

Comparative Analysis of Ensemble Machine Learning Algorithms For Diesel Injector Health Classification Using ECU Sensor Data

^{1a*}Bibek Dhungana, ^{1b}Manish Upreti, ^{1c}Navaraj Dahal, ^{1d}Prabesh Ghimire,
^{1e}Prashanta Dhakal, ^{1f}Lochan Kendra Devkota

¹Department of Automobile and Mechanical Engineering, Thapathali Campus, Institute of Engineering, Tribhuvan University, Kathmandu, Nepal

Email: ^ab_dhungana@tcioe.edu.np, ^bmanishupreti95@gmail.com,

^cnavarajdahall10@gmail.com, ^dprweshdhimire2078@gmail.com,

^eprashantdhakal11@gmail.com, ^flockan.devkota@tcioe.edu.np

*Corresponding Email: ^ab_dhungana@tcioe.edu.np

Abstract

Dysfunction of the diesel fuel injector is a major cause of reduced engine performance, elevated fuel consumption, and increased emission levels in contemporary common-rail diesel engines. Traditional time-based maintenance methods are unable to identify injector faults at an early stage, often remaining undetected until engine failures occur. This paper presents a comparative analysis of four machine learning algorithms, RandomForest (RF), eXtreme Gradient Boosting (XGBoost), Gradient Boosting (GB), and K-Nearest Neighbors (KNN), applied to binary injector health classification using Engine Control Unit (ECU) sensor data. The G-IDSS diagnostic platform was used to collect data from Isuzu 4JA1 diesel vehicles, yielding 43,835 labeled samples across nine engineered features. Hyperparameter optimization was performed using the Optuna framework with the Tree-structured Parzen Estimator (TPE) algorithm under five-fold stratified cross-validation. The optimized RandomForest achieved the highest test accuracy of 85.06% (precision: 0.87, recall: 0.77 for the faulty class), followed by XGBoost at 84.34%. Feature importance analysis identified Engine Speed as the most discriminative parameter (~27% of classification contribution).

Keywords: Predictive Maintenance, Diesel Fuel Injectors, Bayesian Hyperparameter, Optuna, G-IDSS Diagnostics

I. INTRODUCTION

Contemporary common rail-type diesel engines rely on solenoid types of fuel injectors with pressure ranges of 1,500 -2,500 bar forcing specific amounts of atomized fuel. The performance of the injectors directly controls the quality of the combustion, the fuel usage, the power production and the exhausts. The mechanisms of degradation (such as nozzle clogging, internal leakage, needle valve stiction, and solenoid wear) build up over time, not typically creating fault codes in standard OBD-II implementation, until a serious condition develops [1].

Predictive Maintenance (PdM) based on machine learning algorithm offers a data driven alternative that enables fault detection from patterns in existing ECU parameters. The complex, multi-dimensional nonlinear interdependencies among fuel rail pressure, injection quantities, airflow, and engine speed are well suited to tree based ensemble methods, which do not require explicit physical models of degradation process [2].

This study addresses three objectives:

- To identify which ECU parameters most reliably distinguish healthy from degraded injector states.
- To evaluate and compare the classification performance of RandomForests, XGBoost, Gradient Boosting, and KNN on real world vehicle data
- To determine the optimal model configuration through systematic hyperparameter optimization.

II. RELATED WORKS

The scientific article [3] has shown how machine learning models, which are trained on the indirect sensor data, would reliably predict the faults in parts that cannot be directly monitored, which is economically disadvantageous based on its coverage of per component sensors. Their results confirm that the existing ECU outputs would be the only source of data to be used in fault detection.

According to the survey recommended by the [4] , it was proposed that ECU based anomalies fields are critical to the safety of vehicles and that the machine learning models may be trained on the collected CAN data to recognize faults and intrusions. Gradient Boosting (with XGBoost) is explicitly stated in the list of the most effective classifiers to detect Anomaly in CAN buses along with the RandomForest and SVMs [4].

[5] designed a three-stage predictive maintenance pipeline of diesel engines including signal identification, anomaly detection with autoencoders, and performance evaluation and discovered that data preprocessing quality affected the accuracy of detection more of model architecture complexity. This highlights the role of the feature engineering and normalization pipeline employed in the current work.

