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.

An Intelligent Predictive Maintenance Framework for Improving Military Vehicle Readiness Using Machine Learning

Ahmed Alamri
University of Jeddah
Nahla Aljojo
nmaljojo@uj.edu.sa
University of Jeddah
Araek Tashkandi
University of Jeddah
Amjad Al Bishri
University of Jeddah
Ghofran Al Ahmadi
University of Jeddah

Research Article

Keywords: Predictive Maintenance, Military Vehicle Readiness, Machine Learning, Engine Fault Detection, Ensemble Learning

Posted Date: September 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11042037/v1

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

Additional Declarations: No competing interests reported.

An Intelligent Predictive Maintenance Framework for Improving Military Vehicle Readiness Using Machine Learning

[1]Ahmed Alamri, [2]Nahla Aljojo , [3]Araek Tashkandi, [4]Amjad Al Bishri, [5]Ghofran Al Ahmadi
[1,2,3,4,5] *Department of Information Systems & Technology, College of Computer Sciences & Engineering, University of Jeddah, Jeddah, Saudi Arabia*
asalamri4@uj.edu.sa, nmaljojo@uj.edu.sa, , astashkandi@uj.edu.sa, AALBISHRI0104.stu@uj.edu.sa, GALAHMADI0010.stu@uj.edu.sa

Nahla Aljojo (corresponding author)

Abstract
Engines in military vehicles work in very harsh environments and situations, which can cause unexpected failures, unscheduled downtime, higher maintenance costs, and lower mission readiness when typical preventive or corrective maintenance methods are used. This study seeks to provide an intelligent predictive maintenance framework that improves military vehicle readiness by precisely identifying engine defects, reducing undetected failures, improving maintenance scheduling, and facilitating proactive, data-driven logistics decision-making. We used the Automotive Vehicles Engine Health Dataset, which has 19,535 data, to create a structured machine learning pipeline. The process encompassed exploratory data analysis, outlier elimination, feature scaling, class-weighted learning, stratified data partitioning, and supervised model training. We trained and tested four machine learning models—Random Forest, Gradient Boosting, AdaBoost, and K-Nearest Neighbors—using accuracy, precision, recall, F1-score, and ROC-AUC. After tweaking the hyperparameters, Gradient Boosting had the best overall performance, with a recall of 97.4%, an F1-score of 0.816, an accuracy of 70.0%, and a precision of 70.3%. This shows that it was better at finding faults with very few missed engine failures. AdaBoost had the highest recall at 98.9%, but it also had the lowest accuracy at 69.0% and the highest F1-score at 0.814. This means that it was more sensitive but had more false alarms. The F1-score for K-Nearest Neighbors was 0.799, the recall was 88.4%, and the accuracy was 69.0%. The F1-score for Random Forest was 0.770, the recall was 76.0%, and the precision was 78.0%. The Gradient Boosting model also had the most consistent ROC curve, which means it was better at telling the difference between thresholds. This research provides a replicable, comprehensive predictive maintenance framework specifically designed for military vehicle logistics, illustrating how optimal machine learning models may enhance operational readiness, minimize downtime, and facilitate dependable, data-informed repair decisions.

Keywords: Predictive Maintenance; Military Vehicle Readiness; Machine Learning; Engine Fault Detection; Ensemble Learning

1. Introduction

Military logistics is very important for making sure that troops are ready to go, that missions go on without a hitch, and that they are effective on the battlefield. Modern military operations rely largely on ground vehicles that need to work well in very unpredictable and extreme conditions, such as very hot or very cold weather, dust, humidity, big loads, and protracted missions. In these kinds of situations, engines might fail without warning, which can lead to unscheduled vehicle stops, higher maintenance costs, mission delays, and possible dangers to both troops and military assets [4].

Preventive and corrective maintenance are two common maintenance methods that don't always work well since they depend on set timetables or fixing things after they break. Predictive Maintenance (PdM) is a data-driven alternative that uses Artificial Intelligence (AI) [1] and Machine Learning (ML) approaches to predict when problems can happen [2]. PdM uses engine condition data to help make smart maintenance decisions by finding signs of wear and tear early on. Recent developments in ensemble-based machine learning techniques have exhibited significant proficiency in modeling intricate, non-linear interactions within sensor-derived datasets for engine performance evaluation [3], [4].

This study develops a transferable predictive maintenance framework inspired by military vehicle logistics requirements using the publicly available Automotive Vehicles Engine Health Dataset. via machine learning models applied to the Automotive Vehicles Engine Health Dataset. The suggested method follows a full data science pipeline that includes exploratory data analysis, data cleaning, dealing with outliers, fixing class imbalance, preprocessing features, splitting data, training models, setting hyperparameters, and evaluating performance. We look at several machine learning methods, including Random Forest, Gradient Boosting, AdaBoost, and K-Nearest Neighbors (KNN), to find the best way to find engine faults. The major goal is to make defect detection as sensitive as possible while keeping the level of accuracy acceptable.

More and more military groups are using data analytics to make their fleets more ready, cut down on downtime, and organize logistics better. Military vehicle engines are generally serviced or repaired only after they break down, which can cause failures that could have been avoided, make missions less available, and cost too much to maintain. Predictive maintenance solves these problems by using machine learning to predict engine breakdowns before they happen.

This study employs an Engine Health Condition dataset of about 19,000 engine records, which include operational sensor values and categorized engine states. The collection allows for systematic examination of degradation trends and the identification of crucial indicators linked to imminent failures. The

study helps military logistics decision-makers improve the accuracy of failure predictions, optimize maintenance scheduling, and extend engine service life by creating a data-driven predictive maintenance framework. In the end, predictive modeling makes operations safer and makes the military more ready by turning engine health data into useful maintenance information.

Military vehicles have to deal with a lot of tough situations, like high temperatures, sand, humidity, heavy loads, and missions that never end. These conditions make it much more likely that mechanical problems will happen suddenly and often. Current maintenance practices, which are mostly based on schedules or reactions, cause unplanned vehicle downtime during important missions, lower logistical readiness, higher maintenance costs because of emergency repairs, poor spare parts management, and trouble predicting maintenance needs [4]. The main goals of this research are:

To improve the operational readiness of military vehicles by using predictive maintenance approaches to cut down on unplanned breakdowns and unexpected downtime.

To make maintenance more efficient by moving from traditional preventive maintenance to condition-based predictive maintenance.

To cut down on total maintenance expenses by doing fewer emergency repairs and not keeping extra parts on hand that aren't needed.

To prolong the lifespan of equipment by identifying potential engine breakdowns early on, before they become serious problems.

This study offers a thorough, data-driven predictive maintenance system specifically designed for military vehicle engine logistics. It incorporates stringent preprocessing, statistical analysis, machine learning modeling, and automated evaluation in a complete Python-based implementation. The suggested pipeline focuses on controlling feature transition, making it easy to reproduce, and getting ready for deployment. This makes it useful for future engine monitoring and maintenance systems. In general, our work gives military logistics a repeatable computational base that improves predictive maintenance by making it easier to detect faults, cutting down on downtime, and making data more accessible to decision-makers in charge of military vehicle fleets.

The primary contribution of this research is not the invention of a new machine learning algorithm, but the development of an integrated predictive maintenance framework tailored for military logistics environments. The framework combines robust preprocessing, imbalance-aware learning, comparative model evaluation, and fault-sensitive optimization under noisy operational conditions. The study further contributes by emphasizing operational readiness and maintenance decision support in mission-critical contexts."

2. Literature Review

Predictive maintenance research has focused on cars and engines. Recently, the emphasis has turned to works involving the maintenance of vehicles used in military logistics. Conventional maintenance approaches based on fixed schedules by AI, ML, and DL. By employing predictive methods that are datadriven and use sensor and diagnostic data to foresee the failures ahead of time. Current research has demonstrated the application of different models by classical classifiers, such as Logistic Regression and Random Forest. The aim is to advanced ensemble and deep learning techniques for engine fault prediction, estimating life, remaining useful life and fleet management solutions. In the military sector, predictive analysis has proven to be vital, since reliable engine performance is directly linked to success, safety and cost efficiency.

Mykich et al. [1] explores the use of artificial intelligence and machine learning to improve predictive maintenance of Armoured Fighting Vehicle (AFV) engines in military logistics. Researchers used the Automotive Vehicles Engine Health Dataset, which has over 19,000 entries including critical metrics like engine RPM, lubricating oil pressure, fuel pressure, coolant pressure, oil and coolant temperatures and engine health. Several classification models were evaluated, including Random Forest, Decision Tree, Gaussian Naïve 2 Bayes, Logistic Regression, K-Nearest Neighbors, AdaBoost and Support Vector Machine. The performance was based on accuracy, precision, recall, and F1-score. The AdaBoost Classifier achieved the greatest accuracy of 85.93%, followed by SVM at 85.52% and KNN at 85.40%, showing that the ensembled and advanced ML models may increase mission preparation and minimize military vehicle failures.

