iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78906

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Multi Sensor Predictive Maintenance Configurable System for Industry 4.0

Vaishnavi Borkar[1], Rahul Gone[2], Unmesh Kapure[3], Prajwal Gaikwad[4], Shubhangi Deshpande[5]
Information Technology, Pune Vidyarthi Griha's College of Engineering, Technology & Management, Pune, India

Abstract: ***Modern Industry 4.0 plants require maintenance strategies that respond to actual machine condition rather than fixed schedules [1], [2]. Conventional reactive and preventive approaches either wait for failure or over-service equipment, both at significant cost. This paper presents a configurable Multi- Sensor Predictive Maintenance (PdM) system that unifies three strategies — scheduled, usage-based, and ML-driven predictive maintenance — within a single operational platform [3]. The system is validated across three heterogeneous machine types CNC machine, pump, and compressor — each instrumented with five machine-specific sensor modalities and connected via an ESP32 edge node that streams data to a cloud backend over MQTT [4]. For each machine type, a dedicated XGBoost regres- sion model is trained on an engineered feature set comprising raw sensor readings, lag-1 features, threshold-exceedance difference features, a composite health index, and temporal indicators, to estimate Remaining Useful Life (RUL) in days [5], [6]. Evaluated on machine-wise held-out test sets of 31,504 samples each, the models achieve MAE of 3.07, 3.41, and 2.60 days for the CNC, pump, and compressor respectively, demonstrating consistent and actionable prediction accuracy across diverse degradation profiles. Real-time Grafana dashboards and a three-tier alert system (Critical <3 days, Warning <10 days, Normal ≥10 days) translate model outputs into prioritised maintenance actions for role-appropriate stakeholders [7].***

Index Terms: ***Predictive Maintenance, Industry 4.0, XGBoost, Remaining Useful Life (RUL), Multi-Machine Learning, Multi-Sensor Fusion, IIoT, Edge Computing, Condition Monitoring, Feature Engineering, Smart Manufacturing.***

I. INTRODUCTION

A. Background and Motivation

Industry 4.0 has placed dense sensor networks at the heart of modern manufacturing, with machinery such as CNC ma- chines, hydraulic presses, compressors, and pumps continu- ously generating streams of current, vibration, temperature, and pressure data [1], [8]. This shift creates a genuine op- portunity to move beyond time-driven maintenance toward condition-driven decisions [3]. Yet unplanned failures remain costly — production stoppages and emergency repairs in heavy fabrication can run to millions of rupees per incident, and tightly coupled production lines only amplify the damage [2]. Two conventional strategies dominate practice but both fall short. Reactive maintenance waits for breakdown, keeping upfront costs low while accepting unpredictable and often catastrophic downtime. Preventive maintenance avoids the worst failures but routinely replaces healthy components, inflating costs without proportional reliability gains [2], [3]. Predictive Maintenance (PdM) resolves both problems by basing decisions on measured machine health, answering the two questions that matter most operationally: is the machine currently degrading, and how many days remain before inter- vention is needed — the Remaining Useful Life (RUL)? [6]

B. Problem Statement and Proposed Solution

Real factory floors are heterogeneous: CNC machines de- grade through spindle speed drift and cutting force escala- tion, compressors through pressure and power consumption anomalies, pumps through vibration and flow-rate deviations. A single fixed-sensor, single-model PdM system cannot serve this diversity [3]. Equally important, plant engineers need explainable and configurable outputs — requirements that deep learning black-boxes rarely satisfy.

The system proposed here addresses these gaps through five design decisions: (i) *Machine-specific multi-modal sensing* — five sensor modalities per machine type, each paired with calibrated operational thresholds, connected to a dedicated ESP32 edge node [9]; (ii) *Unified maintenance strategies* — scheduled, usage-based, and ML-driven predictive maintenance co- exist in one platform; (iii) *Per-machine XGBoost models* — a separate gradient-boosted regressor is trained for each machine type, capturing machine-specific degradation signatures with full interpretability via feature importance scores [5]; (iv) *Cloud-edge architecture* — MQTT-based streaming from edge to cloud for ingestion, inference, and alert generation [4], [9]; and (v) *Real-time dashboards* — Grafana-based role-specific views with multi-channel alerting [7].