Although earlier research has mostly used simulated or laboratory data, this paper provides a comparative analysis on the test on real-world in-service vehicle data with field realistic noise and missing values filling a gap in the literature.

III. METHODOLOGY

A. Data Acquisition

The data was collected on the ISUZU 4JA1 diesel vehicles in cooperation with Continental Trading enterprise Pvt Ltd., Isuzu Balaju. It was based on a diagnostics platform, i.e. Global Isuzu Diagnostic Service System (G-IDSS, which offers a

dealer level ECU parameter streaming. The data was given in the form of CSV files at the original ECU logging rate. The data sets were comprised of 24582 healthy vehicles data and 19253 faulty vehicles data amounting to 43835 data in total. The labels were coded (0=healthy and 1= faulty) and cross-validated with service maintenance records and G-IDSS fault history to verify the correctness of annotation.

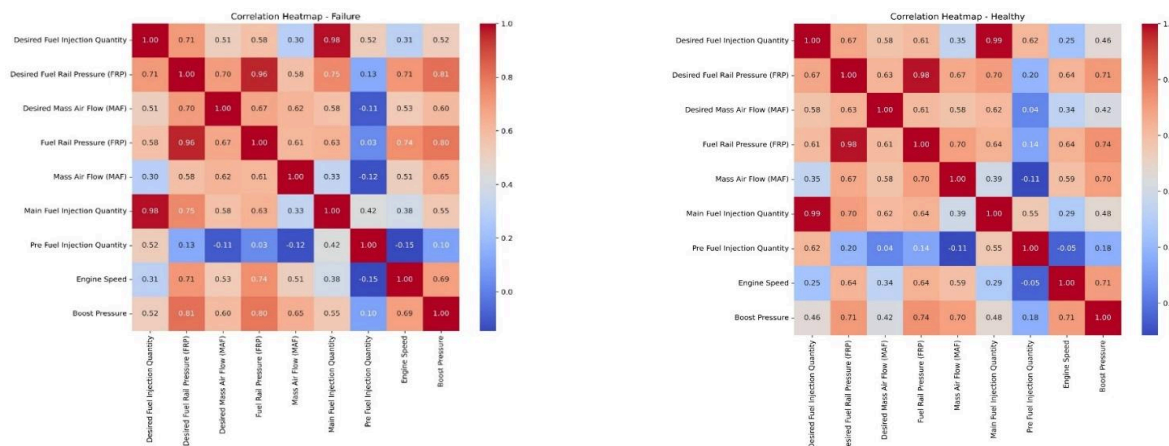
B. Parameter Selection and Feature Engineering

The G-IDSS platform streams 37 raw ECU parameters. An initial screening removed hardware diagnostic indicators (sensor status flags), string-valued categorical fields, and timestamp columns, leaving 17 candidate parameters. The following parameters were subsequently removed for the reasons stated: Idle Speed, REF Averaged MAF, and Vehicle Speed were dropped because they are functionally redundant with Engine Speed and MAF, which capture the same physical phenomena with greater diagnostic fidelity.

Main Injection Timing was removed due to its low diagnostic relevance to the mechanical degradation modes targeted in this study (clogging, leakage, stiction).

Fuel Temperature and Intake Air Temperature were removed following correlation analysis: both exhibited Pearson correlation coefficients $|r| < 0.1$ with the injector health target variable, indicating negligible discriminative power.

Correlation analysis using Pearson coefficients was applied to the remaining features on the training split only, to prevent data leakage. Features with inter-feature correlation $|r| > 0.8$ were flagged. Normally, one feature from each highly correlated pair would be dropped; however, because the chosen classifiers (RandomForest, XGBoost, Gradient Boosting, KNN) are non-parametric and do not assume feature independence, retaining correlated pairs does not violate model assumptions or destabilize decision boundaries.