Narayanan et al. [2] explored the potential of AI-assisted predictive maintenance for Armoured Fighting Vehicle (AFV) engines to improve military readiness and minimize malfunctions during the operations. In training cycles, oil pressure sensors, vibration sensors, and coolant temperature sensors were fitted to Indian Army tanks to gather data. A dataset of 35 tanks was developed, and statistical analysis such as regression and correlation models were undertaken to measure correlations of sensor parameters and engine hours. The models exhibited robust results, with correlation coefficients of 0.9656 (engine hours–vibration), 0.772 (engine hours–coolant temperature), and -0.9215 (engine hours–oil pressure). In addition, the regression analysis yielded an R^2 value of 0.9322, indicating that approximately 93.22% of the variance in vibration can be explained by engine operating hours. The finding proves that AI prediction models are capable in facilitating maintenance and providing a foundation for future AI-based Remaining Useful Life (RUL) estimation in AFVs

Sengupta et al. [3] proposes an ensemble machine learning framework for the predictive maintenance of armoured vehicles to enhance reliability and reduce downtime. The authors utilized the AI4I 2020 Predictive Maintenance Dataset, which contains 10,000 samples and six significant features such as air temperature, process temperature, rotational speed, torque, and tool wear. After preprocessing and balancing the dataset with SMOTE, different classifiers like LightGBM, Random Forest, Decision Tree, Extra Trees, Gradient Boosting, KNN, and AdaBoost were attempted. The performance was evaluated using Accuracy, AUC, precision, recall, and F1-score. The results indicated that the proposed ensemble model was superior to all single classifiers, with an F1-score of 0.9941, 99.03% recall, 99.80% precision, and 98.93% accuracy. This model was determined to be highly effective for the early detection of faults and the scheduling of maintenance for military ground vehicles.

Nagappa et al. [4] improve predictive maintenance in the automotive industry. The database included critical engine parameters, like RPM, lubricating oil pressure, fuel pressure, coolant pressure, and oil and coolant temperatures. The proposed model is Random Forest, Decision Tree, and K-Nearest Neighbors (KNN), with each providing complementary

advantages: Random Forest was 84% accurate, KNN was 80%, and Decision Tree was 76%. In an ensemble, the approach enhanced prediction stability, confidence, and accuracy. It was evaluated based on accuracy, precision, recall, and F1-score, with the ensemble model outperforming individual algorithms, it reflects its value in real-time tracking and proactive scheduling of maintenance in vehicular systems.

Mun et al. [5] introduces an ensemble learning model to improve predictive fault detection and health monitoring of U.S. Army ground vehicles. The data comprised operational sensor measurements and fault codes from more than 3,000 vehicles, along with variables such as battery voltage, lubricant pressure and temperature, and coolant temperature. The authors designed a multivariate LSTM model to improve generalizability. Performance was measured with MAPE, RMSE, and R^2. The findings showed that ensemble LSTM outperformed individual LSTM, reducing average test MAPE to approximately 0.34, and successfully identified battery degradation as early as seven days before replacement. Therefore, the model is effective for fault detection and predictive maintenance.

Dalzochio et al. [6] introduces MILPdM, a military-specific predictive maintenance architecture, which integrates data acquisition, processing, communication, and decision-making. It focused on examination of military vehicle model that had vibration sensors installed to simulate real-world conditions, such as flat tire. The data preprocessed as time-series features and used Random Forest (RF) and Long Short-Term Memory (LSTM) networks. Evaluation was done using Root Mean Squared Error (RMSE) on different prediction horizons. Results shows RF model achieved the lowest error RMSE of 0.05951 when applied to the tire failure scenario, outperforming the LSTM configuration RMSE of 0.0707. Therefore, MILPdM applicable for mission-critical predictive maintenance in military applications.

Kuiper et al. [7] presents a generative learning approach for predictive maintenance in U.S. Army ground vehicle fleets using data from Army's Condition-Based Maintenance (CBM), which consists of millions of time-series sensor readings and fault codes for different types of vehicles. They developed a hybrid model by combining DeepAR, Spatio-Temporal Attention Mechanism (STAM), and a Variational Autoencoder (VAE) to forecasts whether a vehicle fault is likely to occur and estimates time-to-first fault. Evaluation was based on ROC/AUC for fault classification and R^2 for fault-timing regression regression. Results showed that hybrid DeepAR+ STAM+ VAE operated at an AUC of 0.978, outperforming baseline LSTM of 0.520 and STAM-only of 0.458, and an R^2 of 0.77 with fault timing predictions within ±4 minutes. Therefore, it highly effective for early and actionable fault detection in military fleets.

Saraswat et al. [8] research suggests predictive maintenance of automotive engines through real-time sensory data such as RPM, oil pressure, coolant pressure, and temperature parameters. The authors preprocessed the dataset of over 19,000 engine health records of the Automotive Vehicles Engine Health Dataset through normalization, outlier removal, and class balancing using strategies such as SMOTE. Several models were experimented, including Logistic Regression, Random Forest, XGBoost, and LSTM neural networks. The model performance was evaluated by accuracy, F1-score, and ROC-AUC. It was observed that the LSTM model worked best with 95.2% accuracy and a ROC-AUC of 0.968. The result illustrates the potential of deep learning in sequential degradation trends in order to realize reliable early fault detection with high potential for reducing maintenance costs and downtime.

The study by Errezgouny et al. [9] is about a hybrid deep learning approach for vehicle predictive maintenance using On-Board Diagnostics (OBD) data. Thus, the dataset contains over 120,000 time series data points from 14 cars, that include 17 attributes including RPM, engine load, throttle position, and coolant temperature. The researchers used Long Short-Term Memory (LSTM) neural networks in conjunction with K-means clustering to address the issue of unlabeled datasets and to identify operational patterns and temporal relationships. Then, the models have been evaluated by using MAE, MSE, RMSE, and R^2. Therefore, the results indicated that the hybrid LSTM + K-means model performed better than individual LSTM, GRU, and RNN methods, with an R^2 of 97.5% in engine load predictions and the lowest MAE (0.032), indicating better generalization and highlight the effectiveness for real-time condition monitoring and predictive maintenance in automotive systems.

Mahiyudin et al. [10] presents a machine learning model that forecasts vehicle maintenance needs using a dataset of 50,000 vehicle histories. The model was subsequently resampled to 80,996 samples using SMOTE to address class imbalance. The features in the data include sample engine capacity, petrol type, age, mileage, accident history, and service history. Therefore, numerous models were executed, for instance, Decision Tree, Random Forest, Gradient Booster, Naïve Bayes, and KNN, as well as ensemble experiments. Accuracy, precision, recall, and F1-score were employed to evaluate performance. Thus, results showed Decision Tree with the highest accuracy of 99.97% with 100% recall in the imbalanced dataset, while Gradient Booster had an accuracy of 99.58% in the balanced dataset, demonstrating the capability of ML to minimize downtime and optimize maintenance scheduling.

Ojeda and Gonçalves [11]. The study presented a machine learning-based approach to reduce unanticipated delays in automobile manufacture by incorporating predictive maintenance into the production process. The work applied Analytic Hierarchy Process (AHP) for equipment prioritization and Process Failure Mode and Effects Analysis (PFMEA) for detection of high-risk failure modes. Multivariate machine learning algorithms like Gradient Boosting, KNN, SVM, Decision Tree, and Linear Regression were employed by researchers using actual production process data like failures and outages, Mean Time To Repair (MTTR), and Mean Time Between Failures (MTBF). Based on the result, Random Forest was the best-performing model as the model with around 80% accuracy for predicting failure cycles, although the performance of all the models was acceptable. The results suggest the possibility of forecast models to make production more sustainable, increase efficiency, and reduce unplanned outages via the avoidance of wasteful energy and maintenance.

3. Methodology

The study uses a systematic multi-phase technique to ensure data integrity, analytical accuracy, and modeling readiness in the examination of the Automotive Vehicles Engine Health Dataset. Before the model building, the dataset was rigorously examined for any potential quality issues such as missing values, duplicate

records, outliers, and class imbalance. It is crucial to address these difficulties since engine health sensor data is generally degraded by noise, extreme measurements, transient spikes, and inconsistent readings due to hard operating conditions or sensor malfunctions. The technique combines the processes of data collection, data preparation, model building, model testing and model deployment as shown in Figure 1. This end-to-end approach ensures that all the steps in an experiment are the same and helps in building robust and reproducible machine learning models for predictive maintenance.

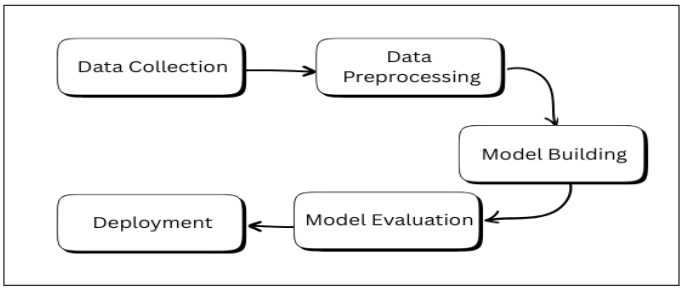


Figure 1: Research Methodology Workflow

3.1 Data Collection and Preparation

The dataset utilized in this research consists of 19,535 labeled observations, featuring seven numerical parameters that denote engine operational characteristics and health status. Six of the qualities are essential measures from engine sensors, while the seventh is the engine status label. Table 1 gives a full description of each attribute.

