



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: <https://doi.org/10.22214/ijraset.2026.85203>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

AutoCare AI: An AI-Assisted Vehicle Service Management System

Om Singh¹, Perseus Bhavnagri²

^{1,2}Student, Department of Information Technology, Wilson College (Autonomous), University of Mumbai, Mumbai, India

Abstract: Independent garages commonly manage repairs with paper job cards, verbal estimates and memory, so the accuracy and efficiency of their service cannot be measured. This paper presents AutoCare AI, a zero-cost web application built with Flutter and Firebase that digitises the garage workflow from walk-in to invoice, estimates a job's fault category, labour time and cost from the customer's free-text complaint using Google Gemini (with a keyword-rule engine for offline demonstration), suggests the least-busy suitable mechanic, predicts each vehicle's next periodic service and records manual, AI and actual outcomes on every job. The study follows a design-science approach with secondary, quantitative data: 31,236 owner complaints from the public NHTSA complaints database, labelled into seven of the application's fault categories and split into 24,988 training and 6,248 test records. A TF-IDF and Logistic Regression classifier reached 90.6 % accuracy (macro-F1 0.907), significantly above an 80 % target (exact binomial $p \approx 1.09 \times 10^{-115}$), whereas the application's keyword rules reached only 57.6 %. The learned classifier was significantly more accurate than the rules (exact McNemar, 2,199 against 136 discordant pairs, $p < 0.001$). In a 250-day simulation, greedy assignment lowered the mean daily spread of mechanic workload to 1.49 h against 4.03 h for random assignment (Wilcoxon $p \approx 3.85 \times 10^{-42}$). Gemini was not evaluated at scale because AI was kept switched off to guarantee zero cost; its few live estimates are reported only as anecdotal evidence.

Keywords: Vehicle service management; fault classification; complaint text; NHTSA; TF-IDF; Logistic Regression; large language models; Gemini; mechanic scheduling; Flutter Web; Firebase.

I. INTRODUCTION

A vehicle that enters an independent garage passes through a sequence of decisions: what is wrong with it, how long the repair will take, what it will cost, which mechanic should do the work and when it should next return for service. In many small Indian garages these decisions are taken informally and are rarely recorded in a form that can be checked later. This study builds and evaluates a vehicle service management system in which an artificial-intelligence (AI) component supports these decisions, and tests on public data the assumptions on which its design rests.

A. Service Management in Independent Garages

The typical small workshop runs on paper job cards, telephone calls and the owner's memory. Estimates of time and cost are verbal guesses, vaguely described faults are triaged by whoever is free, some mechanics are overloaded while others wait, and service reminders are forgotten. Because no structured record exists, the accuracy of an estimate or the efficiency of the workshop cannot be measured. Research on data-driven vehicle maintenance has concentrated on manufacturers and fleets with rich sensor data [1], far removed from the counter of a neighbourhood garage.

B. Artificial Intelligence for Fault Triage

The first information a garage receives is usually a sentence of free text, and classifying it into a fault category is a natural-language processing task. Classic supervised classifiers learn from labelled examples, typically with term-frequency-inverse-document-frequency (TF-IDF) features [2], whereas large language models (LLMs) can perform a task from an instruction alone, without fine-tuning [3]. A garage has no labelled history, which makes an LLM attractive, but its accuracy must be demonstrated rather than assumed.

C. The AutoCare AI Approach

AutoCare AI is a single-owner web application built with Flutter (web target), Riverpod, Firebase Authentication and Cloud Firestore (asia-south1, Mumbai) on Firebase's no-cost Spark plan with no billing account. The owner records customers, vehicles, mechanics, job cards and invoices.

For each new job an estimator predicts the fault category (one of nine), labour hours and cost before GST from the complaint, vehicle age, odometer reading and service type. The production estimator calls Google Gemini (gemini-3.5-flash-lite) through Firebase AI Logic with a strict JSON response schema; an offline keyword-rule engine with a baseline price table is used only in demonstration mode and tests; if Gemini fails, the owner sees a plain message and can enter a manual estimate instead. The application ranks mechanics by open workload with a specialist bonus, predicts each vehicle's next periodic service from time and distance, and stores the manual estimate, AI estimate and actual outcome on every job so that an AI Insights page can report accuracy and error continuously. The owner always reviews the AI estimate before saving.