Specifically, Desired Fuel Rail Pressure and Fuel Rail Pressure ($r = 0.96\text{--}0.98$) and Desired Fuel Injection Quantity and Main Fuel Injection Quantity ($r = 0.98\text{--}0.99$)

were both retained because each pair provides complementary information about the commanded versus actual values, a difference that is diagnostically meaningful for detecting injector faults. The nine final features are listed in Table 1.

Table 1: Important nine features selected from ECU parameters

S.N.	Parameter	Unit	Diagnostic Relevance
1	Desired Fuel Rail Pressure	MPa	Commanded pressure setpoint; deviation from actual indicates injector fault
2	Fuel Rail Pressure	MPa	Actual rail pressure; directly reflects injection system performance
3	Desired Fuel Injection Quantity	mm ³ /st	Commanded injection quantity; baseline for quantity comparison
4	Main Fuel Injection Quantity	mm ³ /st	Actual delivered quantity; discrepancy signals injector wear or clogging
5	Mass Air Flow	g/cyl	Air charge measurement; affected by combustion quality changes
6	Engine Speed	RPM	Rotational dynamics; dominant fault discriminator (~27% importance)
7	Pre Fuel Injection Quantity	mm ³ /st	Pilot injection; sensitive to needle valve condition
8	Desired Mass Air Flow	g/st	Commanded air flow; enables commanded vs. actual comparison
9	Boost Pressure	kPa	Turbocharger output; reflects overall engine load state

Unlike parametric models such as logistic regression or linear discriminant analysis, non-parametric classifiers do not estimate explicit coefficients for each feature. Consequently, multicollinearity among predictor variables does not violate any underlying assumptions, nor does it destabilize the decision boundary estimation. The high inter-feature correlations observed particularly between Desired Fuel Rail Pressure and Fuel Rail Pressure ($r = 0.96/0.98$) and between Desired Fuel Injection Quantity and Main Fuel Injection Quantity ($r = 0.98/0.99$) are therefore retained in the feature space without remediation.

C. Data Preprocessing and Algorithms

StandardScaler was used to scale all the nine features uniformly. Although tree-based models like the RandomForest and XGBoost are scale invariant, standardization was always employed to be fair to make a comparison to the distance sensitive KNN algorithm. The rows that lacked target labels were eliminated. The processed data was

divided into 70 % train and 30 % test blocks through stratified sampling to maintain the balance of classes in both blocks.

Four algorithms were evaluated in this study:

- **RandomForest:** Machine learning algorithm that uses many decision trees to make better predictions. Each tree looks at different random parts of the data, and their results are combined by voting for classification or averaging for regression which makes it as ensemble learning technique. This helps to improve accuracy and reduce errors [6].
- **XGBoost:** This is an eXtreme Gradient Boosting framework which minimizes a regularized loss in a series. It uses second order gradient statistics, sparsity sensitive split finding, and parallelism on hardware to train [7].
- **Gradient Boosting (GB):** It is a sequential boosting algorithm where each successive learner is trained to predict the residual of the former. It uses the shrinkage (low learning rate) and subsampling to enhance generalization in order to reduce the variance [8].
- **K_Nearest Neighbors (KNN):** This is a non-parametric, instance-based classifier that makes a classification decision by voting on the number of nearest training points (k) in the standardized feature space, Euclidean distance P. Cunningham and S.J. Delany. Three configurations were made: k = 5,15 and 50 [9].

D. Hyperparameter Optimization

Optuna was hyperparameter optimized, a methodology of structured Parzen Estimator (TPE) algorithm. In comparison to grid search or random search, TPE is a Bayesian sequential search algorithm, which downloads the connection between hyperparameter settings and their score in past trial and gradually creates the search to high performing areas of parameter space.

Fivefold stratified cross-validation was used to do optimization with the classification accuracy being the objective measure. The optimization of each model was performed separately in the ranges that had been calibrated to the properties of the dataset and the anticipated complexity of injector degradation patterns.

Table 2: Optimized hyperparameters for all four models after Optuna TPE tuning.

