SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Hybrid SVM–Random Forest Approach for Fault Diagnosis, Prognostics and Web-Based Predictive Maintenance Decision Support in Induction Motors

Punam J. Patil[1], Dr. Pankaj Zope[2], Manisha K. Bhole[3]

[1]Instrumentation Department, Bharati Vidyapeeth College of Engineering, Belapur, Navi Mumbai, Maharashtra, India, Email: punam.patil@bvcoenm.edu.in
[2]Department of Computer Engineering and Technology, SSBT's College of Engineering and Technology, Bambhori, Jalgaon, Maharashtra, India, Email: phzope@gmail.com
[3]Instrumentation Department, Bharati Vidyapeeth College of Engineering, Belapur, Navi Mumbai, Maharashtra, India, Email: manisha.bhole@bvcoenm.edu.in

ABSTRACT
Unplanned motor failure remains one of the costliest disruptions in continuous-process industries, and neither breakdown repair nor fixed-interval servicing makes use of the condition information a running motor already generates through its temperature, vibration and load signatures. This work builds a sensor-driven predictive-maintenance pipeline for three-phase induction motors that converts six raw channels — stator, rotor and winding temperature, bearing temperature, vibration and load — into three interpretable outputs: a Health Index, a Remaining Useful Life (RUL) estimate, and a Failure-Risk probability. A radial-basis-function Support Vector Machine assigns each operating snapshot to one of five condition classes, while two independently trained Random Forest regressors refine RUL and failure-risk estimates derived from a deterministic, weighted Health-Index formula. A four-condition rule engine converts these outputs into a specific maintenance recommendation, and the complete pipeline is exposed through a role-based Flask dashboard with live gauges, trend charts and downloadable reports. On a held-out 200-record test partition, the classifier reached 76.0% accuracy, performing well on the majority Critical-Failure and Healthy classes but failing entirely on the underrepresented Overload class — a class-imbalance effect examined in detail alongside a circularity caveat that affects the regression evaluation, since both regression targets are themselves formulas of the same input features. The paper closes by outlining the field-validation, class-balancing and temporal-modelling work needed before deployment.

Keywords: Support Vector Machine; Random Forest Regression; Fault Diagnosis; Remaining Useful Life; Health Index; Class Imbalance; Predictive Maintenance; Flask Dashboard; Induction Motor

I. Introduction

Electric motors sit at the center of almost every continuous-process operation — driving pumps, compressors, conveyors and agitators in manufacturing, power generation, oil and gas, and water-treatment plants. Because they run under variable load and thermal stress around the clock, they accumulate progressive faults such as bearing wear, insulation degradation, rotor eccentricity and winding overheating, and an unplanned stoppage of even one motor can halt an entire line at a cost far above that of a scheduled repair.

Two conventional strategies have long been used to manage this exposure. Reactive maintenance repairs a motor only once it has already failed, which keeps servicing cost low but maximizes downtime and the risk of cascading damage. Preventive maintenance instead services equipment on a fixed calendar regardless of its actual condition, which lowers the chance of a catastrophic failure but routinely performs work on machines that did not need it. Neither approach draws on the temperature, vibration nor load signals a motor is already producing while it runs.

Predictive maintenance (PdM) closes this gap by turning that live sensor stream into an estimate of present health and expected remaining life, so that intervention happens exactly when it is warranted. This paper reports a PdM pipeline built around two classical, comparatively lightweight learning models — an SVM for discrete fault classification and Random Forest regressors for continuous RUL and failure-risk estimation — deployed end-to-end as a browser-based dashboard rather than left as an offline notebook exercise.

The contributions of this paper are: (1) a physically-motivated, weighted Health-Index formulation that fuses six sensor channels into a single 0–1 condition score; (2) a five-class SVM fault classifier evaluated with a full confusion-matrix and per-class breakdown; (3) two Random Forest regressors trained on the same feature set to refine RUL and failure-risk estimates; (4) a four-condition rule-based maintenance-decision engine; (5) a role-based, real-time Flask dashboard with visualization and report generation; and (6) an explicit, evidence-based discussion of the dataset and evaluation limitations — most notably class imbalance and label circularity

— that must be resolved before the system can be considered field-ready.

II. Related Work

A. Sensing Modalities for Motor Fault Diagnosis

Vibration analysis remains the most established route to detecting mechanical faults — bearing degradation, rotor imbalance, shaft misalignment — because each fault type leaves a characteristic frequency signature. Motor Current Signature Analysis (MCSA) offers a non-intrusive alternative built on the same spectral logic applied to the stator current rather than to an accelerometer trace, and is attractive precisely where mounting a vibration sensor is impractical [5]. Subsequent MCSA work has extended this idea to specific fault types, including broken rotor bars [6], [7], and comparative studies confirm that electrical and vibration channels expose different, complementary aspects of the same degradation process rather than duplicating each other.