C. Contributions

This work contributes: (i) a unified three-strategy PdM platform validated across three heterogeneous machine types; (ii) machine-specific sensor instrumentation with threshold- aware feature engineering; (iii) per-machine XGBoost RUL models achieving MAE of 2.60–3.41 days across CNC, pump, and compressor equipment; (iv) a scalable multi-model train- ing and deployment pipeline; and (v) an Industry 4.0-ready deployment stack using MQTT, containerised microservices, and Grafana [1], [8].

II. RELATED WORK

A. Evolution of Maintenance Strategies

Industrial maintenance has moved through three paradigms [2], [3]. Reactive maintenance is cheap to initiate but costly when failures cascade. Preventive maintenance limits catas- trophic events but wastes resources on components still well within their service life. PdM conditions every maintenance decision on real-time sensor evidence, and recent surveys po- sition it as a central enabler of IIoT-integrated manufacturing [1], [3].

B. Sensor-Based Condition Monitoring

Comprehensive fault coverage demands multiple sensor modalities: vibration for bearing and gear faults, current for electrical load anomalies, temperature for lubrication health, and pressure or motor speed for hydraulic and drive-system de- viations [10], [11]. Studies consistently show that single-sensor systems miss failure modes that cross-modal fusion catches, and that machine-specific sensor selection outperforms generic multi-sensor configurations [3], [11].

C. Machine Learning for RUL Prediction

Classical approaches — linear regression, random forests, gradient boosting — are efficient and interpretable on engi- neered tabular features [2]. Among them, XGBoost has deliv- ered strong results on structured industrial datasets, benefiting from built-in regularisation, robust handling of missing values, and scalability [5]. Deep learning models such as LSTMs and Transformers excel on raw sequential data but demand large labelled fault datasets and complex inference pipelines constraints seldom met in real plants where failure events are rare [3], [12].

D. Gaps Addressed by This Work

Most deployed PdM systems are machine-specific but non- configurable, treat the three maintenance paradigms as sep- arate tools, and use fixed generic sensor sets [2], [3]. This paper addresses all three gaps through a unified, configurable platform where each machine type has its own optimised sensor configuration, threshold-aware features, and dedicated XGBoost model — interpretable enough for plant engineers to trust, and lightweight enough for real-time edge inference [5], [9].

III. SYSTEM ARCHITECTURE AND DESIGN

A. Architectural Overview

The system follows a four-layer distributed architecture: Physical Sensing, Edge Processing and Communication, Cloud Analytics and Decision, and Application and Visualization. It is designed around Industry 4.0 principles of modularity, inter- operability, real-time capability, and horizontal scalability [1], [13]. Layers are loosely coupled through standard interfaces, so new machine types can be on-boarded without redesigning any existing backend component.

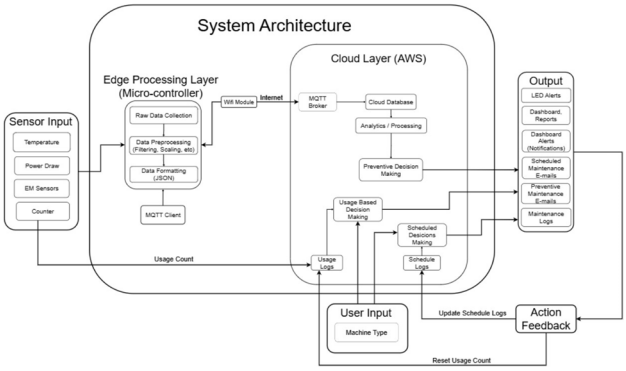


Fig. 1. Four-layer system architecture for the proposed Multi-Sensor PdM platform, supporting heterogeneous machine types with dedicated per-machine sensor nodes and ML models.

B. Physical Sensing Layer

Each machine type carries a dedicated sensor node capturing five modalities chosen to best characterise that machine's dominant failure mechanisms. Table I summarises the sensor configurations and operational thresholds for the three machine types evaluated in this work. A non-intrusive clamp current sensor or power transducer tracks real-time electrical load, while vibration, pressure, temperature, and speed sensors capture complementary degradation indicators [9]. All read- ings are edge-timestamped to preserve inter-modal alignment across the multivariate stream [13].