Model	Hyperparameter	Search Range	Optimized Value
RandomForest	n_estimators	100–600	441
RandomForest	max_depth	5–30	25
RandomForest	min_samples_split	2–10	5
RandomForest	min_samples_leaf	1–5	1
RandomForest	max_features	sqrt / log2	log2
RandomForest	bootstrap	True/False	True
RandomForest	criterion	gini / entropy	gini

XGBoost	n_estimators	100–600	380
XGBoost	max_depth	3–12	7
XGBoost	learning_rate	0.01–0.3	0.08
XGBoost	subsample	0.6–1.0	0.85
XGBoost	colsample_bytree	0.6–1.0	0.78
Gradient Boosting	n_estimators	100–500	320
Gradient Boosting	max_depth	3–8	5
Gradient Boosting	learning_rate	0.01–0.3	0.10
Gradient Boosting	subsample	0.6–1.0	0.80
KNN	n_neighbors	5, 15, 50	6
KNN	metric	euclidean	euclidean

IV. RESULT

A. Comparative Model Performance

Table 3 presents the classification performance of all models on the held-out test set. RandomForest with Optuna-tuned hyperparameters achieved the highest overall accuracy of 85.06%. The F1-score for the faulty class was 0.82, reflecting a balanced trade-off between precision (0.87) and recall (0.77). XGBoost performed comparably, achieving 84.34% accuracy with a slightly higher faulty-class recall of 0.79 and the same F1-score of 0.82, indicating that XGBoost may marginally outperform RF in detecting actual faults (higher sensitivity), while RF demonstrates greater specificity. Gradient Boosting (84.55%) and the best KNN configuration (k=6, 83.62%) achieved competitive results, all exceeding 83% accuracy.

Five-fold cross-validation accuracy (mean $\pm$ std) across models: RandomForest: 84.8% $\pm$ 0.4%, XGBoost: 84.1% $\pm$ 0.5%, Gradient Boosting: 84.3% $\pm$ 0.4%, KNN (k=6): 83.2% $\pm$ 0.6%. The low standard deviations confirm stable generalization and the absence of significant overfitting across all models.

Table 3: Comparative Analysis of different Machine Learning models in terms of evaluation metrics

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-score
RandomForest	0.8506	0.87	0.77	0.82
XGBoost	0.8435	0.85	0.79	0.82
Gradient Boosting	0.8455	0.86	0.78	0.82

KNN (n=6)	0.8362	0.84	0.77	0.81
-----------	--------	------	------	------

All gradient boosting methods yielded identical F1-scores of 0.82 for the faulty class, suggesting that the feature space and dataset size govern the performance ceiling on this task more than algorithm choice alone. Given the safety-critical nature of injector fault detection, the higher recall of XGBoost (0.79 vs. 0.77) is noteworthy: fewer missed fault detections (Type II errors) are preferable in diagnostic applications, which may justify selecting XGBoost for deployment despite its marginally lower accuracy. The choice between RF and XGBoost ultimately depends on the operational priority — specificity versus sensitivity.