II. PROBLEM STATEMENT AND NEED FOR THE STUDY

A garage that relies on verbal estimates and informal assignment cannot tell whether its estimates are accurate, whether its mechanics are evenly loaded or whether customers return for service on time, because none of these quantities is recorded. AI could help, but a small workshop has no labelled data to train models on and cannot pay for cloud services with unpredictable costs, while the obvious low-cost alternative, hand-written fault keywords, has rarely been tested against real complaint text. A system is needed that records every estimate and outcome, uses AI for fault triage and mechanic assignment, operates at no cost and is accompanied by evidence of how accurate its AI component can be.

Need for the Study: This study is undertaken to —

- digitise the garage workflow so that estimates and outcomes become measurable;
- test, on real complaint narratives, whether a learned classifier can triage faults with at least 80 % accuracy;
- compare such a classifier with the keyword rules of the application's offline demo engine;
- examine whether a greedy mechanic-assignment rule spreads workload more evenly than unplanned assignment;
- report honestly the limits of the evidence, including that the production LLM was not evaluated at scale.

III. LITERATURE REVIEW

Vehicle maintenance and service management. Theissler et al. (2021) [1] reviewed machine-learning-based predictive maintenance in the automotive industry and concluded that public data would boost research, that most studies depend on labelled data, and that combining data sources improves accuracy. Mishra et al. (2020) [4] predicted, for each industrial vehicle and date, the days remaining until its next periodic maintenance, addressing non-stationary usage and vehicles with little history. Yıldırım et al. (2025) [5] forecast monthly maintenance attendance at automotive after-sales centres, found XGBoost the most accurate of four algorithms, and used the forecasts for workforce planning. Zhang et al. (2020) [6] noted that priority dispatching rules are widely used because they are fast, intuitive and easy to implement, and showed that rules learned by deep reinforcement learning can outperform existing hand-designed ones.

Maintenance and complaint text. Pavlopoulos et al. (2022) [7] classified free-text fault claims from a vehicle-fleet company across 38 languages and 1,357 classes with a multilingual Transformer, reaching more than 80 % accuracy on high-frequency classes and above 60 % on mid-frequency ones. Akhbardeh et al. (2020) [8] showed that maintenance logbooks are full of domain terms, abbreviations and non-standard spelling that disrupt standard NLP pipelines, and Dima et al. (2021) [9] proposed Technical Language Processing, bringing domain knowledge and human input into NLP for maintenance text. Ihshaish et al. (2026) [10] compared classifiers on different records of the same cases, including NHTSA recalls; macro-F1 differed by 0.46 between the customer's report and the technician's report, more than between architectures, and they recommend evaluating on the text available at the point of decision.

Text classification and LLMs. Kowsari et al. (2019) [2] surveyed the text-classification pipeline from TF-IDF features to Naive Bayes, Logistic Regression and support vector machines. Brown et al. (2020) [3] showed that GPT-3 performs tasks from instructions and a few examples without fine-tuning. Wang et al. (2023) [11] found zero-shot LLMs effective classifiers on three of four datasets compared with traditional and deep-learning methods. Malyi et al. (2026) [12] benchmarked LLMs on wind-turbine maintenance logs, found a clear performance hierarchy and varied calibration, and, since no model was perfect, recommended a human-in-the-loop arrangement.

Evaluation methods. Raschka (2018) [13] recommended stratified splits and cross-validation and described McNemar's test for two classifiers on one test set; Demšar (2006) [14] recommended the Wilcoxon signed-rank test for paired comparisons of two methods.

IV. GAP ANALYSIS

Vehicle-maintenance studies target fleets and dealer networks with sensors and long histories [1], [4], [5], resources a small garage lacks. Fault-text studies use proprietary claims [7] or describe the difficulty of technical text without building a workshop tool [8], [9], and LLM evaluations [11], [12] do not compare against the keyword lists a low-cost system would otherwise use. Scheduling research favours learned dispatching for complex job shops [6] over simple, explainable rules. The gap addressed is a reproducible, zero-cost system for a single garage that records manual, AI and actual outcomes on every job, triages faults from the text available at intake [10], suggests mechanics with a transparent greedy rule, and is evaluated statistically on real public complaint narratives.