TABLE I
MACHINE-SPECIFIC SENSOR CONFIGURATIONS AND THRESHOLDS

Machine	Sensor	Threshold
CNC Machine	Current	15.0 A
	Cutting Force	200.0 N
	Spindle Speed	5000.0 RPM
	Temperature	80.0 °C
	Vibration	50.0 mm/s
Compressor	Humidity	60.0 %
	Motor Speed	4000.0 RPM
	Power Consumption	20.0 kW
	Pressure	150.0 bar
	Temperature	85.0 °C
Pump	5 sensors (machine-specific)	

C. Edge Processing and Communication Layer

The ESP32 microcontroller handles ADC sampling, GPIO interrupt-based stroke counting with debounce logic, and lightweight moving-average filtering to suppress electromag- netic noise before transmission [9]. Structured messages con- taining machine ID, sensor IDs, timestamps, and filtered values are published to machine-specific MQTT topics [4]. QoS levels balance delivery reliability against network overhead. A local circular buffer queues readings during connectivity outages and retransmits on reconnection, preventing data loss in legacy factory environments.

D. Cloud Analytics and Decision Layer

An MQTT broker routes incoming messages to an ingestion service that validates schemas, normalises data, and writes to a time-series database. A relational store holds machine metadata, per-machine thresholds, and maintenance schedules. A configuration service lets operators update scheduled in- tervals, stroke-count thresholds, and RUL alert boundaries without redeploying any service. The analytics engine runs machine-specific feature extraction, routes each data stream to its dedicated XGBoost model, and applies unified decision logic: scheduled logic checks elapsed time since last service; usage logic accumulates strokes; predictive logic maps the per- machine RUL estimate to Critical (<3 days), Warning (<10 days), or Normal (≥10 days), raising an alert whenever any engine signals that service is due [5], [6].

E. Application and Visualization Layer

Grafana dashboards deliver role-appropriate views: opera- tors see live machine status and alerts; engineers drill into per- machine sensor trends and health-index histories; management reviews fleet-level KPIs such as MTBF and maintenance frequency across all three machine types [7]. Alerts reach personnel via web UI, email, and SMS. Integration with CMMS platforms enables automatic work-order generation from model alerts, closing the loop between prediction and execution [13].

F. Scalability and Security

New machine types plug into the existing MQTT topology by registering a unique topic and deploying a machine-specific model artifact — no backend structural changes required.

Analytics microservices scale horizontally via Docker and Kubernetes in response to load [9]. All MQTT traffic uses TLS; edge devices authenticate through certificates or tokens; dashboard access is governed by RBAC; and credentials are managed through encrypted secrets storage.

IV. PREDICTIVE ANALYTICS AND ML METHODOLOGY

A. *Overview of the Predictive Pipeline*

The predictive module operates as a per-machine pipeline: for each of the three machine types (CNC, pump, compres- sor), a dedicated pipeline performs RUL target construction, machine-specific feature engineering, XGBoost model train- ing, and inference with alert generation. Each pipeline is trained and evaluated independently, ensuring that models cap- ture the distinct degradation characteristics of their respective equipment [5], [6].

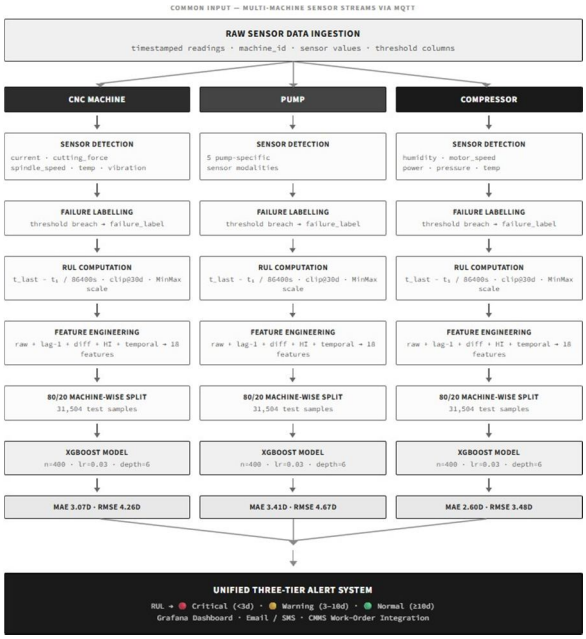


Fig. 2. End-to-end per-machine ML pipeline for XGBoost-based RUL estimation, applied independently to CNC, pump, and compressor datasets.

B. *Datasets and RUL Target Construction*

Three machine-type datasets are used: *cnc_machine.csv*, *pump_machine.csv*, and *compressor_machine.csv*. Each con- tains timestamped five-sensor readings sorted chronologically per machine ID, along with sensor-threshold columns. The failure label is computed dynamically — a timestep is labelled as a failure event if any sensor reading exceeds its correspond- ing threshold, ensuring consistent labelling across machines regardless of pre-existing annotations.

