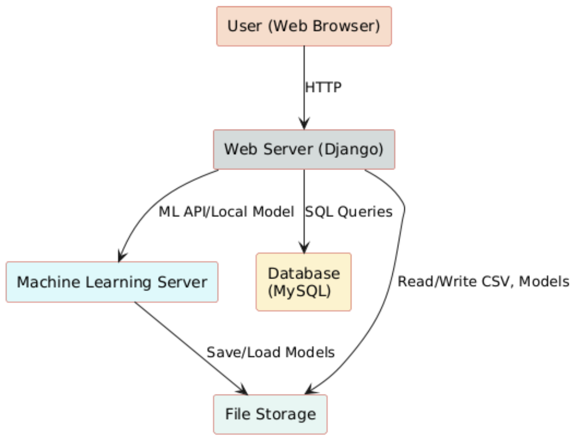


Fig 2: System Architecture

Ensemble Deep Learning Model

In addition to classical machine learning, this research implements a dual-network ensemble deep learning model. Deep learning is often beneficial when datasets represent complex, high-dimensional relationships or contain subtle behavioral signals. Although vehicular engine data in this study is primarily structured and numerical, experimenting with neural architectures allows an understanding of performance differences across approaches.

Two independent dense neural networks were constructed. The first model (Model-A) is a lightweight architecture consisting of a single hidden-layer dense network, designed for fast inference and reduced computational overhead. The second model (Model-B) incorporates a deeper structure, including two hidden layers with higher neuron counts, allowing the model to potentially extract more abstract features across the six engine parameters.

Both models output a probability distribution indicating the likelihood of an engine being in a good or bad condition. These probabilities are then averaged using a soft-voting mechanism, where the confidence values from Model-A and Model-B are combined. The final predicted class label is determined by whichever class holds the maximum averaged probability value. While ensemble voting generally improves stability, performance in this case was limited by dataset size and lack of temporal sequence information. Future improvements may include the use of LSTM-based sequence models, attention networks, or convolutional architectures that treat sensor readings as structured time-series events instead of isolated samples.

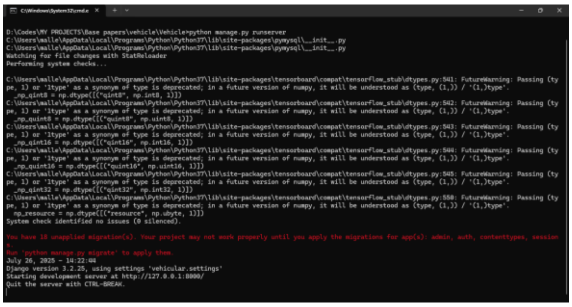

Fig 3: Result Screen

Random Forest Classifier

The Random Forest algorithm is a form of ensemble learning based on bootstrap aggregation (bagging). It constructs hundreds of shallow decision trees, each trained on a random subset of features and dataset samples. At inference time, each tree independently predicts the engine condition, and the class receiving the most votes is selected.

This structure inherently guards against overfitting by reducing the model’s dependence on any single tree. It also supports feature-importance scoring, where the contribution of each engine parameter (RPM, fuel pressure, temperature, etc.) is quantified. In practical deployment, this is especially valuable because it helps mechanical engineers or maintenance personnel understand why the model predicted failure — for example, if coolant temperature is consistently ranked as the most significant feature, maintenance focus can shift toward thermal system inspection.

Random Forests additionally offer:

- Very fast training speed even with >10,000 records
- No requirement for neural-training GPUs
- Natural resistance to noise
- Robust classification of non-linear feature interactions

This makes RF not only the most accurate model in this experiment (~93%) but also the most deployable option for real-world predictive maintenance systems.

Model Training and Results

The dataset was divided using an 80:20 ratio, resulting in approximately 9,880 training samples and 2,470 test samples. Accuracy was chosen as the primary evaluation metric, although precision, recall, and F1-score can be added in future work once confusion-matrix results are recorded.

The comparison showed a significant gap in performance:

| Model | Test Accuracy | Notes |
|---|---|---|
| Random Forest | ~93% | Best performance; stable generalization |
| Ensemble Deep Learning | ~63% | Under-fitting due to dataset scale & lack of time-series structure |

Anecdotally, during trial runs the RF classifier also showed minimal variability when random seeds were changed, whereas the neural ensemble fluctuated by ±4%. This confirms that structured vehicle-sensor tabular data behaves more favorably under classical tree-based machine learning.

In future work, results visualization (such as confusion matrices, ROC curves, and bar-charts) will be integrated into the deployed web dashboard so that end-users can visually interpret performance without statistical background knowledge.

