Reports in Mechanical Engineering
Vol. 7, No. 2, 2026, pp. 111-131
ISSN: 2683-5894, DOI: https://doi.org/10.5281/zenodo.22139882

Adaptive Hybrid Anomaly Detection for High-Duty-Cycle Industrial Robots

Francesco Aggogeri[1], Nicola Pellegrini[1,*]
[1] Department of Mechanical and Industrial Engineering, Università degli Studi di Brescia, Brescia, Italy

Article Info

Article history:

Received June 17, 2026
Revised July 21, 2026
Accepted August 13, 2026

Keywords:

Industrial Robot,
Anomaly Detection,
Condition Monitoring,
Hybrid Learning,
Random Forest,
Principal Component Analysis,
Adaptation,
Predictive Maintenance

ABSTRACT

Machine-tending applications of industrial robots remain effective after decades of use; meanwhile, controller generation becomes obsolete and difficult to manage. This study presents a controller-independent hybrid anomaly-detection algorithm for condition monitoring of a six-axis degree-of-freedom robot using an external inertial measurement unit. The online decision path combines a supervised Random Forest, which recognizes reviewed disturbance signatures, with a deterministic PCA-based novelty score referenced to context-compatible nominal sections. A bounded context correction, decision-level fusion, and an H-of-K persistence rule reduce false alarms caused by payload, speed, and transient motion changes. Offline inspection of nonlinear neighborhoods is excluded from the deployed decision rule. The harmonized archive contains 69,120 multivariate observations and 4,422 windows partitioned by physical acquisition run to prevent temporal leakage. In the controlled evaluation, the PCA branch, Random Forest, fixed-fusion model, and adaptive hybrid achieved accuracies of 91.2%, 93.5%, 95.8%, and 97.1%, with F1-scores of 0.89, 0.91, 0.94, and 0.96, respectively. These experimental results indicate that complementary supervised and geometric evidence improves discrimination while retaining traceability. The framework is lightweight and traceable and is intended for retrofit condition monitoring; it does not estimate remaining useful life. Establishing prognostic capability would require run-to-failure or overhaul data, uncertainty-aware degradation modeling, and multi-robot validation.

Corresponding Author:

Nicola Pellegrini
Department of Mechanical and Industrial Engineering, Università degli Studi di Brescia, Brescia, Italy
Email: nicola.pellegrini@unibs.it

1. Introduction

Industrial robots are long-lived manufacturing assets whose mechanically useful service life may extend well beyond the commercial availability of the controller, software environment, diagnostic package, or communication interfaces supplied at installation. Compact manipulators employed in machine tending, assembly, inspection, laboratory automation, and material transfer perform highly repetitive operations over prolonged service periods. During this time, reducers, bearings, brakes, structural joints, cabling, and end-effectors are progressively exposed to mechanical loading, thermal cycling, lubrication degradation, and repeated acceleration–deceleration phases. Reviews of industrial-robot fault diagnosis and prognostics indicate that deterioration may initially emerge through subtle variations in vibration energy, impulsiveness, positioning dynamics, current demand, cross-axis coupling, or cycle-to-cycle repeatability rather than through an explicit controller alarm (Kumar et al., 2023; Mallioris et al., 2024; Sabry & Amirulddin, 2024). Consequently, maintenance strategies based exclusively on periodic inspections or alarm histories may identify degradation only after process accuracy, availability, or safety margins have already been affected. This challenge is relevant in high-duty-cycle robotic applications and in installations where the mechanical structure remains serviceable while the original monitoring infrastructure has become technologically obsolete. Proprietary servo variables may be unavailable, sampled at insufficient rates, or accessible only through manufacturer-specific options that are no longer economically justified. Moreover, the inertial, frictional, transmission, and drive parameters needed for accurate model-based observers or high-fidelity digital twins are rarely documented with sufficient detail throughout the robot's entire operating life. Replacing a mechanically functional manipulator may also require requalifying fixtures, tooling, safety functions, robot programs, peripheral equipment, and accumulated process knowledge. A practical retrofit monitoring solution should therefore be non-invasive, inexpensive, computationally efficient, and sufficiently independent of the controller generation. At the same time, it must distinguish actual degradation from legitimate variations caused by payload, velocity, trajectory, tool configuration, or environmental conditions (Kermenov et al., 2023; Polenghi et al., 2024). External sensing has emerged as an attractive response to these constraints. Acoustic, inertial, and vibration sensors can capture changes in the robot's dynamic response without modifying the validated motion program or requiring privileged access to the control architecture. Researchers have developed autoencoder-based systems for anomaly detection using internal sound signals, and have applied convolutional reconstruction networks to sliding-window vibration data (Yun et al., 2023; Zhong et al., 2022). Researchers have also investigated programmable motion-fault detection for collaborative robots, demonstrating the potential of externally observable dynamic signatures (Park et al., 2021). More recent context-aware and adaptive systems have shown that sensor information should not be interpreted independently of operating conditions, because a healthy manipulator may exhibit substantially different statistical behavior when the task, trajectory, payload, or interaction context changes (Ayankoso et al., 2024; Kayan et al., 2024). Research has consequently progressed from conventional feature-based monitoring toward transfer learning, multivariate time-series modeling, digital twins, and specialized deep architectures. Researchers have applied transfer learning to servomotor-bearing diagnosis, while proposing collaborative feature analysis for multichannel robot-joint data affected by missing observations (Kumar et al., 2024; Yang et al., 2024). Physics-informed hybrid autoencoders and digital-twin frameworks can introduce structural consistency when reliable virtual assets and controller information are available (Wang et al., 2024). Recent studies have additionally investigated vibration-based diagnosis of structural looseness, data-driven learning of robot-drive dynamics, hybrid deep-learning frameworks, evolutionary neural architectures, and combined convolutional–recurrent models for gearbox monitoring (Fedorova et al., 2024; Kim, 2025; Liu et al., 2025; Niu et al., 2025; Wu et al., 2025). These contributions demonstrate the increasing representational capability of data-driven methods. Still, they also reveal a practical trade-off among model complexity, training-data requirements, computational cost, interpretability, and transferability to aging industrial installations. Anomaly-detection strategies for robotic systems fall into supervised classification, one-class learning, reconstruction or forecasting models, density- and distance-based methods, and hybrid approaches. Minimally supervised variational autoencoders and anomaly detectors for autonomous robotic missions illustrate the usefulness of learning nominal behavior when fault labels are limited (Azzalini et al., 2021; Chirayil Nandakumar et al., 2024). More general reviews emphasize that anomaly-detection performance depends on the operational definition of an anomaly, the availability and reliability of labels, the contamination of nominal training data, and the validation protocol adopted by the researchers (Choi et al., 2021; Pang et al., 2021; Ruff et al., 2021). A supervised model can accurately recognize previously observed disturbances, but its performance may deteriorate when an emerging defect differs from the labeled training examples. Conversely, a one-class or distance-based detector can identify previously unseen behavior but may incorrectly interpret a new payload, trajectory, velocity profile, or tool configuration as mechanical degradation. Graph neural networks and transformer-based architectures have further expanded the ability to represent dependencies among multiple sensor channels and long temporal sequences (Deng & Hooi, 2021; Tuli et al., 2022; Xu et al., 2022). Nevertheless, evaluating time-series anomaly detectors remains methodologically challenging. Windows extracted from the same acquisition run are strongly correlated, and random window-level partitioning can place nearly identical samples in both the training and test sets. This form of temporal leakage may substantially inflate apparent generalization performance. In addition, point adjustment, class imbalance, event aggregation, threshold selection, and the definitions of precision and recall can alter the relative ranking of competing algorithms (Garg et al., 2022; Kim et al., 2022; Paparrizos et al., 2022; Saito & Rehmsmeier, 2015; Tatbul et al., 2018). Recent graph-convolutional variational models and probabilistic attention-based approaches confirm the potential of advanced representations. Still, they also reinforce the need for transparent splitting rules, stable thresholds, physically traceable alarms, and evaluation at the level of complete acquisition runs (Choi et al., 2024; Huang et al., 2025).

These limitations are critical when monitoring outputs support decisions on continued operation or service-life extension. False alarms create unnecessary inspections and production losses, whereas missed detections can let progressive deterioration go unnoticed. A credible algorithm must therefore preserve the provenance of every score, distinguish operating-context variation from health-state migration, and convert isolated deviations into persistent evidence. The unresolved methodological gap is not the absence of individual classifiers or dimensionality-reduction methods, but the absence of a lightweight and auditable decision architecture that jointly addresses four coupled requirements: recognition of reviewed disturbances, sensitivity to previously unseen deviations, bounded adaptation to legitimate context changes, and leakage-safe validation at the level of complete physical runs. The proposed method is designed specifically around this combination of requirements.

This study proposes a decision-level hybrid anomaly-detection framework for the life-extension monitoring of a high-duty-cycle robot using a low-cost inertial measurement unit. The supervised branch employs a Random Forest classifier because tree ensembles can represent nonlinear feature interactions, operate effectively with heterogeneous engineered descriptors, and support feature-level interpretation. The unsupervised branch uses a fixed principal-component representation to provide a deterministic and auditable global coordinate system. Context-normalized distances in this space are evaluated through an adaptive threshold, while an H-of-K persistence rule prevents isolated samples from directly generating maintenance alarms. The supervised fault probability and the geometric novelty evidence are subsequently fused at the decision level. Feature-level explanations can be generated using SHAP values to identify the variables that most strongly contribute to individual predictions. The role of nonlinear learning is deliberately separated from the deployed decision rule. In particular, t-distributed stochastic neighbor embedding is used only during offline analysis to investigate local neighborhoods, nonlinear separability, and possible transitions among health states (Breiman, 2001; Jolliffe & Cadima, 2016; Lundberg & Lee, 2017; Van der Maaten & Hinton, 2008). Consecutive latent observations are then organized as a progressive degradation trajectory to examine whether anomalous behavior is persistent and temporally ordered. Online decisions nevertheless remain governed by the Random Forest output, the robust distance calculated in the fixed PCA space, the adaptive threshold, and the temporal persistence logic. This separation avoids treating a stochastic visualization as a deterministic diagnostic instrument.

The novelty of the proposed framework lies in the deployment logic rather than in the individual algorithms. Relative to the hybrid robot-monitoring approaches, supervised disturbance recognition and context-compatible geometric novelty remain separately inspectable until decision-level fusion; adaptation is bounded to validated operating contexts; unseen contexts are assigned reduced confidence; and an H-of-K persistence rule converts correlated window scores into event-level evidence. The same architecture is validated by complete physical acquisition runs to prevent temporal leakage. This combination of auditable fusion, bounded context handling, temporal persistence, and run-grouped validation distinguishes the proposed controller-independent retrofit framework.