B. Feature Engineering and Learning Approaches

Time-domain statistics (RMS, kurtosis, skewness), frequency-domain features (FFT peaks, harmonics) and time–frequency representations (short-time Fourier, wavelet transforms) are the three feature families most consistently reported across the fault-diagnosis literature, and combining channels from more than one domain generally improves diagnostic robustness. On the modelling side, Support Vector Machines [1] and Random Forests [2] — both implemented here via scikit-learn [3] — remain competitive baselines because they perform well on modest, hand-engineered feature sets and stay interpretable through mechanisms such as RF feature importance, whereas deep sequence models (1-D CNNs, LSTMs, autoencoders) typically need far larger labelled datasets to outperform them and are harder to justify in a safety-relevant maintenance decision. LSTM-based RUL estimators such as Zheng et al. [12] illustrate the sequence-model alternative that becomes attractive once run-to-failure trajectories, rather than single snapshots, are available.

C. Predictive Maintenance as a Systems Problem

Broader systematic reviews of the PdM literature — Carvalho et al. [9] across manufacturing generally, and Zonta et al. [10] and Serradilla et al. [11] with an explicit Industry 4.0 and deep-learning lens — converge on two recurring gaps. The first is domain generalization: models trained on simulated or bench data frequently fail to transfer to field conditions where load, ambient temperature and sensor placement vary in ways a synthetic dataset cannot capture. The second is integration: a large share of published studies stop at the classification or regression stage without demonstrating an operator-facing decision layer. A further, less-discussed issue — severe class imbalance among fault types, for which resampling techniques such as SMOTE [13] are a standard remedy — recurs across simulated PdM datasets and is central to the results reported in Section VII. The present study is deliberately scoped to close the integration gap end-to-end while confronting the generalization and imbalance gaps directly, rather than setting them aside as future work.

III. Problem Statement And Design Objectives

Without continuous condition monitoring, a maintenance team cannot tell a motor that is quietly degrading from one that is genuinely healthy, which forces a choice between the excess cost of over-servicing and the excess risk of a reactive breakdown. The problem addressed here is therefore to build a system that ingests motor sensor data continuously, converts it into an interpretable health condition together with a quantitative RUL and failure-probability estimate, and translates those estimates into a specific maintenance instruction without requiring an expert to read raw sensor traces.

Three design targets follow from this problem statement: high classification accuracy across all fault categories — not just the dominant ones — so as to minimize both false alarms and missed faults; continuously updated Health-Index and RUL estimates visible at a glance; and rule-based decision support that names a specific action rather than only a risk number. Section VII returns to each target in light of the measured results, including where performance fell short.

IV. Dataset And Feature Description

The pipeline is developed and evaluated on a structured motor-operations dataset representing a three-phase induction motor — the most common industrial motor type owing to its ruggedness and low maintenance overhead. Each record is one operating snapshot; Table I lists the schema after dropping the non-informative identifier columns. The six raw sensor channels — three temperature zones, bearing temperature, vibration and load — form the shared input feature vector for every downstream model, on the reasoning that rising stator/rotor/winding temperature signals insulation stress, elevated bearing temperature or vibration signals mechanical wear, and abnormal load amplifies all of the above. The dataset comprises approximately 1,000 labelled records, split 80/20 into training and test partitions stratified by health-status class, leaving a held-out 200-record test set for the evaluation in Section VII.

TABLE I. DATASET SCHEMA (SELECTED COLUMNS)

Feature / Label	Type	Interpretation
Stator / Rotor / Winding Temperature	Float, °C	Insulation stress or overload indicator (3 channels)
Bearing Temperature	Float, °C	Mechanical wear or lubrication indicator
Vibration	Float, mm/s	Mechanical wear or misalignment indicator
Load Percentage	Float, %	Mechanical loading applied to the motor
Health Index	Float, 0–1	Derived overall condition score
RUL Hours	Integer	Derived remaining useful life
Failure Risk	Float, 0–1	Derived probability of failure
Motor_Health_Status	Categorical	Healthy / Warning / Critical, refined into 5 fault classes
Maintenance_Action	Categorical	Recommended action (Table III)

V. Proposed Methodology

A. Pipeline Overview

Fig. 1 summarizes the seven functional stages of the framework: sensor data acquisition, pre-processing, SVM-based condition classification, deterministic Health-Index computation, refinement of RUL and failure-risk via two Random Forest regressors, the rule-based decision engine, and the Flask dashboard that surfaces every downstream output. Each stage is implemented as an independently re-runnable script that persists its output to an intermediate file, which keeps the pipeline auditable and makes it straightforward to substitute a field-collected dataset later without re-architecting the downstream stages.

Fig. 1. Seven-stage predictive-maintenance pipeline, from raw sensor ingestion to the operator-facing dashboard.

B. Pre-Processing

Identifier columns are dropped, and the six raw sensor channels form the shared feature set for every model. Because kernel methods are sensitive to feature magnitude, the SVM classifier is trained on standardized features (zero mean, unit variance); the Random Forest regressors, being scale-invariant, are trained directly on the raw values. An 80/20 train–test split with a fixed random seed and stratification on the health-status label preserves class proportions across the partition.

C. SVM-Based Condition Classification

Motor condition is classified with an RBF-kernel SVM (C = 10, kernel coefficient "scale"), chosen for its ability to model the non-linear decision boundaries that separate fault classes not linearly separable in the raw six-dimensional sensor space. The target label is encoded prior to training, and performance is reported via overall accuracy, a full confusion matrix, and per-class precision/recall/F1 (Section VII-A).