RUL is computed per machine instance as the time remain- ing until the last recorded timestamp:

$$\mathrm{RUL\_days}_i = \frac{t_{\mathrm{last}} - t_i}{86400\ \mathrm{s}}$$

Values are clipped at 30 days to keep the model focused on the operationally relevant planning window, then MinMax- scaled to [0, 1] for training. Predictions are inverse-transformed at inference time to recover day-unit estimates [6]. After feature engineering, each dataset yields 154,656 records across all machine instances.

C. *Feature Engineering*

Four feature groups are constructed for each machine type, producing 18 engineered features in total [5], [14]:

1) *Raw Sensor Features (5):* The direct readings from all five machine-specific sensor modalities at each timestep.
2) *Lag-1 Features (5):* For each sensor, the value from the immediately preceding timestep is appended (e.g., vibration_lag1). This encodes short-term rate-of- change, allowing the model to detect accelerating degra- dation trends without a dedicated sequence model.
3) *Threshold-Difference Features (5):* For each sensor s, a signed exceedance feature is computed as $s - s_{\mathrm{threshold}}$. Positive values indicate the sensor is operating above its safe operating limit, providing a direct, threshold-aware degradation signal that is machine-specific and physically interpretable.
4) *Composite Health Index (1):* The per-sensor exceedances are clipped to zero (negative values discarded) and summed:

$$HI = \sum_{s} \max(0, s - s_{threshold})$$

This scalar aggregates the total threshold violation burden across all sensors at each timestep. Higher values indicate the machine is simultaneously stressing multiple opera- tional limits — a strong predictor of imminent failure.

5) *Temporal Features (2):* Hour-of-day and day-of-week encode cyclical patterns such as shift changes and peak- load periods that influence sensor baselines independently of machine health state [14].

Rows with NaN values introduced by the lag-1 shift on each machine's first record are dropped before training. Fig. 3 illustrates the complete feature engineering pipeline.

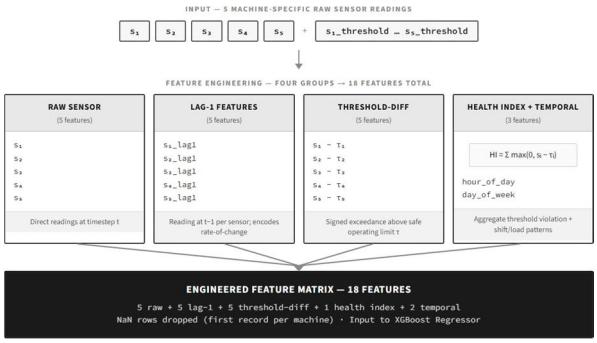


Fig. 3. Feature engineering pipeline: raw readings, lag features, threshold- difference features, composite health index, and temporal features feed the per-machine XGBoost regressor.

D. Train-Test Split

Each dataset is partitioned machine-instance-wise: 80% of machine IDs are assigned to the training set and the remaining20% to the test set. All timestamped records for a given ma- chine fall entirely in one partition, preventing data leakage and ensuring that evaluation metrics reflect genuine generalisation to machine instances not seen during training [2]. This split yields 31,504 test samples per machine type.

E. XGBoost Regression Model

A separate XGBoost regression model is trained per ma- chine type, predicting the MinMax-scaled RUL from the 18- feature engineered matrix [5]. XGBoost is selected because the feature matrix is tabular and explicitly constructed, failure- event data is sparse, inference must be low-latency, and feature importance scores provide the transparency that plant engineers require for root-cause analysis [2], [5].

TABLE II
XGBOOST HYPERPARAMETERS (IDENTICAL ACROSS ALL THREEMACHINE MODELS)

Hyperparameter	Value
n_estimators	400
learning_rate	0.03
max_depth	6
subsample	0.9
colsample_bytree	0.9
reg_alpha (L1)	0.1
reg_lambda (L2)	1.0

The same hyperparameter configuration is applied to all three models, demonstrating that a single well-tuned archi- tecture generalises across machine types when paired with machine-specific sensor sets and threshold-aware features. The low learning rate (0.03) paired with 400 estimators promotes stable convergence; row and column subsampling (both 0.9) add stochastic regularisation that guards against overfitting on machines with limited degradation history [5].

F. Model Persistence and Deployment

Each trained model is serialised alongside its RUL scaler, feature column list, and sensor threshold dictionary. This four- artifact bundle is sufficient for complete inference on a new machine stream — the cloud analytics engine loads the bundle corresponding to the incoming machine type and performs all preprocessing and prediction steps deterministically and reproducibly [6], [9].