Table 1: Summary of Related Work in Predictive Maintenance for Vehicles.

Feature	Description
Engine rpm	Engine rotational speed (revolutions per minute),
Lub oil	Lubricating oil pressure, reflecting the lubrication
Fuel pressure	Fuel supply pressure, representing fuel injection
Coolant	Coolant system pressure, associated with thermal
Lub oil	Temperature of the lubricating oil, indicating
Coolant	Engine coolant temperature, representing overall
Engine	Target variable indicating engine health status (0 =

All attributes are continuous numerical variables except for the target, which is categorical but stored as an integer, where 0 indicates normal engine operation and 1 indicates a faulty engine condition. The data used in this study are publicly available in Kaggle and serve as a research benchmark for predictive maintenance modelling. Table 2 illustrates a sample of the dataset, where each observation represents a snapshot of engine operating behavior derived from multiple sensor measurements.

Table 2. Engine Condition Value Counts

Index	Engine rpm	Lub oil pressure	Fuel pressure	Coolant pressure	Lub oil temp	Coolant temp	Engine Condition
0	700	2.493592	11.790927	3.178981	84.144163	81.632187	1
1	876	2.941606	16.193866	2.464504	77.640934	82.445724	0
2	520	2.961746	6.553147	1.064347	77.752266	79.645777	1
3	473	3.707835	19.510172	3.727455	74.129907	71.774629	1
4	619	5.672919	15.738871	2.052251	78.396989	87.000225	0

The dataset replicates engine performance and failure scenarios pertinent to military vehicle logistics; nevertheless, it does not derive from classified or active military systems. Due to national security concerns, legal limits, and operational sensitivity, real-world maintenance logs for military vehicles are not available to the public. So, the dataset was made up to look like real engine behavior and fault conditions for study reasons.

Figure 2 demonstrates that the target variable (Engine Condition) is not uniformly distributed among the classes. The collection contains roughly 12,317 defective engine samples, which is 63% of the total. There are roughly 7218 samples of normal operating conditions. This is about 37% of the total. If this discrepancy is not corrected, it can bias machine learning models toward the majority class and make detection of defects difficult. The first evaluations of the data quality found that the dataset has no missing values and no duplicate records which indicates structural completeness. However, exploratory research found a considerable incidence of outliers specifically in temperature and pressure associated variables. Such anomalous figures are common in data sets coming from sensors and may be true errors or merely noise from the measuring procedure. It is crucial to discriminate between relevant fault-related anomalies and spurious sensor spikes, because the latter can either mask relevant failure signatures or generate noise that compromises model trustworthiness.

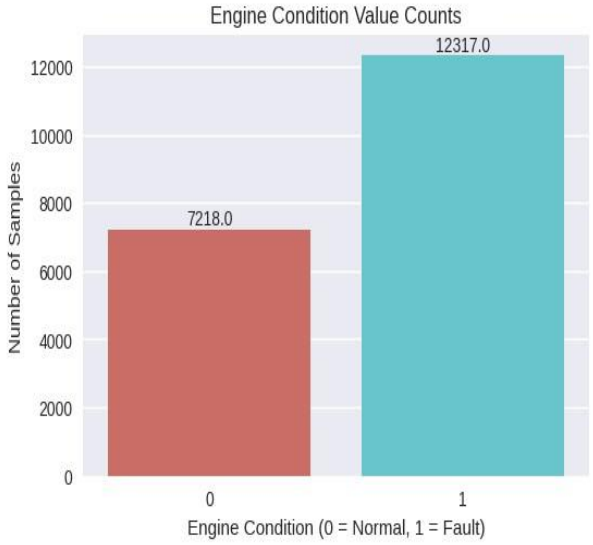


Figure 2: Engine Condition Value Counts

Before building the model, Exploratory Data Analysis (EDA) was done to look at the statistical qualities, distributional characteristics, and data quality of the Automotive Vehicles Engine Health Dataset. We used descriptive statistics to characterize central tendency, dispersion, and range for all

variables. You can see these in Table 3. The results show that engine operational characteristics, especially measurements of engine speed, pressure, and temperature, vary a lot. This is because of different operating situations and possible fault scenarios.

The dataset was first checked for structural problems to see if the data was reliable. There were no missing values or duplicate records, which means that the dataset is full and ready for machine learning study. Feature distributions showed that the ranges were not uniform and the levels of dispersion were different. This is normal for sensor-based engine telemetry collected in different operating circumstances.

We used one of the most common methods in exploratory research to find and show outliers, following the statistical rules set out by Bluman, Allan G. [13]. The Hampel filter is another method. It is a decision-based filter that finds and replaces outliers [14]. We used boxplots and the interquartile range (IQR) to look at all the continuous variables to find any extreme or strange data. Initial exploratory analysis indicated that most extreme values were operationally plausible and potentially associated with fault conditions. However, during preprocessing, a stricter hybrid filtering strategy combining IQR and Hampel filtering was applied to reduce excessive sensor noise and improve model generalization. This means that the extreme results are likely to be within acceptable operational limits and not noise or measurement errors. The situation of class imbalance was handled in accordance with [15].

Overall, the EDA confirmed that the dataset was statistically sound, showed that it was good for predictive maintenance modeling, and helped with the next steps of preprocessing and choosing a model.

Table 3 Descriptive Statistics

Statistic	Engine rpm	Lub oil pressure	Fuel pressure	Coolant pressure	Lub oil temp	Coolant temp	Engine Condition
Count	19,535	19,535	19,535	19,535	19,535	19,535	19,535
Mean	791.24	3.3	6.66	2.34	77.64	78.43	0.63
Std	267.61	1.02	2.76	1.04	3.11	6.21	0.48
Min	61	0	0	0	71.32	61.67	0
25%	593	2.52	4.92	1.6	75.73	73.9	0
50% (Median)	746	3.16	6.2	2.17	76.82	78.35	1
75%	934	4.06	7.74	2.85	78.07	82.92	1
Max	2239	7.27	21.14	7.48	89.58	195.53	1

Figure 3 shows histogram plots that show how each feature in the Automotive Vehicles Engine Health Dataset is spread out. The distributions show clear non-normal patterns for most sensor variables. This is because the engine data were collected in a variety of operating circumstances. The distribution of engine rpm is right-skewed, with most observations happening at lower to mid-range speeds and fewer samples happening at higher speeds. This means that engines mostly work under modest loads and don't run at high speeds very often.

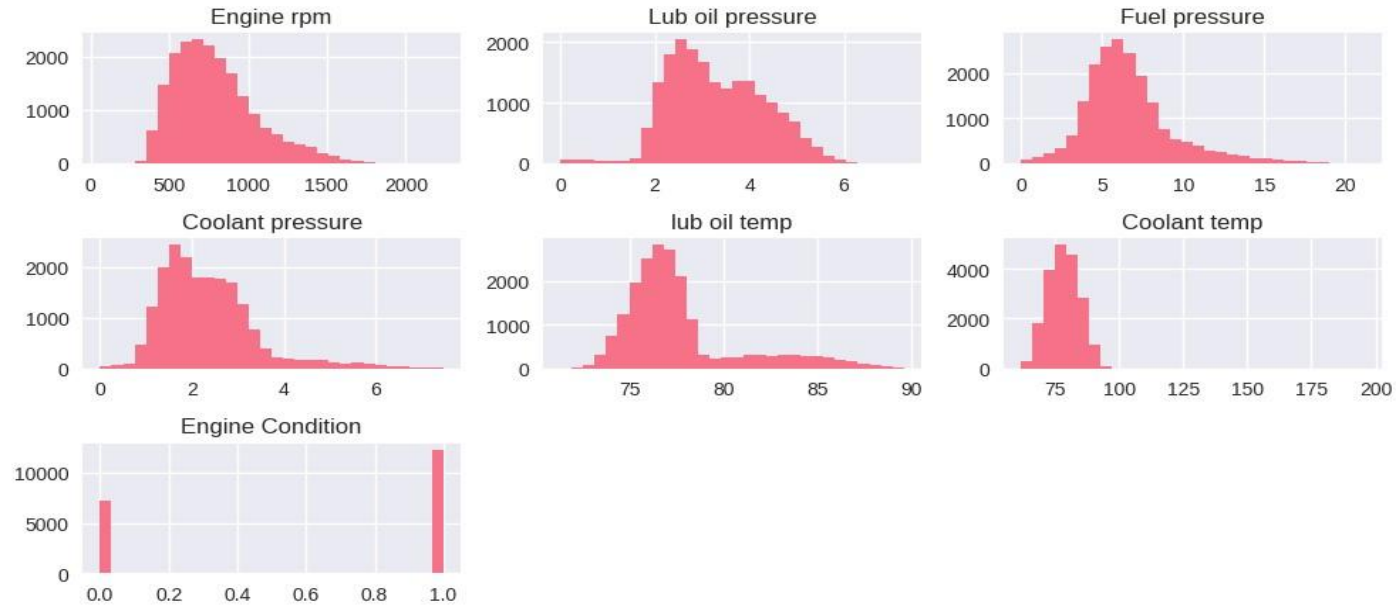


Figure 3 Feature Distribution Analysis

The distributions of lubricating oil pressure and coolant pressure are also positively skewed, with most values falling within the normal working limits and long tails leading to higher values. These tails probably have to do with temporary stress circumstances or strange operating states. The temperature of the lubricating oil also exhibits a right-skewed pattern, which means that the thermal behavior is steady during normal operation, but there are times when the temperature rises, which could mean that the oil is degrading or that there is a malfunction.

gasoline pressure has a fairly skewed distribution with a wider

range. This shows that the amount of gasoline delivered changes depending on the load on the engine. The temperature of the coolant has a high peak around normal operating values, followed by a lengthy right tail. This suggests that the thermal regulation is working well, although there are times when it becomes too hot. The target variable, Engine Condition, has two possible values: normal and defective. In general, the fact that these features are skewed shows how important it is to use strong preprocessing and non-linear machine learning models that can find complex operating patterns and deviations associated to faults.