D. Health-Index Formulation

Ahead of either regression model, a deterministic Health Index (HI) is built directly from sensor readings, giving the system an interpretable baseline that does not depend on a trained model. Each channel is first mapped to a normalized risk in [0, 1] via a min–max clip:

$$\text{risk}(x) = \text{clip}\left[(x - x_{min}) / (x_{max} - x_{min}), 0, 1 \right] \quad (1)$$

using the ranges in Table II, with the three temperature channels averaged into one temperature-risk term

before the four risk components are combined by fixed weight:
HI = 1 − (0.35·Temp_Risk + 0.25·Bearing_Risk + 0.25·Vibration_Risk + 0.15·Load_Risk) (2)

TABLE II. SENSOR NORMALISATION RANGES AND HEALTH-INDEX WEIGHTS

Channel	Normalisation Range	Risk Component	Weight
Stator / Rotor / Winding Temperature	40–130 / 45–135 / 60–155 °C	Temperature Risk (avg. of 3)	0.35
Bearing Temperature	40–120 °C	Bearing Risk	0.25
Vibration	0.5–18 mm/s	Vibration Risk	0.25
Load Percentage	20–110 %	Load Risk	0.15

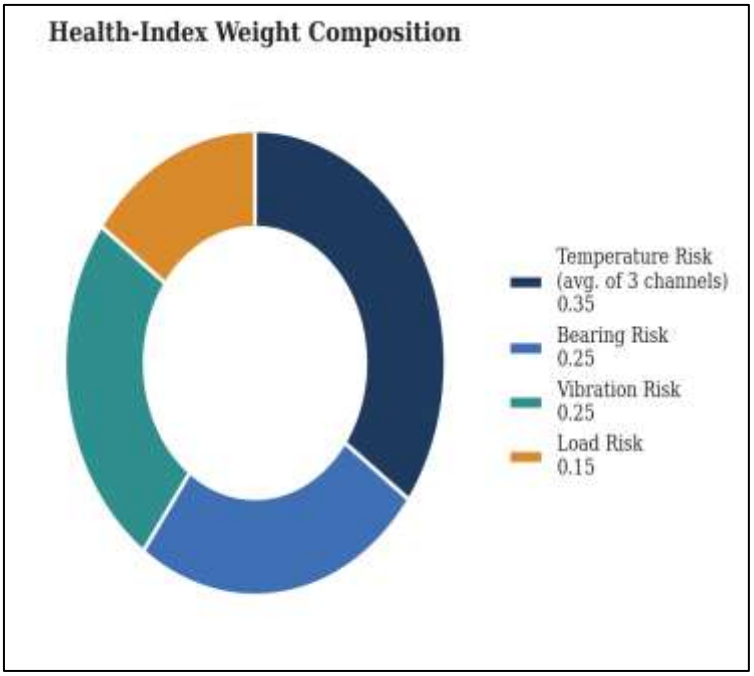


Fig. 2. Relative contribution of each risk component to the Health Index.

A value near 1 indicates a near-nominal motor; values approaching 0 indicate compounded stress across temperature, bearing condition, vibration and load simultaneously. This score is the foundation for both regression targets described next.

E. Remaining Useful Life Estimation

A baseline RUL label is first obtained by scaling HI against an assumed maximum operating life of 20,000 hours: RUL = HI × 20,000 (3). Table III and Fig. 3 show representative output rows — a Health Index of 0.339 corresponds to 6,780 hours of estimated remaining life, while 0.497 corresponds to 9,940 hours — illustrating the direct proportionality built into the baseline label.

TABLE III. REPRESENTATIVE HEALTH INDEX → RUL MAPPING

Health Index	RUL (hours)
0.339	6,780
0.437	8,740
0.365	7,300
0.497	9,940
0.429	8,580

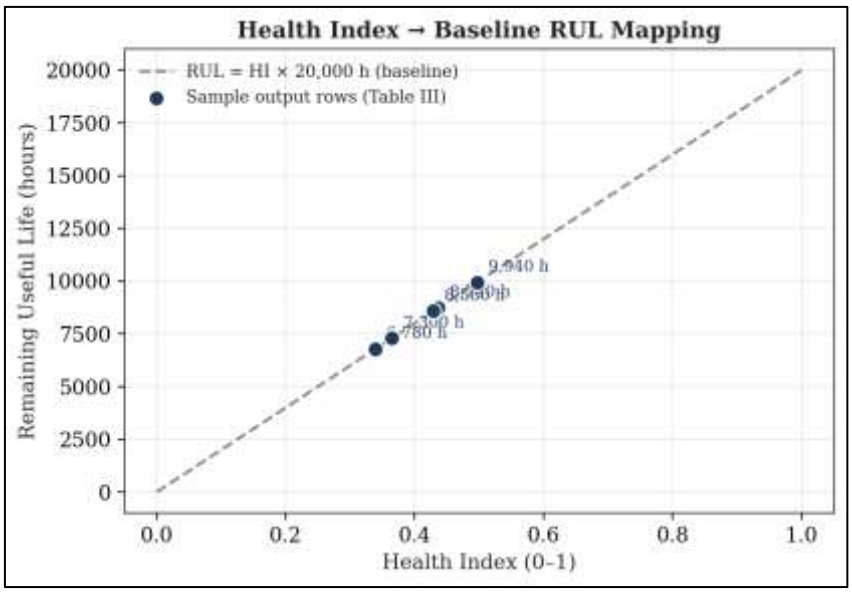


Fig. 3. Health-Index-to-RUL relationship: sample output rows (Table III) against the HI × 20,000 baseline.