V. OBJECTIVES, VARIABLES AND HYPOTHESIS

Objectives. This study aims to — digitise the garage workflow from walk-in to invoice; predict fault category, labour time and cost from the complaint with an AI estimator, plus an offline keyword engine for demonstration; suggest the least-busy suitable mechanic with a greedy rule and specialist bonus; predict each vehicle's next periodic service; record manual, AI and actual outcomes on every job; and operate on free tiers with owner-only data security.

Variables. Table I lists the variables of the study.

TABLE I
VARIABLES OF THE STUDY

Type	Variable	Measurement
Independent	Classification method (baseline, keyword rules, three TF-IDF models)	Applied to the same narrative
Independent	Assignment strategy (greedy, round-robin, random)	Simulation condition
Dependent	Fault-category prediction	Correct or not; accuracy, macro-F1
Dependent	Workload balance	Daily SD of mechanics' actual hours; makespan
Control	Test set and job stream	6,248 complaints; seed 42

Hypotheses. The 80 % threshold is the accuracy target shown on the application's AI Insights page.

- H1. H0: the accuracy of a learned classifier on complaint narratives is at most 80 %. H1: it exceeds 80 %.
- H2. H0: the learned classifier and the keyword rules have the same probability of a correct classification. H1: the learned classifier is more accurate.
- H3. H0: greedy assignment does not reduce the daily dispersion of mechanic workload compared with random assignment. H1: it yields a lower dispersion.

VI. METHODOLOGY

A. Research Design

The study combines secondary, quantitative research with the design-science approach of Hevner et al. [15], in which knowledge is produced by building and rigorously evaluating an artefact. The secondary research was conducted in October 2026; the unit of analysis is one owner complaint. The original synopsis proposed Django, scikit-learn and three user roles; the scope was deliberately narrowed to one garage-owner user, Flutter Web with Firebase and Gemini, so that one student could build, run at zero cost and defend it. No questionnaire, interview, survey or user study was conducted. The demonstration data inside the application are synthetic and are not used as evidence.

B. Data Collection

Owner complaints for 23 popular car and SUV models, model years 2015–2022, were downloaded from the public NHTSA complaints API [16] and saved as CSV. Each record holds make, model, year, reported component(s), a free-text narrative, dates and crash and fire flags. The narrative was used as input because it is closest to what a customer says at the counter [10]. Single-component complaints were mapped to the application's categories (e.g., SERVICE BRAKES to Brakes, POWER TRAIN to Clutch & transmission); multi-component, unmapped and duplicate complaints, narratives under 20 characters and the mixed "ENGINE

AND ENGINE COOLING" component were excluded. Five methods were compared: a majority baseline; the application's keyword rules, ported line for line from its source code and applied without training; and TF-IDF (unigrams and bigrams) with Multinomial Naive Bayes, class-balanced Logistic Regression and a linear SVM, built with scikit-learn [17]. The data were split 80/20 with stratification (seed 42), and the trained models were also assessed with stratified 5-fold cross-validation [13].

C. Sample Size

Cochran's formula [18] gives the minimum sample for estimating a proportion: $n = Z^2 \times p \times (1 - p) / e^2 = (1.96^2 \times 0.5 \times 0.5) / 0.05^2 = 384.16 \approx 384$, with $Z = 1.96$ (95 % confidence), $p = 0.5$ and $e = 0.05$. The test set of 6,248 complaints far exceeds this minimum. The sample is purposive: 23 models sold in the United States, and only owners who chose to complain.

VII. DATA ANALYSIS AND FINDINGS

A. Description of the Data Set

Of 67,618 complaints downloaded, 46,684 named a single component and 31,236 could be labelled. Seven of the nine categories are represented: AC & cooling has no matching component and General service does not occur, since NHTSA records defects, not maintenance. Narratives are long (mean 99.4 words, SD 74.0, median 83, range 1–576). Table II shows the class distribution; the imbalance (Engine 32.2 %) is why macro-F1 is reported with accuracy.