Figure 4 shows a correlation matrix that shows how the features in the Engine Health dataset are related in a linear way. The results show that the sensor variables are mostly linearly independent, as seen by the low correlation coefficients that are close to zero. This indicates that the majority of traits provide unique information and are not significantly linearly correlated with each other.

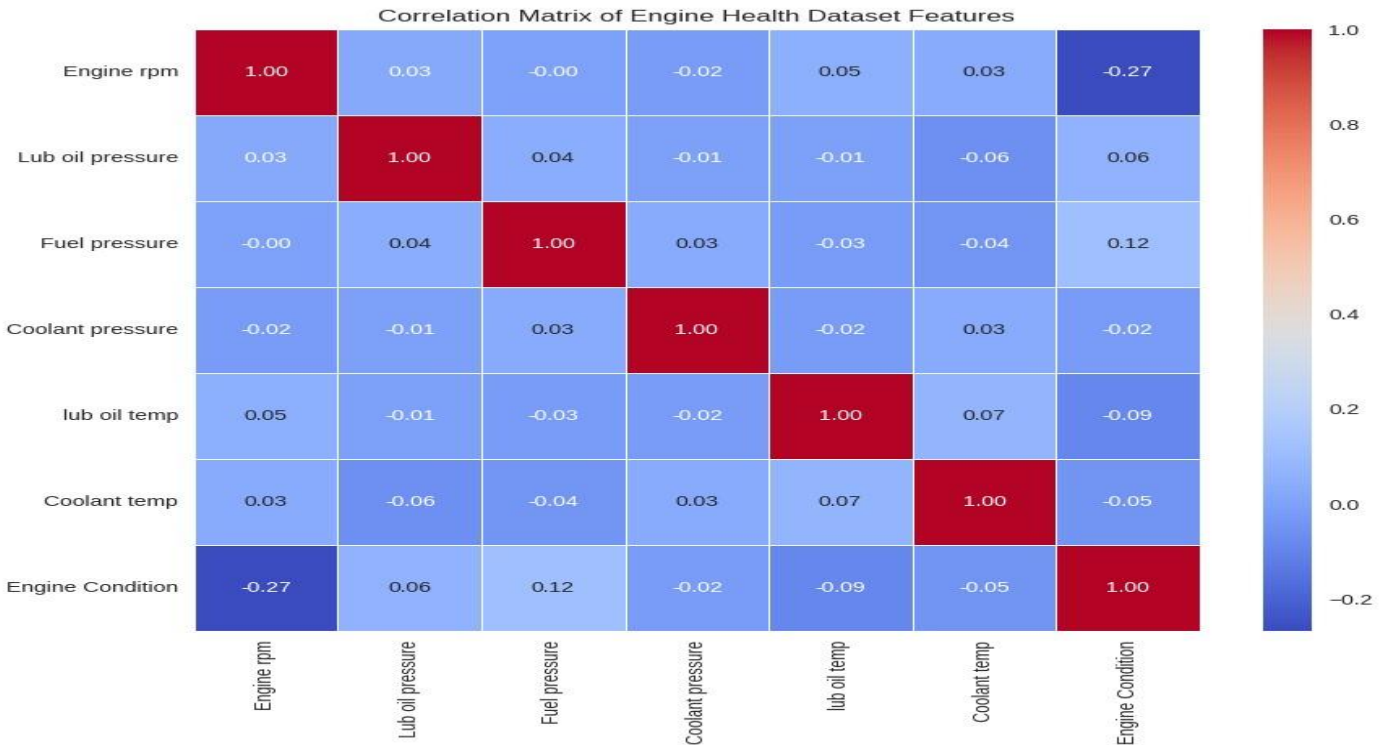


Figure 4. The Correlation Analysis Results of the Features

The objective variable, Engine Condition, has weak linear connections with the input features as well. Engine rpm has the strongest link to the target of all the variables, with a mild negative correlation of −0.27. This negative association means that the more the engine speed goes up, the less likely it is that the engine will be called faulty. Fuel pressure has the second strongest link to Engine Condition, with a modest positive correlation of 0.12. This means that when fuel pressure goes up, the chance of a defect happening goes up somewhat. There is a slight negative correlation of −0.09 between the temperature of the lubricating oil and the health of the engine. This means that the two are not directly related.

The weak correlations suggest that linear models would not be enough to effectively estimate engine condition based only on data from one sensor. Instead, it's more likely that there are complicated, non-linear interactions between features. This is why we will use better and more tailored machine learning models in the next stages of this work.

The correlation heatmap is a visual diagnostic tool that uses color intensity to show how strong the association is. Dark blue areas show negative relationships, whereas dark red areas show positive links. Together, these areas help with feature interpretation and model selection.

3.2 Justification of Adopting Models

The chosen models—Random Forest, Gradient Boosting, AdaBoost, and K-Nearest Neighbors—were used to represent several types of learning behaviors, such as ensemble, boosting, and distance-based methods. This combination makes it possible to accurately describe non-linear sensor interactions, compare performance fairly, and find faults reliably in complicated, unbalanced predictive maintenance datasets.

3.2.1 Random Forest

Random Forest is an ensemble learning technique that uses more than one decision tree to make classification more reliable and general [16–17]. It was chosen because it worked well with organized sensor data, didn't get too good at it, and could predict how engine parameters interacted with each other in non-linear ways.

AdaBoost

AdaBoost is an ensemble strategy that uses boosting to improve classification performance by focusing on data that were misclassified before [16]. It was chosen because it is very good at finding minority and hard-to-detect problem cases, which is important for predictive maintenance applications that want to avoid missing engine breakdowns..

K-Nearest Neighbors (KNN)
K-Nearest Neighbors is a classifier that doesn't need parameters and uses distance to assign labels depending on how similar a sample is to its neighbors [18]. It was used as a clear baseline to test how well engine condition data could be separated naturally and to compare performance to more sophisticated ensemble models.

Gradient Boosting
Gradient Boosting is a method that develops models one after the other to reduce prediction errors [19]. It was chosen because it can find small, non-linear patterns of degradation in engine behavior and because it works well with complicated, noisy, and high-variability predictive maintenance datasets.

3.3 Data Preprocessing
Data pretreatment was done to make sure that the dataset was good for machine learning analysis that was accurate and fair. The steps in the process were checking the quality of the data, dealing with outliers, splitting the dataset into parts, scaling the features, and dealing with class imbalance [20-23]. To keep the data safe and the modeling consistent, these processes were done in a certain order..

3.2.1 Experimental Environment
We did all of the preprocessing, modeling, and assessment work in a Python-based environment utilizing Jupyter Notebook and Google Colab [24]. We used standard data science packages to do numerical processing, statistical analysis, visualization, and machine learning [25]. An integrated pipeline structure was used to make sure that preprocessing was done the same way in both the training and testing phases and to make it possible to reproduce results.

Hyperparameter adjustment was done to make the model work better, be more stable, and be able to generalize better [26-27]. Using randomized search, we thoroughly refined the key parameters for each algorithm to find the optimum configurations that balance bias and variance (see Table 4). Adding more trees (to 424) and limiting tree depth (to 19) made the Random Forest model more stable and less likely to overfit. The Gradient Boosting model worked best with 154 estimators, which gave it the right amount of complexity without too much volatility. AdaBoost needed a lot of weak learners (398) with a learning rate of 0.38, which made it possible to fix samples that were incorrectly classified. Setting the neighborhood size to 24 for K-Nearest Neighbors made classification more stable by making it less sensitive to noise. These improved settings made sure that the model comparisons were fair and that the predictions were accurate.

Table 4. The Models Hyperparameter

Model	Hyperparameter
Random Forest	n_estimators=424, max_depth=19
Gradient Boosting	n_estimators=154
AdaBoost	learning_rate=(0.38), n_estimators=398
KNN	n_neighbors=24

3.2.2 Data Cleaning and Preparation
Initial data quality checks confirmed that the dataset contained no missing values and no duplicate records**,** indicating structural completeness [28]. Consequently, data cleaning focused on handling outliers, scaling features, and addressing class imbalance (see Table 5).