This baseline label is then used as the training target for a Random Forest Regressor (200 estimators, fixed seed) fitted on the six raw sensor features. The motivation for this second, learned stage is that a tree ensemble can in principle capture non-linear channel interactions the fixed-weight formula cannot; because the training target is itself an exact function of the same inputs, however, the regressor is expected to recover the underlying formula closely rather than to add independent predictive information — a point examined in Section VII-B.

F. Failure-Risk Prediction

A second, independently trained Random Forest Regressor (200 estimators) predicts Failure Risk, a continuous value in [0, 1], from the same six-channel feature set (with Motor_Type, Motor_Health_Status and Failure_Risk itself excluded from the inputs). Model quality is assessed with Mean Absolute Error and R^2, consistent with standard regression practice.

G. Rule-Based Maintenance-Decision Engine

Predicted RUL and Failure Risk are combined by a rule engine (Table IV) that maps the two continuous outputs onto one of four maintenance recommendations. The rules are deliberately conservative: either a low RUL or a high failure risk alone is sufficient to escalate the action, so a motor that is not yet operationally exhausted but is showing an acute risk spike is still flagged.

TABLE IV. MAINTENANCE DECISION RULES

Condition (RUL or Failure Risk)	Recommended Action
RUL < 1,000 h OR Risk > 0.8	Immediate Maintenance / Shutdown
RUL < 3,000 h OR Risk > 0.6	Schedule Maintenance
RUL < 8,000 h OR Risk > 0.3	Inspection Recommended
Otherwise	No Maintenance Needed

H. Real-Time Dashboard

The pipeline is deployed as a Flask web application with role-based login ("admin" and "viewer"). Selecting a motor recomputes its risk terms, Health Index, RUL, failure risk, colour-coded health status and recommended action live, alongside a projected maintenance date obtained by converting RUL into calendar days under an assumption of continuous operation. The interface presents a semi-circular health gauge, a rolling health-trend chart over the last fifteen queries, and a normalised feature-importance bar chart, and supports CSV/PDF export (via ReportLab) and a multi-motor comparison view; an alert is raised automatically whenever failure risk exceeds 60%.

VI. Implementation Stack

The system is implemented entirely in Python (Table V) and developed in Visual Studio Code. Data moves through a chain of intermediate CSV artefacts — raw sensor data, an SVM-labelled file, an RUL-augmented file, a failure-risk-augmented file, and a final scheduled file consumed by the dashboard — with each stage re-runnable in isolation, which keeps the pipeline transparent and makes it straightforward to swap in field-collected data without redesigning the downstream stages.

TABLE V. SOFTWARE STACK

Component	Role in the Pipeline
Pandas / NumPy	Data loading, cleaning, normalization and numerical computation
Scikit-learn	SVM classifier and Random Forest regressors; preprocessing and metrics
Matplotlib / Seaborn	Offline evaluation visualization (confusion matrix, scatter, bar charts)
Flask	Web dashboard backend, routing and session-based role login
ReportLab	Programmatic generation of downloadable PDF motor-health reports

VII. Results and Discussion

A. Fault Classification Performance

The five-class SVM classifier reached an overall accuracy of 76.0% (152 of 200 test records correct). Fig. 4 shows the confusion matrix and Fig. 5 the per-class precision/recall/F1 breakdown (Table VI). The model is strong on the two most populous classes — Critical Failure (F1 = 0.821) and Healthy (F1 = 0.800), together 66% of the test set — weaker on Bearing Fault (F1 = 0.572), and fails completely on Overload Condition, producing zero correct predictions and, in fact, no Overload predictions at all. This is a direct consequence of class imbalance: Overload Condition is only 1.5% of the test set, not a sign that the fault type is inherently unrepresentable, and it means the measured accuracy falls short of the 90% target set in Section III. The matrix also shows meaningful cross-talk between Bearing Fault and both Critical Failure and Healthy, and between Overheating and Critical Failure — consistent with the Section II observation that adjacent fault categories can share overlapping sensor signatures when one underlying stressor (e.g. sustained overload) raises several channels at once.

