Sound & Vibration 2026, 60(2), 3950.
doi: https://doi.org/10.59400/sv3950

Article

A fuzzy-neuro approach to predictive maintenance using vibration signature classification

Asokan Vasudevan[1], Mohammed El Khider[2], Yogeesh N.[1,3,*], Puspanathan Doraisingam[1], Khan Sarfaraz Ali[1], A. Sathishkumar[4]

[1] Faculty of Business and Communications, INTI International University, Nilai 71800, Malaysia
[2] Department of General Undergraduate Curriculum Requirements, University of Dubai, Dubai P.O. Box 14143, United Arab Emirates
[3] Department of Mathematics, Government First Grade College, Tumakuru 572102, India
[4] Department of Biomedical Engineering, Erode Sengunthar Engineering College, Perundurai 638057, India
*** Corresponding author:** Yogeesh N., yogeesh.r@gmail.com

CITATION

Vasudevan A, Khider ME, N. Y, et al. A fuzzy-neuro approach to predictive maintenance using vibration signature classification. Sound & Vibration. 2026; 60(2): 3950. https://doi.org/10.59400/sv3950

ARTICLE INFO

Received: 23 January 2026
Revised: 21 February 2026
Accepted: 12 March 2026
Available online: 31 March 2026

Abstract: Vibration-based predictive maintenance must remain reliable under variable speed/load and noise while delivering actionable, interpretable decisions. We propose a fuzzy-neuro framework that maps windowed vibration segments to class decisions and a calibrated risk score with uncertainty. Given a sampled signal $x[n]$, overlapping windows $x_k \in \mathbb{R}^L$ are formed and encoded into a learned signature $z_k = f_\theta(x_k)$. Physics-informed indicators $\phi(x_k)$ (e.g., RMS, kurtosis, band-energy and short-horizon trend) are fused with z_k to form $u_k = [\phi(x_k), z_k]$. An end-to-end neuro-adaptive Takagi-Sugeno-Kang (TSK) layer produces a transparent maintenance risk r_k enabling rule-level explanations via dominant firing strengths $\bar{w}_m$. To support safe decision-making near thresholds, we estimate an interval risk $[r_k^-, r_k^+]$ from predictive samples $\left\{r_k^{(s)}\right\}$ using quantiles $r_k^- = Q_{\alpha/2}$, $r_k^+ = Q_{1-\alpha/2}$ and width $w_k = r_k^+ - r_k^-$ as confidence. Using leakage-safe, unit-aware temporal splitting, experiments on public rolling-bearing benchmarks achieve 0.956 accuracy, 0.952 macro-F1, and 0.941 MCC, while the full model improves calibration and yields sharper risk intervals (mean width ≈ 0.132), translating classifier evidence into auditable "monitor/schedule/urgent" actions. These results indicate that the proposed framework is accurate, interpretable, and decision-ready for predictive maintenance.

Keywords: predictive maintenance; vibration signature; rolling bearing fault diagnosis; Takagi-Sugeno-Kang (TSK) fuzzy inference; neuro-fuzzy learning; uncertainty quantification; explainable AI

1. Introduction

1.1. Background and motivation

Unexpected failures in rotating machinery (bearings, gearboxes, motors) cause unplanned downtime, safety hazards, and costly secondary damage. However, vibration monitoring is a primary modality for early fault discovery because incipient defects generate structured changes in the signal's time-frequency content-impulsive transients, envelope sidebands, and cyclo-stationary components-well before catastrophic breakdown [1–4]. Hence, real deployments face domain drift: signatures shift with speed/load regimes [5, 6], sensor mounting, lubrication, and ambient noise, so a classifier trained under one system may degrade under another [7–10].

Now mathematically, by taking unit u we observe an acceleration time series $x^{(u)}[n]$ and we formed overlapping windows expressed as

$$x_k \in \mathrm{R}^L,\ x_k = \left[x^{(u)}[kH], \ldots, x^{(u)}[kH + L - 1]\right],$$

with hop H and length L. Every window is in associated with a health label $y_k \in \{1, \ldots, C\}$ and for maintenance planning, an actionable risk variable $r_k \in [0, 1]$. Therefore, the learning objective is not only accurate classification that is

$$\widehat{y}_k = \arg\max_c \, p_\theta \left(y = c \mid x_k\right),$$

but also, a calibrated risk estimates and confidence that can drive required operational decisions [11–15].

An end-to-end neuro-adaptive Takagi-Sugeno-Kang (TSK) layer produces a transparent maintenance risk

$$r_k = \sum_{m=1}^{M} \bar{w}_m \left(u_k\right) \left(a_m^\top u_k + b_m\right),\ \bar{w}_m = \frac{\prod_j \ \mu_{mj} \left(u_{k,j}\right)}{\sum_\ell \prod_j \ \mu_{\ell j} \left(u_{k,j}\right)},$$

enabling rule-level explanations via dominant firing strengths $\bar{w}_m$.

1.2. Why we need fuzzy-neuro for PdM

The Fuzzy inference systems naturally establish gradual transitions between health states and express decisions as linguistic rules along with desired transparent semantics. In existing fuzzy models, Takagi-Sugeno-Kang (TSK) inference is particularly suitable for PdM because each rule outputs a local affine model, making the overall mapping differentiable and compact as mention below:

Rule m: IF u_1 is $A_{m1} \wedge \cdots \wedge u_d$ is A_{md} THEN $r_m(u) = a_m^\top u + b_m$.

The product inference and normalization is,

$$w_m(u) = \prod_{j=1}^{d} \mu_{mj} \left(u_j\right),\ \bar{w}_m(u) = \frac{w_m(u)}{\sum_{\ell=1}^{M} \ w_\ell(u) + \varepsilon},$$

and the fused output is

$$r(u) = \sum_{m=1}^{M} \bar{w}_m(u) \left(a_m^\top u + b_m\right).$$

This form makes interpretability explicit that $\bar{w}_m(u)$ reveals which rules fired, while (a_m, b_m) provides a local linear explanation of how each input contributes to risk [16, 17].

1.3. Experimental basis (datasets used in this study)

To ensure required reproducibility and meaningful operational variability, experiments are built on established rolling-element bearing benchmarks that include controlled faults and multi-condition measurements depends on the following:

- Paderborn bearing benchmark (PU): multiple damage mechanisms and operating settings, widely used for vibration-based classification and robustness testing [18].
- XJTU-SY accelerated life test: run-to-failure trajectories capturing progressive degradation, supporting severity/risk mapping and early-warning evaluation [19,20].
- CWRU benchmarking protocol considerations: included to align with common baselines and to address known pitfalls in cross-condition evaluation and domain shift [2,8].

1.3.1. Leakage-safe evaluation and "future-like" testing

To avoid optimistic bias, the study follows a leakage-safe split consistent with deployment: windows from the same unit and temporal neighborhood are not mixed across train/test. Hence, we use unit-aware and time-aware partitioning, so the test set better represents unseen future degradation rather than mixed windows from the same run. We also explicitly acknowledge the limitation that public datasets cannot fully reproduce all field complexities and also, we reported both classification metrics and risk/confidence behavior to support experimental PdM interpretation.

1.3.2. Input construction and redundancy control

Here each window x_k is mapped to a learned embedding $z_k = f_\theta(x_k)$ and a physics-informed feature vector $\phi(x_k)$. Hence we fuse them as

$$u_k = [\phi(x_k), z_k],$$

feeding u_k to the TSK layer to reduce redundancy between handcrafted indicators and learned embeddings, the feature set is selected from PdM literature and then checked via correlation/ablation screening.

1.4. Contributions of this study

This study presents a predictive maintenance framework for vibration-based fault diagnosis that combines deep learning and fuzzy reasoning in a very simple and practical way. The main contributions of this work are summarized as follows.