Table 5: Summary of Data Preprocessing Steps

Step	Method Applied	Outcome
Missing & Duplicate Check	Structural validation	No missing or duplicate records
Outlier Handling	IQR + Hampel filter	~30% noisy samples removed
Dataset Size	Before / After cleaning	19,535 → 13,688 samples
Data Splitting	Stratified 80:20	Balanced train–test distribution
Feature Scaling	Z-score normalization	Comparable feature magnitudes
Class Imbalance	Model-internal class weighting	Equalized class contribution

Outlier analysis showed that there were a lot of extreme values, especially in aspects linked to pressure and temperature, which is common in engine data from sensors. To tell the difference between real fault-related extremes and false sensor noise, a hybrid outlier treatment technique that combined the interquartile range (IQR) rule and Hampel filtering was used. Around 30% of the dataset was found to be noisy or not representative and was taken out, leaving a cleaned dataset with 13,688 samples. This method kept important fault characteristics while getting rid of noise that could make the model less generalizable.

After dealing with outliers, the dataset was split into two parts: a training set and a testing set. The split was 80:20 so that both sets had class distributions that were typical of the whole dataset. Then, z-score normalization (StandardScaler) was used to scale the features so that distance-based and gradient-based learning algorithms could work. There were no missing values, however median imputation was kept in the preprocessing pipeline to make sure it was strong enough for deployment.

To fix the problem of class imbalance, we used model-internal class weighting instead of oversampling or making fake data. This method gave normal and faulty engine classes the same effective importance throughout training. This made it easier to find faults without changing the data in a way that wasn't real.

4. Presentation of the Results
The Random Forest model demonstrates balanced but moderate predictive performance. An accuracy of 68.5% indicates acceptable overall classification capability on unseen data (see Table 6). The relatively high precision (77.8%) shows that when the model predicts an engine fault, it is usually correct, minimizing false alarms. The recall value of 75.7% reflects a reasonable ability to identify faulty engines, although a notable proportion of fault cases remain undetected. The F1-score (76.7%) confirms a stable trade-off between precision and recall. Overall, the Random Forest model performs reliably as a baseline classifier but exhibits limitations in maximizing fault detection sensitivity, which is critical in predictive maintenance applications.

Table 6. Performance Metrics of the Random Forest Model

Metric	Accuracy	Precision	Recall	F1-score
Score	0.685	0.778	0.757	0.767

The confusion matrix reveals that the Random Forest model correctly identifies a substantial number of faulty engines (1,419 true positives), but also misclassifies 456 faulty cases as normal, indicating missed fault detections (see Figure 5). Additionally, 406 normal engines are incorrectly classified as faulty, contributing to false alarms. This pattern explains the model's strong precision but moderate recall.

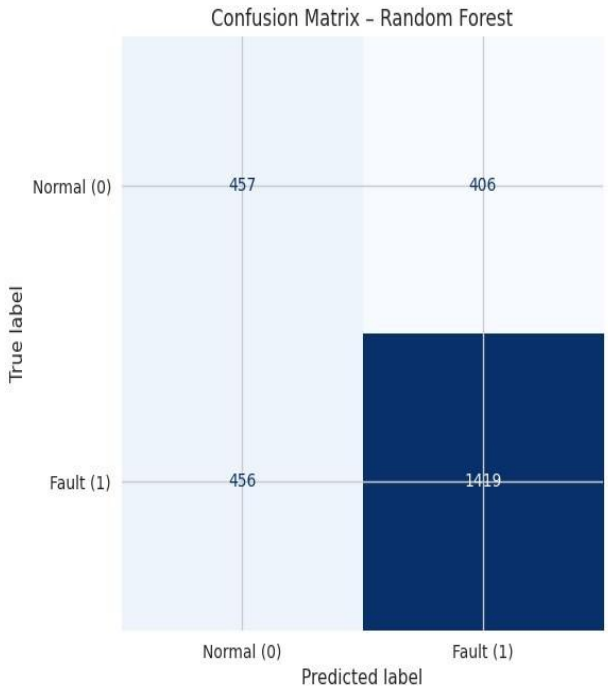


Figure 5: The Confusion Matrix of Random Forest

The ROC curve further illustrates the model's discriminative capability, achieving an AUC of 0.72, which indicates acceptable but not strong separation between normal and faulty classes (see Figure 6). While the curve deviates from random guessing, it highlights the model's limited ability to consistently distinguish between classes across different decision thresholds.

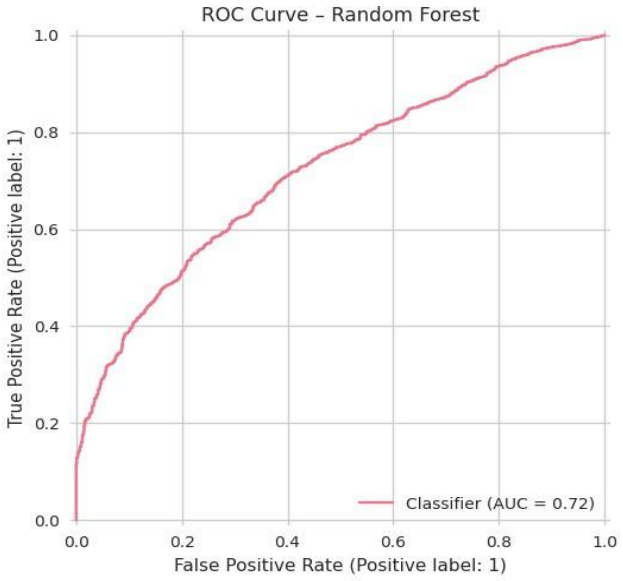


Figure 6: The ROC Curve of Random Forest

The Gradient Boosting model demonstrates strong predictive capability, particularly in fault detection. An accuracy of 70.0% reflects stable overall classification performance. The precision of 70.3% indicates a moderate rate of false alarms, which is acceptable in safety-critical predictive maintenance contexts (see Table 7). Most notably, the recall of 97.4% highlights the model's exceptional ability to identify faulty engines, minimizing missed fault cases. This high recall is reflected in the strong F1-score of 0.816, indicating an effective balance between sensitivity and precision. These results confirm that Gradient Boosting is well suited for predictive maintenance applications where early and reliable fault detection is prioritized over strict false-alarm minimization.

Table 7. Performance Metrics of the Gradient Boosting Model

Metric	Accuracy	Precision	Recall	F1-score
Score	0.700	0.703	0.974	0.816

The confusion matrix indicates that the Gradient Boosting model successfully identifies 1,826 faulty engines, with only 49 fault cases misclassified as normal, demonstrating a very low rate of missed detections (see Figure 7). This high sensitivity is essential in predictive maintenance, where undetected faults can lead to severe operational consequences. Conversely, 773 normal engines are incorrectly classified as faulty, which accounts for the model's moderate precision and increased false-alarm rate. This trade-off reflects a deliberate bias toward fault detection rather than fault avoidance, a desirable characteristic in mission-critical military maintenance environments where early identification of potential engine failures is prioritized over minimizing false alerts.

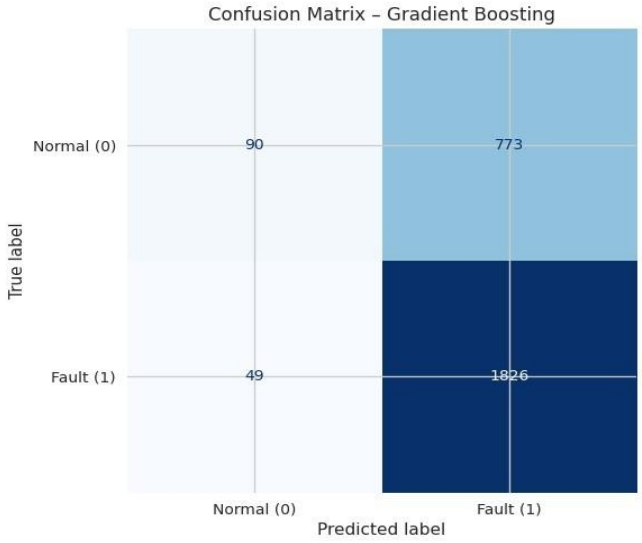


Figure 7: The Confusion Matrix of Gradient Boosting

The ROC curve further confirms the model's discriminative capability, achieving an AUC of 0.72, which indicates a consistent ability to separate normal and faulty engine conditions across different decision thresholds (see Figure 8). Although the performance is not optimal, the curve demonstrates that the model performs substantially better than random classification and remains reliable under varying operational conditions, supporting its robustness for predictive maintenance applications.

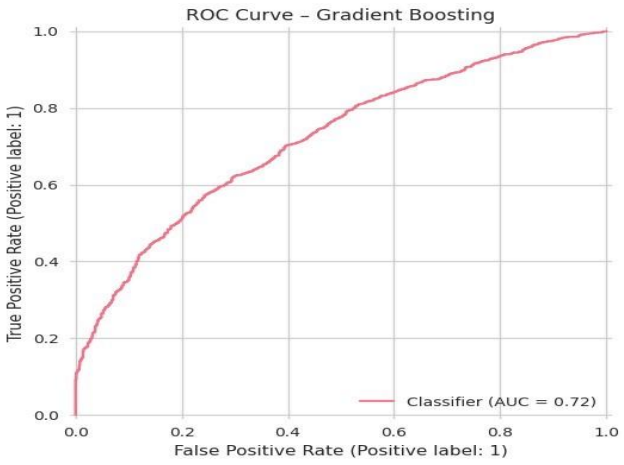


Figure 8. The ROC Curve of Gradient Boosting

The AdaBoost model demonstrates extremely strong fault detection capability, as reflected by its very high recall of 98.9%, indicating that almost all faulty engines are correctly identified. This makes AdaBoost particularly suitable for predictive maintenance scenarios where missing a fault carries high operational risk. The overall accuracy (69.0%) and precision (69.0%) are moderate, suggesting an increased number of false alarms (see Table 8). Nevertheless, the F1-score of 0.814 confirms a well-balanced trade-off between sensitivity and precision. Overall, AdaBoost prioritizes fault sensitivity over false-alarm reduction, aligning well with mission-critical maintenance requirements in military vehicle operations.