TABLE II
LABELLED COMPLAINTS BY FAULT CATEGORY

Category	Records	Test set
Engine	10,065	2,013
Clutch & transmission	6,488	1,298
Battery & electrical	4,520	904
Body & paint	3,828	766
Suspension & steering	3,738	748
Brakes	1,867	373
Tyres & wheels	730	146
Total	31,236	6,248

Table III and Fig. 1 compare the five methods. Logistic Regression and the linear SVM are practically equal; Logistic Regression was taken forward because it yields class probabilities for a confidence display and had the most stable folds of the learned models.

TABLE III
CLASSIFICATION ON THE TEST SET AND 5-FOLD CV

Method	Accuracy	Macro-F1	CV accuracy
Majority baseline	0.322	0.070	0.322 ± 0.000
Keyword rules (demo)	0.576	0.429	—
TF-IDF + Naive Bayes	0.826	0.807	0.815 ± 0.006
TF-IDF + Logistic Reg.	0.906	0.907	0.902 ± 0.003
TF-IDF + Linear SVM	0.909	0.910	0.905 ± 0.004

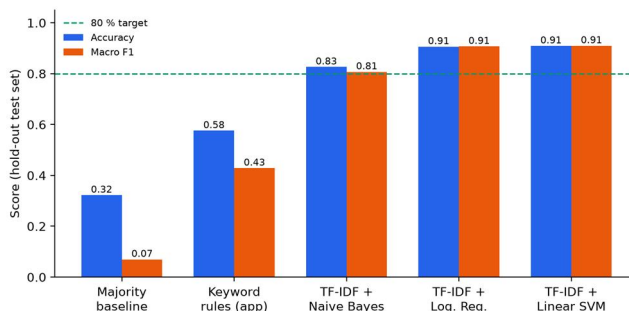


Fig. 1 Accuracy and macro-F1 of the five methods on the test set

B. Hypothesis 1

H0: accuracy $\leq 80\%$; H1: accuracy $> 80\%$. The Logistic Regression model was applied once to the 6,248 test complaints and tested with a one-sided exact binomial test ($\alpha = 0.05$). It classified 5,661 correctly (90.6 %); the lower bound of the one-sided 95 % Wilson interval is 0.8998 and $p = 1.09 \times 10^{-115}$, so H0 is rejected (Table IV). The keyword rules (3,598 correct, 57.6 %, $p = 1.0$) do not meet the target. Per-class F1 ranged from 0.840 (Battery & electrical) to 0.952 (Suspension & steering), and the heaviest terms included the misspelling "breaks" for Brakes.

TABLE IV
HYPOTHESIS TESTS ($\alpha = 0.05$)

Hypothesis	Test used and statistic	p-value	Decision
H1 (Logistic Reg.)	Exact binomial, one-sided; 5,661 / 6,248 = 90.6 %	1.09×10^{-115}	Reject H0
H1 (keyword rules)	Exact binomial, one-sided; 3,598 / 6,248 = 57.6 %	1.0	Fail to reject H0
H2	Exact McNemar, two-sided; discordant pairs 2,199 vs 136	< 0.001	Reject H0
H3	Wilcoxon signed-rank, one-sided; W = 177	3.85×10^{-42}	Reject H0

C. Hypothesis 2

H0: equal probability of a correct classification; H1: the learned classifier is more accurate. On the same 6,248 complaints, an exact two-sided McNemar test [13] on the discordant pairs gave 2,199 complaints where Logistic Regression was right and the rules wrong, against 136 the other way ($p < 10^{-300}$, reported as $p < 0.001$); H0 is rejected. The rules failed systematically: they predicted Engine for 2,936 complaints although only 2,013 were Engine, partly because the 7.6 % of complaints with no keyword hit fall back to Engine, and they assigned 365 complaints to General service and AC & cooling, which do not occur in the data. On the application's short, synthetic sample complaints, written to match the keywords, the same rules look highly accurate, so that in-app figure is only an illustration.