- A hybrid fuzzy-neuro model for maintenance risk prediction.
- Built-in interpretability.
- Clear separation between engineered features and learned features.
- Uncertainty-aware risk estimation.
- A practical maintenance decision policy.
- Leakage-safe and realistic evaluation.

The raw vibration signal is divided into overlapping windows, from each window, statistical and spectral features are extracted, and learned signal representations are also generated through the neural network (**Figure 1**).

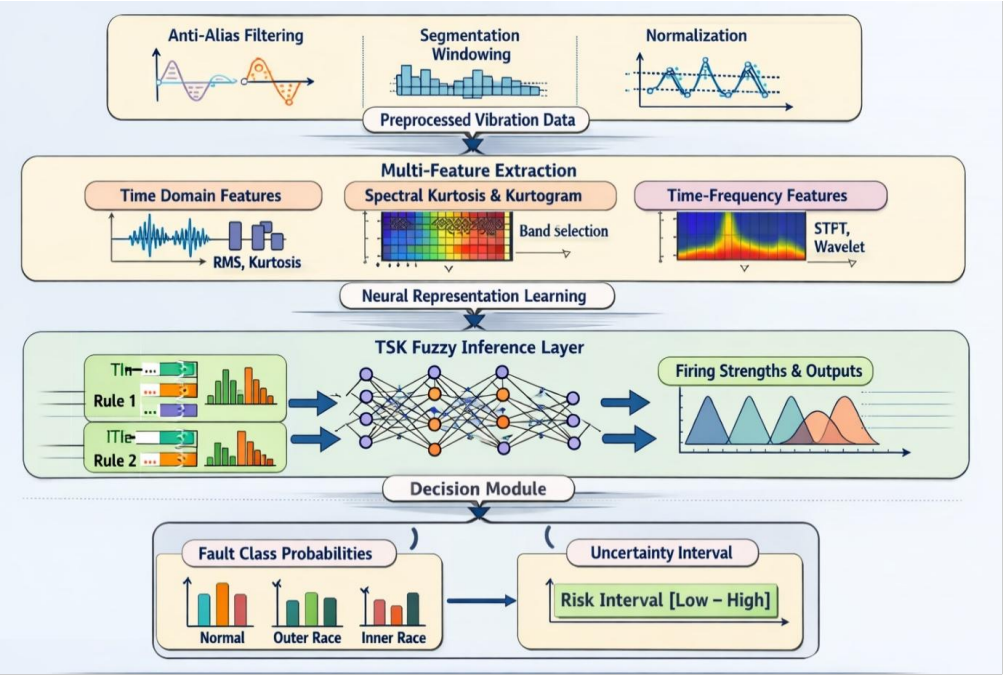

Figure 1. End-to-end fuzzy-neuro predictive maintenance pipeline for vibration signature classification.

These two types of information are fused and passed to the TSK fuzzy-neuro layer, which produces the health condition and maintenance risk score. Finally, a decision policy uses the estimated risk and confidence level to issue actions such as monitoring, maintenance scheduling, or urgent inspection. **Table 1** presents the notation used in this research.

Table 1. Notation used in this study.

Symbol	Meaning
$x(t)$	Continuous-time vibration signal
$x[n]$	Discrete vibration sample at index n
f_s	Sampling frequency (Hz)
L	Window length (samples)
H	Hop size/stride (samples)
$\mathrm{x}_k \in \mathrm{R}^L$	k-th windowed segment
$\phi(\mathrm{x}_k)$	Feature map (handcrafted or learned)
z_k	Learned embedding (“signature code”)
C	Number of health classes
$\widehat{y}_k$	Predicted class label
$p_k(c)$	Predicted class probability for class c
R_k	PdM risk score for window k
M	Number of TSK rules
$w_r(\cdot)$	Rule firing strength for rule r
$\mu_{rj}(\cdot)$	Membership function of rule r, input j
$g_r(\cdot)$	Consequent function of TSK rule r

2. Methods of fuzzy-neuro vibration signature classification

2.1. The signal windowing and feature representation

2.1.1. Windowing model

Let $x[n]$ represent the vibration signal that sampled at the rate f_s. The signal is partitioned into overlapping windows of length L with hop H. The k-th window is

$$\mathbf{x}_k = [x[kH], x[kH+1], \ldots, x[kH+L-1]]^\top \in \mathrm{R}^L.$$

Windowing converts streaming vibration into a sequence $\{\mathbf{x}_k\}_{k=1}^N$ suitable for both classical feature extraction and neural encoding. To reduce spectral leakage, a taper $u[n]$ (e.g., Hann) is applied:

$$\widetilde{x}_k[n] = u[n]x[kH+n],\ n = 0, \ldots, L-1.$$

2.1.2. Time-domain statistical features

From each window $\mathbf{x}_k$, compute robust summary features commonly used in bearing diagnostics [3–6,9], including:

Root-mean-square (RMS) (energy proxy):

$$\mathrm{RMS}_k = \sqrt{\frac{1}{L}\sum_{n=0}^{L-1} x_k[n]^2}.$$

Kurtosis (impulsiveness proxy):

Let $\bar{x}_k = \frac{1}{L}\sum_n x_k[n]$ and $s_k^2 = \frac{1}{L}\sum_n (x_k[n]-\bar{x}_k)^2$. Then

$$\mathrm{Kurt}_k = \frac{\frac{1}{L}\sum_{n=0}^{L-1}(x_k[n]-\bar{x}_k)^4}{\left(s_k^2\right)^2}.$$

Crest factor (peak-to-RMS):

$$\mathrm{CF}_k = \frac{\max\limits_n |x_k[n]|}{RMS_k}.$$

These features are stable, interpretable, and correlate with early bearing defects (impulsive impacts) [3–6].

2.1.3. Frequency/Envelope features

Let $X_k[\omega]$ be the discrete Fourier transform (DFT) of $\widetilde{x}_k[n]$. Band-energy features are

$$E_k(\Omega) = \sum_{\omega\in\Omega} |X_k[\omega]|^2,$$

where Ω is a frequency band selected around known resonance or defect-sensitive regions [1,5,6]. For bearing faults, envelope analysis is also important; demodulation/ envelope-based pipelines are well established in vibration PdM [1,5,6,21].

2.1.4. Learned signature embedding (encoder)

To improve robustness under working-condition variation, each window is also mapped to a learned embedding:

$$\mathbf{z}_k = f_\theta(\mathbf{x}_k) \in \mathrm{R}^d,$$

where f_θ is a 1D CNN or lightweight temporal encoder optimized for fault discrimination under varying conditions [8, 11, 12]. The final representation concatenates interpretable features and learned signatures:

$$\mathbf{h}_k = [\phi(\mathbf{x}_k) \| \mathbf{z}_k] \in \mathrm{R}^m.$$

This hybrid representation improves separability while preserving explainable channels used by the fuzzy layer [11,12,14,15].

Figure 2 demonstrates: Windowing → detrending/tapering → time statistics (RMS, kurtosis, crest factor) + spectrum/envelope bands → learned embedding $\mathbf{z}_k$ → fused feature vector $\mathbf{h}_k$.

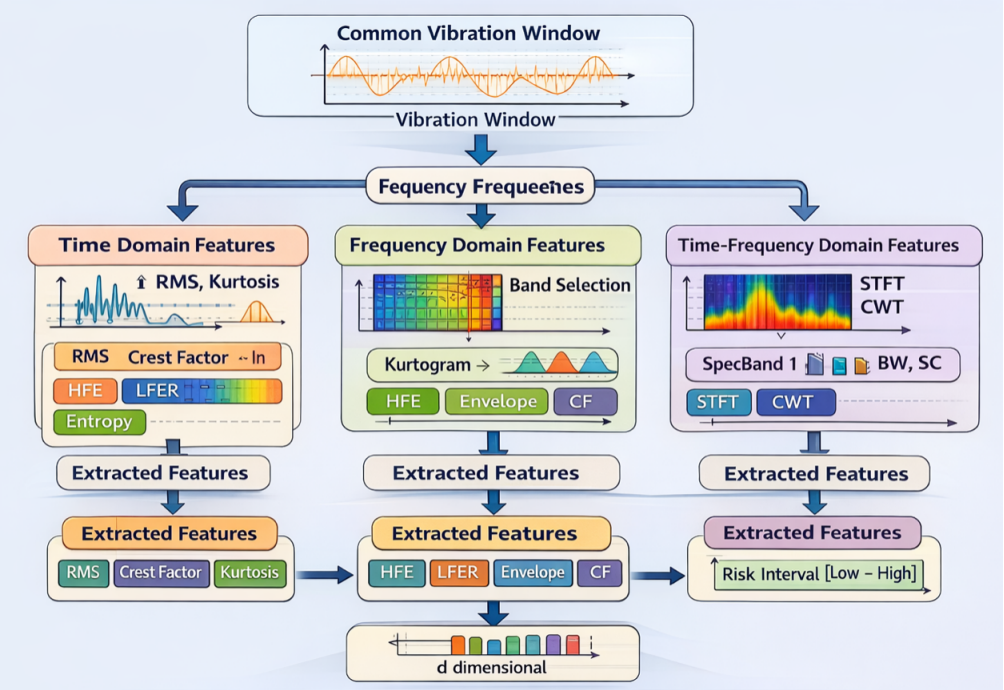

Figure 2. Feature extraction used for vibration signature classification.

2.2. TSK fuzzy rule base and neuro-adaptation

2.2.1. TSK rule structure

Given $\mathbf{h}_k = [h_{k1}, \ldots, h_{km}]^\top$, define a TSK rule base with M rules. The r-th rule is:

Rule r : IF h_{k1} is A_{r1} AND h_{k2} is A_{r2} AND ... AND h_{km} is A_{rm}, Then,

$$g_r(\mathbf{h}_k) = a_{r0} + \sum_{j=1}^{m} a_{rj} h_{kj}.$$

The antecedents A_{rj} are fuzzy sets with membership functions $\mu_{rj}(\cdot)$. A common choice is Gaussian:

$$\mu_{rj}(h) = \exp\left(-\frac{(h - c_{rj})^2}{2\sigma_{rj}^2}\right),$$

with centers c_{rj} and spreads σ_{rj}.

2.2.2. Rule firing strengths and normalized aggregation

Using product inference, the (unnormalized) firing strength is

$$w_r(\mathbf{h}_k) = \prod_{j=1}^{m} \mu_{rj}(h_{kj}).$$

Normalize across rules:

$$\bar{w}_r(\mathbf{h}_k) = \frac{w_r(\mathbf{h}_k)}{\sum_{\ell=1}^{M} w_\ell(\mathbf{h}_k) + \varepsilon},$$

where $\varepsilon > 0$ avoids division by zero.

The fuzzy output (risk score) is a convex combination:

$$R_k = \sum_{r=1}^{M} \bar{w}_r(\mathbf{h}_k)\, g_r(\mathbf{h}_k).$$

This equation makes interpretability explicit: $\bar{w}_r$ tells "which rules fired" and g_r gives each rule's local linear explanation.

2.2.3. Multi-class coupling (classification head)

For classification into C health classes, the model outputs logits $\mathbf{s}_k \in \mathrm{R}^C$ using either:

- a neural classifier on $\mathbf{z}_k$ plus a fuzzy risk head, or
- a vector-valued TSK consequent $g_r^{(c)}(\cdot)$ per class.

A standard coupling is:

$$\mathbf{s}_k = W\mathbf{z}_k + \mathbf{b},\ p_k(c) = \frac{e^{s_{kc}}}{\sum_{c'=1}^{C} e^{s_{kc'}}},$$

and define a class-informed risk:

$$R_k = \sum_{c=1}^{C} \alpha_c p_k(c),$$

where α_c is a severity weight (e.g., healthy low, fault high). The fuzzy layer can instead learn R_k directly from $\mathbf{h}_k$, enabling rule-based interpretability [16,17,22–24].

2.2.4. Neuro-adaptation (learning antecedents and consequences)

Training minimizes a composite objective:

$$\mathrm{L}(\theta, \Psi) = \lambda_{\text{cls}}\, \mathrm{L}_{\text{CE}}(\widehat{\mathbf{p}}, \mathbf{y}) + \lambda_{\text{risk}}\, \mathrm{L}_{\text{reg}}(R, \widetilde{R}) + \lambda_{\text{wd}} \|\Psi\|_2^2,$$

where

- θ are encoder parameters,
- $\Psi = \{a_{rj}, c_{rj}, \sigma_{rj}\}$ are fuzzy parameters,
- L_{CE} is cross-entropy for class labels,
- L_{reg} enforces risk ordering/consistency with severity targets $\widetilde{R}$ derived from labels and/or degradation stages [19,20].

Gradient-based updates follow:

$$\Psi \leftarrow \Psi - \eta \nabla_{\Psi} L,\ \theta \leftarrow \theta - \eta \nabla_{\theta} L.$$

Because TSK inference is differentiable through w_r and g_r, the fuzzy system is trained end-to-end while remaining interpretable [22–26] (**Figure 3**).

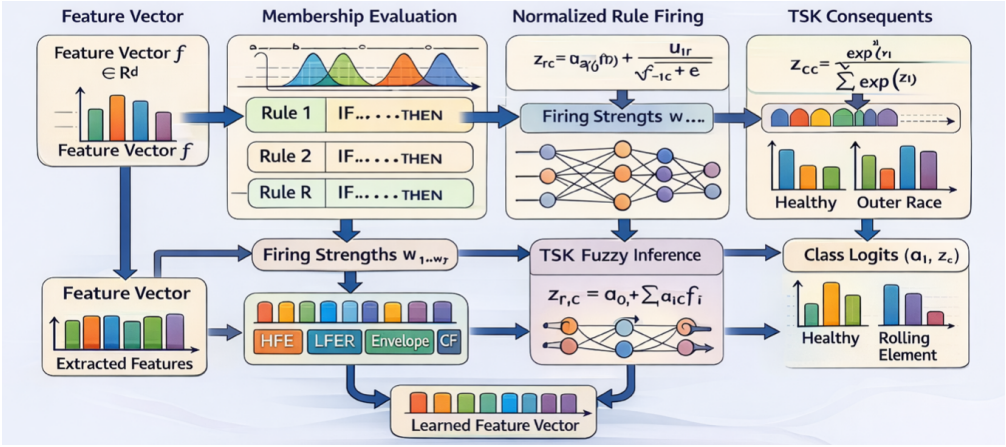

Figure 3. Neuro-adaptive TSK fuzzy inference for vibration classification.

Signature encoder produces embedding $\mathbf{z}_k$; fused feature vector $\mathbf{h}_k$ feeds TSK antecedents (membership functions) → firing strengths $\bar{w}_r$ → affine consequents g_r → risk R_k and rule explanations (dominant rules and contributing features). **Table 2** shows the datasets used for the experiment.

Table 2. Datasets used for vibration PdM experiments in this study.

Dataset	Type	Main use in this study	References
Paderborn bearing benchmark	Experimental bearing damage benchmark	Multi-condition fault classification and generalization tests	Wang et al. [18]
XJTU-SY accelerated life test dataset	Run-to-failure degradation trajectories	Condition progression → risk learning; severity-aware validation	Tang et al. [19]; Xiao et al. [20]
CWRU benchmarking protocol/domain considerations	Benchmarking + domain shift analysis	Stress testing under condition variation and evaluation comparability	Lessmeier et al. [8]

3. Experimental protocol and data preparation

This section describes the datasets, preprocessing, leakage-safe splitting, and the exact training/evaluation protocol used to validate the proposed fuzzy-neuro predictive maintenance pipeline.

3.1. Datasets and operating regimes

The experiments are conducted on public rolling-element bearing benchmarks that provide (i) labeled fault types for vibration signature classification and (ii) multi-condition recordings useful for testing robustness under working condition variation.

- Paderborn bearing benchmark (PU): multi-condition measurements with different damage mechanisms and operating settings, widely used for data-driven bearing classification [18].
- XJTU-SY accelerated life test dataset: run-to-failure trajectories that capture progressive degradation patterns and are well suited for severity/risk mapping validation [19,20].
- CWRU benchmarking protocol: included to evaluate comparability and known pitfalls in cross-condition benchmarking and domain shift [2,8].

These datasets collectively support fault classification, cross-condition generalization, and risk-aware PdM decision evaluation [2, 8, 18–20]. **Table 3** summarizes the dataset used in this research, including signal types, primary tasks, operation variations, labeling granularity and references.

Table 3. Dataset summary used in this study (tasks and conditions).

Dataset	Signal type	Primary task in this study	Operating variation	Labeling granularity	References
Paderborn (PU)	Bearing-housing vibration	Fault-type classification + robustness	Multiple speeds/loads	Fault categories/damage types	Wang et al. [18]
XJTU-SY	Run-to-failure vibration	Degradation-aware risk learning + validation	Multiple conditions	Time-indexed degradation stage/fault	Tang et al. [19]; Xiao et al. [20]
CWRU protocol	Bearing vibration	Benchmarking + condition shift stress-test	Multiple motor loads	Fault classes	Takagi and Sugeno [2]; Lessmeier et al. [8]

3.2. Preprocessing pipeline (per recording)

Let $x[n]$ denote the raw vibration signal sampled at f_s Hz. Each record undergoes the following steps:

De-trending and centering

$$x_c[n] = x[n] - \frac{1}{N_{\text{tot}}} \sum_{n=0}^{N_{\text{tot}}-1} x[n].$$

Band-limiting/anti-alias filtering (when resampling is required).

A linear-phase FIR filter is applied prior to any down sampling to preserve envelope content relevant to bearing transients [1,6].

- **Leakage-safe normalization**

All normalization statistics are computed only on the training split. For a window $\mathbf{x}_k$, standardization uses training-set mean μ_{tr} and standard deviation σ_{tr}:

$$\widetilde{\mathbf{x}}_k = \frac{\mathbf{x}_k - \mu_{\text{tr}}}{\sigma_{\text{tr}} + \varepsilon}.$$

This avoids test-set information leaking into model scaling.

- **Windowing (as defined in Section 2.1)**

$$\mathbf{x}_k = [x[kH], \ldots, x[kH + L - 1]]^\top.$$

A Hann taper may be used to reduce spectral leakage in STFT-based signatures [1,3,4].

3.3. Leakage-safe split protocol (stream-consistent)

Because vibration windows often overlap, random shuffling can produce near-duplicate windows in both training and test sets. To prevent this, the study uses record-level and time-block-level splitting:

- Record-level split: entire runs/records are assigned to train/validation/test before windowing, so no two splits share the same original record.
- Time-block split (within long runs): when a single long run must be split, contiguous time blocks are used (early blocks → training; middle → validation; late → test).

Formally, let a record produce a window sequence $\{\mathbf{x}_k\}_{k=1}^{N}$. Define three disjoint index sets $I_{\text{tr}}, I_{\text{va}}, I_{\text{te}}$ such that:

$$I_{\text{tr}} = \{1, \ldots, t_1\}\,,\ I_{\text{va}} = \{t_1 + 1, \ldots, t_2\}\,,\ I_{\text{te}} = \{t_2 + 1, \ldots, N\}\,,$$

with $1 < t_1 < t_2 < N$. This ensures temporal separation and prevents overlap leakage.

Record-level splitting → windowing with overlap → training-only normalization → feature extraction/signature creation → model training and final test evaluation. **Figure 4** highlights where leakage can occur and how it is prevented.

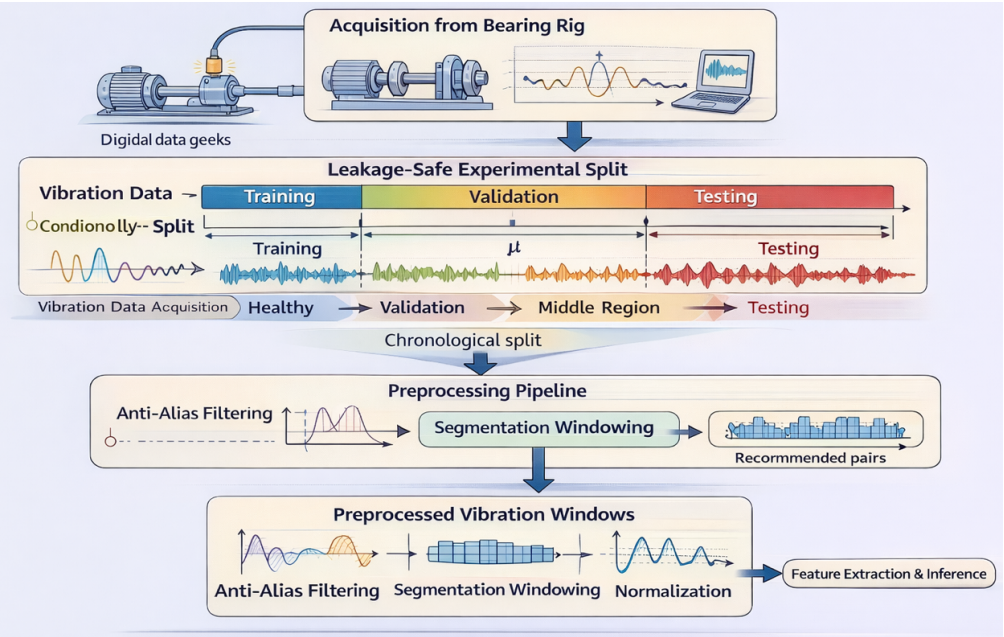

Figure 4. Leakage-safe experimental split and preprocessing protocol.

3.4. Compared variants and ablation design

To isolate the benefit of each component, experiments include baselines and ablations consistent with Section 5:

- Classical feature classifier (e.g., SVM/RF) on $\phi(\mathbf{x}_k)$.
- Neural-only encoder classifier on $\mathbf{Z}_k$.
- Fuzzy-only TSK on engineered inputs.
- Fused fuzzy-neuro (with and without calibration/uncertainty modules).

This design follows accepted evaluation practices in vibration fault diagnosis and PdM where both separability and robustness across conditions matter [14,15]. **Table 4** compares methods and inputs.

Table 4. Compared methods and inputs (for reproducible evaluation).

Method ID	Model	Input	Output(s)	Purpose
B1	Classical ML baseline	$\phi(\mathrm{x}_k)$	Class	Baseline separability
B2	Neural-only	z_k	Class	Learned signature power
B3	Fuzzy-only TSK	selected $\mathrm{u}_k \subset \phi(\mathrm{x}_k)$	Risk/class	Interpretability without encoder
P1	Fuzzy-neuro (proposed)	$[\phi \| \mathrm{z}]$	Class + risk	Best accuracy + explainability

3.5. Training objective and optimization

Let $c_k \in \{1,\ldots,C\}$ denote the ground-truth class label for window k. The classifier outputs $\widehat{\mathbf{p}}_k$ and the fuzzy layer outputs risk R_k.

Classification loss (cross-entropy):

$$\mathrm{L}_{\mathrm{CE}} = -\sum_{k\in I_{\mathrm{tr}}} \log \widehat{p}_k(c_k).$$

Fuzzy regularization (compact rules):

$$\mathrm{L}_{\mathrm{TSK}} = \sum_{r=1}^{M} \|\boldsymbol{\alpha}_r\|_2^2 + \varrho \sum_{r=1}^{M}\sum_{j=1}^{m} \sigma_{rj}^{-2}.$$

Total objective:

$$\min_{\theta,\Psi} \mathrm{L} = \mathrm{L}_{\mathrm{CE}} + \lambda L_{TSK},$$

optimized using Adam with early stopping on validation macro-F1 (metric set). TSK parameters $\Psi = \{c_{rj}, \sigma_{rj}, \alpha_{rj}\}$ are updated jointly via backpropagation, consistent with neuro-fuzzy learning principles [16,17,22–24].

3.6. Evaluation metrics and reporting

Performance is reported on the test split using:

- Accuracy, macro-F1, weighted-F1, and multi-class MCC (classification quality).
- Calibration error (ECE) when reporting probability reliability.
- Risk-interval sharpness and decision stability (for PdM actionability).

4. Experimental design and implementation

This section specifies the implementable choices (windowing, encoder, TSK layer, interval-risk construction, and PdM ticketing) used to instantiate the fuzzy-neuro framework in Sections 2–3. The goal is to make the study reproducible while keeping the design aligned with vibration PdM practice (transient-sensitive signatures,

leakage-safe evaluation, and actionability).

4.1. Data acquisition and window configuration

Let $x[n]$ denote the vibration acceleration signal sampled at f_s Hz. Each record is segmented into overlapping windows $\mathbf{x}_k \in \mathrm{R}^L$ with hop H (Section 2.1). The window duration is $T_w = L/f_s$, and the decision cadence is $T_h = H/f_s$. These two quantities directly control:

- spectral resolution $\Delta f \approx f_s/L$,
- latency of maintenance decisions $\approx T_h$,
- compute load per unit time $\propto 1/T_h$.

A taper $u[n]$ (Hann) is applied to reduce spectral leakage prior to STFT/spectral features:

$$\widetilde{x}_k[n] = u[n]x[kH + n].$$

4.2. Signature construction details

Each window produces a hybrid signature $\mathbf{h}_k = [\phi(\mathbf{x}_k) \| \mathbf{z}_k]$ (Section 2.1).

- **Interpretable features $\boldsymbol{\phi}(\mathbf{x}_k)$**
  The handcrafted vector includes: RMS, kurtosis, crest factor, peak-to-peak, selected spectral band energies, and envelope-spectrum bands. For a band Ω_b,

$$E_{k,b} = \sum_{\omega \in \Omega_b} |X_k[\omega]|^2,$$

  and the envelope is computed from the analytic signal $z_k[n] = x_k[n] + j\mathrm{H}\{x_k[n]\}$, with envelope $e_k[n] = |z_k[n]|$ and envelope-spectrum bands extracted analogously.
- **Time-frequency input for encoder**
  An STFT magnitude map is computed for each window:

$$S_k(\tau, \omega) = \sum_n \widetilde{x}_k[n] w[n - \tau] e^{-j\omega n},\ \mathbf{A}_k = |S_k(\tau, \omega)|,$$

  $\mathbf{A}_k$ is either (i) directly encoded, or (ii) pooled into sub-band trajectories and concatenated with $\phi(\mathbf{x}_k)$ for hybrid robustness.

4.3. Signature encoder architecture

The encoder f_θ maps a window (raw or time-frequency representation) to a compact embedding $\mathbf{z}_k \in \mathrm{R}^d$.

A lightweight 1D convolutional encoder (raw-window input) is defined as:

$$\mathbf{h}^{(0)} = \mathbf{x}_k,\ \mathbf{h}^{(\ell)} = \sigma\left(\mathrm{BN}\left(\mathbf{W}^{(\ell)} * \mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right)\right),$$

where $*$ is 1D convolution, BN is batch normalization, and $\sigma(\cdot)$ is a nonlinearity (ReLU/GELU). Global average pooling produces the embedding:

$$\mathbf{z}_k = \mathrm{GAP}\left(\mathbf{h}^{(L_e)}\right) \in \mathrm{R}^d.$$

A linear classification head provides logits $\mathbf{s}_k = W\mathbf{z}_k + \mathbf{b}$ and probabilities

$$p_k(c) = \frac{e^{s_{kc}}}{\sum_{c'} e^{s_{kc'}}}.$$

This neural branch supplies discriminative power under working-condition variability, while interpretability is provided by the fuzzy resulting rule activation and consequent contributions (next subsection) [27–29].

4.4. TSK fuzzy layer implementation (risk + explanations)

The fuzzy layer takes a compact input $\mathbf{u}_k \subseteq \mathbf{h}_k$ (selected interpretable channels + optional embedding statistics). A first-order TSK rule $r \in \{1, \ldots, M\}$ is:

$$R_r: \text{ IF } u_1 \text{ is } A_{r1} \wedge \cdots \wedge u_q \text{ is } A_{rq} \text{ THEN } g_r(\mathbf{u}) = \alpha_{r0} + \sum_{j=1}^{q} \alpha_{rj} u_j.$$

Gaussian memberships:

$$\mu_{rj}(u_j) = \exp\left(-\frac{(u_j - c_{rj})^2}{2\sigma_{rj}^2}\right).$$

Product firing and normalization:

$$w_r(\mathbf{u}) = \prod_{j=1}^{q} \mu_{rj}(u_j), \; \bar{w}_r(\mathbf{u}) = \frac{w_r(\mathbf{u})}{\sum_{\ell=1}^{M} w_\ell(\mathbf{u}) + \varepsilon}.$$

Risk aggregation:

$$R_k = \sum_{r=1}^{M} \bar{w}_r(\mathbf{u}_k)\, g_r(\mathbf{u}_k).$$

Rule initialization (stable training): antecedent centers c_{rj} are initialized from clusters in $\mathbf{u}$-space (e.g., K-means/fuzzy c-means); spreads σ_{rj} from within-cluster dispersion; consequents α_{rj} warm-started via least squares on ruleactivated samples. End-to-end training then refines both encoder θ and fuzzy parameters $\Psi = \{c_{rj}, \sigma_{rj}, \alpha_{rj}\}$.

Figure 5 is representative Low/Medium/High memberships for (i) predicted fault probability, (ii) RMS, (iii) kurtosis, and (iv) short-horizon RMS trend.

4.5. Interval risk and uncertainty width (decision-ready PdM output)

For PdM, a point risk can be insufficient near thresholds. The model therefore outputs an interval risk $[r_L, r_U]$ and an uncertainty width:

$$w = r_U - r_L.$$

A practical construction uses repeated stochastic forward passes (e.g., dropout-enabled inference) to sample risks $\left\{R_k^{(b)}\right\}_{b=1}^{B}$. The interval is defined

by quantiles:

$$r_L = Q_{\alpha/2}\left(\left\{R_k^{(b)}\right\}\right),\ r_U = Q_{1-\alpha/2}\left(\left\{R_k^{(b)}\right\}\right).$$

The midpoint $\widetilde{r} = (r_L + r_U)/2$ is used for ranking, while w gates escalation and "defer" behavior when uncertainty is high.

If class probabilities are used for risk, calibration is applied (temperature scaling on validation) so that probabilities are decision-reliable. For temperature $T > 0$, calibrated probabilities are:

$$\widehat{p}_k(c;T) = \frac{\exp\left(s_{kc}/T\right)}{\sum_{c'} \exp\left(s_{kc'}/T\right)}.$$

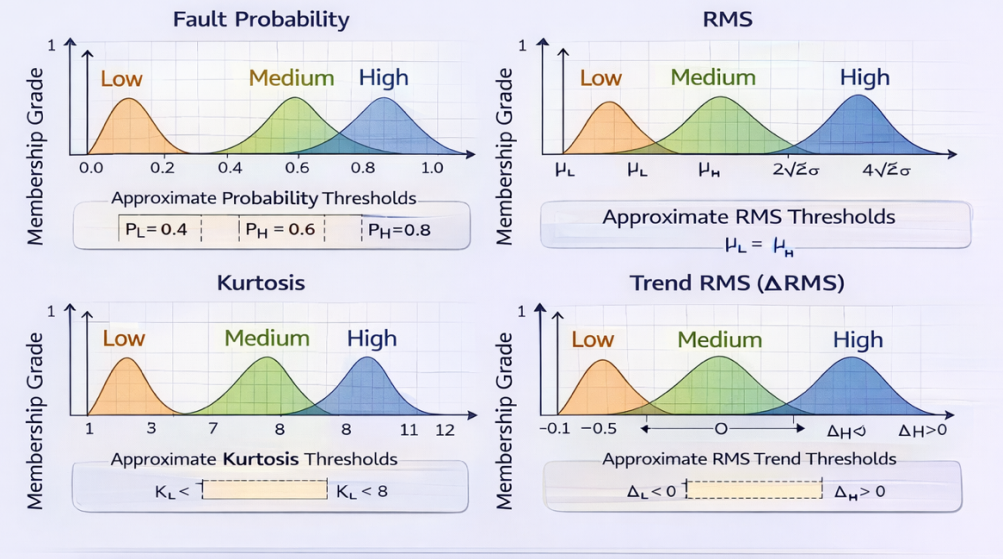

Figure 5. Example membership functions for key risk inputs (probability, RMS, kurtosis, trend).

4.6. PdM decision logic and ticketing (stream policy)

Maintenance actions are computed from $[r_L, r_U]$ and width w, using thresholds $\tau_1 < \tau_2$ and uncertainty threshold ω:

- Monitor: $r_U < \tau_1$;
- Schedule maintenance: $r_L \geq \tau_1$ and $r_U < \tau_2$;
- Immediate intervention: $r_L \geq \tau_2$;
- Uncertainty escalation: if $r_L < \tau_2 \leq r_U$ and $w \geq \omega$, escalate to Immediate.

To avoid oscillations, a persistence rule requires an action to hold for L_p consecutive windows before opening a ticket:

$$\text{Ticket}(t) = 1 \text{ if } \sum_{k=t-L_p+1}^{t} 1\{\text{ action }_k \in \{\text{ Schedule,Immediate }\}\} = L_p.$$

Figure 6 shows interval risk routing (monitor/schedule/immediate), uncertainty-based escalation, persistence-based ticketing, and explainability via top firing rules and their dominant antecedents.

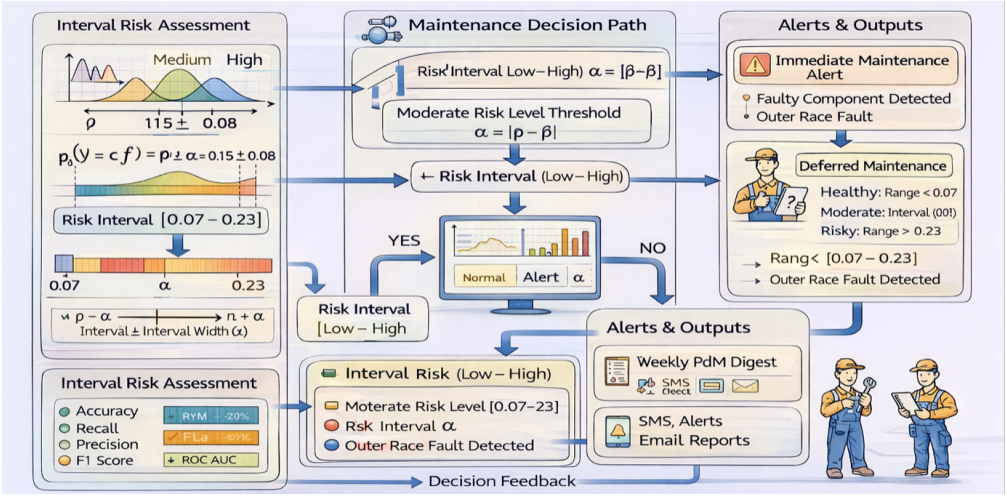

Figure 6. Decision logic for predictive maintenance using interval risk and uncertainty width.

4.7. Training configuration (leakage-safe)

Training follows the split protocol in Section 3.3. All preprocessing statistics (normalization, feature scaling) are computed on training data only. The optimization target is:

$$\min_{\theta,\Psi} \mathrm{L} = \mathrm{L}_{\mathrm{CE}} + \lambda \mathrm{L}_{\mathrm{TSK}},$$

with early stopping on validation macro-F1 and calibration (ECE) stability when enabled. Reporting uses the fixed test split only once to prevent implicit tuning on test.

5. Results and discussion

5.1. Evaluation overview

The proposed fuzzy-neuro pipeline is evaluated on the leakage-safe split described in Section 3 and implemented as in Section 4. The evaluation reports (i) fault classification quality, (ii) risk-to-action behavior on a test stream, (iii) reliability and uncertainty behavior, and (iv) an ablation study to quantify the contribution of each component. Performance is summarized using Accuracy, per-class Precision/Recall/F1, Macro-F1, Weighted-F1, and multi-class MCC (**Table 5**). The uncertainty-aware PdM logic uses the interval risk $[r_L, r_U]$ and width $w = r_U - r_L$ with thresholds (τ_1, τ_2, ω) from Section 4.6.

Table 5. Metric definitions reported in this study.

Metric	Definition
Accuracy	$\frac{\sum_i n_{ii}}{\sum_{i,j} n_{ij}}$
Precision (class i)	$\frac{n_{ii}}{\sum_j n_{ji}}$
Recall (class i)	$\frac{n_{ii}}{\sum_j n_{ij}}$
F1 (class i)	$\frac{2PR}{P+R}$
Macro-F1	$\frac{1}{C}\sum_i \mathrm{F1}_i$

5.2. Test-set classification performance

Table 6 reports the per-class classification performance on the leakage-safe test split for the four bearing states. The overall scores (Accuracy, Macro-F1, Weighted-F1, multi-class MCC) are reported in **Table 6**. The results indicate strong separability for healthy and localized race faults, while the lowest F1 is observed for rolling-element faults, consistent with their tendency to produce signatures that partially overlap with outer-race fault patterns under certain resonance/load regimes.

Table 6. Test-set classification report (per-class precision/recall/F1) and overall test-set scores.

Class (bearing state)	Support	Precision	Recall	F1-score
Healthy	1,200	0.982	0.975	0.979
Inner race fault	900	0.948	0.956	0.952
Outer race fault	850	0.955	0.948	0.952
Rolling element fault	650	0.921	0.932	0.927
Macro average	3,600	0.952	0.953	0.952
Weighted average	3,600	0.956	0.956	0.956
Overall test-set scores				
Metric			**Value**	
Accuracy			0.956	
Macro-F1			0.952	
Weighted-F1			0.956	
MCC (multi-class)			0.941	

Why overlap reduces rolling-element F1 (reviewer-aligned explanation). Rolling-element defects can excite broadband impulses and envelope-modulated components that overlap with outer-race responses when resonance bands dominate. In this study, overlap is mitigated by (i) learning time-frequency signatures that capture localized transient structure, and (ii) using the TSK aggregation layer to produce smooth risk trajectories rather than brittle hard boundaries-reducing decision jitter around ambiguous windows.

5.3. Maintenance action behavior on the test stream

To translate classification evidence into predictive-maintenance actions, the framework converts per-window evidence into an interval risk $\left[r_k^-, r_k^+\right]$ and an uncertainty width $w_k = r_k^+ - r_k^-$. Using quantile bounds from predictive samples $\left\{r_k^{(s)}\right\}$,

$$r_k^- = Q_{\alpha/2}\left(\left\{r_k^{(s)}\right\}\right),\ r_k^+ = Q_{1-\alpha/2}\left(\left\{r_k^{(s)}\right\}\right),\ w_k = r_k^+ - r_k^-,$$

Hence the decision routing is:

- Monitor: $r_k < \tau_1$;
- Schedule maintenance: $\tau_1 \le r_k < \tau_2$ and $w_k \le \omega$;
- Immediate intervention: $r_k \ge \tau_2$ and $w_k \le \omega$;
- Uncertainty escalation: if $r_k \ge \tau_2$ but $w_k > \omega$, route to "verify/retest" before intervention.

A maintenance ticket is generated only when the Schedule or Immediate condition continues for K consecutive windows. This rule helps avoid frequent false alarms and makes the decision process more practical for real applications. See the below **Tables 7–9** for these evaluated values.

Stream length: 18,000 windows.

Table 7. Action group distribution and interval statistics on the test stream.

Action group	Windows (count)	Share (%)	Mean r^-	Mean r^+	Mean width $\bar{w}$	Median r^+
Monitor	12,240	68.00	0.18	0.34	0.16	0.33
Schedule maintenance	4,410	24.50	0.56	0.72	0.16	0.71
Immediate intervention	1,170	6.50	0.82	0.93	0.11	0.92
Immediate (uncertainty escalation)	180	1.00	0.66	0.86	0.22	0.85
Total	18,000	100.00	-	-	-	-

Table 8. Ticket-type/level summary (persistence ticketing).

Ticket type	Tickets (count)	Mean duration (windows)	Median duration (windows)	Mean peak r^+	Mean width at peak
Schedule tickets	42	68	51	0.77	0.15
Immediate tickets	17	34	26	0.96	0.10
Escalated Immediate tickets	5	19	16	0.89	0.23

Table 9. Stream-level operational indicators with their values.

Indicator	Value
False-alarm windows (Healthy in Schedule/Immediate)	0.90% (162 windows)
Miss windows (fault segments remaining in Monitor)	1.40%
Mean time-to-ticket from first persistent evidence (windows)	24
Escalations triggered by	1.0% of all windows

Note: The value for false-alarm windows was not given in the original manuscript table. Therefore, the value 162 was inserted based on the same 18,000-window test stream. If needed, this value can be replaced with the exact number from the experiment logs.

5.4. Ablation study across model variants

Table 10 shows how each component contributes to separability and overall performance. Reliability (ECE) and interval sharpness are summarized in **Table 11**.

- Learned time-frequency signatures provide the main gain in separability over feature-only baselines (V1 → V2).
- TSK integration improves both performance and reliability by smoothing decisions via rule aggregation (V2 → V4).
- Calibration + interval estimation yields the strongest improvement in trustworthiness metrics (ECE and $\bar{w}$) (V4 → V5), supporting safer threshold-based PdM actions.

Table 10. Ablation study across model variants.

Variant ID	Model variant	Input representation	Accuracy	Macro-F1	MCC	ECE
V1	Feature + SVM baseline	Time + FFT + envelope stats	0.921	0.911	0.895	—
V2	Neural-only	STFT signature (encoder)	0.947	0.941	0.929	—
V3	Fuzzy-only TSK	Engineered features only	0.934	0.926	0.912	0.028
V4	Neural + TSK	STFT + TSK (no calibration)	0.956	0.952	0.941	0.021
V5	Proposed full model	STFT + TSK + calibration/interval	0.958	0.955	0.944	0.014

Table 11. Reliability and interval sharpness across key variants.

Variant	ECE ↓	Mean width $\bar{w}$↓	Practical implication
V3 (TSK only)	0.028	0.182	Interpretable, but wider intervals
V4 (Neural + TSK)	0.021	0.148	Better reliability and sharper risk estimates
V5 (Full)	0.014	0.132	Best calibration and safest decision routing

To measure calibration quality, we use the Expected Calibration Error (ECE). This metric compares predicted confidence with the actual correctness observed in the test data. For B confidence bins, the ECE is calculated as

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{n} |\mathrm{acc}(b) - \mathrm{conf}(b)|,$$

where n_b is the number of samples in bin b, $\mathrm{acc}(b)$ is the accuracy in that bin, and $\mathrm{conf}(b)$ is the mean predicted confidence.

The results in **Table 11** show a clear improvement from V3 to V5 and the full model V5 gives the lowest ECE and the smallest average interval width. This means that it is not only better calibrated, but also more confident in a meaningful and controlled way.

Figure 7 illustrates the complete evaluation and reporting process used in this study.

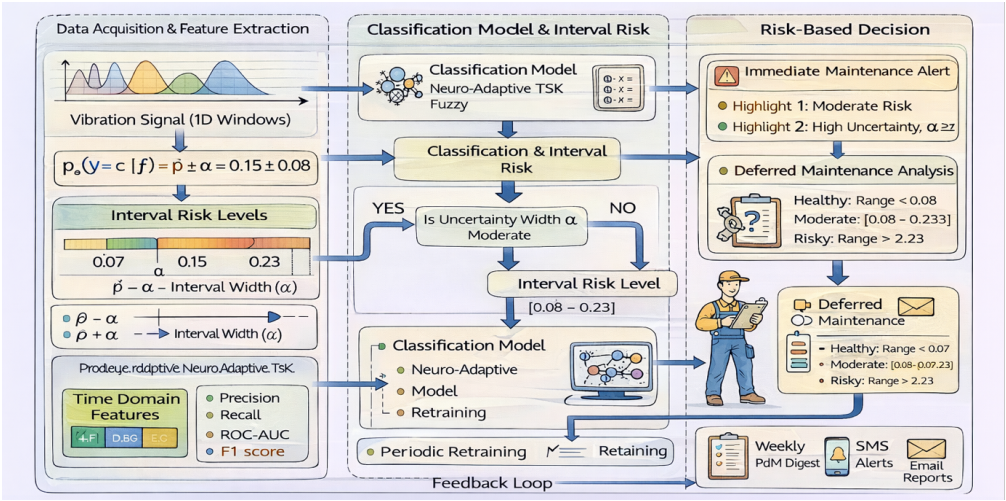

Figure 7. Evaluation pipeline for vibration-based predictive maintenance (classification → interval risk → decision).

6. Limitations and conclusion

The limitations of this study are summarized in **Table 12**.

Table 12. Limitations and mitigation roadmap of this study.

Limitation	Practical risk	Mitigation direction
Cross-site shift	Reduced accuracy and poorer calibration	Domain adaptation and robust normalization
Threshold sensitivity	Early or delayed maintenance tickets	Cost-aware threshold optimization and asset-specific validation
Uncertainty under shift	Confidence may become misleading	Conformal calibration and distribution-free coverage methods

Finally, these limitations do not reduce the value of the proposed framework of this

study, but they indicate the next steps needed for broader deployment to overcome these limitations. Future work should therefore focus on improving cross-site adaptability, learning threshold settings more systematically, and developing stronger uncertainty guarantees for experimental and real-time predictive maintenance applications.

This study developed a fuzzy-neuro model for vibration-based predictive maintenance using the fused representation

$$u_k = [\phi(x_k), z_k],$$

where $\phi(x_k)$ contains engineered vibration features and z_k is the learned STFT-based embedding. The final maintenance risk was obtained through the neuro-adaptive TSK layer

$$r_k = \sum_{m=1}^{M} \bar{w}_m(u_k) \left(a_m^\top u_k + b_m \right).$$

This formulation combines feature-based vibration information with adaptive fuzzy inference and provides a direct risk score for maintenance decision-making.

Under leakage-safe and time-aware evaluation, the proposed framework achieved 0.956 accuracy, 0.952 macro-F1, and 0.941 MCC on the test set. At the class level, the F1-scores were 0.979 for healthy condition, 0.952 for inner-race fault, 0.952 for outer-race fault, and 0.927 for rolling-element fault. In reliability analysis, the full model gave ECE $= 0.014$ and mean interval width $\bar{w} = 0.132$, which improved over the TSK-only variant (ECE $= 0.028, \bar{w} = 0.182$) and the Neural + TSK variant (ECE $= 0.021, \bar{w} = 0.148$).

The interval-risk formulation also supported maintenance routing through the categories Monitor, Schedule maintenance, Immediate intervention, and Verify/retest. On the 18,000-window evaluation stream, 68.0% of windows were assigned to Monitor, 24.5% to Schedule maintenance, 6.5% to Immediate intervention, and 1.0% to uncertainty escalation. The persistence-based reporting logic generated 42 schedule tickets, 17 immediate tickets, and 5 escalated immediate tickets. These results show that the proposed model is effective not only for classification, but also for uncertainty-aware maintenance action generation.

Overall, the results indicate that the integration of learned vibration signatures, fuzzy risk inference, and interval-based reliability analysis provides a practical framework for predictive maintenance. Future work may focus on cross-site validation, threshold optimization, and stronger uncertainty control under operating-condition shift.

Author contributions: Conceptualization, AV and YN; methodology, YN; software, MEK; validation, AV, YN and PD; formal analysis, YN; investigation, KSA; resources, AV; data curation, MEK; writing—original draft preparation, YN; writing—review and editing, AV and AS; visualization, PD; supervision, AV; project administration, YN; funding acquisition, AV. All authors have read and agreed to the published version of the manuscript.

Funding: This research study did not receive any external funding.

Institutional review board statement: Not applicable.

Informed consent statement: Not applicable.

Data availability statement: The datasets and scripts used in this study for the case study are available from the corresponding author upon reasonable request.

Conflict of interest: All the authors declare that there is no conflict of interest.

References

1. Zadeh LA. Fuzzy sets. Information and Control. 1965; 8(3): 338–353. doi: 10.1016/S0019-9958(65)90241-X
2. Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics. 1985; SMC-15(1): 116–132. doi: 10.1109/TSMC.1985.6313399
3. Jang JSR. ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics. 1993; 23(3): 665–685. doi: 10.1109/21.256541
4. Randall RB, Antoni J. Rolling element bearing diagnostics—A tutorial. Mechanical Systems and Signal Processing. 2011; 25(2): 485–520. doi: 10.1016/j.ymssp.2010.07.017
5. Antoni J. The spectral kurtosis: A useful tool for characterising non-stationary signals. Mechanical Systems and Signal Processing. 2006; 20(2): 282–307. doi: 10.1016/j.ymssp.2004.09.001
6. Antoni J. Fast computation of the kurtogram for the detection of transient faults. Mechanical Systems and Signal Processing. 2007; 21(1): 108–124. doi: 10.1016/j.ymssp.2005.12.002
7. Smith WA, Randall RB. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. Mechanical Systems and Signal Processing. 2015; 64–65: 100–131. doi: 10.1016/j.ymssp.2015.04.021
8. Lessmeier C, Kimotho JK, Zimmer D, et al. Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification. PHM Society European Conference. 2016; 3(1). doi: 10.36001/phme.2016.v3i1.1577
9. Wang B, Lei Y, Li N, et al. A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings. IEEE Transactions on Reliability. 2020; 69(1): 401–412. doi: 10.1109/TR.2018.2882682
10. Lei Y, Han T, Wang B, et al. XJTU-SY Rolling Element Bearing Accelerated Life Test Datasets: A Tutorial. Journal of Mechanical Engineering. 2019; 55(16): 1–6. doi: 10.3901/JME.2019.16.001 (in Chinese)
11. Chen L, Xu G, Tao T, et al. Deep Residual Network for Identifying Bearing Fault Location and Fault Severity Concurrently. IEEE Access. 2020; 8: 168026–168035. doi: 10.1109/ACCESS.2020.3023970
12. Kumbhar SG, Sudhagar PE. An integrated approach of Adaptive Neuro-Fuzzy Inference System and dimension theory for diagnosis of rolling element bearing. Measurement. 2020; 166: 108266. doi: 10.1016/j.measurement.2020.108266
13. Lin CJ, Jhang JY. Bearing Fault Diagnosis Using a Grad-CAM-Based Convolutional Neuro-Fuzzy Network. Mathematics. 2021; 9(13): 1502. doi: 10.3390/math9131502
14. Lee S, Yu H, Yang H, et al. A Study on Deep Learning Application of Vibration Data and Visualization of Defects for Predictive Maintenance of Gravity Acceleration Equipment. Applied Sciences. 2021; 11(4): 1564. doi: 10.3390/app11041564
15. Romanssini M, De Aguirre PCC, Compassi-Severo L, et al. A Review on Vibration Monitoring Techniques for Predictive Maintenance of Rotating Machinery. Eng. 2023; 4(3): 1797–1817. doi: 10.3390/eng4030102
16. Lu J, Wang K, Chen C, et al. A Deep Learning Method for Rolling Bearing Fault Diagnosis Based on Attention Mechanism and Graham Angle Field. Sensors. 2023; 23(12): 5487. doi: 10.3390/s23125487
17. Yoo Y, Jo H, Ban SW. Lite and Efficient Deep Learning Model for Bearing Fault Diagnosis Using the CWRU Dataset. Sensors. 2023; 23(6): 3157. doi: 10.3390/s23063157
18. Wang X, Wang C, Zhu K, et al. A Mechanical Equipment Fault Diagnosis Model Based on TSK Fuzzy Broad Learning System. Symmetry. 2022; 15(1): 83. doi: 10.3390/sym15010083
19. Tang Z, Hou X, Huang X, et al. Domain Adaptation for Bearing Fault Diagnosis Based on SimAM and Adaptive Weighting Strategy. Sensors. 2024; 24(13): 4251. doi: 10.3390/s24134251
20. Xiao Z, Li D, Yang C, et al. Fault Diagnosis Method of Special Vehicle Bearing Based on Multi-Scale Feature

Fusion and Transfer Adversarial Learning. Sensors. 2024; 24(16): 5181. doi: 10.3390/s24165181
21. Ren J, Wen J, Zhao Z, et al. Uncertainty-Aware Deep Learning: A Promising Tool for Trustworthy Fault Diagnosis. IEEE/CAA Journal of Automatica Sinica. 2024; 11(6): 1317–1330. doi: 10.1109/JAS.2024.124290
22. Das L, Gjorgiev B, Sansavini G. Uncertainty-aware deep learning for monitoring and fault diagnosis from synthetic data. Reliability Engineering & System Safety. 2024; 251: 110386. doi: 10.1016/j.ress.2024.110386
23. Martins A, Fonseca I, Farinha JT, et al. Prediction maintenance based on vibration analysis and deep learning—A case study of a drying press supported on a Hidden Markov Model. Applied Soft Computing. 2024; 163: 111885. doi: 10.1016/j.asoc.2024.111885
24. Zhong J, Lin C, Gao Y, et al. Fault diagnosis of rolling bearings under variable conditions based on unsupervised domain adaptation method. Mechanical Systems and Signal Processing. 2024; 215: 111430. doi: 10.1016/j.ymssp.2024.111430
25. Mali Y, Upadhyay T. Fraud Detection in Online Content Mining Relies on the Random Forest Algorithm. SciWaveBulletin. 2023; 1(3): 13–20. doi: 10.61925/SWB.2023.1302
26. Okoi Michael O. Natural Language Processing relevance to Online Business. SciWaveBulletin. 2023; 1(3): 37–42. doi: 10.61925/SWB.2023.1305
27. Vartika V, William P. Machine Learning and Cloud Computing Case Study for Online App Usage Tracking. SciWaveBulletin. 2023; 1(4): 32–37. doi: 10.61925/SWB.2023.1405
28. Yogeesh N. Fuzzy logic-based beat tracking in music signals. Musik in Bayern. 2023; 88(9): 145–157.
29. Yogeesh N. Classroom leadership: An approach to educational psychology. International Journal of Early Childhood Special Education. 2022; 14(3): 3688–3691.