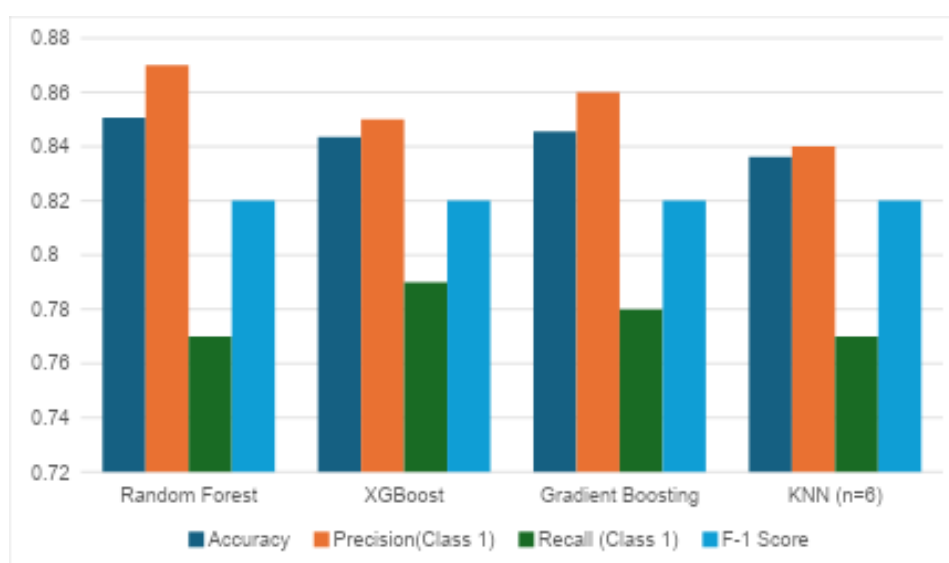


Figure 3: Comparative Analysis of ensemble ML model

B. XGBoost Confusing matrix analysis

The RandomForest classifier was evaluated on a test set comprising 13,151 samples, of which 7,375 (56.08%) belonged to the Healthy class and 5,776 (43.92%) to the Failure class. The model correctly classified 6,714 Healthy instances (51.05% of total) and 4,473 Failure instances (34.02% of total), yielding an overall accuracy of 85.06%. Type I errors (False Positives) accounted for 661 instances (5.03%) wherein Healthy engine states were misclassified as Failure. More critically, Type II errors (False Negatives) amounted to 1,303 instances (9.91%), representing Failure conditions incorrectly predicted as Healthy a particularly consequential error in safety-critical diagnostic systems.

Table 4: Confusion Matrix of RandomForest

	True	True
--	------	------

	Positive	Negative
Predicted Positive	6714	661
Predicted Negative	1303	4473

C. Feature Importance Analysis

RandomForest feature importance analysis, designed via mean decrease in Gini impurity, revealed Engine Speed as the most discriminative feature, contributing approximately 27% to the classification decision suggesting that engine fault conditions essentially manifest as deviations in rotational dynamics. Desired Fuel Rail Pressure and Fuel Rail Pressure followed closely, contributing ~19% each, collectively establishing the fuel pressure subsystem as the dominant fault signature domain (~65% combined with Engine Speed). Boost Pressure contributed a moderate 14%, consistent with the differential correlation patterns identified in the heatmap analysis. Particularly, Pre-Fuel Injection Quantity despite exhibiting the most pronounced correlation shift between Healthy and Failure states ranked as the least important feature (< 1%), suggesting that its discriminative information is either captured unnecessarily by higher-ranked features or insufficient in magnitude for impurity-based node splitting.

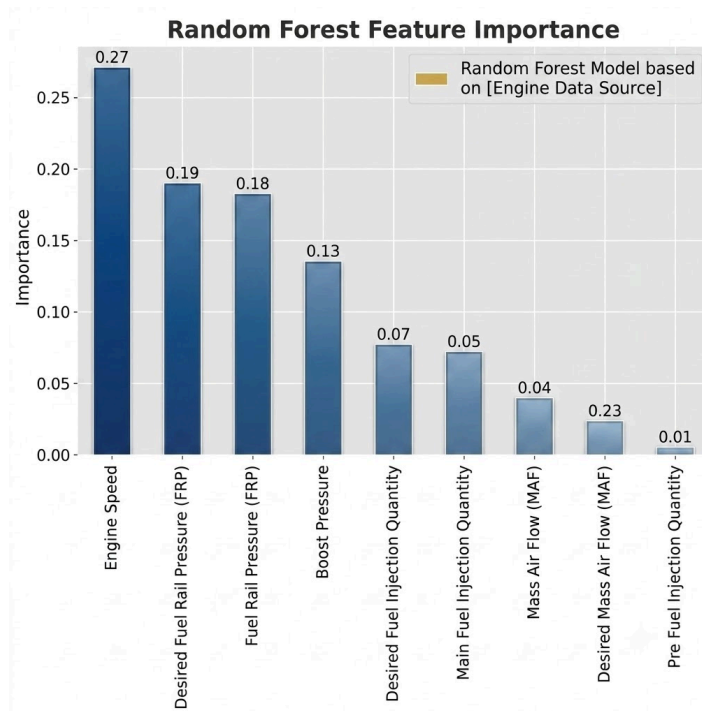


Figure 4: Feature importance analysis of RandomForest

V. CONCLUSION

This study evaluated four machine learning classifiers — RandomForest, XGBoost,

Gradient Boosting, and KNN for binary diesel injector health classification using real-world ECU sensor data from Isuzu 4JA1 vehicles. Optuna-based hyperparameter optimization with five-fold cross-validation consistently improved model performance over default configurations.