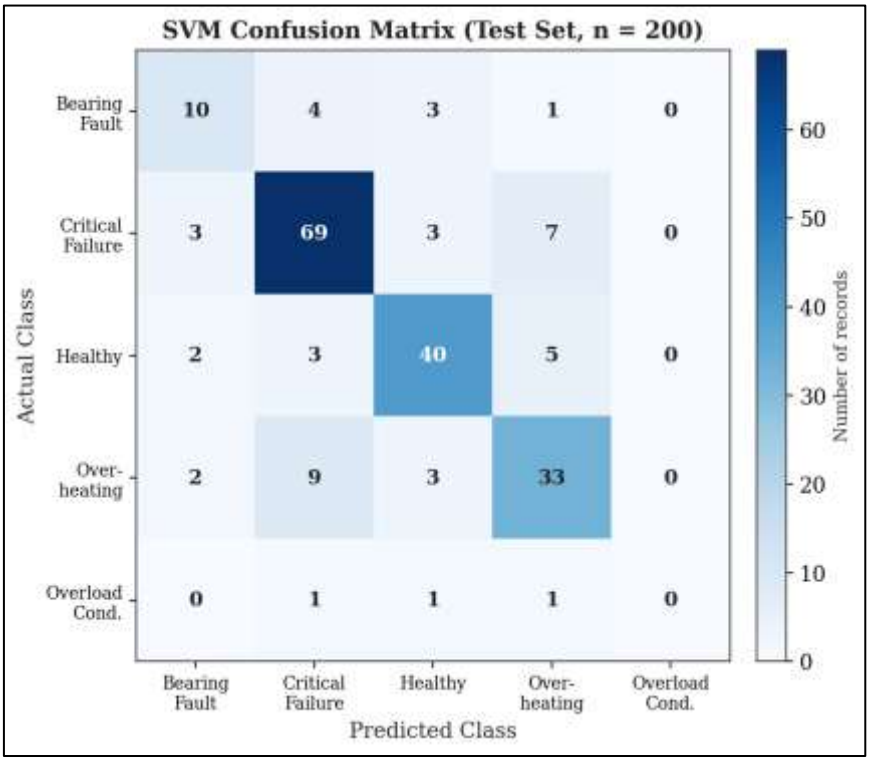


Fig. 4. Confusion matrix for the five-class SVM classifier (test set, n = 200).

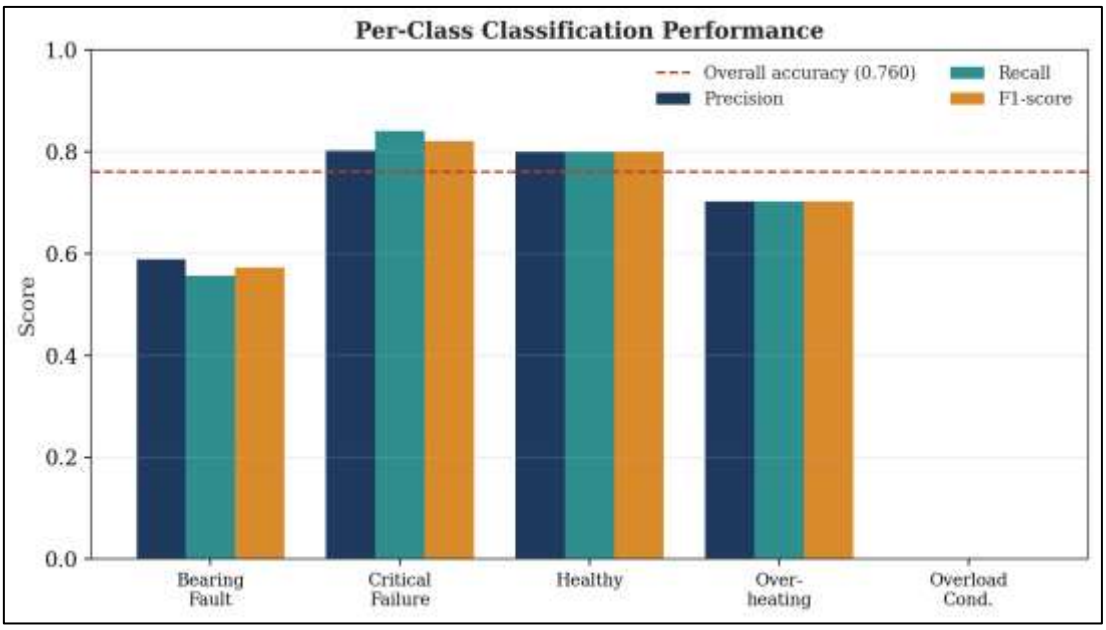


Fig. 5. Precision, recall and F1-score by class, against the 0.760 overall accuracy line.

TABLE VI. PER-CLASS CLASSIFICATION METRICS

Class	Support	Precision	Recall	F1-score
Bearing Fault	18	0.588	0.556	0.572
Critical Failure	82	0.802	0.841	0.821
Healthy	50	0.800	0.800	0.800
Overheating	47	0.702	0.702	0.702
Overload Condition	3	N/A (0 predicted)	0.000	N/A

B. RUL and Failure-Risk Regression Behaviour

The RUL regressor's predictions track the derived RUL labels closely across the observed 4,000–16,000-hour range, lying almost exactly on the actual-versus-predicted diagonal with only minor scatter at the extremes; the Failure-Risk regressor shows a similar pattern across the 0.1–0.9 range, with visibly larger scatter in the 0.4–0.7 mid-range. Both results need the same caveat: since each regression target is itself a deterministic function of the same six sensor inputs supplied to the regressor (Sections V-E, V-F), a tree ensemble is expected to recover that mapping almost exactly, so the reported fit is evidence of internal pipeline consistency rather than of validated accuracy against an independent, field-observed run-to-failure ground truth. Genuine validation would require RUL and risk labels drawn from actual time-to-failure records — run-to-failure rigs or maintenance logs — rather than from a formula computed on the model's own inputs. The wider mid-range scatter on Failure Risk is nonetheless a useful diagnostic: it marks the moderate-to-elevated risk boundary as the region of greatest model uncertainty, which is exactly where a maintenance planner would most want a reliable number, making it a priority area once independently labelled data is available.