G. Inference and Alert Generation

At inference time, the most recent feature vector for each machine instance is extracted and passed through the cor- responding machine-type model. The scaled prediction is inverse-transformed to days and clipped to [0, 30], then classi- fied by the three-tier policy shown in Table III and illustrated in Fig. 4.

TABLE III
RUL -BASED MAINTENANCE DECISION POLICY

Predicted RUL	Status	Action
< 3 days	CRITICAL	Immediate maintenance required
3–10 days	WARNING	Schedule maintenance soon
≥ 10 days	NORMAL	No immediate action needed

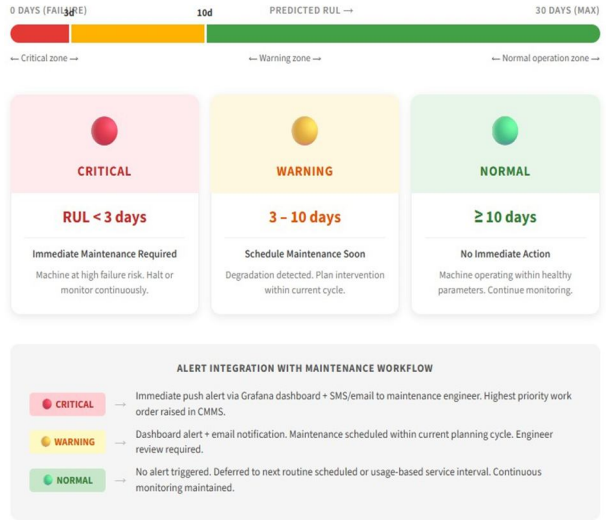


Fig. 4. Three-tier alert system mapping predicted RUL to maintenance actions. Thresholds are user-configurable via the cloud configuration service.

Alert thresholds are user-configurable through the cloud configuration service, allowing maintenance engineers to cali- brate sensitivity per machine type and operational context [6], [9].

V. RESULTS AND PERFORMANCE EVALUATION

A. Experimental Setup

The multi-machine pipeline was implemented in Python using XGBoost and scikit-learn [5], [14]. Three machine- type datasets were processed independently through the same pipeline: *cnc_machine.csv*, *pump_machine.csv*, and *compres- sor_machine.csv*. Each dataset was sorted chronologically per machine, failure labels were derived from threshold breaches, RUL targets were computed and clipped at 30 days, and machine-wise 80/20 splits were applied. All runs used random seed 42 for reproducibility. Post-feature engineering, each dataset contains 154,656 records with 18 features.

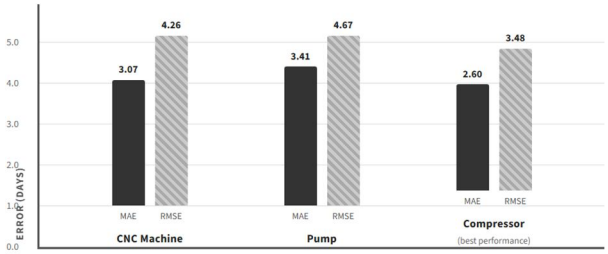
B. Per-Machine XGBoost Results

Table IV summarises prediction performance for each machine type on the held-out test set of 31,504 samples. Fig. 5 visualises the MAE and RMSE values across all three machine types.

TABLE IV
PER-MACHINE XGBOOST RUL PREDICTION PERFORMANCE

Machine Type	MAE (days)	RMSE (days)	Test Samples
CNC Machine	3.07	4.26	31,504
Pump	3.41	4.67	31,504
Compressor	**2.60**	**3.48**	31,504

The compressor model achieves the best performance (MAE 2.60 days, RMSE 3.48 days), likely owing to the more structured and predictable degradation profile of pressure and motor speed signals in compressor operation. The pump model shows the highest error (MAE 3.41 days), consistent with the more stochastic nature of pump wear driven by fluid dynamics and cavitation effects. The CNC model (MAE 3.07 days) performs in between, reflecting the moderately complex interplay of cutting force, spindle speed, and vibration in machining operations. All three models produce predictions within a practically useful window for a 30-day maintenance planning horizon [2], [5], [6].