Table 8: Performance Metrics of the AdaBoost Model

Metric	Accuracy	Precision	Recall	F1-score
Score	0.69	0.69	0.989	0.814

The confusion matrix shows that AdaBoost correctly identifies 1,854 faulty engines, while only 21 faulty cases are misclassified as normal, confirming its exceptionally low missed-fault rate (see Figure 9). However, 828 normal engines are incorrectly classified as faulty, explaining the reduced precision and higher false-alarm rate. This behavior reflects AdaBoost's strong bias toward detecting hard-to-classify fault cases.

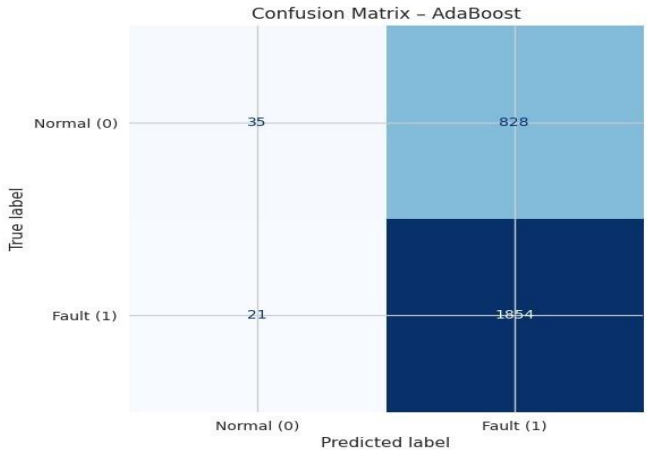


Figure 9: The Confusion Matrix of AdaBoost

The ROC curve further supports this observation, achieving an AUC of 0.71, which indicates reliable discrimination between normal and faulty engine conditions across thresholds (see Figure 10). Although slightly lower than Gradient Boosting, this AUC confirms AdaBoost's robustness and effectiveness in fault-sensitive predictive maintenance applications.

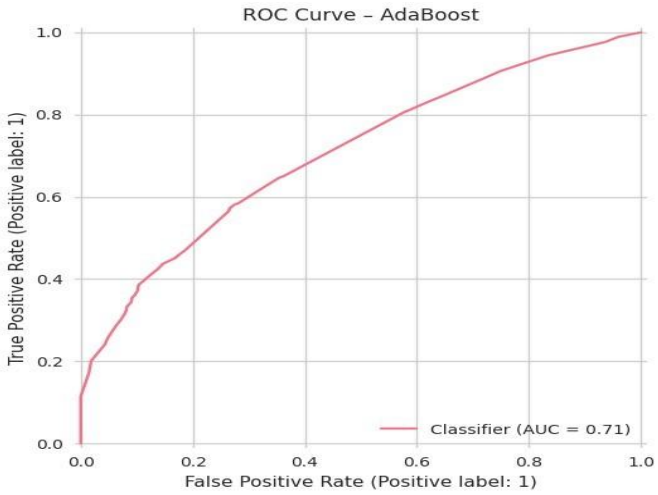


Figure 10: The ROC Curve of AdaBoost

The KNN model demonstrates moderate overall classification performance with an accuracy of 69.5%. The precision of 72.8% indicates a reasonable ability to limit false alarms when predicting faulty engines (See Table 9). A recall value of 88.4% highlights the model's strong sensitivity to engine faults, enabling it to detect most failure cases. The resulting F1-score of 0.799 confirms a balanced trade-off between precision and recall. Compared to ensemble-based models, KNN shows competitive fault detection capability but slightly lower robustness, reflecting its reliance on local neighborhood patterns rather than learned global decision structures.

Table 9: Performance Metrics of the K-Nearest Neighbors (KNN) Model

Metric	Accuracy	Precision	Recall	F1-score
Score	0.695	0.728	0.884	0.799

The confusion matrix shows that KNN correctly identifies 1,657 faulty engines, while 218 faulty cases are misclassified as normal, indicating some missed fault detections (see Figure 11). Additionally, 618 normal engines are incorrectly classified as faulty, contributing to moderate precision. This behavior reflects KNN's sensitivity to local data density and class overlap in the feature space.

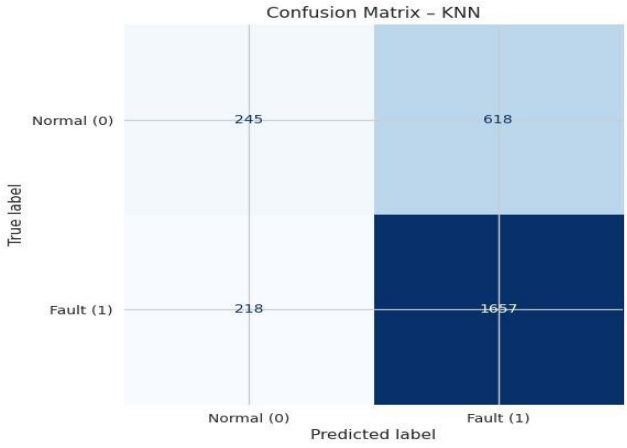


Figure 11. The Confusion Matrix of KNN

The ROC curve further illustrates the model's discriminative ability, achieving an AUC of 0.69, which is lower than the ensemble-based models but still above random classification (see Figure 12). This indicates that while KNN can capture fault-related patterns, its separation capability is limited in complex engine health datasets.

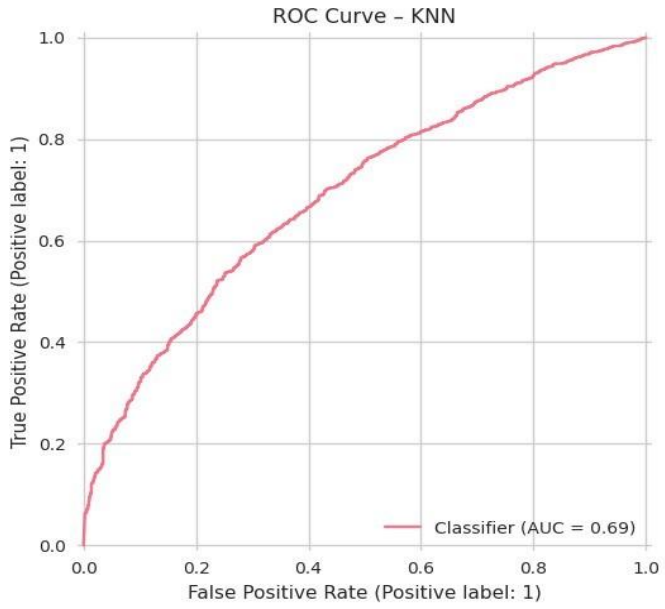


Figure 12: The ROC Curve of KNN

Table 10 shows how Gradient Boosting, AdaBoost, KNN, and Random Forest compare in terms of accuracy, precision, recall, and F1-score. In terms of recall, Gradient Boosting and AdaBoost are clearly better than the other models, with values close to 0.97–0.99. This means they are better at finding problematic engines and reducing missed failures. This makes both models very good for predictive maintenance that is very important to the mission. KNN has a high recall (around 0.88) and a good F1-score, but it is not as powerful overall because it is sensitive to local data distributions. Random Forest has the most precise results of all the models, which means it has less false alarms. However, its recall is lower, which means it might overlook certain faults. In general, ensemble boosting methods find the best balance between fault sensitivity and stability. Gradient Boosting is the most balanced model because it has the highest F1-score, high recall, and reasonable precision. This makes it the best choice for making military vehicles more ready.

Table 10: Comparative Performance of Selected Machine Learning Models

Model	Accuracy	Precision	Recall	F1-score
Gradient Boosting	0.700	0.703	0.974	0.816
AdaBoost	0.690	0.690	0.989	0.814
KNN	0.695	0.728	0.884	0.799
Random Forest	0.685	0.778	0.757	0.767

5. Discussion

This research originated from the necessity to enhance military vehicle preparedness via dependable, data-driven predictive maintenance. Traditional maintenance procedures have a hard time when operating circumstances are not clear. This is why machine learning techniques that can find engine problems early while dealing with loud, unbalanced, and complex sensor data are being used more and more.

The experimental findings indicate that all assessed machine learning models exhibited satisfactory performance in forecasting engine condition, with significant variations in sensitivity and stability. Models based on ensembles always did better than distance-based learning, which shows that they are good at simulating non-linear interactions between engine sensor variables. Gradient Boosting and AdaBoost had the best recall rates (97% and 98%, respectively), which means they were quite good at finding faults. KNN had a good recall rate but wasn't as reliable, while Random Forest had the best precision but missed more fault cases. In general, the results show that there is a clear trade-off between fault sensitivity and false-alarm control, which is a key part of making decisions about predictive maintenance.