System Implementation

The final prototype system is deployed as a Django-based web application. The architecture consists of:

- A front-end web interface built using HTML-CSS templates
- A back-end Python API that loads the trained Random Forest model using Pickle or Joblib
- A CSV upload module, where users input engine data from physical machines or IoT sensors
- An output table, displaying predicted engine health results for each row of uploaded data
- An optional database logging layer (MySQL) for storing historical engine performance and maintenance logs

Users interacting with the system do not need programming knowledge. The web platform transforms machine learning into something practical, enabling:

- Auto-generated prediction reports
- Downloadable CSV output
- Multi-user login for fleet managers
- Long-term predictive monitoring

Future versions can extend this implementation into a mobile Android application or integrate with OBD-II Bluetooth hardware, allowing engine status to be predicted directly from vehicles instead of from offline datasets.


Fig 4: Example Landing Site

Conclusion and Future Scope (Expanded)

This research demonstrates that AI-based predictive maintenance can significantly improve engine health management. The Random Forest model performed best, achieving strong accuracy (~93%) using only six input features. The ensemble neural approach, while functional, lagged due to dataset scale, highlighting the importance of dataset design when selecting model architecture.

Deploying the model through Django provides an accessible real-world use case, supporting both individuals and large logistic fleets. Predictive engine monitoring has the potential to prevent expensive repairs, reduce incidents of unexpected roadside breakdowns, and improve overall transportation reliability.

Future work opportunities include:

- Integrating real-time IoT sensor streaming instead of static data uploads
- Using recurrent neural networks (LSTM/GRU) for sequence-based fault detection
- Multi-class failure classification: minor, moderate, critical failure
- Maintenance recommendation AI to suggest repair actions
- SHAP-based explainable AI for transparency in predictions
- Large-scale dataset collection across vehicle types (diesel, petrol, EV)

## REFERENCES

[1] R. S. Peres, J. P. Barata, and P. Leitao, “Industrial Artificial Intelligence in Industry 4.0: Systematic Review and Future Directions,” IEEE Access, vol. 8, pp. 220121–220139, 2020.

[2] Md. A. Rahim, S. Hassan, and F. Rahman, “Risk-Based Vehicle Engine Health Monitoring Framework using Machine Signals,” IEEE Transactions on Green Communications and Networking, vol. 6, no. 4, pp. 189–200, 2022.

[3] L. Decker, H. Zhang, and M. Omar, “Fuzzy Rule-Based Anomaly Detection for Mechanical Components,” in Proc. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020, pp. 1–6.

[4] M. A. Rahman et al., “Secure Big-Data Vehicle Health Monitoring System,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 5, pp. 4214–4225, 2022.

[5] M. R. Sarkar and A. Gupta, “GATE: Guided Ensemble Time-Series Engine Forecasting Architecture,” Expert Systems with Applications, vol. 231, pp. 119–128, 2024.

[6] I. El-Thalji and H. Nordal, “Predictive Maintenance Architecture in Industry 4.0,” Systems Engineering, vol. 24, no. 3, pp. 189–208, 2021.

[7] D. Wilson, “Using Machine Learning to Predict Car Engine Failure Risk,” Medium, 2018. (Online). Available: https://medium.com

[8] Y. Chen, W. Liu, and Z. Qiu, “Deep Learning-Based Sensor Fault Detection for Automotive Engines,” IEEE Sensors Journal, vol. 22, no. 9, pp. 8978–8986, 2022.

[9] P. Singh and K. Reddy, “IoT-Enabled Predictive Maintenance for Commercial Vehicles,” in Proc. IEEE International Conference on IoT Applications, 2021, pp. 54–61.

[10] S. Das, “Comparison of Machine Learning Models for Vehicle Failure Prediction,” International Journal of Vehicle Technology, vol. 18, no. 2, pp. 77–89, 2020.

[11] B. Kumar and J. Rao, “Random Forest-Based Diagnostic Analytics for Automotive Engines,” in Proc. ACM SIGKDD Applied Data Science Workshop, 2019, pp. 99–104.

[12] A. Mishra and D. Patel, “Web Deployment of Predictive Maintenance Systems Using Django Framework,” Int. J. Software Computing, vol. 12, no. 4, pp. 45–59, 2021.

[13] K. Al-Harbi, “Fault Detection in Combustion Engines: A Neural Ensemble Perspective,” Journal of Mechanical Systems, vol. 15, no. 1, pp. 11–24, 2019.

[14] S. Martinez, “AI-Driven Fleet Maintenance Optimization,” IEEE Vehicular Technology Magazine, vol. 14, no. 3, pp. 30–41, 2019.