MACHINE TYPE	MAE (DAYS)	RMSE (DAYS)	TEST SAMPLES	FEATURES
CNC Machine	3.07	4.26	31,504	18
Pump	3.41	4.67	31,504	18
Compressor	**2.60**	**3.48**	31,504	18

Fig. 5. Per-machine XGBoost RUL prediction error (MAE and RMSE, in days) across CNC machine, pump, and compressor on held-out test sets of 31,504 samples each.

C. Prediction Distribution Analysis

Fig. 6 shows predicted versus actual RUL scatter plots for all three machine models. In all three cases, predictions follow the ideal diagonal closely, with the bulk of points concentrated near the regression line. The characteristic fan- shaped dispersion — tighter near 0 days and wider near 30 days — is expected behaviour for RUL regression: near-failure states are strongly constrained by threshold breaches across multiple sensors, while healthy mid-life states have wider possible trajectories [5], [14]. The compressor plot shows the tightest clustering, corroborating its best quantitative metrics.

D. Per-Machine Alert Classification

Following inference, each machine instance receives a pre- dicted RUL mapped to one of three alert states. Table V illustrates the classification outcomes.

TABLE V
MACHINE HEALTH STATUS BASED ON PREDICTED RUL

Machine Status	Predicted RUL	Alert State
Degraded Instance	< 3 days	CRITICAL
Moderately Worn	3–10 days	WARNING
Healthy Instance	≥ 10 days	NORMAL

This classification lets maintenance planners act immedi- ately on CRITICAL machines, schedule WARNING machines within the current planning cycle, and defer NORMAL ma- chines to routine service — directly translating probabilistic model output into a prioritised maintenance workload across all three machine types [6], [9].

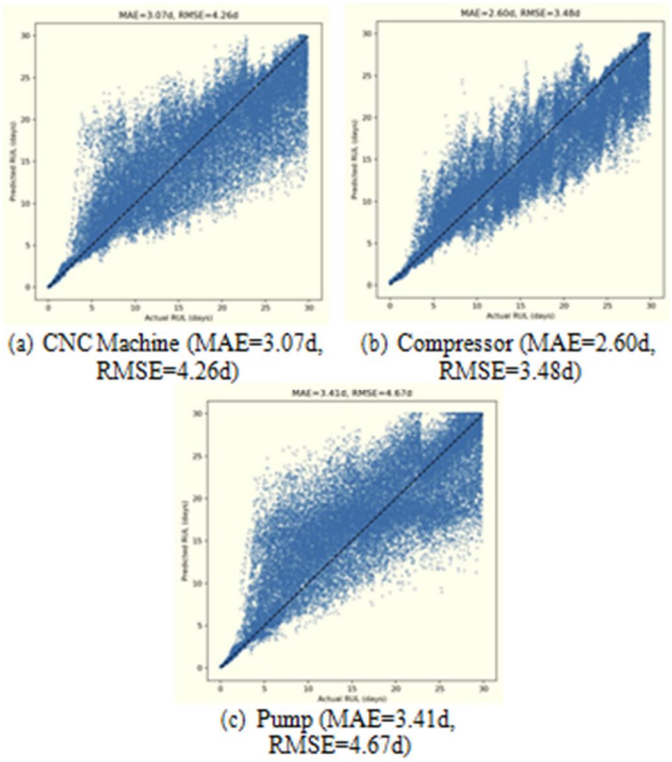


Fig. 6. Predicted vs. actual RUL (days) for all three machine-type models on held-out test sets. The dashed line indicates ideal prediction. Fan-shaped dispersion widens at higher RUL values, consistent with greater trajectory uncertainty in healthy machine mid-life.

E. Discussion of Results

The threshold-difference features are a key driver of model performance: by expressing each sensor reading relative to its operational limit rather than as a raw value, the model receives a signal that is calibrated to machine-specific risk rather than absolute magnitude. This makes the same XG- Boost architecture transferable across machine types without hyperparameter retuning — only the sensor set and threshold values change [5], [14]. The machine-wise train-test split ensures that reported metrics reflect generalisation to unseen machine instances, which is the relevant criterion for industrial deployment where new machines are regularly commissioned [2], [3].

VI. PRACTICAL CONSIDERATIONS

A. Data Quality and Labeled Failure Data

Labeled fault events are inherently rare on real factory floors, and maintenance records are often incomplete. The system mitigates this in two ways: failure labels are computed automatically from threshold breaches rather than relying on pre-existing annotations, and RUL is defined as a continuously computable regression target rather than a categorical fault label [3], [6]. This design dramatically reduces the manual labelling burden while still producing a rich supervision signal for training. Standardised data collection and threshold calibration protocols remain a worthwhile organisational investment for maximising long-term model accuracy.