Gradient Boosting was the best classifier since it did well on all of the evaluation metrics. Like the research of Owusu-Adjei et al. [29] and Ayodele [30] It had a recall rate of 97%, the greatest F1-score (82%), and a consistent accuracy rate of 70%, showing that it was good at finding faults with acceptable false-alarm rates. In military maintenance, it's more important to reduce missed engine issues than false alerts, which is why Gradient Boosting is a good choice. It may find little changes in engine behavior over time since it can fix earlier mistakes over and over again. The results suggest that Gradient Boosting can help with proactive maintenance scheduling, cut down on unexpected failures, and make mission-critical settings more ready for operations.

The overall results show that predictive maintenance based on machine learning is a good way to keep an eye on engine health in situations that are complicated and changeable. Models that can capture non-linear feature interactions, especially boosting-based ensembles, did a better job of finding defective engine states. The relatively low accuracy values seen in this study (about 70%) do not mean that the model is bad; they just show that the dataset is hard to work with because it has a lot of noise, variability, and overlapping class distributions.

The dataset utilized in this study is different from the highly controlled datasets used in previous studies since it focuses on scalability and algorithmic robustness instead of tweaking for specific domains. Consequently, models were assessed in more realistic scenarios, where flawless differentiation between healthy and defective states is improbable. The high recall values attained by Gradient Boosting and AdaBoost show that it is possible to find faults early, even with these limitations. The results as a whole show that tailored ensemble learning approaches work well for predictive maintenance (see Table 11). They also show how important it is to put sensitivity ahead of sheer accuracy in military logistics applications where safety is a top priority.

Table 11: Comparative Performance with Related Studies

Study	Model	Accuracy	Precision	Recall	F1-score
Mykich et al. [1]	AdaBoost	86%	–	–	–
Nagappa et al. [4]	Random Forest	84%	–	–	–
Saraswat et al. [8])	LSTM	95.40 %	–	–	0.95
Mahiyudin et al. [10]	Gradient Boosting	99.50 %	99.90 %	99.10 %	99.50 %
This Study	Gradient Boosting	70%	70%	97%	82%

This study includes a number of flaws, even though the results are intriguing. First, the dataset is artificial and doesn't directly connect to real military mission profiles, operational circumstances, or vehicle-specific limits. Second, only tabular sensor features were taken into account. Temporal dependencies that could improve failure prediction were not. Third, the study looked at binary engine condition categorization instead of estimating the severity of faults at several levels or the remaining usable life. Finally, the costs of processing and the limits on real-time deployment were not clearly looked at, which could make it harder to use in embedded military systems.

Subsequent research ought to augment this study by integrating real-world or semi-real military datasets when feasible, encompassing time-series sensor streams and contextual operational data. To find patterns of temporal degradation, you may look into advanced deep learning models like LSTM or hybrid CNN–LSTM architectures. Also, making the work more useful in real life by adding remaining usable life forecast and multi-class fault diagnostics would be helpful. Combining explainable AI methods with frameworks for real-time deployment would make it easier for military logistics to make reliable decisions and use new technologies.

5.1 Validation and Deployment

To support real world deployments and long-term maintainability, the best performing model (Gradient Boosting) and the preprocessing pipeline were saved using joblib. This ensures reproducibility, simplifies integration into external systems, and allows consistent preprocessing during inference. A validation step was conducted by reloading the saved model and pipeline, then applying them to a real sample from test dataset as shown In Figure 13.

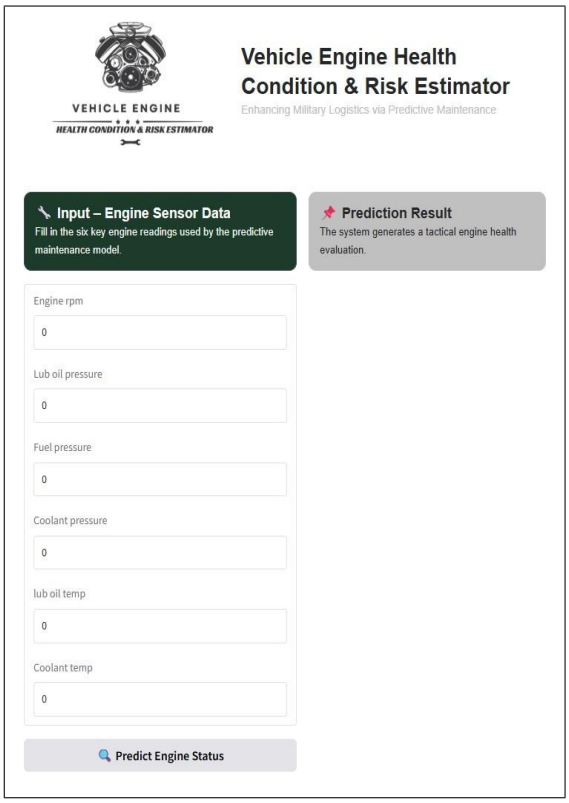


Figure 13: Deployment Architecture

The system successfully produced both a class prediction and a fault probability, demonstrating correct end-to-end functionality of the pipeline. Interpretation logic was added to translate predictions into actionable maintenance recommendations:

- Class 1 (Fault): "Engine at risk, schedule maintenance".
- Class 0 (Normal): "Engine Healthy, continue routine monitoring".

This validation confirms that the pipeline can be reliably deployed within a monitoring system or embedded into military vehicle diagnostics platforms. The deployment workflow consists of: First, Preprocessing new sensor data via the saved pipeline. Second, running the Gradient Boosting model to predict engine condition. Third, interpreting the prediction and producing maintenance guidance and last, optionally logging the input and output for continuous monitoring and retraining.

This structure ensures that the predictive maintenance framework is capable of operating automatically and consistently, enabling real-time decision support in military logistics environments. Refer to the result of the deployment in Figure 14 and Figure 15.