RandomForest achieved the highest test accuracy (85.06%, F1 = 0.82 for the faulty class), while XGBoost demonstrated marginally superior recall (0.79 vs. 0.77), which may be preferable for safety-critical deployment. The low cross-validation variance across all models ($\leq 0.6\%$) confirms stable generalization on this dataset and engine type.

Engine Speed emerged as the dominant discriminative feature ($\sim 27\%$ importance), providing a physically interpretable basis for the model's decisions. The preprocessing pipeline, standardization, correlation-based feature screening reduced to nine parameters, and train-only scaler fitting, proved essential for consistent performance on field-collected data.

The present results should be interpreted within the scope of a single engine family (Isuzu 4JA1) and a single diagnostic platform (G-IDSS). Generalization to other engine architectures or fuel injection systems cannot be assumed without further validation. Additionally, the binary classification framework does not distinguish between fault types.

Future work will target: (1) multi-class fault classification (clogging, leakage, nozzle wear); (2) validation across multiple engine families and injection architectures; (3) integration with real-time ECU data streams for online fault monitoring; and (4) exploration of cost-sensitive learning to reduce the False Negative rate.

REFERENCES

- [1] S.B Abhishek, A.U Omkar, S.V Gawali, K Arun "Common Rail Direct Injection," *International Research Journal of Engineering and Technology*, 2020.
- [2] Aftab Ahmed Soomro, Ayesha Noreen, Shafaq Naz, Jawad Ali Arshad, Muhammad Kashif Majeed, Nasir Rafique, Abdul Karim Kashif Baig, Syed Tahseenullah, & Basit Ahmad. (2025). DATA-DRIVEN PREDICTIVE MAINTENANCE OF DIESEL ENGINES USING ADVANCED MACHINE LEARNING AND AI-BASED REGRESSION ALGORITHMS FOR ACCURATE FAULT DETECTION AND REAL-TIME CONDITION MONITORING. *Spectrum of Engineering Sciences*, 3(7), 408–429.
- [3] Bode, G., Thul, S., Baranski, M., & Müller, D. (2020). Real-world application of machine-learning-based fault detection trained with experimental data. *Energy*, 198, 117323. <https://doi.org/10.1016/j.energy.2020.117323>
- [4] Övgü Özdemir, M. Tuğberk İşyapar, Pınar Karagöz, Klaus Werner Schmidt, Demet Demir, & N. Alpay Karagöz. (n.d.). *A Survey of Anomaly Detection in In-Vehicle Networks*.

- [5] Kohrt, E., Moon, M., Sullivan, M., Das, S., Thurston, M., & Nenadic, N. G. (2024). Data-driven Detection of Engine Faults in Infrequently-driven Ground Vehicles. *Annual Conference of the PHM Society*, 16(1). <https://doi.org/10.36001/phmconf.2024.v16i1.3899>
- [6] L. Breiman, "Random Forests," *Machine Learning*, pp. 5-32, 2001.
- [7] C. G. Tianqi Chen, "XGBoost. A Scalable Tree Boosting System," *ACM*, pp. 785-794, 2016.
- [8] Kamau, B. N., Malenje, B., Wamwea, C., & Onyango, L. A. (2025). Comparative Study of Extreme Gradient Boosting (XGBOOST), K-Nearest Neighbors (KNN), and Random Forest for Migraine Classification. *American Journal of Mathematical and Computer Modelling*, 10(1), 19–28. <https://doi.org/10.11648/j.ajmcm.20251001.13>
- [9] S. J. D. Padraig Cunningham, "k-Nearest Neighbour Classifier-A Tutorial," *ACM Computer Survey*, 2021.
- [10] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, & Masanori Koyama. (n.d.). *Optuna | Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM Conferences. Retrieved June 2, 2026, from <https://dl.acm.org/doi/10.1145/3292500.3330701>