D. Hypothesis 3

H0: greedy assignment does not reduce daily workload dispersion compared with random; H1: it does. With no real workload data available, 250 working days (seed 42) of 12 jobs each were simulated for the five mechanics of the demonstration garage. Job categories followed the application's synthetic sample mix, estimated hours were the application's baseline hours, and actual hours were the estimate times a uniform factor between 0.7 and 1.4. Jobs were assigned by the greedy rule (lowest estimated open hours, less 1.5 h for a specialty match), round-robin or randomly, and the daily SDs of actual hours under greedy and random assignment were compared with a one-sided Wilcoxon signed-rank test [14]. Table V and Fig. 2 show the outcome; H0 is rejected ($W = 177$, $p = 3.85 \times 10^{-42}$), and makespan was also lower ($W = 576$, $p = 4.19 \times 10^{-40}$). This is a simulation of the rule's behaviour, not a measurement in a real garage.

TABLE V
SIMULATED MECHANIC WORKLOAD (250 DAYS)

Strategy	Mean daily SD (h)	Makespan (h)	Specialist share
Greedy (app)	1.49	8.90	41.2 %
Round-robin	2.23	10.04	12.9 %
Random	4.03	12.79	13.3 %

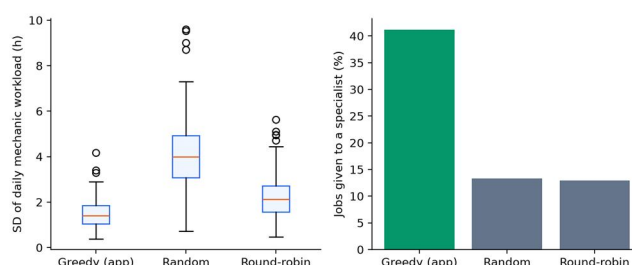


Fig. 2 Daily workload dispersion (left) and share of jobs given to a specialist (right) under three assignment strategies

E. Supporting Evaluation of the Artefact

Gemini was not evaluated on the data set: AI was kept switched off after development to guarantee zero cost and prevent quota misuse. A handful of live calls in October 2026 are reported as anecdotal observations only. Every call returned the correct category, Brakes: "Grinding noise from the front brakes and the pedal feels soft" (2016 Swift, 95,562 km) gave confidence 0.98, 2 hours and ₹3,500, and the misspelt "squeaking noise in brakes" gave 0.98, 2 hours and ₹3,200. Latency ranged from 2.1 s to 19.7 s, so the ≈ 2 s target was met only occasionally on the free tier. In one real job, 2 hours of labour (₹1,000) were recorded against the AI's 2 hours, and the invoice of ₹1,180 including 18 % GST was produced as a PDF in the browser. All 55 automated tests passed, and a live check confirmed that each user can read only their own Firestore data. Time and cost accuracy could not be evaluated, because NHTSA records contain neither.

VIII. RESULT

H1 is supported: a learned TF-IDF and Logistic Regression model triaged 90.6 % of real complaint narratives correctly, significantly above 80 %, with macro-F1 0.907. H2 is supported: it was correct on 2,199 complaints the keyword rules missed, against 136 the other way, so keywords work offline at no cost but classify real narratives poorly, which supports the premise that an AI component materially improves fault triage. H3 is supported for the rule's behaviour: in simulation, greedy assignment cut the daily workload spread from 4.03 h to 1.49 h and sent over three times as many jobs to a specialist. The production LLM was correct in every live test, but these were few and all concerned brakes, so its accuracy is not established.

IX. RECOMMENDATIONS BASED ON STUDY

- 1) Replace the keyword engine with a compact TF-IDF and Logistic Regression model, also usable as an automatic fallback when Gemini is unavailable; it was 33 percentage points more accurate and works offline at no cost.
- 2) Evaluate Gemini on the same 6,248 test complaints when quota allows, and compare all engines with McNemar's test.
- 3) Build a labelled corpus of real Indian job cards (short, often Hinglish, including routine service) to train and test in the intended setting.
- 4) Measure time and cost errors on several months of real job cards, which the application already records.
- 5) Keep the owner in the loop, as Malyi et al. [12] recommend, and retain the greedy rule with accurate hour estimates.