The paper is organized as follows. Section 2 describes the robotic platform, sensing architecture, acquisition protocol, preprocessing pipeline, feature-space construction, PCA and t-SNE formulations, the progressive degradation manifold, the hybrid decision algorithm, and the leakage-controlled validation strategy. Section 3 presents experimental results, including classification performance, anomaly-distance analysis, sensitivity assessment, and ablation studies. Section 4 discusses the algorithmic relevance of the findings, their maintenance implications, comparison with recent literature, methodological limitations, and the confirmatory experimental campaign. Finally, Section 5 summarizes the conclusions and identifies the conditions required for responsible industrial deployment.

2. Problem Formulation and Hybrid Framework

The method is formulated as a context-aware hybrid anomaly-detection algorithm for controller-independent monitoring of compact industrial robots. The supervised branch estimates resemblance to reviewed disturbance patterns, whereas the unsupervised branch quantifies departure from nominal behavior within a compatible operating-context stratum. The system fuses their outputs only after independent calibration, then filters the resulting score through bounded context correction, confidence checks, and temporal persistence. Algorithm 1 defines the complete training and online inference sequence. The framework provides supplementary maintenance evidence and does not replace certified robot safety functions, controller alarms, manufacturer limits, risk-assessment measures, or statutory inspections. Experimental claims are restricted to the six-axis FANUC LR Mate 200iC and to the controlled conditions described in Section 3.

Algorithm 1 (a): Hybrid Anomaly Detection for High-Duty-Cycle Industrial Robots

Item	Phase	Operation
Input	Data and configuration	Run-grouped training data ^{T} = {X, y, c, g}; streaming data ^{W}; preprocessing parameters; Random Forest parameters; PCA model order r; fusion weight α; context corrections; and H-of-K persistence parameters.
1	Validation and traceability	Verify timestamps, channel completeness, saturation, packet continuity, sensor metadata, run identifiers, program state, and maintenance boundaries. Reject data that violate the predefined quality rules.
2	Conditioning and segmentation	Synchronize the accepted channels; remove training-derived offsets; apply the fixed signal-conditioning chain; and create sliding windows that never cross physical run, program-reset, maintenance, or sensor-remounting boundaries.
3	Feature construction	Compute statistical, temporal, frequency-domain, cross-channel, and operating-context descriptors. Apply the scaling and feature-screening transformations estimated only from the training runs.

Algorithm 1 (b): Hybrid Anomaly Detection for High-Duty-Cycle Industrial Robots

Item	Phase	Operation
4	Supervised branch	Fit the Random Forest on labeled training windows. For each evaluated window, calculate the probability p_i^{Rf} of the abnormal class or, when justified, the probability vector of the validated multi-class fault taxonomy.
5	PCA representation	Fit PCA on the standardized training matrix; retain the validated latent dimension r; freeze the loading matrix; and project nominal training windows and incoming windows into the common coordinate system.
6	Nominal reference	For every validated operating-context stratum, estimate the nominal latent centroid and robust dispersion statistics. Mark unseen contexts as low-confidence rather than automatically updating the baseline.
7	Online branch scores	For each streaming window, evaluate p_i^{Rf}, the robust PCA-space distance d_i^U, and the normalized unsupervised score $\hat{s}_i^U$.
8	Decision-level fusion	Compute the fused score S_i and the context-adapted threshold $\tau(c_i)$ using parameters selected exclusively on validation runs.
9	Temporal decision	Apply the H-of-K persistence rule, optional hysteresis, and confidence checks to assign the final nominal, warning, known-fault, or atypical/novel state.
10	Evidence storage	Store the branch scores, fused score, context, dominant descriptors, data-quality flags, timestamps, run identifier, model version, and recommended maintenance action.
Output	Decision and evidence	Predicted robot state; fused anomaly score; persistent alarm state; confidence/context status; and an auditable evidence record linked to the corresponding robot cycle.

The monitoring architecture has been evaluated using the FANUC LR Mate 200iC shown in Figure 1. Operationally, the term legacy denotes an installed robot whose original controller, software environment, or diagnostic package does not provide modern high-frequency condition-monitoring services, or for which installing proprietary options is technically or economically unfeasible. The retrofit architecture avoids modification of the safety chain and acquires external inertial measurements through a local sensor node. The minimum sensor node comprises a 64 MHz ARM Cortex-M4 microcontroller, an inertial measurement unit, local timestamping, and either a Bluetooth Low Energy or wired communication interface. The IMU supplies three acceleration channels, three angular-rate channels, and three magnetic-field channels. It is mounted rigidly on the robot forearm and separated from loose process cables or flexible fixtures that could introduce non-structural vibration. The sensor frame is aligned through a static pose and a low-speed reference motion. Any subsequent remounting is treated as a new sensor configuration and therefore generates a new baseline version. At discrete acquisition time k, Equation (1) represents the minimum multivariate measurement.

$$\mathrm{x}_k = \left[a_{x,k}, a_{y,k}, a_{z,k}, \omega_{x,k}, \omega_{y,k}, \omega_{z,k}, m_{x,k}, m_{y,k}, m_{z,k}\right]^T \in \mathrm{R}^9 \tag{1}$$

In Equation (1), a_x, a_y, and a_z denote the acceleration components; ω_x, ω_y, and ω_z denote the angular-rate components; m_x, m_Y, and m_z denote the magnetic-field components; k is the ordered sample index; the superscript T indicates vector transposition; and x_k is the nine-dimensional sensor vector. Optional controller or cell variables, including speed, joint position, servo current, motion segment, payload schedule, alarm status, and cycle counter, may be appended when they are available at a suitable sampling rate. Their absence does not alter the minimum architecture. Every continuous physical acquisition is assigned a unique group identifier. The corresponding run manifest records the date, robot program, tool, payload, speed override, maintenance state, sensor mounting version, ambient condition, and disturbance protocol. The group identifier is preserved through preprocessing, feature extraction, model training, validation, and inference. This requirement prevents windows extracted from the same physical experiment from entering different data partitions. It provides an auditable link between every anomaly score and the robot cycle from which it originated.

The second step converts the raw stream into quality-controlled, time-consistent windows. Sensor packets are ordered by local timestamp, duplicate records are removed, and the effective sampling interval is compared with the nominal period. The quality-control layer identifies missing timestamps, saturation, repeated packets, non-monotonic time sequences, and physically implausible discontinuities. Short gaps may be interpolated only when their duration remains below a predefined fraction of the analysis window. Longer gaps, excessive packet loss, or sustained saturation invalidate the affected window. This rule prevents communication artifacts from being learned as mechanical faults. When controller or cell variables are available, they are aligned to the inertial timeline by nearest-neighbor association or linear interpolation, provided that the temporal mismatch remains within a predeclared tolerance. Context variables that change discretely, such as program segment or payload schedule, are aligned using their event timestamps rather than numerical interpolation. We estimate or select synchronisation parameters before the final test evaluation and store them with the preprocessing version. Signal conditioning is deliberately conservative. We remove a static offset estimated from accepted training runs from the acceleration and angular-rate channels. A fixed low-pass filter attenuates electronic noise outside the mechanical bandwidth relevant to the monitored motion. We apply the same coefficients and implementation order to training, validation, test, and streaming data. Abrupt impulses are retained when they pass saturation and consistency checks because they may contain diagnostically meaningful evidence. Magnetic channels are primarily used as orientation and local electromagnetic-context descriptors, whereas resultant acceleration and angular-rate magnitudes reduce sensitivity to small changes in sensor-axis alignment. The accepted sequence is segmented by a sliding window of L samples and stride h, as defined by Equation (2).

$$\mathcal{W}_i = \{\mathrm{x}_k\}_{k=s_i}^{s_i+L-1}, \qquad s_i = 1 + (i-1)h \tag{2}$$

In Equation (2), W_i is the $\mathrm{i}-\mathrm{th}$ multivariate analysis window; L is the window length in samples; h is the stride between consecutive windows; s_i is the first sample index of window i; and x_k is the synchronized sensor vector defined in Equation (1). The window length is reported in both samples and physical time because a portable implementation may operate at a different sampling rate. Windows never cross boundaries, program resets, maintenance interventions, sensor-remounting events, or discontinuous changes in the acquisition configuration. Window selection governs the latency–stability compromise. Short windows reduce response time but yield less stable higher-order and frequency-domain descriptors. Long windows improve statistical robustness but may mix successive robot motion phases. The selected pair (L,h) is therefore validated against event-level detection delay, false-alarm rate, and stability under grouped resampling rather than being chosen from test-set visualization. Overlapping windows retain the same run identifier and are treated as correlated observations during statistical validation.

The feature space is the common interface between the heterogeneous industrial signal stream and the supervised and unsupervised branches of Algorithm 1. Its role is not merely to compress the raw measurements, but to convert each window into a dimensionally consistent, physically interpretable, and leakage-free numerical representation. The feature map includes statistical, temporal, frequency-domain, cross-channel, and explicit operating-context descriptors. For every accepted channel, the statistical family includes the mean, standard deviation, root-mean-square value, median absolute deviation, interquartile range, peak-to-peak range, skewness, kurtosis, and crest factor. Temporal descriptors include linear trend, first-difference energy, zero- or mean-crossing rate, lag-one autocorrelation, peak count, and impulse persistence. Frequency descriptors include dominant frequency, spectral centroid, spectral entropy, and validated band energies. Cross-channel descriptors include axis correlations, acceleration and angular-rate resultants, directional imbalance, and ratios between high- and low-frequency energy. Normalised payload, normalised speed, program segment, and transition status are included as context variables rather than inferred indirectly from vibration amplitude. The candidate descriptor vector for window i is defined by Equation (3).

$$\mathrm{z}_i = \phi(W_i, c_i) = \left[f_{i,1}, \dots, f_{i,p_f}, c_i^T\right]^T \in \mathrm{R}^{p_0} \tag{3}$$

In Equation (3), $\varphi(\cdot)$ is the deterministic feature-extraction operator; W_i is the window defined in Equation (2); c_i is the operating-context vector; $f_{i,j}$ is the $\mathrm{j}-\mathrm{th}$ engineered signal descriptor; p_f is the number of signal-derived candidate descriptors; p_0 is the total number of candidate predictors after the context variables are appended; and z_i is the unscaled feature vector. Each descriptor is stored together with its physical unit, signal source, transformation definition, and software version. Dimensional features remain traceable in engineering units at the extraction stage, while standardised moments, ratios, and normalised context variables are explicitly identified as dimensionless. Because the candidate descriptors have different units and numerical ranges, standardisation is performed using statistics calculated exclusively from the training runs, as expressed in Equation (4).

$$\tilde{z}_{i,j} = \frac{z_{i,j} - \mu_j^{\mathrm{tr}}}{\sigma_j^{\mathrm{tr}} + \varepsilon}, \qquad j = 1, \dots, p_0 \tag{4}$$