C. Maintenance-Action Distribution

Applying the Table IV rule engine across the full dataset produces the distribution in Fig. 6: over 40% of records are flagged for immediate maintenance or shutdown, a share far higher than would be expected in a healthy operating fleet and best read as a property of how the simulated dataset was generated — deliberately weighted toward faulty instances to support classifier training — rather than as a realistic failure prevalence estimate. The distribution nonetheless functions as a sanity check on the rule engine itself: the thresholds partition the health spectrum monotonically and correctly reserve the narrow “Schedule Maintenance” band for motors sitting on the inspection/immediate boundary.

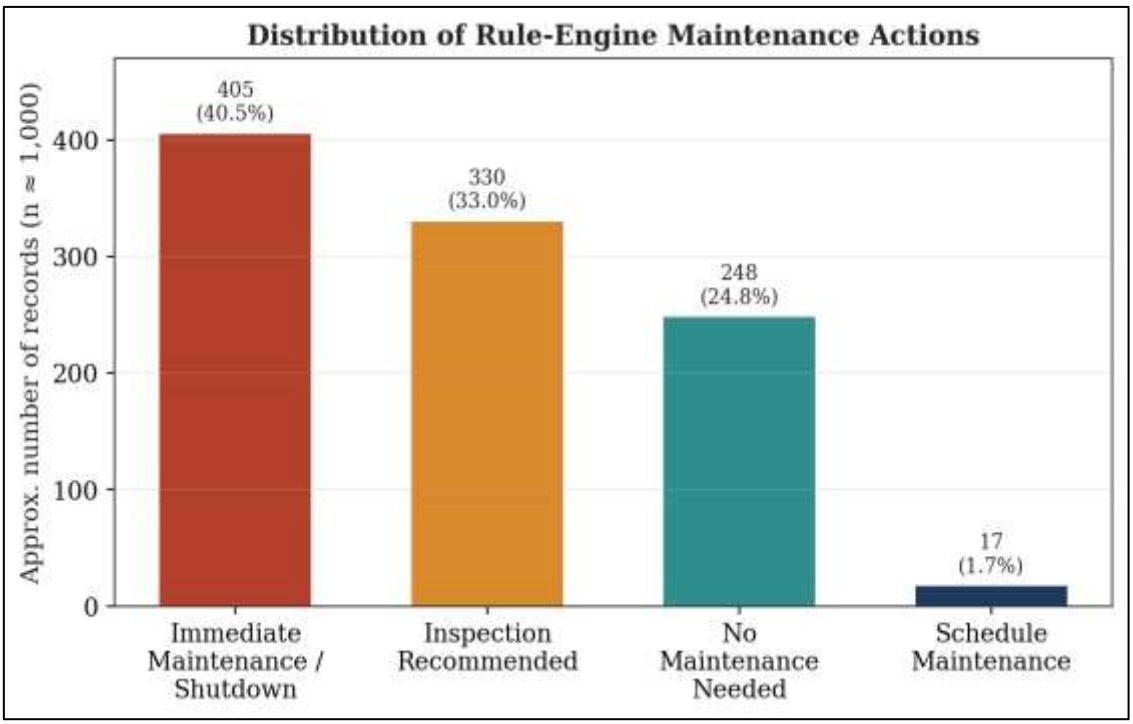
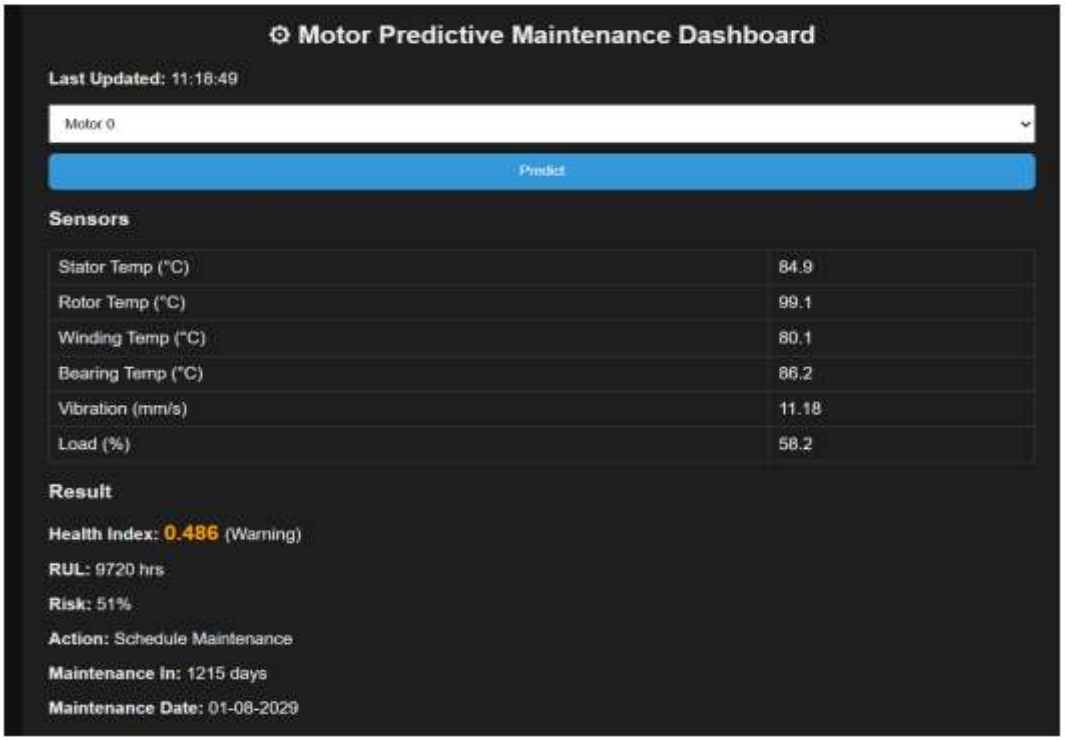
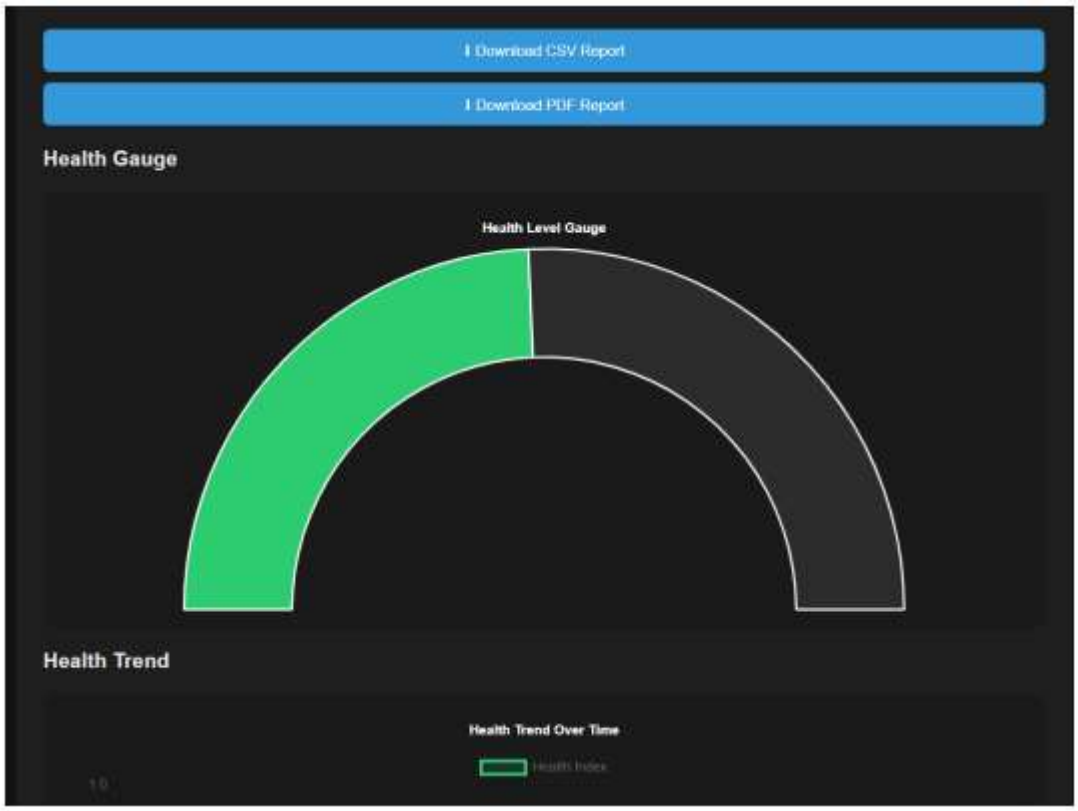


Fig. 6. Distribution of maintenance actions produced by the rule engine across the dataset (n ≈ 1,000).

D. Dashboard Verification

The Flask dashboard was exercised interactively across the test set and correctly reproduced, for each selected motor, its sensor readings, Health Index, RUL, failure risk, color-coded status, recommended action and projected maintenance date, alongside the gauge, trend chart and feature-importance chart of Section V-H. CSV/PDF export, role-based login, and the multi-motor comparison view all functioned as designed, confirming that the modelling components of Sections V-C through V-G operate as a coherent, operator-facing system rather than remaining isolated offline scripts.





E. Limitations

Synthetic ground truth: both RUL and Failure Risk training labels are deterministic functions of the same sensor inputs used for prediction, so the strong regression fit reflects pipeline self-consistency rather than validated field accuracy.

Class imbalance: Overload Condition is under 2% of the dataset and was never correctly classified, pulling overall accuracy below the 90% target and pointing to resampling (e.g. SMOTE [13]), class weighting, or targeted data collection as remedies.

Single motor archetype: the dataset represents one induction-motor configuration; generalization to other sizes, pole counts or duty cycles is untested.

No temporal modelling: each record is an independent snapshot; a sequence-aware model could exploit trend information ahead of any single-snapshot threshold breach.

Simulated rather than field data: reported metrics should be read as a proof-of-concept upper bound, not a deployment-ready figure.

VIII. Practical Benefits And Applications

Relative to reactive or calendar-based servicing, the framework enables early fault detection directly from sensor data without requiring manual trace interpretation, reduces unplanned downtime and its cascading costs, and replaces a single binary alarm with quantitative Health-Index, RUL and failure-risk estimates. Because the rule engine only escalates when a threshold is actually crossed, unnecessary preventive servicing is reduced alongside the manual-inspection burden, while the gauge, trend and feature-importance visuals make condition interpretable at a glance and the CSV/PDF export supports shift handover and audit trails. The same pipeline architecture extends to additional motors or sensor channels without redesigning the core models, making it a plausible fit for manufacturing plants, power-generation auxiliaries, oil-and-gas pumps and compressors, HVAC systems, automotive shop floors, material-handling conveyors, water/wastewater pumping, and chemical or textile processes requiring continuous operation.

IX. Future Work

The most consequential next step is replacing the deterministic-formula-derived RUL and failure labels with independently sourced ground truth — run-to-failure rigs or historical maintenance logs — so the regressors can be validated against outcomes not computed from their own inputs. Class-imbalance mitigation through targeted data collection, resampling such as SMOTE [13], or cost-sensitive training would directly address the Overload Condition recall failure of Section VII-A. On the modelling side, sequence-aware architectures such as LSTM or 1-D CNN networks [12] could exploit temporal trends across successive readings rather than treating

each snapshot independently, drawing on the broader deep-learning PdM literature [11], and a digital-twin simulation could support what-if testing of maintenance policies before they touch physical equipment. On the deployment side, integration with real IoT sensors and cloud infrastructure, a companion mobile app with SMS/email alerting, and additional sensor channels — current signature [5]–[7], voltage, and acoustic signals — would widen diagnostic coverage in line with the multi-sensor fusion approaches identified in Section II.

X. Conclusion

This paper presented a hybrid SVM–Random Forest pipeline for the predictive maintenance of industrial induction motors, combining an SVM fault classifier, two Random Forest regressors for RUL and failure risk, a deterministic Health-Index formulation, a four-condition rule-based decision engine, and a role-based Flask dashboard exposing all outputs to plant operators in real time. On a held-out test set the classifier reached 76.0% accuracy, strong on the dominant fault classes but unable to recognise the severely underrepresented Overload class, while the RUL and failure-risk regressors tracked their training targets closely — a result that must be read alongside the circularity inherent in labels derived from the same features used for prediction. Presented alongside these limitations rather than in spite of them, the system constitutes a working, vertically integrated diagnosis-to-decision proof-of-concept whose principal remaining gap is validation against independently sourced, field-collected run-to-failure data; with that step, together with the class-imbalance and temporal-modelling extensions of Section IX, the framework offers a credible foundation for a deployable Industry 4.0 predictive-maintenance solution.

References

1. C. Cortes and V. Vapnik, “Support-Vector Networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
2. L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
3. F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
4. R. K. Mobley, An Introduction to Predictive Maintenance, 2nd ed. Oxford, U.K.: Butterworth-Heinemann, 2002.
5. M. El Hachemi Benbouzid, “A review of induction motors signature analysis as a medium for faults detection,” IEEE Transactions on Industrial Electronics, vol. 47, no. 5, pp. 984–993, 2000.
6. K. Vinoth Kumar, “A Review of Voltage and Current Signature Diagnosis in Industrial Drives,” International Journal of Power Electronics and Drive Systems, vol. 1, no. 1, 2011.
7. S. Halder, S. Bhat, D. Zychma, and P. Sowa, “Broken Rotor Bar Fault Diagnosis Techniques Based on Motor Current Signature Analysis for Induction Motor — A Review,” Energies, vol. 15, no. 22, 8569, 2022.
8. Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, “Machinery health prognostics: A systematic review from data acquisition to RUL prediction,” Mechanical Systems and Signal Processing, vol. 104, pp. 799–834, 2018.
9. T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. P. Francisco, J. P. Basto, and S. G. S. Alcala, “A systematic literature review of machine learning methods applied to predictive maintenance,” Computers & Industrial Engineering, vol. 137, 106024, 2019.
10. T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, “Predictive maintenance in the Industry 4.0: A systematic literature review,” Computers & Industrial Engineering, vol. 150, 106889, 2020.
11. O. Serradilla, E. Zugasti, and U. Zurutuza, “Deep learning models for predictive maintenance: A survey, comparison, challenges and prospect,” Applied Intelligence, 2020.
12. S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, “Long short-term memory network for remaining useful life estimation,” in Proc. 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 88–95, 2017.
13. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.