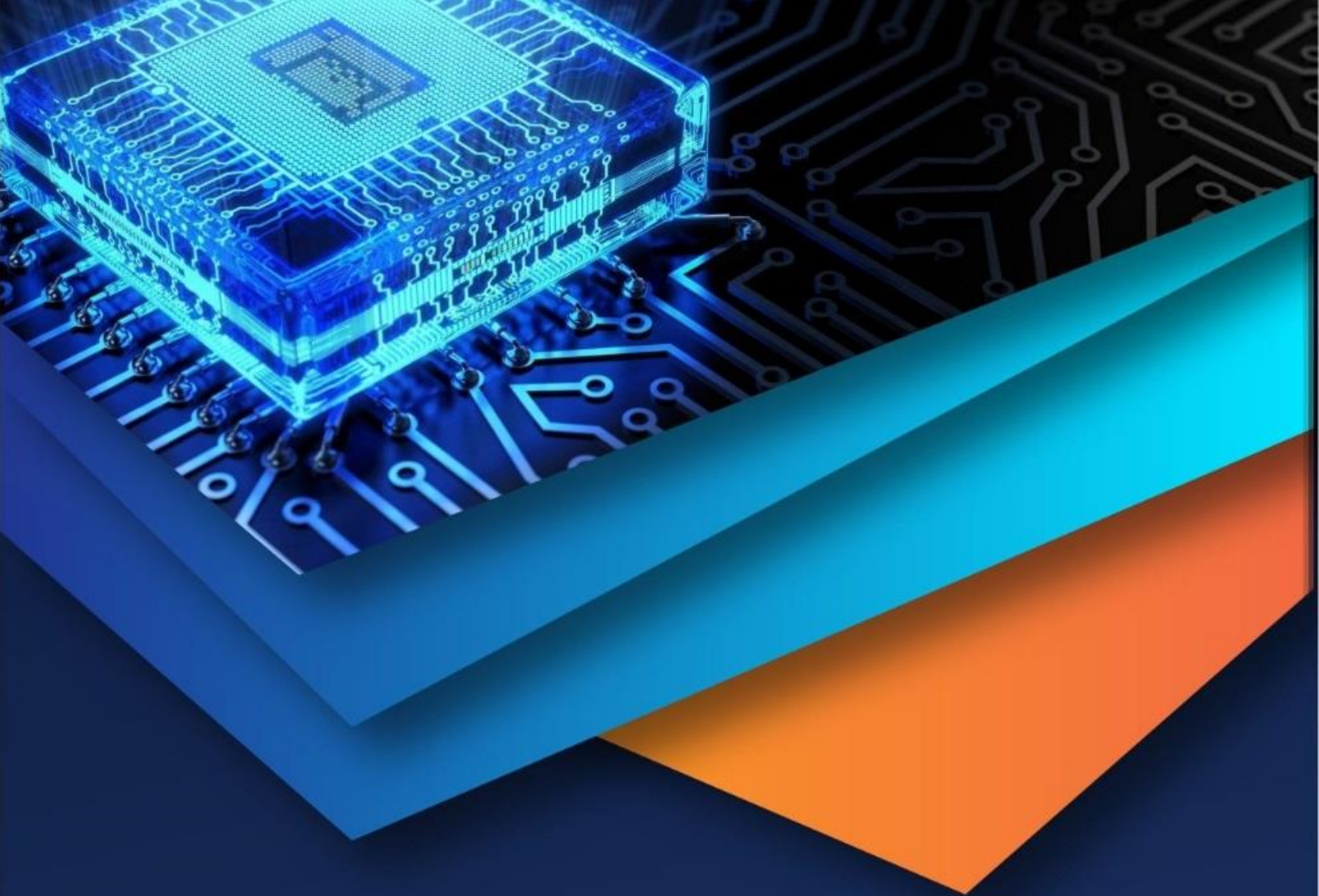
X. CONCLUSION

This study presented AutoCare AI, a zero-cost web application that digitises an independent garage's workflow, estimates fault category, time and cost from the customer's complaint, suggests mechanics, predicts the next service and records every estimate against the actual outcome.