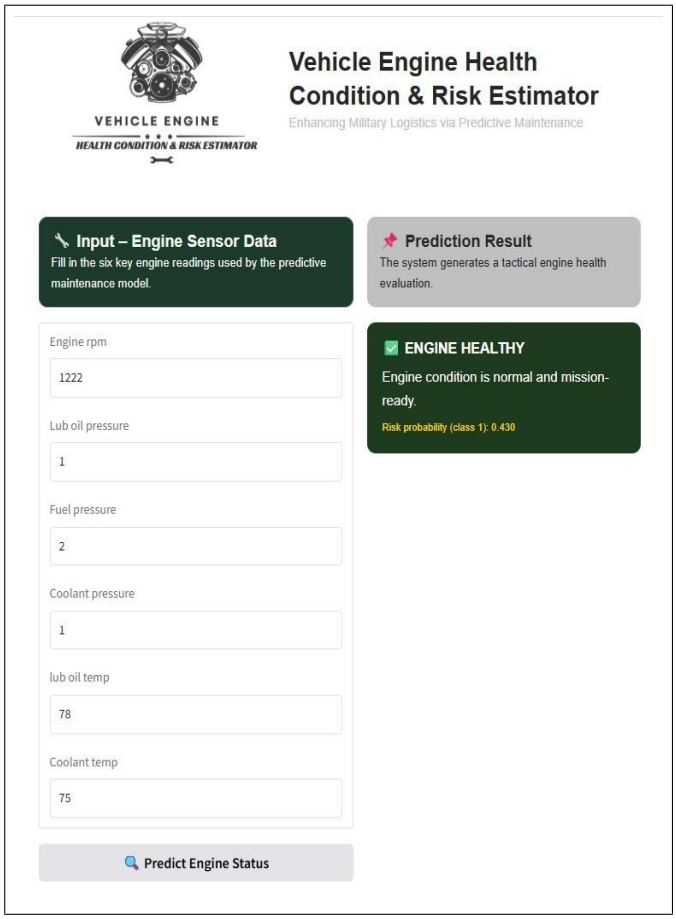


Figure 14: The result of the Deployment in case of Healthy Engine

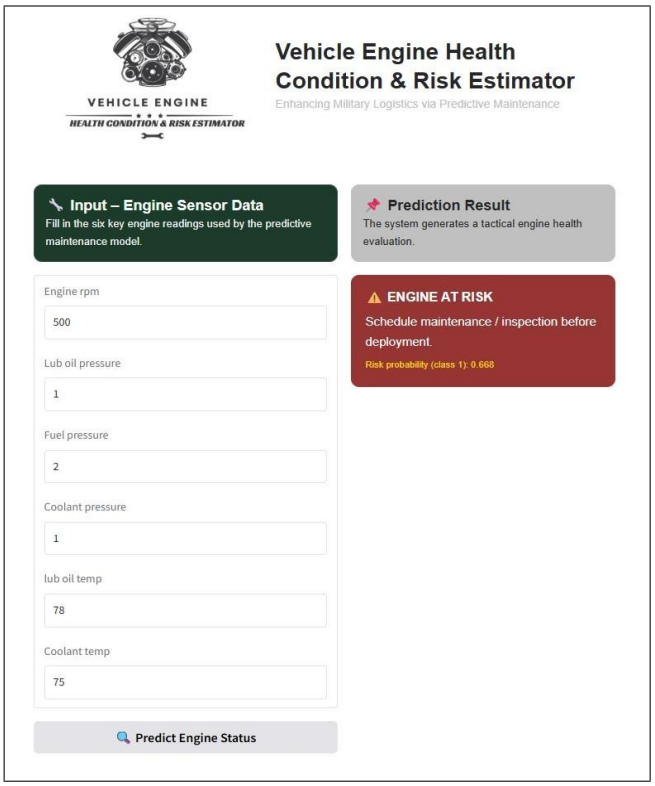


Figure 15: The result of the Deployment in case of Unhealthy Engine

6. Conclusion

This study shows that predictive maintenance based on data can greatly improve military logistics by making it easier to find problems early, lowering the number of unexpected vehicle breakdowns, and helping maintenance decisions that are proactive and focused on readiness. The chosen machine learning models work well with complicated, noisy engine sensor data. They can find non-linear connections and class imbalance, and they can be used in operations since they are scalable and easy to understand. Gradient Boosting was the best classifier among all the models that were tested. It had a recall rate of 97%, which means it was very good at finding broken engines and reducing missed detections, which is very important in mission-sensitive contexts. The model also had the greatest F1-score of 82%, which shows that it had a good balance between precision and recall. The overall accuracy and precision were only 70% each, which is an acceptable trade-off for predictive maintenance because being sensitive to faults is more important than reducing false alarms. The results show that boosting-based ensemble learning can reliably find small changes in engine behavior even when the data is hard to work with and not balanced. This study assesses models using a more demanding, general-purpose benchmark dataset characterized by greater unpredictability and class overlap, in contrast to previous studies that indicated elevated accuracies using strictly regulated or domain-specific datasets. The recall achieved is far better than the accuracy gained, which focuses on operational reliability instead of optimistic performance measurements. This change shows how important it is to put strong fault detection and scalability ahead of inflated accuracy. This makes the proposed architecture more realistic for real-world military logistics uses. The research depends on synthetic, non-temporal data, which could restrict its applicability to real-world implementations and long-term degradation prediction..

Acknowledgements

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-25-DR-1167). Therefore, the authors thank the University of Jeddah for its technical and financial support

Data Availability

Data of this study is available; the corresponding author will provide it on demand

Funding

The author received funding for this research from the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-25-DR-1167).

Competing Interests

The authors declare that they have no competing interests

References

[1] K. Mykich, I. Zavushchak, and A. Savka, “Predictive maintenance for automotive vehicle engines in military logistics,” in Proceedings of the 6th International Workshop on Modern Machine Learning Technologies (MoMLeT 2024), Lviv–Shatsk, Ukraine, 2024, CEUR Workshop Proceedings.

[2] Narayanan, T. S. A., & Padhy, S. C. (2023). Artificial Intelligence for Predictive Maintenance of Armoured Fighting Vehicles Engine. Indian Journal of Artificial Intelligence and Neural Networking (IJAINN), 3(5), August 2023. DOI: 10.54105/ijainn.E1071.083523.

[3] Sengupta, P., Mehta, A., & Rana, P. S. (2023). Predictive Maintenance of Armoured Vehicles using Machine Learning Approaches. arXiv preprint arXiv:2307.14453v1 [cs.LG].

[4] Nagappa, G., Abdulla, G., Nizamuddin, S., Sandeep, G., & Rajesh, G. (2025). An ensemble deep learning model for vehicular engine health prediction. International Journal of Engineering and Science Invention (IJESI), 14(4), 47–65. https://doi.org/10.35629/6734-14044765.

[5] Mun, S., Hwang, J., Bian, L., Falls, T. C., & Bond, W. G. (2023). Ensemble learning for fault condition prediction and health status monitoring in military ground vehicles. In K. Babski-Reeves, B. Eksioglu, & D. Hampton (Eds.), Proceedings of the IISE Annual Conference & Expo 2023 (Abstract ID: 1479). Institute of Industrial and Systems Engineers (IISE).

[6] Dalzochio, J., Kunst, R., Barbosa, J., Pignaton de Freitas, E., & Binotto, A. (2023). MILPdM: A predictive maintenance architecture for the military domain. In 2023 International Conference on Machine Learning and Applications (ICMLA). IEEE.

[7] Kuiper, P., Lin, S., Blanchet, J., & Tarokh, V. (2024). Generative learning for simulation of vehicle faults. In H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, & M. D. Rossetti (Eds.), Proceedings of the 2024 Winter Simulation Conference. arXiv:2407.17654.

[8] Saraswat, A., Peshawari, D., Ray, S., & Agarwal, R. K. (2025). Predictive maintenance of automotive engines using machine learning and deep learning techniques. International Journal of Innovative Research in Technology, 12(1), 446–455. https://doi.org/10.2349/IJIRT.179974.

[9] Errezgouny, A., Chater, Y., González, C. D. B., & Cherkaoui, A. (2025). An integrated deep learning approach for predictive vehicle maintenance. Decision Analytics Journal, 16. https://doi.org/10.1016/j.dajour.2025.100597.

[10] Mahiyudin, G., Hussain, M., & Dewi, D. D. (2025). A

Comprehensive Study on Predicting the Need for Vehicle Maintenance Using Machine Learning. Engineering Proceedings, 107(1), 89. https://doi.org/10.3390/engproc2025107089.

[11] Ojeda, J. C. O., & Gonçalves, M. C. (2025). Application of a Predictive Model to Reduce Unplanned Downtime in Automotive Industry Production Processes: A Sustainability Perspective. Sustainability, 17(9), 3926. https://doi.org/10.3390/su17093926.

[12] Modi, P. (n.d.). Automotive vehicles engine health dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/parvmodi/automotivevehicles-engine-health- dataset?resource=download

[13] Bluman, Allan G. Elementary statistics : a step by step approach, page 162. 8th edition. McGraw-Hill

[14] Sung, S.-H.; Seo, C.-S.; Pokojovy, M.; Kim, S. A Comparative Analysis of Preprocessing Filters for Deep Learning-Based Equipment Power Efficiency Classification and Prediction Models. Appl. Sci. 2025, 15, 11277. https://doi.org/10.3390/app152011277

[15] Yang, Y.; Iqbal, M.Z. Cost-Optimised Machine Learning Model Comparison for Predictive Maintenance. Electronics 2025, 14, 2497.https://doi.org/10.3390/electronics14122497

[16] Zhi-Hua Zhou ."Ensemble Methods: Foundations and Algorithms". CRC Press (an imprint of the Taylor & Francis Group) (2012). Machine Learning & Pattern Recognition Series .Boca Raton, FL .978-1-4398-3005-5

[17] Hernández-Lobato, D. (2009). Prediction based on averages over automatically induced learners: Ensemble methods and Bayesian techniques (Doctoral dissertation, Universidad Autónoma de Madrid).

[18] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, 2nd ed. New York, NY, USA: Wiley, 2001.

[19] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189–1232.https://doi.org/10.1214/aos/1013203451

[20] Tawakuli A, Havers B, Gulisano V, Kaiser D, Engel T. Survey: Time-series data preprocessing: A survey and an empirical analysis. Journal of Engineering Research. 2025 Jun 1;13(2):674-711.

[21] Alnawmasi N, Mannering F. A note on random parameters models of crash injury severities with k-means clustering for data preprocessing. Analytic Methods in Accident Research. 2025 Oct 12:100408.

[22] Wang J, Che J, Li Z, Gao J, Zhang L. Hybrid wind speed optimization forecasting system based on linear and nonlinear deep neural network structure and data preprocessing fusion. Future Generation Computer Systems. 2025 Mar 1;164:107565.

[23] Zhao H, Alachiotis N. Data preprocessing methods for selective sweep detection using convolutional neural networks. Methods. 2025 Jan 1;233:19-29.

[24] Tang W, Chen J, Koo S. Soft wearable robot design based on deep learning with Colab. International Journal of Fashion Design, Technology and Education. 2025 Jan 1:1-2.

[25] Raschka S, Patterson J, Nolet C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information. 2020 Apr;11(4):193.

[26] Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, Thomas J, Ullmann T, Becker M, Boulesteix AL, Deng D. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2023 Mar;13(2):e1484.

[27] a Ilemobayo J, Durodola O, Alade O, J Awotunde O, T Olanrewaju A, Falana O, Ogungbire A, Osinuga A, Ogunbiyi D, Ifeanyi A, E Odezuligbo I. Hyperparameter tuning in machine learning: A comprehensive review. Journal of Engineering Research and Reports. 2024 Jun 7;26(6):388-95.

[28] Côté PO, Nikanjam A, Ahmed N, Humeniuk D, Khomh F. Data cleaning and machine learning: a systematic literature review. Automated Software Engineering. 2024 Nov;31(2):54.

[29] Owusu-Adjei M, Ben Hayfron-Acquah J, Frimpong T, Abdul-Salaam G. Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems. PLOS Digital Health. 2023 Nov 30;2(11):e0000290.

[30] Ayodele A. A comparative study of ensemble learning techniques for imbalanced classification problems. World Journal of Advanced Research and Reviews. 2023;19(1):1633-43.