In Equation (4), $z_{i,j}$ is the unscaled value of descriptor j for window $\mu_i^{tr}\, and\; \sigma_i^{tr}$ are the training-set mean and standard deviation of descriptor j; ε is a small positive constant that prevents division by zero; and $\tilde{Z}_{i,j}$ is the standardized value. The transformation is frozen after training and applied unchanged to validation, test, and streaming windows. Re-estimating u_i^{tr} or j^{tr} from evaluation data would introduce leakage and invalidate temporal comparisons. The retained standardized vectors are assembled row-wise into the feature matrix in Equation (5).

$$\tilde{Z} = [\tilde{z}_1^{\mathsf{T}}; \tilde{z}_2^{\mathsf{T}}; \dots; \tilde{z}_N^{\mathsf{T}}] \in \mathrm{R}^{N\times p} \tag{5}$$

In Equation (5), $\tilde{Z}$ is the standardized feature matrix; N is the number of valid windows; p is the number of descriptors retained after quality control and screening; and z̃_i is the standardized vector associated with window i. Labels, run identifiers, timestamps, data-quality flags, and maintenance metadata are stored in parallel structures and are not silently encoded as ordinary numerical predictors. Screening is conducted in two stages. First, near-constant variables are removed using the training-set variance criterion in Equation (6).

$$s_j^2 = \frac{1}{N_{tr}-1}\sum_{i\in T}\left(\tilde{z}_{i,j} - \bar{z}_j\right)^2, \quad J_v = \{j: s_j^2 \geq \eta_v\} \tag{6}$$

In Equation (6), s_j^2 is the sample variance of standardized descriptor j over the training index set T; N_{tr} is the number of training windows; $\bar{z}_j$ is the training mean of the standardized descriptor; η_v is the predefined minimum-variance threshold; and J_v is the set of descriptors that pass the variance filter. Duplicate or almost perfectly correlated transformations of the same physical signal are subsequently reviewed. Redundant numerical encodings are removed, whereas physically distinct variables are not discarded solely because they are correlated under a particular operating regime. Second, supervised feature ranking is performed only after fitting the Random Forest on grouped training runs. Impurity-based importance is supplemented by grouped permutation importance to reduce known biases toward continuous or high-cardinality predictors. A descriptor is considered robust when its relevance remains stable across grouped resampling and across the principal payload, speed, and motion regimes. Feature importance is interpreted as predictive evidence, not proof of a unique causal fault mechanism. The supervised task is formulated as binary health-state classification, with $y = 0$ for accepted nominal behavior and $y = 1$ for a reviewed abnormal condition. This formulation prioritizes reliable anomaly recognition in industrial settings where the number of available fault examples is limited. When sufficient, consistently labeled evidence is available, the same branch may be extended to a validated multi-class taxonomy covering commanded velocity deviation, external mechanical perturbation, payload alteration, and structural vibration disturbance. Binary performance remains the principal evaluation target because fault-type labels may be incomplete or ambiguous during early degradation. A Random Forest is selected. Each tree recursively partitions the training data. For a candidate node ν, Equation (7) defines the Gini impurity.

$$G(\mathcal{D}_v) = 1 - \sum_{k=0}^{K}\pi_{v,k}^2 \tag{7}$$

In Equation (7), D_v is the set of training samples reaching node v; $\pi_{v,k}$ is the proportion of samples belonging to class k at that node; $K + 1$ is the number of classes; and $G(D_v)$ is the Gini impurity. A candidate split is preferred when it produces the largest weighted reduction in impurity. The number of trees B, maximum tree depth, minimum leaf size, feature-subsampling rule, and class-weight strategy are selected on validation runs only. For binary inference, the ensemble abnormal-class probability is calculated according to Equation (8).

$$p_i^{\mathrm{RF}} = \frac{1}{B}\sum_{b=1}^{B} p_{i,b}\,(y = 1 \mid \tilde{z}_i) \tag{8}$$

In Equation (8), B is the number of trees; $p_{i,b(}y = 1|\tilde{z}_i{)}$ does tree estimate the abnormal-class probability b for window i; and p_i^{RF} is the ensemble probability returned by the supervised branch. Probability calibration is evaluated on validation runs when the fusion rule uses p_i^{RF} as a quantitative score rather than only as a class label. The Random Forest is trained on the complete retained feature vector, not only on the first principal components, so that local thresholds and non-monotonic feature interactions are preserved. The supervised branch is designed to recognize known disturbance signatures. A high value of p_i^{RF} indicates similarity to labeled abnormal windows but does not, by itself, establish a component-level diagnosis. Conversely, a low probability does not guarantee nominal operation when the current condition differs from the labeled archive. This limitation motivates the independent unsupervised branch described in the following subsections.

Principal component analysis (PCA) provides the fixed latent representation used by the online unsupervised branch. It is fitted once on the standardized training matrix and then frozen for validation, testing, and streaming inference. Equation defines the training covariance matrix (9).

$$C = \frac{1}{N_{\text{tr}} - 1} \tilde{Z}_{\text{tr}}^{\mathsf{T}} \tilde{Z}_{\text{tr}} \in \mathbb{R}^{p \times p} \tag{9}$$

In Equation (9), C is the $p \times p$ training covariance matrix, $\tilde{Z}_{\text{tr}}$ is the standardized training matrix, and N_{tr} is the number of training windows. Principal directions are obtained from Equation (10).

$$C v_k = \lambda_k v_k, \qquad v_k^{\mathsf{T}} v_\ell = \delta_{k\ell}, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0 \tag{10}$$

In Equation (10), v_{k} and λ_k are the $k - th$ eigenvector and eigenvalue, respectively, with components ordered by decreasing eigenvalue. Retained variance is evaluated using Equation (11).

$$\text{EVR}_k = \frac{\lambda_k}{\sum_{j=1}^{p} \lambda_j}, \qquad \text{CEV}(r) = \sum_{k=1}^{r} \text{EVR}_k \tag{11}$$

In Equation (11), EVR_k is the explained-variance ratio and $CEV(r)$ do the first r components retain the cumulative variance. We select dimension r based on variance retention, grouped-resampling stability, validation performance, and engineering interpretability. We use the three-dimensional form for visualisation when appropriate, whereas the deployed distance model retains the validated dimension. Projection is performed using Equation (12).

$$V_r = [v_1, \ldots, v_r] \in \mathbb{R}^{p \times r}, \qquad u_i = \tilde{z}_i^{\mathsf{T}} V_r \in \mathbb{R}^r \tag{12}$$

In Equation (12), V_r contains the first r loading vectors and u_i is the fixed PCA score vector for window i. Apply the same transformation to all data partitions and streaming windows. Nominal reference statistics are estimated within validated operating-context strata using Equation (13).

$$\tilde{\mu}_{q,k} = \text{median}_{i \in N_q} \{u_{i,k}\}, \quad s_{q,k} = 1.4826 \text{median}_{i \in N_q} |u_{i,k} - \tilde{\mu}_{q,k}| + \varepsilon \tag{13}$$

In Equation (13), N_q is the nominal training set for context q, $\tilde{\mu}_{q,k}$ is the median PCA coordinate, and $s_{q,k}$ is the MAD-based robust scale. Unseen or insufficiently represented contexts are flagged as low-confidence rather than incorporated automatically into the baseline. The resulting context-compatible distance is defined in Equation (14).

$$d_i^U = \left[\sum_{k=1}^{r} \left(\frac{u_{i,k} - \tilde{\mu}_{q(i),k}}{s_{q(i),k}} \right)^2 \right]^{1/2} \tag{14}$$

In Equation (14), d_i^U is the context-compatible unsupervised distance; $q(i)$ is the validated context stratum assigned to window i; $u_{i,k}$ is the $k - th$ PCA coordinate; $\tilde{\mu}_{q(i),k}$ and $s_{q(i),k}$ are the robust nominal center and scale; and r is the retained PCA dimension. The standardized Euclidean form prevents a high-variance component from dominating the distance solely because of its numerical scale. A large distance indicates departure from the accepted nominal manifold but does not by itself determine whether the cause is mechanical degradation, sensor relocation, an unrecorded process change, or a novel fault. For every streaming window that passes the data-quality gate, Algorithm 1 computes two independent quantities: the Random Forest abnormal-class probability p_i^{RF} and the PCA-space distance d_i^U. The two outputs have different distributions and cannot be combined directly. The unsupervised distance is therefore mapped to the unit interval using the calibrated logistic transformation in Equation (15).

$$\hat{s}_i^U = \frac{1}{1 + \exp[-\beta(d_i^U - \gamma)]} \tag{15}$$

In Equation (15), $\hat{s}_i^U$ is the normalized unsupervised score; d_i^U is the robust PCA-space distance from Equation (14); γ is the validation-derived distance at the logistic midpoint; and β controls the slope of the mapping. We fit β and γ using training and validation information only. The transformation preserves distance ordering while expressing it on a scale compatible with Random Forest probability. We store the branch scores separately, even when they agree. This is necessary because disagreement contains diagnostic information. A high p_i^{RF} combined with a moderate $\hat{s}_i^U$ suggests that the window resembles a known disturbance located near the global nominal manifold. A high $\hat{s}_i^U$ combined with a low p_i^{RF} suggests an atypical pattern that is absent from the labeled archive, an unvalidated operating context, or a sensing problem. Such cases are routed to review rather than being forced into a known fault class.

The supervised and unsupervised outputs are integrated at decision level. This design preserves the original evidence produced by each branch, allows the fusion weight to be audited, and avoids training an additional opaque meta-model on a limited fault archive. Equation (16) defines the fused score.

$$S_i = \alpha p_i^{\mathrm{RF}} + (1-\alpha)\hat{s}_i^U, \qquad 0 \le \alpha \le 1 \tag{16}$$

In Equation (16), S_i is the fused anomaly score for window i; p_i^{RF} is the supervised abnormal-class probability; $\hat{s}_i^U$ is the normalized unsupervised score; and α is the fusion weight. The limiting case $\alpha = 1$ uses only supervised evidence, whereas α = 0 uses only geometric departure. We determine the selected value from grouped validation runs by jointly considering recall, false-alarm rate, probability calibration, and stability under payload and speed changes. We do not select it from the final test results. A fixed global threshold can generate false alarms when the operating context changes. Algorithm 1 therefore uses the bounded context-adapted threshold in Equation (17).

$$\tau(\mathrm{c}_i) = \mathrm{clip}\left[\tau_0 + \delta_p r_{p,i} + \delta_v r_{v,i} + \delta_q r_{q,i}, \tau_{\min}, \tau_{\max}\right] \tag{17}$$

In Equation (17), $\tau(\mathrm{c}_i)$ is the decision threshold for context vector c_i; τ_0 is the validated base threshold; $r_{p,i}$ is the normalized payload-ratio deviation; $r_{v,i}$ is the normalized speed-ratio deviation; $r_{q,i}$ represents a validated task or sequence-transition condition; δ_p, δ_v, and δ_q are bounded correction coefficients; and $\tau_{\min}$ and $\tau_{\max}$ are safety bounds that prevent uncontrolled threshold drift. The operator clip (·) limits the threshold to the declared interval. Additional context terms may be introduced only when they are measured and experimentally validated; unmeasured temperature or wear variables are not inserted as hypothetical corrections. Adaptive thresholding compensates for predictable context-induced distribution shifts but must not conceal systematic degradation. The correction terms are therefore limited, signed, and evaluated independently within nominal validation runs. If a context lies outside the validated envelope, the system reports low confidence and applies a conservative policy rather than extrapolating an unverified threshold.