On 31,236 public NHTSA complaints, a learned classifier triaged faults with 90.6 % accuracy, significantly better than the application's keyword rules (57.6 %), and a 250-day simulation showed that greedy assignment spreads workload far more evenly than random assignment. The limitations are stated plainly: Gemini was not evaluated at scale because AI was kept switched off to guarantee zero cost; the data are U.S. defect complaints that differ from Indian garage complaints and lack two of the nine categories; time and cost accuracy could not be tested; the in-app sample data are synthetic; the scheduling result is simulated; and no user study was conducted. The next step is to evaluate the LLM and the time and cost estimates on real job cards.

REFERENCES

- [1] A. Theissler, J. Pérez-Velázquez, M. Kettelgerdes, and G. Elger, "Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry," *Reliability Engineering & System Safety*, vol. 215, art. no. 107864, Nov. 2021, doi: 10.1016/j.res.2021.107864.
- [2] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, art. no. 150, 2019, doi: 10.3390/info10040150.
- [3] T. B. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, pp. 1877–1901.
- [4] S. Mishra, L. Vassio, L. Cagliero, M. Mellia, E. Baralis, R. Loti, and L. Salvatori, "Machine learning supported next-maintenance prediction for industrial vehicles," in *Proc. Workshops of the EDBT/ICDT 2020 Joint Conference (DARLI-AP 2020)*, *CEUR Workshop Proceedings*, vol. 2578, Copenhagen, Denmark, 2020.
- [5] Ş. Yıldırım, A. D. Yücekaya, M. Hekimoğlu, M. Ucal, M. N. Aydın, and İ. Kalafat, "AI-driven predictive maintenance for workforce and service optimization in the automotive sector," *Applied Sciences*, vol. 15, no. 11, art. no. 6282, 2025, doi: 10.3390/app15116282.
- [6] C. Zhang, W. Song, Z. Cao, J. Zhang, P. S. Tan, and C. Xu, "Learning to dispatch for job shop scheduling via deep reinforcement learning," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, pp. 1621–1632.
- [7] J. Pavlopoulos, A. Romell, J. Curman, O. Steinert, T. Lindgren, and M. Borg, "Automotive multilingual fault diagnosis," *arXiv preprint arXiv:2210.06918*, 2022.
- [8] F. Akhbardeh, T. Desell, and M. Zampieri, "MaintNet: A collaborative open-source library for predictive maintenance language resources," in *Proc. 28th Int. Conf. on Computational Linguistics: System Demonstrations (COLING 2020)*, Barcelona, Spain (Online), Dec. 2020, pp. 7–11, doi: 10.18653/v1/2020.coling-demos.2.
- [9] A. Dima, S. Lukens, M. Hodkiewicz, T. Sexton, and M. P. Brundage, "Adapting natural language processing for technical text," *Applied AI Letters*, vol. 2, no. 3, art. no. e33, 2021, doi: 10.1002/ail2.33.
- [10] H. Ihshaish, P. Mayhew, T. M. A. Zayet, and A. Del Amo, "The record is part of the task: Matched-record evaluation of text classifiers across maintenance, safety and recall reporting," *arXiv preprint arXiv:2609.16267*, 2026.
- [11] Z. Wang, Y. Pang, and Y. Lin, "Large language models are zero-shot text classifiers," *arXiv preprint arXiv:2312.01044*, 2023.
- [12] M. Malyi, J. Shek, A. McDonald, and A. Biscaya, "A comparative benchmark of large language models for labelling wind turbine maintenance logs," *IET Renewable Power Generation*, vol. 20, no. 1, art. no. e70212, 2026, doi: 10.1049/rpg2.70212.
- [13] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint arXiv:1811.12808*, 2018 (v3, 2020).
- [14] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [15] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.
- [16] National Highway Traffic Safety Administration (NHTSA), "NHTSA Datasets and APIs — Complaints," U.S. Department of Transportation, 2026. [Online]. Available: <https://www.nhtsa.gov/nhtsa-datasets-and-apis>
- [17] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [18] W. G. Cochran, *Sampling Techniques*, 3rd ed. New York, NY, USA: John Wiley & Sons, 1977.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)