B. Model Robustness and Retraining

Operating conditions shift with production schedules, sea- sonal factors, and cumulative wear. Each machine-type model may drift over time as its operational envelope changes [2], [5]. The system's feedback loop ingests recorded maintenance events back into the training pipeline, enabling periodic re- training. Because models are stored as lightweight serialised artifacts (model, scaler, feature columns, thresholds), rede- ployment after retraining requires only replacing the relevant bundle without modifying any backend infrastructure [9]. Transfer-learning initialisation from a related machine type can shorten the cold-start period for new equipment [3].

C. Interpretability and Deployment

XGBoost's native feature importance scores let engineers verify that predictions are driven by physically meaningful signals — threshold-breaching sensors or rising health index rather than artefacts of the training data [5]. Surfacing these rankings in the Grafana dashboard builds operator trust and supports root-cause analysis across all three machine types [7]. On the deployment side, the entire inference stack per machine requires only four small files, making it far lighter than neural network checkpoints and well-suited to on-premise or edge- side inference in bandwidth-constrained factory environments [9].

VII. CONCLUSION AND FUTURE WORK

This paper presented a configurable Multi-Sensor PdM system for Industry 4.0, validated across three heterogeneous machine types — CNC machine, pump, and compressor. By training dedicated XGBoost models per machine type on threshold-aware, lag-enriched feature sets, the system achieves MAE of 2.60–3.41 days across all three machines on held- out test sets of 31,504 samples each [5], [6]. The three-tier alert system and Grafana dashboards connect model output directly to actionable maintenance decisions, while the unified platform simultaneously supports scheduled, usage-based, and ML-driven maintenance strategies [4], [9].

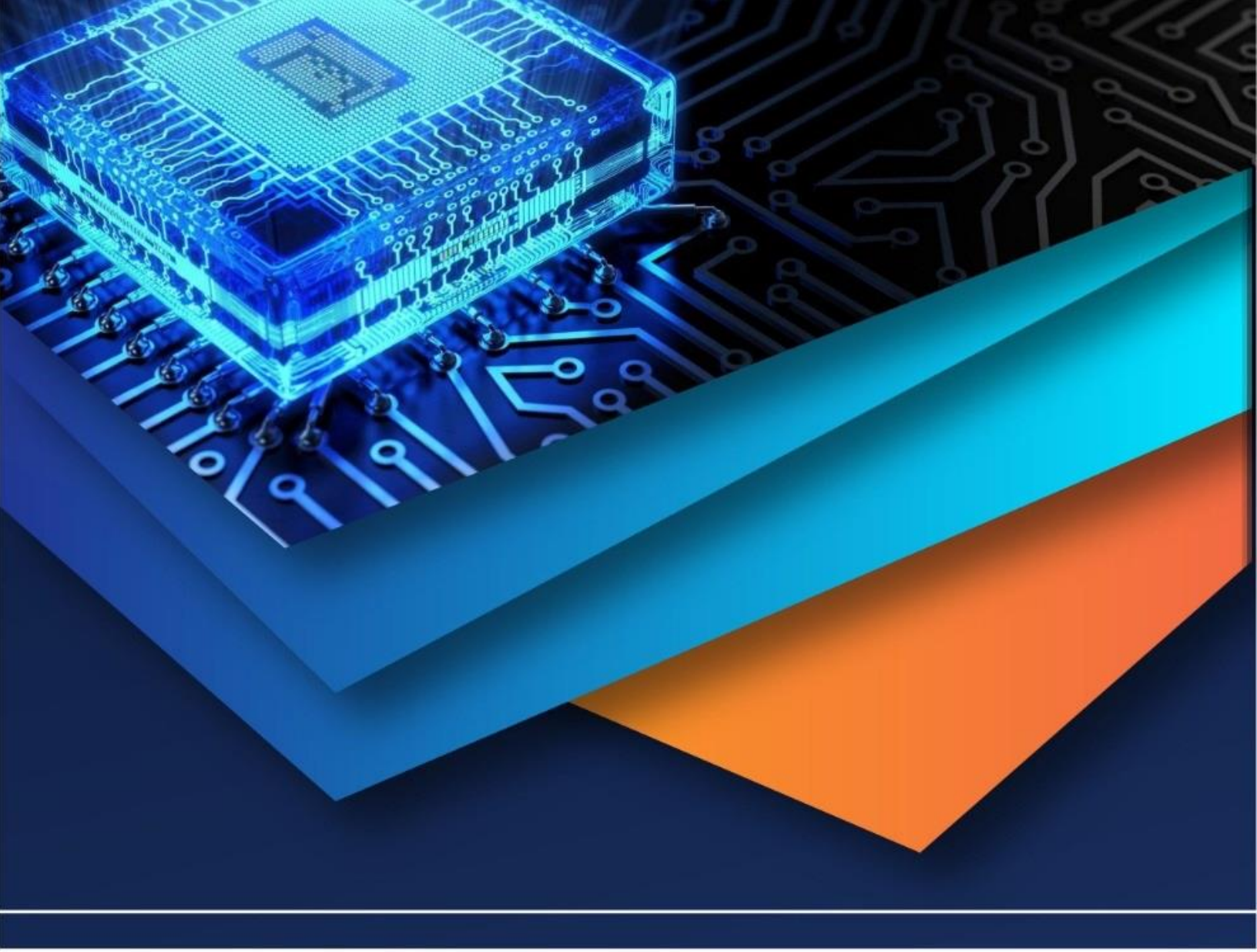
Future work will pursue: large-scale longitudinal deploy- ment on a live factory floor to quantify actual reductions in downtime and maintenance cost; digital twin integration to simulate degradation scenarios and generate synthetic fault data for machines with limited failure history [1]; federated learning across multiple sites to share model improvements without exposing raw operational data [3]; hyperparameter optimisation per machine type to further improve model ac- curacy; expanded sensing (acoustic emission, flow rate) for richer fault observability; and CMMS integration for automatic work-order generation [13]. Together, these extensions move the framework toward fully autonomous, self-improving main- tenance for next-generation smart manufacturing [1], [3].