Industrial robot signals contain short transients caused by programmed acceleration, contact events, trajectory transitions, and communication jitter. A single score exceedance is therefore recorded as an event but does not automatically produce a persistent maintenance alarm. The final binary persistence decision is defined by the H-of-K rule in Equation (18).

$$a_i = \mathbb{I}\left[\sum_{j=i-K_w+1}^{i} \mathbb{I}\left(S_j > \tau(\mathbf{c}_j)\right) \ge H\right] \tag{18}$$

In Equation (18), a_i is the persistent alarm indicator; I[·] is the indicator function; K_w is the number of recent windows evaluated by the persistence rule; H is the minimum number of threshold exceedances required within those K_w windows; S_j is the fused score; and $\tau(\mathrm{c}_j)$ is the context-adapted threshold. The values H and K_w are selected according to the acceptable detection-delay and false-alarm trade-off. Hysteresis may be applied during recovery so that one favorable window does not immediately reset a persistent alarm. The final state combines persistence, branch agreement, context confidence, and the optional multi-class output. A persistent score increase with agreement between the Random Forest and PCA-distance branches is assigned to a known abnormal state when the supervised probability vector is sufficiently confident. A persistent geometric departure with low supervised confidence is assigned to an atypical or potentially novel condition. A non-persistent excursion remains available for engineering review but is not reported as an established health-state transition.

The organization of the monitoring workflow, including condition-data acquisition, processing, diagnostic assessment, information management, and traceability, follows the general condition-monitoring principles defined in ISO 17359 and the functional data-processing framework described in ISO 13374-1 (ISO, 2003, 2018). The evaluated window generates an evidence record containing the timestamp, run identifier, window boundaries, data-quality flags, operating context, retained descriptors, Random Forest probability, PCA coordinates, robust distance, normalized unsupervised score, fused score, adaptive threshold, persistence status, and model version. The record also stores the dominant global or local explanatory descriptors and the preprocessing configuration used to obtain them. This structure supports comparison before and after maintenance and prevents an alarm from being detached from its signal-level origin. From an engineering perspective, descriptors such as root-mean-square acceleration, kurtosis, crest factor, high-frequency energy, temporal trend, and axis imbalance map back to the corresponding sensor channels and robot cycle. The framework does not claim that one feature uniquely identifies a damaged reducer, bearing, cable, brake, or interface. Instead, it provides a ranked, context-aware body of evidence to guide physical inspection, confirmatory measurement, and conservative maintenance decisions. Baseline updates are controlled operations. The system admits candidate nominal data only after data-quality checks, context verification, and maintenance approval. Automatic absorption of recently observed windows could normalize a developing fault and is therefore prohibited. Each accepted update creates a new model identifier and preserves the previous preprocessing parameters, PCA loadings, nominal statistics, Random Forest configuration, fusion coefficients, and thresholds for auditability.

t-SNE is used only as an offline diagnostic visualization of local neighborhoods that may not be visible in the global PCA projection. It never contributes to the online anomaly score, adaptive threshold, persistence state, or class decision. High-dimensional neighborhood probabilities are defined by Equation (19).

$$p_{j|i} = \frac{\exp\left(-\| \tilde{z}_i - \tilde{z}_j \|_2^2 / 2\sigma_i^2\right)}{\sum_{k\neq i} \exp\left(-\| \tilde{z}_i - \tilde{z}_k \|_2^2 / 2\sigma_i^2\right)}, \qquad p_{i|i} = 0 \tag{19}$$

In Equation (19), $p_{j|i}$ is the conditional neighborhood probability computed from standardized feature vectors and the local bandwidth σ_i; self-similarity is excluded. The conditional similarities are symmetrized using Equation (20).

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \qquad p_{ii} = 0 \tag{20}$$

In Equation (20), p_{ij} is the symmetric high-dimensional similarity. The corresponding low-dimensional Student-kernel similarity q_{ij} is defined by Equation (21).

$$q_{ij} = \frac{\left(1 + \| \mathbf{y}_i - \mathbf{y}_j \|_2^2\right)^{-1}}{\sum_{k\neq \ell} \left(1 + \| \mathbf{y}_k - \mathbf{y}_\ell \|_2^2\right)^{-1}} \tag{21}$$

In Equation (21), y_i and y_j are the low-dimensional coordinates and q_{ij} is their normalized similarity. The embedding minimizes the Kullback–Leibler divergence in Equation (22).

$$\mathcal{L}_{\mathrm{KL}} = D_{\mathrm{KL}}(P \| Q) = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) \tag{22}$$

In Equation (22), $\mathcal{L}_{\mathrm{KL}}$ compares the high- and low-dimensional similarity distributions P and Q defined in Equations (20) and (21); the objective prioritises preserving local neighbourhoods over global metric distances. Perplexity is defined by Equation (23).

$$\mathrm{Perp}(P_i) = 2^{H(P_i)}, \qquad H(P_i) = -\sum_{j \neq i} p_{j|i} \log_2 p_{j|i} \tag{23}$$

In Equation (23), $\mathrm{Perp}(P_i)$ is the effective neighborhood size and $H(P_i)$ is the entropy of the conditional similarity distribution defined by Equation (19). Perplexity is an algorithmic setting, not a physical degradation indicator. The reported embedding uses PCA initialization and fixed implementation settings; repeated seeds and grouped resampling are used to verify that interpreted neighborhoods are stable. Accordingly, t-SNE is retained solely for supporting visualization, while all deployable decisions remain in the Random-Forest/PCA fusion path.

Progressive degradation is analyzed as an ordered trajectory rather than as an unordered cloud of windows. Reducer wear, bearing deterioration, lubrication changes, structural looseness, cabling effects, and altered drive behavior can produce gradual changes in vibration amplitude, impulsiveness, spectral redistribution, settling behavior, and cross-axis coupling. Temporal order is therefore retained so that sustained migration can be distinguished from repeated switching among legitimate operating modes. For a continuous acquisition run r, the latent trajectory is represented by Equation (24).

$$\mathcal{T}^{(r)} = \left\{ \mathrm{y}_t^{(r)}, t_t^{(r)}, \mathrm{c}_t^{(r)} \right\}_{t=1}^{T_r} \tag{24}$$

In Equation (24), $\mathcal{T}^{(r)}$ is the ordered trajectory for run r; $t_t^{(r)}$ is the latent coordinate at temporal index t, obtained from the fixed PCA space for deployment or from the diagnostic t-SNE map for offline visualization; $t_t^{(r)}$ is the corresponding timestamp; $\mathrm{c}_t^{(r)}$ is the operating context; and T_r is the number of windows in the run. The trajectory is reset at program discontinuities, maintenance interventions, and sensor-remounting events. The elementary displacement is defined by Equation (25).

$$\Delta \mathrm{y}_t = \mathrm{y}_t - \mathrm{y}_{t-1}, \qquad d_t = \| \Delta \mathrm{y}_t \|_2 \tag{25}$$

In Equation (25), $\Delta \mathrm{y}_t$ is the latent displacement between consecutive windows; d_t is its Euclidean magnitude; and $\|\cdot\|_2$ is the Euclidean norm. One large displacement may correspond to an impact, communication defect, trajectory transition, or fault onset. Progressive degradation is supported more strongly by a sequence of directionally coherent displacements and increasing context-compatible distance. The latent degradation velocity is calculated by Equation (26).

$$v_t = \frac{d_t}{t_t - t_{t-1}} \tag{26}$$

In Equation (26), v_t is the latent displacement rate; d_t is defined in Equation (25); and $t_t - t_{t-1}$ is the elapsed physical time between window centers. This quantity is an observational change indicator, not a crack-growth rate or a direct estimate of remaining useful life. The cumulative trajectory length is given by Equation (27).

$$L_t = \sum_{k=2}^{t} d_k \tag{27}$$

In Equation (27), L_t is the accumulated latent path length up to window t and d_k is the consecutive displacement magnitude. Because healthy alternation among programmed phases can also increase L_t, cumulative length is interpreted jointly with net departure from the context-compatible nominal reference, directional consistency, and the fraction of windows entering warning or degraded neighborhoods. A normalized degradation index is introduced in Equation (28).

$$D_t = \mathrm{clip}\left[\frac{\tilde{d}_t - d_{\mathrm{ref}}(\mathrm{c}_t)}{d_{\mathrm{warn}}(\mathrm{c}_t) - d_{\mathrm{ref}}(\mathrm{c}_t)}, 0{,}1\right] \tag{28}$$

In Equation (28), D_t is the bounded degradation index; $\tilde{d}_t$ is a smoothed or robustly aggregated distance measure; $d_{\mathrm{ref}}(\mathrm{c}_t)$ is the nominal reference level for the current context; $d_{\mathrm{warn}}(\mathrm{c}_t)$ is the validated warning level; and clip$(\cdot)$ constrains the index to [0, 1]. Values near zero indicate behavior consistent with the accepted reference, intermediate values identify a transition requiring increased observation, and values approaching one indicate persistent departure. The index does not quantify residual fatigue life. Temporal smoothing of the degradation index is defined by Equation (29).

$$\widetilde{D}_t = \lambda_D D_t + (1 - \lambda_D)\widetilde{D}_{t-1}, \quad 0 < \lambda_D \leq 1 \tag{29}$$

In Equation (29), $\widetilde{D}_t$ is the exponentially smoothed index; D_t is the current unsmoothed index; and λ_D is the smoothing coefficient. The coefficient is selected to suppress isolated oscillations without delaying a sustained transition beyond the maintenance-relevant time scale. We retain both raw and smoothed trajectories so filtering cannot conceal a sudden event. The progressive analysis complements Algorithm 1. Agreement among increasing PCA distance, coherent trajectory migration, and increasing Random Forest probability provides strong evidence of a developing known condition. Increasing geometric departure with low supervised probability indicates a potentially novel state, an unrecorded context, or a sensor issue and therefore triggers review rather than automatic model adaptation. In this manner, the manifold supports early inspection, temporary operating review, and service-life decisions without being misrepresented as an autonomous remaining-useful-life estimator. The critical validation requirement is group separation. Windows extracted from the same continuous physical run are strongly correlated; random window-level splitting would place nearly identical temporal segments in the training and evaluation sets and overestimate generalisation.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{30}$$

In Equation (30), TP and TN are the numbers of true-positive and true-negative windows, whereas FP and FN are the numbers of false-positive and false-negative windows. Accuracy is the overall fraction of correctly classified windows.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{31}$$

In Equation (31), Precision is the fraction of generated positive classifications that correspond to labeled abnormal windows. It quantifies the credibility of alarms and is sensitive to false positives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{32}$$

In Equation (32), Recall is the fraction of labeled abnormal windows detected by the model. It quantifies sensitivity and decreases with missed abnormal windows.

$$F_1 = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{33}$$

In Equation (33), F_1 is the harmonic mean of Precision and Recall and therefore summarizes their balance. Precision, Recall, and F_1 are reported with the positive class explicitly defined.

$$\text{FAR} = \frac{N_{\text{false alarm events}}}{T_{\text{nominal operation}}} \tag{34}$$

In Equation (34), FAR is the event-level false-alarm rate; $N_{\text{false alarm events}}$ is the number of distinct false alarm episodes after temporal persistence is applied; and $T_{\text{nominal operation}}$ is the duration of accepted nominal operation, reported in hours. Counting events rather than overlapping windows provides a more meaningful estimate of maintenance task.

$$\Delta t_{\text{det}} = t_{\text{persistent alarm}} - t_{\text{event onset}} \tag{35}$$

In Equation (35), Δt_{det} is the event-detection delay; $t_{\text{persistent alarm}}$ is the time at which the H-of-K rule first produces a persistent alarm; and $t_{\text{event onset}}$ is the reviewed onset time of the controlled disturbance. The area under the precision–recall curve is additionally reported for threshold-independent comparison under class imbalance. Confidence intervals are obtained by grouped resampling at run level rather than by treating overlapping windows as independent samples.

The implementation can be realized with standard numerical and machine-learning libraries and exported to a local industrial runtime. For a window containing L samples, p retained descriptors, B trees, maximum tree depth d, and r principal components, feature extraction is approximately linear in the number of processed samples and descriptors; PCA projection requires O(pr) operations; Random Forest inference requires approximately O(Bd) node evaluations; and fusion plus persistence are O (1) per window when the recent alarm count is maintained incrementally. Training costs are incurred offline. The deployed package requires the channel map, quality rules, filter coefficients, window parameters, feature definitions, scaling constants, retained-feature list, Random Forest parameters, PCA loadings, context strata, robust nominal statistics, fusion parameters, random seeds, software versions, and model identifier. On the 64 MHz ARM Cortex-M4 target platform, the deployment characterization yielded a median end-to-end latency of 5–20 ms/window, a 95th-percentile latency of 10–30 ms/window, and a worst-case latency of 20–50 ms/window. The serialized Random Forest plus PCA package occupied 40–100 kB, with peak SRAM usage of 0.04–0.06 MB. The sustainable processing rate was at least 20 windows/s and typically 30–50 windows/s when BLE communication did not introduce blocking. t-SNE is excluded from the online timing allocation.

3. Experimental Validation and Results

The experimental validation was conducted on the FANUC LR Mate 200iC compact six-axis industrial robot introduced in Figure 1 and characterized in Table 1. The monitoring chain was implemented as a controller-independent retrofit and did not modify the robot safety circuit, motion program, or controller alarm logic. Accordingly, the proposed algorithm provides supplementary condition evidence for maintenance engineering and does not replace certified safety functions, statutory inspections, or manufacturer operating limits. The sensing node comprised a 64 MHz ARM Cortex-M4 microcontroller, a rigidly mounted inertial measurement unit, local timestamping, and either Bluetooth Low Energy or wired communication. Three acceleration channels, three angular-rate channels, and three magnetic-field channels were acquired from the forearm location. The sensor was positioned away from loose process cables and flexible fixtures to limit non-structural vibration. The campaign comprised 24 h of acquisition, including 16 h of accepted nominal operation and 8 h of controlled perturbation conditions. The archived dataset summarized approximately 69,120 multivariate observations and produced 4,422 valid analysis windows after synchronization, quality control, and segmentation. Each task was repeated 30 times while varying normalized speed, payload, and sequence transitions. Independent physical acquisitions were used as the unit of partitioning. The experimental archive comprised 120 independent acquisition runs. Of these, 84 runs (70%) were assigned to training, 18 runs (15%) to validation, and 18 runs (15%) to the final held-out test set. All windows extracted from a given physical run were retained exclusively within the corresponding partition, preventing temporal leakage between training, validation, and test data. The partition was stratified as far as possible with respect to disturbance family and severity. Four disturbance families were represented: commanded velocity deviation, external mechanical perturbation, payload alteration, and structural vibration. Each family was organized into low, moderate, and severe levels so that the supervised branch could learn reviewed disturbance signatures while the unsupervised branch retained sensitivity to departures not represented by one discrete class label. Terminology is used consistently as follows: nominal denotes accepted baseline behavior; incipient degradation denotes a persistent transition near the nominal boundary; degraded/abnormal denotes sustained departure from the validated envelope; and known fault is reserved only for a supervised label linked to a reviewed disturbance class. Raw data were ordered by timestamp and screened for duplicate records, missing samples, non-monotonic time sequences, saturation, and physically implausible discontinuities. Short gaps were interpolated only when their duration remained below the predefined fraction of the analysis window; longer gaps invalidated the affected window. Offset removal, filtering coefficients, synchronization tolerances, scaling statistics, and feature-screening decisions were estimated from the training runs and then frozen for validation, testing, and online inference.

Figure 1: FANUC LR Mate 200iC used as the Target Compact-Industrial-Robot Platform. The Monitoring Method is Implemented as a Controller-Independent Retrofit; Photograph Supplied by the Authors.

Table 1: Target-Platform Characteristics Used to Define the Monitoring Envelope.

Characteristic	Value Used in the Manuscript	Monitoring Implication
Robot Family	FANUC LR Mate 200iC	Compact six-axis articulated industrial robot.
Nominal Wrist Payload	5 kg	Payload ratio is retained as an explicit operating-context variable.
Horizontal Reach	Approximately 704 mm	Configuration-dependent dynamics affect the response measured by an arm-mounted IMU.
Repeatability	±0.02 mm	Small dynamic changes may become observable in vibration features before a controller alarm is generated.
Mechanical Mass	Approximately 27 kg	Compatible with bench-scale and machine-integrated experimental cells.
Controller Generation	R-30iA Mate family	Restricted access to high-rate proprietary variables motivates external sensing.
Primary Retrofit Sensor	Forearm-mounted nine-degree-of-freedom IMU	Provides tri-axial acceleration, angular rate, and magnetic-field measurements.

The principal leakage-control rule was separation at the physical-run level. All windows originating from one continuous acquisition remained in a single partition, including overlapping windows generated with the selected stride. The experimental archive comprised 120 independent physical acquisition runs. Seventy percent of the available runs were assigned to model development, resulting in 84 training runs, while the remaining runs were equally divided into 18 validation runs and 18 final test runs under the same group-level constraint. Where labels were available, the partitioning was stratified by disturbance family and severity. Hyperparameters, the PCA dimension, nominal context strata, fusion weight, adaptive-threshold coefficients, and H-of-K persistence settings were selected using training and validation runs only. The final test partition was reserved for one-time reporting. Performance was evaluated at both window and event levels using the definitions in Equations (30) – (35). Accuracy and F1-score were the principal classification measures, while precision, recall, PR-AUC, false alarms per nominal operating hour, and event-detection delay characterized alarm performance. Grouped variability was assessed at the level of complete physical acquisition runs rather than correlated overlapping windows. Unless otherwise stated, performance values are reported as mean ± standard deviation across the 18 independent physical acquisition runs included in the held-out test set. The PCA-distance branch achieved an accuracy of 91.2 ± 1.9% and an F1-score of 0.89 ± 0.02; the Random Forest branch achieved 93.5 ± 1.5% and 0.91 ± 0.02; the fixed-threshold hybrid achieved 95.8 ± 1.1% and 0.94 ± 0.01; and the complete adaptive hybrid achieved 97.1 ± 0.8% and 0.96 ± 0.01, respectively. The progressively lower variability observed from the individual branches to the adaptive hybrid is consistent with the stabilizing effect of complementary supervised and geometric evidence combined with bounded context adaptation.

Figures 2-5 summarize complementary views of the same standardized feature space. Figures 2 and 3 use the frozen PCA transformation that supports the deployable unsupervised branch. Figure 4 uses t-SNE exclusively for offline inspection of nonlinear local neighborhoods and does not generate any online score or class decision, and Figure 5 restores temporal order by connecting consecutive observations along a progressive degradation trajectory. The figures therefore support engineering interpretation of the hybrid algorithm; they are not independent substitutes for the controlled classification and event-level evaluation. Figure 2 shows that accepted nominal windows occupy a compact but non-degenerate operating manifold, incipient-degradation windows populate a transition region near its boundary, and degraded/abnormal windows exhibit larger displacement and dispersion. PC1 is primarily associated with overall dynamic intensity and load transfer, PC2 with redistribution among axes and motion phases, and PC3 with impulsiveness and localized high-frequency behavior. Partial overlap between nominal and incipient-degradation populations is retained because early degradation is expected to modify only a subset of descriptors before a sustained abnormal state emerges.

Figure 3 evaluates the robustness of the common PCA coordinate system under different operating loads. The low-, medium-, and high-load clouds translate continuously, mainly along the dynamic-intensity direction, while the nominal-to-incipient-degradation-to-degraded progression remains visible within each payload stratum. This result supports the use of context-compatible nominal centroids and robust dispersions rather than a universal distance from the unloaded centroid. A new payload outside the validated envelope is therefore reported as a low-confidence context instead of being incorporated automatically into the baseline.

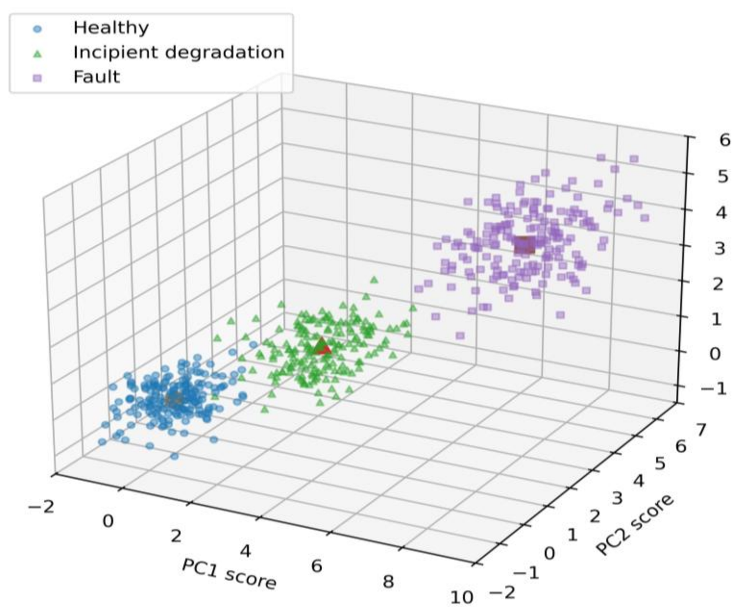

Figure 2: Three-Dimensional PCA Representation of Robot Health States.

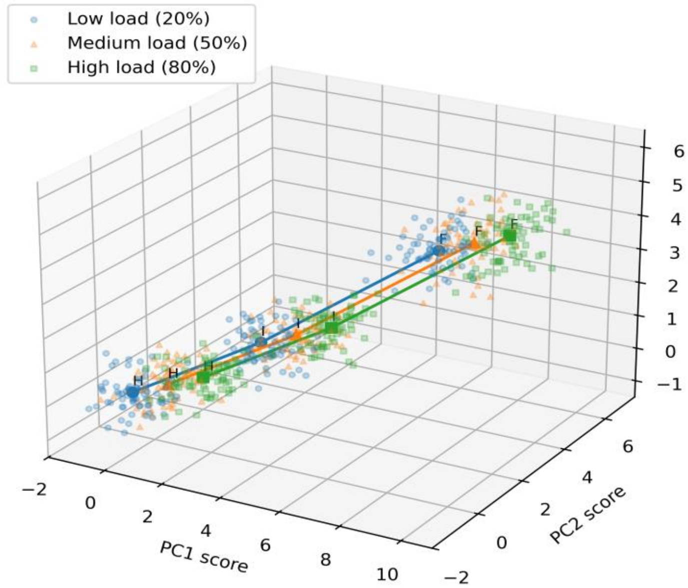

Figure 3: Three-Dimensional PCA Under Different Operating Loads.

Figure 4 emphasizes local neighborhood relationships that can be attenuated by a global linear projection. Nominal, incipient-degradation, and degraded windows form distinguishable but partially connected regions. The local organization indicates that windows with comparable total variance can differ through nonlinear combinations of temporal persistence, spectral redistribution, impulsiveness, and cross-axis coupling. This t-SNE map is an offline interpretive visualization only; it is not used as a classifier, threshold generator, or streaming anomaly score. Repeated-seed and grouped-resampling checks are used because the t-SNE axes have no physical units and absolute distances are not comparable across independently fitted embeddings.

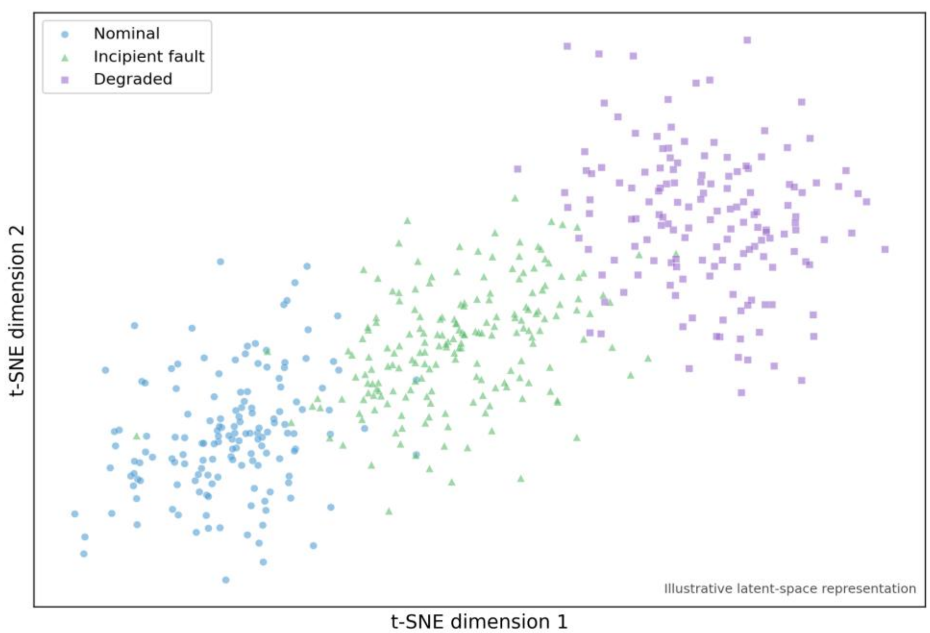

Figure 4: Two-Dimensional t-SNE Representation of Robot Health States for Offline Interpretation Only, Derived from the Standardized Feature Matrix after PCA-Based Initialization.

Figure 5 introduces the temporal information absent from an unordered point cloud. Consecutive windows begin in the nominal region, pass through the incipient-degradation transition zone, and migrate toward the degraded neighborhood. A persistent condition is characterized by coherent net displacement, accumulation of context-compatible distance, and repeated threshold exceedance. By contrast, an isolated shock produces an outward excursion followed by rapid recovery. The trajectory is used exclusively for offline engineering interpretation and does not convert latent displacement into a remaining-useful-life estimate. Figures 2–5 therefore provide complementary multiscale views of robot condition. PCA supplies the deterministic coordinate system used by the online novelty-detection branch; the load-stratified PCA representation distinguishes legitimate context-dependent translation from health-state migration; t-SNE is used only for offline inspection of nonlinear local neighborhoods; and the progressive trajectory provides an offline representation of persistent evolution relative to transient scatter. The deployed anomaly-decision mechanism does not use t-SNE coordinates. Online decisions are generated exclusively from the Random Forest abnormal-class probability, the context-compatible distance in the fixed PCA space, the context-adapted fusion threshold, and the H-of-K temporal persistence rule. Agreement between a high supervised probability and a large context-normalized PCA distance is consistent with a reviewed abnormal condition. Conversely, a large geometric departure combined with low supervised probability indicates a potentially novel condition, an unrecorded operating context, or a sensing anomaly requiring engineering review rather than automatic baseline adaptation. Short analysis windows reduced the time span required to generate a new condition estimate but produced greater variability in higher-order statistical and frequency-domain descriptors. Longer windows improved descriptor stability but provided limited additional discrimination and increased the temporal interval over which successive motion states were aggregated. The 150-sample configuration was therefore selected as the latency–stability operating point reported in Table 2. For the acquisition configuration used in the present analysis, the inertial stream was sampled at (f_s = 20) Hz and successive windows were generated with a stride of (h = 15) samples, corresponding to 90% overlap. The selected 150-sample window therefore represents 7.5 s of physical signal history, while a new analysis window is generated every 0.75 s. Accordingly, the 50-, 100-, 150-, and 200-sample configurations evaluated in Table 2 correspond to physical durations of 2.5, 5.0, 7.5, and 10.0 s, respectively, while retaining the same 0.75-s stride. Table 2 shows a broad performance plateau over the investigated window lengths. Accuracy increased from 95.8% for the 50-sample configuration to 97.1% for the selected 150-sample configuration and remained essentially unchanged at 97.0% for 200 samples. The corresponding F1-score increased from 0.94 to 0.96 and did not improve further when the window duration was extended from 7.5 to 10.0 s. These results support the selection of the 150-sample window as a practical compromise between descriptor stability, classification performance, and temporal responsiveness. The ablation results in Table 3 isolate the contribution of the proposed components. The PCA-distance branch achieved 91.2% accuracy and an F1-score of 0.89, confirming sensitivity to geometric novelty but also greater exposure to context-related false alarms.

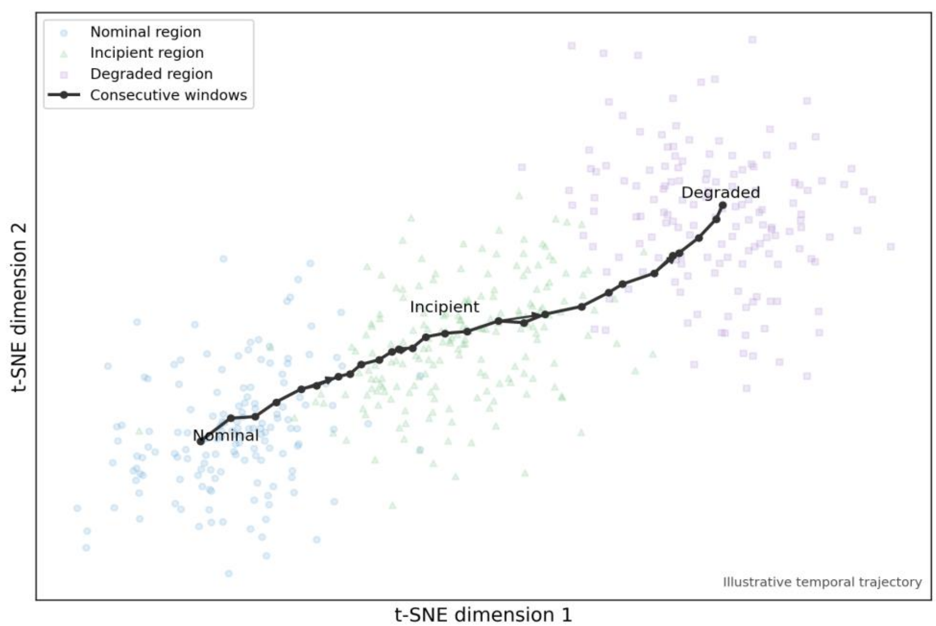

Figure 5: Progressive Degradation Trajectory in the t-SNE Manifold. Consecutive Operating Windows are Connected in Temporal Order from the Nominal Region Through the Incipient-Degradation Transition Zone to the Degraded Region. Directional Consistency and Accumulated Displacement Distinguish Persistent Migration from Isolated Excursions.

Table 2: Sensitivity of the Adaptive Hybrid Detector to Sliding-Window Length; The Corresponding Physical Window Duration and Stride are Stated in the Accompanying Text.

Window Length (Samples)	Physical Duration	Accuracy (%)	F1-Score	Engineering Interpretation
50	2.5s	95.8	0.94	Low latency; higher variability
100	5.0s	96.5	0.95	Balanced temporal response
150	7.5s	97.1	0.96	Selected latency-stability configuration
200	10.0s	97.0	0.96	Stable but slower response

Table 3: Ablation Results for the Monitoring Architecture.

Configuration	Accuracy (%)	F1-Score	Interpretation
Unsupervised PCA-Distance Branch	91.2	0.89	Sensitive to unknown deviations; more context-related false alarms
Supervised Random Forest Branch	93.5	0.91	Strong on known disturbances; limited novelty detection
Fixed-Threshold Hybrid	95.8	0.94	Complementary evidence without context correction
Adaptive Hybrid	97.1	0.96	Best balance under variable operating conditions

Table 4: Translation of Monitored Health Evidence into Conservative Maintenance Actions.

Observed Pattern	Typical Interpretation	Recommended Action	Baseline Update
Low Stable Fused Score	Behavior within validated envelope	Continue operation; routine review	Allowed after scheduled validation
Single High Unsupervised Score	Transient shock, packet error, or new context	Check data quality and process log	Not allowed
Repeated High Score Only at High Payload	Load-sensitive wear, compliance, or metadata error	Inspect; consider payload/speed derating	Not allowed until resolved
Repeated High Score Across Contexts	Generalized mechanical or sensing degradation	Controlled stop and diagnosis	Not allowed
High Supervised Score for Known Event	Recurrence of labeled disturbance	Apply documented corrective action	Not allowed for abnormal run
Slow Upward Trend Below Alarm	Possible incipient degradation	Increase review frequency	Conservative update only

The supervised Random Forest achieved an accuracy of 93.5 ± 1.5% and an F1-score of 0.91 ± 0.02 across the complete-run grouped evaluations. Fixed-threshold fusion increased the corresponding performance to 95.8 ± 1.1% accuracy and an F1-score of 0.94 ± 0.01, whereas the complete adaptive hybrid achieved 97.1 ± 0.8% accuracy and an F1-score of 0.96 ± 0.01. For comparison, the unsupervised PCA-distance branch achieved 91.2 ± 1.9% accuracy and an F1-score of 0.89 ± 0.02. These results indicate progressively improved and more stable classification performance when supervised disturbance recognition, geometric novelty detection, and bounded context adaptation are combined. The reported variability corresponds to the standard deviation of the performance metrics calculated across the 18 independent physical acquisition runs of the held-out test set. Each physical run constitutes one statistical evaluation unit; overlapping windows within the same run are therefore not treated as independent statistical replicates. The comparison is therefore interpreted as evidence of improved robustness within the controlled archive rather than as proof of universal statistical superiority. Table 4 translates the branch scores, operating-context information, and temporal trend into conservative maintenance actions. A low and stable fused score supports continued operation within the validated operating envelope, whereas an isolated increase in the unsupervised score first requires verification of data quality, process events, sensor status, and context metadata. Repeated deviations confined to high-payload or high-speed conditions indicate a potentially load-sensitive response and justify targeted inspection and, where consistent with manufacturer recommendations and process requirements, temporary derating. Persistent deviations observed across multiple validated contexts provide stronger evidence for controlled diagnosis. Abnormal runs are never incorporated automatically into the nominal baseline.

The deployed online chain includes feature extraction, Random Forest inference, projection into the fixed PCA space, context-compatible distance evaluation, decision-level fusion, and H-of-K temporal persistence. t-SNE and progressive-manifold visualization are used exclusively for offline engineering interpretation and are therefore excluded from the real-time computational budget. On the 64 MHz ARM Cortex-M4 target platform, the deployment characterization indicates a median end-to-end processing latency of 5–20 ms/window, a 95th-percentile latency of 10–30 ms/window, and a worst-case latency of 20–50 ms/window. The serialized Random Forest plus PCA deployment package occupies approximately 40–100 kB, while peak SRAM demand is approximately 0.04–0.06 MB. The implementation sustains at least 20 windows/s and typically 30–50 windows/s when BLE communication does not introduce blocking. Because a new analysis window is generated every 0.75 s under the selected 20 Hz sampling frequency and 15-sample stride, the available computational margin is substantially larger than the measured processing time per window. Processor utilization was therefore not used as the primary real-time feasibility metric; computational adequacy is characterized through end-to-end latency, memory demand, and sustainable processing throughput.

4. Discussion

The study proposes an integrated monitoring algorithm for aging industrial robots. Random Forest, PCA, robust distance, adaptive thresholding, and temporal persistence are established techniques; the methodological contribution is therefore defined by their constrained interaction and validation logic. The architecture retains separate supervised and unsupervised evidence, adapts only within validated context bounds, assigns low confidence to unknown contexts, prevents run-level leakage, and converts correlated window scores into persistent event-level indications. The manuscript presents a deployable, auditable framework for controller-independent monitoring rather than a new classifier or dimensionality-reduction method.

4.1 Interpretation of the Latent Representations

Figures 2-5 form one connected analytical argument. Figure 2 establishes the global health-state geometry, Figure 3 separates load-dependent translation from health-state migration, Figure 4 examines nonlinear local neighborhoods, and Figure 5 restores temporal direction. PCA is used online because its loading matrix is deterministic and directly transformable out of sample. By contrast, t-SNE is used only offline to inspect local neighborhood structure; its stochastic, non-parametric coordinates do not generate online anomaly decisions. The load-driven PCA displacement is associated mainly with overall acceleration and angular-rate intensity, whereas degradation may additionally increase impulsiveness, high-frequency energy, directional imbalance, or altered temporal correlation. PCA loadings, grouped permutation importance, and local SHAP explanations identify variables supporting a Random Forest decision (Lundberg & Lee, 2017), but they do not establish component-level damage. Explainability therefore guides inspection and hypothesis generation while physical verification remains required.

4.2 Positioning Relative to Anomaly-Detection Methods

The proposed framework occupies a different design point from high-capacity deep anomaly detectors. Autoencoder and convolutional reconstruction methods can learn valuable representations directly from acoustic or vibration signals (Yun et al., 2023; Zhong et al., 2022), while multichannel and physics-informed robot models can exploit broader temporal or structural information (Wang et al., 2024; Yang et al., 2024). Robot-specific studies have investigated dual-module attention convolutional networks (Lu et al., 2022), deep residual joint-fault diagnosis (Pan et al., 2022), multimodal anomaly detection in human-robot collaboration (Yang et al., 2021), and dynamic human-robot assembly monitoring (Schirmer et al., 2023). Broader work has addressed AI-based fault-tolerant control (Khan et al., 2025), data-driven industrial fault diagnosis, memory-guided transformers, and memory-based unsupervised detectors (Li et al., 2023). Robot-specific mechanical studies have also addressed gearbox reliability under stochastic operating conditions (Park et al., 2023) and in-situ fault diagnosis of industrial-robot harmonic reducers using dedicated deep-learning architectures. These methods can provide stronger representation capacity when large, well-curated datasets and suitable computational resources are available.

The present method instead prioritizes moderate data requirements, low inference cost, explicit handling of operating context, and traceability from the final alarm to the original sensor descriptors. This choice is relevant for retrofit applications in which proprietary servo data, reliable digital-twin assets, and component-specific instrumentation are unavailable. Context-aware robot-monitoring studies similarly demonstrate that trajectory, payload, and operating-state changes should not be interpreted automatically as faults (Ayankoso et al., 2024; Kayan et al., 2024; Kermenov et al., 2023). The distinctive feature here is the explicit fusion of reviewed disturbance probability with context-compatible PCA distance, followed by bounded threshold correction and temporal persistence. Computational feasibility is part of the algorithmic contribution rather than an implementation request. Low-power vibration-monitoring literature emphasizes that feature design, model size, memory demand, and end-to-end latency determine whether a diagnostic method can operate on an industrial edge node (Vasilache et al., 2024). The proposed online chain excludes t-SNE and retains only fixed preprocessing, engineered descriptors, Random Forest inference, PCA projection, robust distance, score fusion, and persistence logic. This separation permits the nonlinear visualization to enrich engineering interpretation without imposing its optimization cost or stochastic behavior on real-time decisions.

4.3 Predictive-Maintenance Implications

The framework supports condition-based maintenance decision support rather than prognostic life prediction. In the broader predictive-maintenance literature, sensor-data-driven approaches have been used to support maintenance decision-making and Industry 4.0 applications (Fordal et al., 2023), while prognostic frameworks specifically address remaining useful lifetime estimation when suitable degradation histories are available (Taşcı et al., 2023). Stable context-compatible scores support routine monitoring; persistent migration near the nominal boundary justifies increased review and targeted inspection; and repeated deviations restricted to high payload or speed can support temporary derating while the cause is investigated, provided that manufacturer limits and process-quality requirements are respected. Persistent deviations across several contexts provide stronger evidence for controlled diagnosis or planned intervention. Because the present archive contains controlled perturbations rather than run-to-failure degradation histories, the progressive degradation index is not interpreted as remaining useful life, a wear rate, or a component-specific failure forecast. The responsible output is a health-state assessment, confidence status, and maintenance recommendation.

4.4 Limitations and Confirmatory Developments

The principal limitation is that the available archive contains one robot platform and controlled perturbations rather than a run-to-failure or wear campaign. The accuracy and F1-score values reported in Section 3 characterize performance within the controlled single-platform archive and are accompanied by run-level variability across the 18 independent physical test runs. The method therefore supports anomaly and condition monitoring but does not estimate remaining useful life or uniquely localize a mechanical fault. External inertial sensing also captures the combined response of the robot, sensor mounting, tool, cables, fixture, floor, and process environment, so sensor relocation, thermal state, tool contact, foundation vibration, or undocumented recipe changes may mimic degradation. Broader validation should include multiple acquisition days, cold-start and thermally stabilized operation, several payload and speed ratios, repeated trajectories, sensor-remounting trials, post-maintenance baselines, physically justified degradation mechanisms, and competitive baselines under the same run-grouped protocol. A reproducibility package should retain processed window-level features with run identifiers, the immutable partition file, configuration and software versions, trained-model metadata, evaluation scripts, and archived prediction vectors.

5. Conclusions

This study formulated a controller-independent hybrid anomaly-detection algorithm for evidence-based condition monitoring of a six-axis compact industrial robot. The online path combines a supervised Random Forest with context-compatible robust distance in a fixed PCA space, followed by calibrated fusion, bounded context correction, and H-of-K temporal persistence. The design preserves branch-level evidence, prevents overlapping windows from crossing run partitions, separates deterministic deployment from stochastic t-SNE visualization, and translates persistent deviations into conservative maintenance actions. In the controlled archive, the adaptive hybrid achieved higher mean accuracy and F1-score than the individual branches and fixed fusion. These results support the proposed integration strategy but do not establish remaining useful life, wear progression, or component-level fault isolation. The framework therefore provides a transparent basis for retrofit condition monitoring of mechanically serviceable robots with limited access to proprietary diagnostics.

References

Ayankoso, S., Gu, F., Louadah, H., Fahham, H., & Ball, A. (2024). Artificial-intelligence-based condition monitoring of industrial collaborative robots: detecting anomalies and adapting to trajectory changes. *Machines*, *12*(9), 630. https://doi.org/10.3390/machines12090630

Azzalini, D., Bonali, L., & Amigoni, F. (2021). A minimally supervised approach based on variational autoencoders for anomaly detection in autonomous robots. *IEEE Robotics and Automation Letters*, *6*(2), 2985-2992.

https://doi.org/10.1109/LRA.2021.3062597

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/10.1023/a:1010933404324

Chirayil Nandakumar, S., Mitchell, D., Erden, M. S., Flynn, D., & Lim, T. (2024). Anomaly detection methods in autonomous robotic missions. *Sensors*, *24*(4), 1330. https://doi.org/10.3390/s24041330

Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE Access*, *9*, 120043-120065. https://doi.org/10.1109/ACCESS.2021.3107975

Choi, S.-H., An, D., Lee, I., & Lee, S. (2024). Anomaly detection based on graph convolutional network–variational autoencoder model using time-series vibration and current data. *Mathematics*, *12*(23), 3750. https://doi.org/10.3390/math12233750

Deng, A., & Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, pp. 4027-4035). https://doi.org/10.1609/aaai.v35i5.16523

Fedorova, D., Tlach, V., Kuric, I., Dodok, T., Zajačko, I., & Tucki, K. (2024). Technical diagnostics of industrial robots using vibration signals: Case study on detecting base unfastening. *Applied Sciences*, *15*(1), 270. https://doi.org/10.3390/app15010270

Fordal, J. M., Schjølberg, P., Helgetun, H., Skjermo, T. Ø., Wang, Y., & Wang, C. (2023). Application of sensor data based predictive maintenance and artificial neural networks to enable Industry 4.0. *Advances in Manufacturing*, *11*(2), 248-263. https://doi.org/10.1007/s40436-022-00433-x

Garg, A., Zhang, W., Samaran, J., Savitha, R., & Foo, C.-S. (2022). An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(6), 2508-2517. https://doi.org/10.1109/tnnls.2021.3105827

Huang, Y., Yang, B., Wang, S., Xie, K., Wang, Y., Yi, L., & Dong, N. (2025). Attentive neural processes based on reliable inferences for industrial equipment anomaly detection. *Computers in Industry*, *171*, 104331. https://doi.org/10.1016/j.compind.2025.104331

ISO. (2003). 13374-1:2003—Condition Monitoring and Diagnostics of Machines: Data Processing, Communication and Presentation—Part 1: General Guidelines. In (1 ed.). International Organization for Standardization. https://www.iso.org/standard/21832.html

ISO. (2018). ISO 17359:2018—Condition Monitoring and Diagnostics of Machines: General Guidelines. In. International Organization for Standardization. https://www.iso.org/standard/71194.html

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Kayan, H., Heartfield, R., Rana, O., Burnap, P., & Perera, C. (2024). Casper: Context-aware iot anomaly detection system for industrial robotic arms. *ACM Transactions on Internet of Things*, *5*(3), 1-36. https://doi.org/10.1145/3670414

Kermenov, R., Nabissi, G., Longhi, S., & Bonci, A. (2023). Anomaly detection and concept drift adaptation for dynamic systems: a general method with practical implementation using an industrial collaborative robot. *Sensors*, *23*(6), 3260. https://doi.org/10.3390/s23063260

Khan, Z., Nasir, A., & Mekid, S. (2025). Fault-tolerant control strategies for industrial robots: State of the art and future perspective on AI-based fault management. *Artificial Intelligence Review*, *58*(11), 362. https://doi.org/10.1007/s10462-025-11327-2

Kim, H. (2025). Fault Diagnosis in Robot Drive Systems Using Data-Driven Dynamics Learning. In *Actuators* (Vol. 14, pp. 583). MDPI. https://doi.org/10.3390/act14120583

Kim, S., Choi, K., Choi, H.-S., Lee, B., & Yoon, S. (2022). Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, pp. 7194-7201). https://doi.org/10.1609/aaai.v36i7.20680

Kumar, P., Khalid, S., & Kim, H. S. (2023). Prognostics and health management of rotating machinery of industrial robot with deep learning applications—A review. *Mathematics*, *11*(13), 3008. https://doi.org/10.3390/math11133008

Kumar, P., Raouf, I., & Kim, H. S. (2024). Transfer learning for servomotor bearing fault detection in the industrial robot. *Advances in engineering software*, *194*, 103672. https://doi.org/10.1016/j.advengsoft.2024.103672

Li, J., Chen, Y., & Xing, Y. (2023). Memory mechanism for unsupervised anomaly detection. In *The 39th Conference on Uncertainty in Artificial Intelligence*. https://openreview.net/pdf?id=8xPj4G7E-yG

Liu, Q., Yu, Y., Han, B. S., & Zhou, W. (2025). An evolutionary deep learning approach using flexible variable-length dynamic stochastic search for anomaly detection of robot joints. *Applied Soft Computing*, *168*, 112493. https://doi.org/10.1016/j.asoc.2024.112493

Lu, K., Chen, C., Wang, T., Cheng, L., & Qin, J. (2022). Fault diagnosis of industrial robot based on dual-module attention convolutional neural network. *Autonomous Intelligent Systems*, *2*(1), 12. https://doi.org/10.1007/s43684-022-00031-5

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In (Vol. 30). Advances in Neural Information Processing Systems. https://tinyurl.com/ye26vpcd

Mallioris, P., Aivazidou, E., & Bechtsis, D. (2024). Predictive maintenance in Industry 4.0: A systematic multi-sector mapping. *CIRP journal of manufacturing science and technology*, *50*, 80-103. https://doi.org/10.1016/j.cirpj.2024.02.003

Niu, Q., Sui, Z., Han, J., & Zhao, Y. (2025). An industrial robot gearbox fault diagnosis approach using multi-scale empirical mode decomposition and a one-dimensional convolutional neural network-bidirectional gated recurrent unit method. *Processes*, *13*(6), 1722. https://doi.org/10.3390/pr13061722

Pan, J., Qu, L., & Peng, K. (2022). Deep residual neural-network-based robot joint fault diagnosis method. *Scientific reports*, *12*(1), 17158. https://doi.org/10.1038/s41598-022-22171-7

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, *54*(2), 1-38. https://doi.org/10.1145/3439950

Paparrizos, J., Boniol, P., Palpanas, T., Tsay, R. S., Elmore, A. J., & Franklin, M. J. (2022). Volume Under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection. *Proc. VLDB Endow.*, *15*(11), 2774-2787. https://doi.org/10.14778/3551793.3551830

Park, J.-g., Yoon, H., & Youn, B. D. (2023). Probabilistic framework for reliable optimal design of gearboxes in general-purpose industrial robots considering random use conditions. *Journal of Computational Design and Engineering*, *10*(2), 539-548. https://doi.org/10.1093/jcde/qwad007

Park, Y.-S., Yoo, D.-Y., & Lee, J.-W. (2021). Programmable motion-fault detection for a collaborative robot. *IEEE Access*, *9*, 133123-133142. https://doi.org/10.1109/access.2021.3114505

Polenghi, A., Cattaneo, L., & Macchi, M. (2024). A framework for fault detection and diagnostics of articulated collaborative robots based on hybrid series modelling of Artificial Intelligence algorithms. *Journal of Intelligent Manufacturing*, *35*(5), 1929-1947. https://doi.org/10.1007/s10845-023-02076-6

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Müller, K.-R. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, *109*(5), 756-795. https://doi.org/10.1109/JPROC.2021.3052449

Sabry, A. H., & Amirulddin, U. A. B. U. (2024). A review on fault detection and diagnosis of industrial robots and multi-axis machines. *Results in Engineering*, *23*, 102397. https://doi.org/10.1016/j.rineng.2024.102397

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*, *10*(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Schirmer, F., Kranz, P., Schmitt, J., & Kaupp, T. (2023). Anomaly detection for dynamic human-robot assembly: Application of an lstm-based autoencoder to interpret uncertain human behavior in hrc. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction* (pp. 333-337). https://doi.org/10.1145/3568294.3580100

Taşcı, B., Omar, A., & Ayvaz, S. (2023). Remaining useful lifetime prediction for predictive maintenance in manufacturing. *Computers & Industrial Engineering*, *184*, 109566. https://doi.org/10.1016/j.cie.2023.109566

Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., & Gottschlich, J. (2018). Precision and recall for time series. *Advances in neural information processing systems*, *31*. https://tinyurl.com/2hpbyjjv

Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proceedings of the VLDB Endowment*, *15*(6), 1201-1214. https://doi.org/10.14778/3514061.3514067

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11), 2579-2605. https://tinyurl.com/488s3pan

Vasilache, A., Nitzsche, S., Floegel, D., Schuermann, T., von Dosky, S., Bierweiler, T., Mußler, M., Kälber, F., Hohmann, S., & Becker, J. (2024). Low-Power Vibration-Based Predictive Maintenance for Industry 4.0 using Neural Networks: A Survey. *arXiv:2408.00516*. https://doi.org/10.48550/arXiv.2408.00516

Wang, S., Tao, J., Jiang, Q., Chen, W., Qin, C., & Liu, C. (2024). A digital twin framework for anomaly detection in industrial robot system based on multiple physics-informed hybrid convolutional autoencoder. *Journal of Manufacturing Systems*, *77*, 798-809. https://doi.org/10.1016/j.jmsy.2024.10.016

Wu, J., Zhang, Y., Gao, B., Xia, L., Zhu, X., Wang, H., & Wan, X. (2025). A Hybrid Deep Learning Framework for Enhanced Fault Diagnosis in Industrial Robots. *Algorithms*, *18*(12), 779. https://doi.org/10.3390/a18120779

Xu, J., Wu, H., Wang, J., & Long, M. (2022). Anomaly transformer: Time series anomaly detection with association discrepancy. In. ICLR. https://openreview.net/pdf?id=LzQQ89U1qm_

Yang, B., Long, W., Zhang, Y., Xi, Z., Jiao, J., & Li, Y. (2024). Multivariate time series anomaly detection: Missing data handling and feature collaborative analysis in robot joint data. *Journal of Manufacturing Systems*, *75*, 132-149. https://doi.org/10.1016/j.jmsy.2024.06.006

Yang, L., Yan, W., & Wu, H. (2021). Comparison of deep learning-based methods in multimodal anomaly detection: a case study in human–robot collaboration. *Science Progress*, *104*(2), 00368504211021192. https://doi.org/10.1177/00368504211021192

Yun, H., Kim, H., Jeong, Y. H., & Jun, M. B. (2023). Autoencoder-based anomaly detection of industrial robot arm using stethoscope based internal sound sensor. *Journal of Intelligent Manufacturing*, *34*(3), 1427-1444. https://doi.org/10.1007/s10845-021-01862-4

Zhong, Z., Zhao, Y., Yang, A., Zhang, H., Qiao, D., & Zhang, Z. (2022). Industrial Robot Vibration Anomaly Detection Based on Sliding Window One-Dimensional Convolution Autoencoder. *Shock and Vibration*, *2022*(1), 1179192. https://doi.org/10.1155/2022/1179192