VIII. ACKNOWLEDGMENT

The authors sincerely thank their project guide and fac- ulty mentors for their invaluable guidance, technical insights, and constructive feedback throughout this work. They extend appreciation to the Department of Information Technology for providing the necessary infrastructure, computational re- sources, and development tools. Special thanks to industry pro- fessionals and domain experts whose practical insights helped align the system with real-world Industry 4.0 requirements. The authors also acknowledge peers and collaborators for their discussions, testing, and critical review of the system design and methodology.

REFERENCES

[1] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.
[2] J. Lee, B. Bagheri, and H. A. Kao, "A cyber-physical systems archi- tecture for industry 4.0-based manufacturing systems," Manufacturing Letters, vol. 3, pp. 18–23, 2015.
[3] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 812–820, 2015.
[4] P. J. R. Gama, E. Zio, and R. L. M. Silva, "Remaining useful life estimation: A review," Mechanical Systems and Signal Processing, vol. 115, pp. 1–20, 2019.
[5] U. Khan, D. Cheng, F. Setti, F. Fummi, M. Cristani, and L. Capogrosso, "A comprehensive survey on deep learning-based predictive mainte- nance," ACM Transactions on Embedded Computing Systems, vol. 24, no. 4, pp. 1–42, 2025.
[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.
[7] A. Banks and R. Gupta, "Mqtt version 3.1.1," OASIS Standard, 2014.
[8] A. Rajkumar, M. Fuchs, and C. Busch, "Edge computing for predictive maintenance in industry 4.0," IEEE Access, vol. 9, pp. 124 721–124 735, 2021.
[9] F. Pedregosa et al., "Scikit-learn: Machine learning in python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[10] R. Yan, R. X. Gao, and X. Chen, "Wavelets for fault diagnosis of rotary machines: A review with applications," Signal Processing, vol. 96, pp. 1–15, 2014.
[11] T. Januska, J. Sedlacek, and J. Krejci, "Grafana for industrial monitoring and visualization," in Proceedings of the IEEE International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 2019.
[12] P. Zhang, Y. Yan, and H. Gao, "Deep learning for smart manufacturing: Methods and applications," IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 1245–1256, 2021.
[13] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," NASA Ames Prognostics Data Repository, 2008.
[14] Z. Zhao, Y. Wang, R. Yan, and J. Mao, "Machine health monitoring using local feature-based gated recurrent unit networks," IEEE Transac- tions on Industrial Electronics, vol. 65, no. 2, pp. 1539–1548, 2018.
[15] "Iso 13374-1:2019, condition monitoring and diagnostics of machines – data processing, communication and presentation," ISO, Geneva, Switzerland, 2019.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY