

When Real Faults Are Too Rare to Wait For: A Fault Injection Methodology for Labelled Diagnostic Data

Idriss El-Thalji^{1,2}, Makafui Kwame Botsoe¹

¹Department of Engineering Management, Prince Sultan University, Riyadh, Saudi Arabia

²Department of Mechanical and Structural Engineering and Materials Science, University of Stavanger, Stavanger, Norway

E-mail: ielthalji@psu.edu.sa, makafui.k.botsoe@uis.no

Abstract. Data-driven fault diagnosis is limited by the scarcity of labelled faulty data. Faults are rare, slow to develop or unsafe to reproduce, while healthy records are abundant. Training algorithms only on healthy data, as faulty data do not exist, may enable Anomaly detection, which flags deviations but cannot help with the entire diagnostic process or identify the type, location or severity of the fault. There is limited research work on having a general automated procedure that turns a historical healthy record and prior engineering knowledge into a labelled, validated data set that can be used for full diagnostic training. This paper proposes a framework that injects time-evolving, operating-condition-dependent faults into real healthy signals at the raw-waveform level. It specifies design rules for failure modes, degradation anchors, onset, co-occurring faults and speed coverage, and it releases a dataset only after a reliability gate, together with a validity statement of its scope and assumptions. A five-level ablation is scored against the real degradation of IMS Test. The error against the real curve falls from 15.8% of its range for a random walk to 4.8% for stage structure combined with a snapshot of a real-world curve. No level reproduces the long hidden-degradation plateau of the real bearing, and the assumed failure level changes the error about four-fold, which makes the definition of anchors and scale inputs the main open problem.

1 Introduction

Rotating machinery is increasingly monitored through sensor networks, which has motivated data-driven fault diagnosis in which models learn to detect, classify, and localize faults from measured signals [1]. A well-recognized obstacle is the scarcity of labelled faulty data relative to the abundance of healthy data [2, 3]. Faults in critical assets are rare by design, and letting them develop under instrumented conditions is often unsafe, expensive, or impractical. Anomaly detection, trained on healthy data only [4], can show that something has changed but not the fault type, location, or severity, whereas supervised diagnosis needs exactly the labelled faulty examples that are hardest to obtain. Fault injection bridges this gap by introducing a controlled, labelled fault into otherwise healthy data in order to train the diagnostic and prognostic algorithms, as illustrated in Figure 1.

Generating labelled failure data without waiting for real failures has been pursued in several communities. In software, automotive, and IT systems, fault injection is the established way to produce such data, on hardware-in-the-loop benches or in cloud environments [5, 6, 7, 8]. In

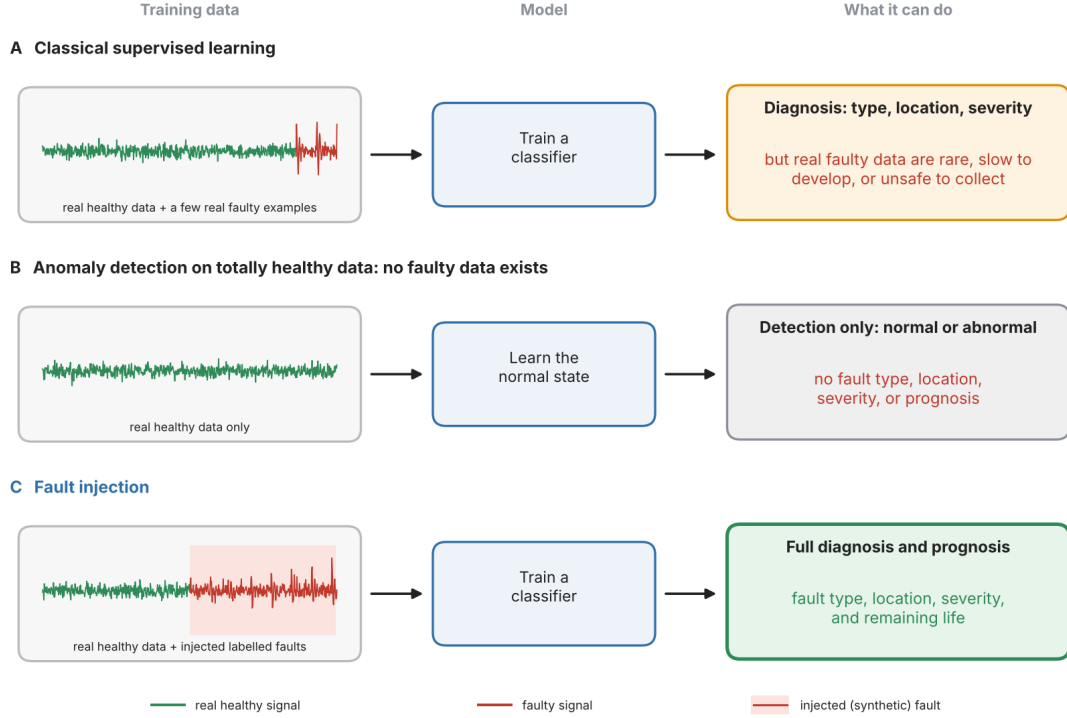


Figure 1: Three approaches to training machine learning algorithms.

rotating machinery and drives, the need has been met by synthetic fault pulse trains added to healthy bearing vibration and aligned with real faults through domain adaptation [9], by physics-based simulation with domain adaptation [10, 11], and by conditional generative adversarial networks that synthesise fault samples at other operating conditions [12]. In prognostics, run-to-failure data come from damage-propagation models and high-fidelity simulators [13, 14] and from physics-constrained generative models [15]. What this work does not appear to offer is a general, automated procedure that takes a historical healthy record and prior engineering knowledge about degradation and returns a labelled, validated dataset whose reliability can be quantified. The pipelines we found are tied to a specific system, describe a fault as a type or severity level instead of a time-evolving process, and report realism qualitatively or through a single downstream score. The design questions any such procedure must answer: how many failure modes, how many injection or anchor points, when faults overlap, how speed varies, and how reliability is tested.

Existing approaches to generating labelled fault data fall into four groups: digital-twin and simulation-based injection, signal synthesis and generative models, fault injection in software, automotive, and IT systems, and synthetic data for rotating machinery and prognostics. Digital twins perturb a calibrated model of the asset and re-simulate it. They can exceed 95% accuracy on real benchmarks when well calibrated, but show a pronounced gap when not [16], and they depend on a faithful model [1]. Signal synthesis interpolates real healthy samples with fault knowledge, but the distribution often deviates from actual faults [17], and conditional GANs correct class imbalance but learn from real fault examples [18]. In software, automotive, and IT systems, injection environments create diagnostic datasets [5, 6, 7] and a failure-prediction thesis stresses the need for a procedure to process injected data [8], but they act on running systems, not on recorded waveforms. In rotating machinery, the closest work adds expert-defined

fault pulse trains to healthy bearing vibration and closes the gap with domain adaptation [9], or simulates faults and adapts them to limited real data [10, 11], and in prognostics, run-to-failure data come from simulators and a physics-informed GAN [13, 14, 15]. In all of them, faults are types or severity levels, and any mismatch is handled after injection, not by calibration.

This paper proposes a framework for the fault injection process to generate a training dataset for diagnostic and prognostic purposes. The framework works on real healthy recordings, treats a fault as a time-evolving, condition-dependent process calibrated against a reference curve, states explicit design rules, and ties dataset release to a reliability gate. To build an automated fault-injection framework, we first test how much prior knowledge the injection requires. We started with three levels of guidance: an unguided random walk, a stage-informed trajectory from a five-stage wear model, and the shape of a real-world degradation curve. Each level injects an outer-race fault at the raw-waveform level into the real healthy record of IMS Test 2, bearing 1, with the same healthy and failure levels, so that only the guidance differs. We then compare each level with the real degradation curve of the same bearing, which none of them uses to set a parameter, and express the error as a percentage of the real range.

2 Framework and design rules

Turning the injection model into a repeatable procedure requires the remaining decisions to be explicit. This section states them as design rules.

2.1 Framework overview

The framework takes six inputs: a historical healthy record; a failure-mode library (physical signature, condition scaling, and injection operator per mode); degradation anchors that define the shape of the primary fault’s severity trajectory; two scale inputs, the healthy level and the failure level, between which that shape is mapped; an operating-condition schedule; and coupling rules stating which modes may co-occur. Four stages follow, with an assumption log alongside. The healthy record is screened for early degradation; a scenario is sampled (onset, stage durations, perturbed anchors, speed profile, secondary faults); the injection engine generates the waveform and calibrates the fault amplitude to the target trajectory; and a reliability gate decides whether the result is released, returning control to scenario sampling with revised anchors or parameters if it fails. Every assumption made at stages 1–3 is written to the log. The output is a labelled dataset with stage, continuous severity, speed, and a flag and severity for every active mode in each window, together with its provenance (configuration, seeds, validation report) and validity statement.

2.2 Design rules

Failure modes come from the asset’s failure-mode and effects analysis and are ranked by criticality (severity, occurrence, detectability). They are included until a user-set fraction of total criticality is covered, and the uncovered fraction is reported. Modes that are easily confused with an included mode, and common confounders, are added regardless of rank. A mode enters the library only if its signature, condition scaling and operator are known. For a rolling-element bearing the library holds the four defect locations and the shaft modes imbalance, misalignment and looseness, and this paper implements the outer-race mode with imbalance and misalignment as low-level confounders. Degradation anchors are pairs of monitoring time and target indicator value joined by shape-preserving interpolation, with one anchor per turning point plus the start (three for a monotone wear curve, nine for this paper’s reference curve). The smallest set beyond which calibration error and task performance stop changing is adopted, and anchor times and amplitudes are perturbed within bounds to prevent a model from memorising one

trajectory. Onset and stage durations are drawn from distributions, and the healthy record is screened against control limits from its own early portion, because an undetected incipient fault would contaminate every label (here the carrier is assumed healthy). Co-occurring faults get their own trajectories and independently drawn onsets, are restricted to plausible pairs from a compatibility matrix, are labelled multi-label, and are superposed additively unless an interaction is known. Operating conditions combine set points, drift and transients, with modes scaled by their physics, coverage judged by a leave-one-speed-out test, and other speeds obtained for fixed-speed data by resampling the time axis, which shifts frequencies but does not reproduce rotordynamic changes. Table 1 lists each framework component, its design rule, and its status in this study.

Table 1: Framework components and their status.

Component	Design rule	Status here
Failure-mode library	Criticality coverage; add confusable modes	Outer race, imbalance, misalignment
Degradation anchors	One anchor per turning point; perturb; convergence test	Nine anchors, one curve; perturbation and convergence not run
Onset and screening	Distributions; screen the healthy record	Not implemented
Co-occurring faults	Compatibility matrix; staggered onsets; multi-label	Two constant, additive secondary faults
Operating conditions	Physical scaling; set points, drift, transients	One speed in the ablation
Reliability gate	Calibration, spectral plausibility, discrimination, transfer, repeatability, held-out assets	Calibration on real carrier and RMS-level comparison with real Test 2 curve run; waveform-level tests not run

3 Validation methodology

Because synthetic fault data can look plausible while failing to transfer [17], injection is coupled with tiered validation. *Statistical similarity* compares distributional and spectral properties and health-index trends (RMS, kurtosis, spectral kurtosis) with any real faults [21]. *Physical plausibility* checks the fault frequency and sideband spacing against shaft speed. *Downstream performance* trains a model only on synthetic data and tests it on held-out real faults, mirroring the calibration sensitivity reported for digital twins [16]. *Sensitivity* perturbs the trajectory parameters θ to confirm that performance degrades smoothly. To avoid circularity, a calibration set fits parameter ranges and a held-out run-to-failure instance scores them.

Repeatability regenerates the data set with different seeds and reports the spread of the calibration error and the tier metrics. Synthetic-versus-real discrimination trains a classifier to separate synthetic from real faulty windows of the same mode, speed, and severity range; an accuracy well above chance shows artefacts that a diagnostic model could exploit. This test needs real faulty data, and it is necessary but not sufficient. Transfer to real faults compare a model trained on synthetic data with an upper reference trained on real data and a lower reference trained on an unguided baseline. Held-out assets repeat the tests on assets that were not used to choose the anchors or tolerances. Sufficiency: anchor convergence, leave-one-speed-out transfer, and the fraction of criticality covered show how many anchors, speeds, and modes

Table 2: Validity statement of the dataset produced in this study.

Item	Content
Modes covered	Outer-race defect; imbalance and misalignment at constant low severity. Other modes not covered
Assumptions	Carrier free of early degradation; additive overlay; one speed; one reference curve; severity in RMS units; failure level from one Test 1 snapshot
Tests passed	Calibration on the real Test 2 healthy carrier (mean error 0.0035 times healthy RMS); RMS-level comparison of five guidance levels with the real Test 2 degradation curve
Tests not run (waveform level)	Spectral plausibility, discrimination, transfer, repeatability, held-out assets
Appropriate use	Pipeline development and comparing guidance levels, not claims on real faults

are enough. The ablation applies the lower-reference idea at the RMS level, against the real Test 2 degradation curve, and the calibration is checked on the real healthy carrier.

The case study dataset is the IMS bearing dataset (University of Cincinnati, NASA Prognostics Data Repository), which holds three run-to-failure experiments at 2000 RPM and a radial load of 6000 lbs, sampled at 20 kHz [22, 23]. Test 1 ended in an inner-race defect on bearing 3 and a rolling-element defect on bearing 4, and Test 2 in an outer-race defect on bearing 1 [22]. The outer-race fault is therefore injected onto a healthy carrier from Test 2, bearing 1, so that the real degradation of the same bearing is available for comparison. IMS has a single operating condition; PRONOSTIA, with three speed and load combinations [24], is a candidate for the condition-dependence component.

4 Implementation and a five-level ablation on prior knowledge

The ablation asks how much prior knowledge a fault-generation procedure needs before its output resembles real degradation. Five levels are compared. Each receives the same real IMS healthy carrier, outer-race mode, injection engine, and scale inputs, and differs only in what it is told about how the fault evolves between those inputs.

- **Level 1a, unguided:** a random walk in log-RMS space with no trend, direction, or structure.
- **Level 1b, monotone growth:** the only knowledge is that a fault grows; the trajectory rises exponentially from the healthy to the failure level. This is a stronger baseline than the random walk.
- **Level 2, stage-informed:** the five-stage taxonomy (healthy, initiation, propagation, fake healing, critical) with ordering and timing chosen before any reference curve or real record was consulted.
- **Level 3, real-world curve snapshot:** the shape of a generic acceleration-RMS degradation curve from another asset, as nine anchors, without stage information. Only the shape is supplied, not any raw signal.

- **Level 4, stage structure plus snapshot:** the snapshot fixes the timing and amplitude of the stage turning points and the stage model fixes the form within each stage: a smooth rise at initiation, accelerating growth in propagation, a sine-eased dip during fake healing, and a plateau at the critical stage. It also outputs a stage label for every window.

All are compared against the real degradation curve of Test 2, bearing 1, which none uses to set any parameter.

4.1 Scale inputs and reference curve

A first version gave each level its own amplitude assumptions (none for the random walk, a guessed 9 times healthy for Levels 1b and 2, the drawn amplitudes for the generic curve), so levels differed in scale as well as shape. In the version reported here every level receives the same two scale inputs, and a shape function $\sigma(t) \in [0, 1]$ is mapped onto them, $r(t) = r_h + (r_f - r_h) \sigma(t)$. The healthy level r_h is the RMS of the healthy record (0.077 g). The failure level r_f is the RMS at which the asset is regarded as failed; in practice an alarm or trip level or the end-of-life level of similar assets. It is taken from IMS Test 1 as the mean end-of-record ratio of the two channels of failed bearing 3, 3.96 times healthy. This rests on one end-of-record snapshot of an inner-race failure, so it is indicative only; the end-of-record ratio of Test 2's bearing 1 is 9.4. Test 2 never sets the failure level, and a variant that uses it is reported only as a diagnostic.

The shape for Levels 3 and 4 comes from a generic RMS degradation curve of a rolling-element bearing, a snapshot from earlier work that is redrawn in Figure 2: a central trend and the spread of its source points were extracted from the original figure, expressed relative to the healthy level ($= 1.0$), and plotted anew without its annotations. Nine anchors (n_i, r_i) on the 1975-sample axis are read from the central trend and joined by a shape-preserving cubic Hermite interpolant: (0, 1.0), (1070, 1.16), (1110, 1.69), (1270, 1.92), (1420, 4.97), (1500, 6.37), (1650, 3.65), (1900, 6.73), (1975, 6.73). The first peak lies just below the second climb, and a self-healing dip separates them, consistent with the literature where debris redistribution and surface smoothing make vibration indicators fall temporarily during worsening degradation, including a documented suspect period in IMS [25]. The stage taxonomy follows the descriptive model of wear evolution in rolling bearings (running-in, steady state, defect initiation, defect propagation, damage growth), in which the dynamic impact rises at initiation, settles as defect edges are smoothed and debris redistributes, and rises again at damage growth [26].

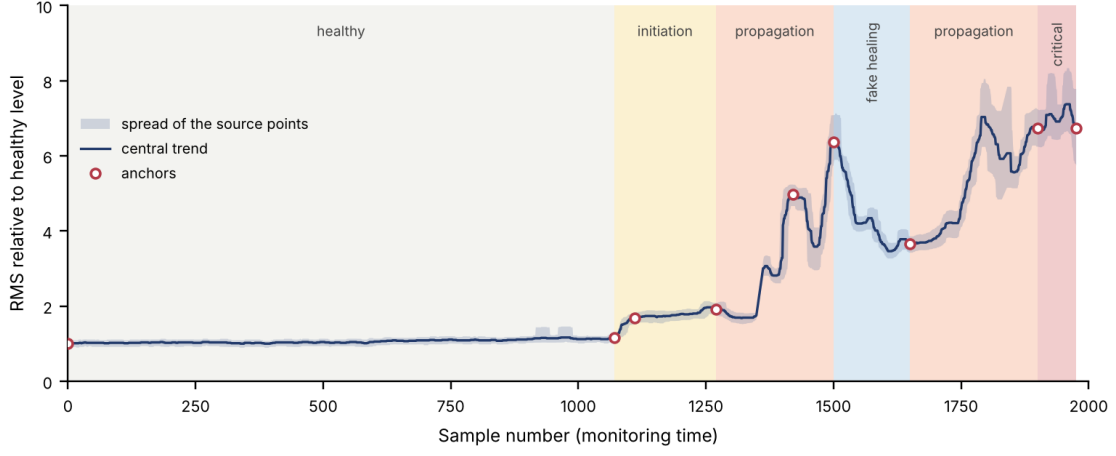


Figure 2: Real-world degradation curve snapshot used by Levels 3 and 4, redrawn from a generic bearing RMS curve.

Baselines. Level 1a is $r(n) = \exp(\sum_{i \leq n} \epsilon_i)$ with $\epsilon_i \sim \mathcal{N}(0, \sigma_w^2)$ and $\sigma_w = 0.03\sqrt{2000/N}$ for $N = 984$ windows, min-max normalized and mapped to the scale inputs; it is reported as mean and standard deviation over 200 realizations. Level 1b is r_f^t in relative units, with t the life fraction. Level 2 places stage boundaries at 10%, 25%, 55%, 65%, and 72% of life with ramps, a sine-eased dip, and an accelerating critical stage, then min-max normalizes the shape. These stages are a simplified realization of the wear-evolution model of [26] (illustrated in Figure 3): the model's rise at defect initiation and its dip before damage growth become the initiation ramp and the fake-healing dip, and damage growth becomes the accelerating critical stage.

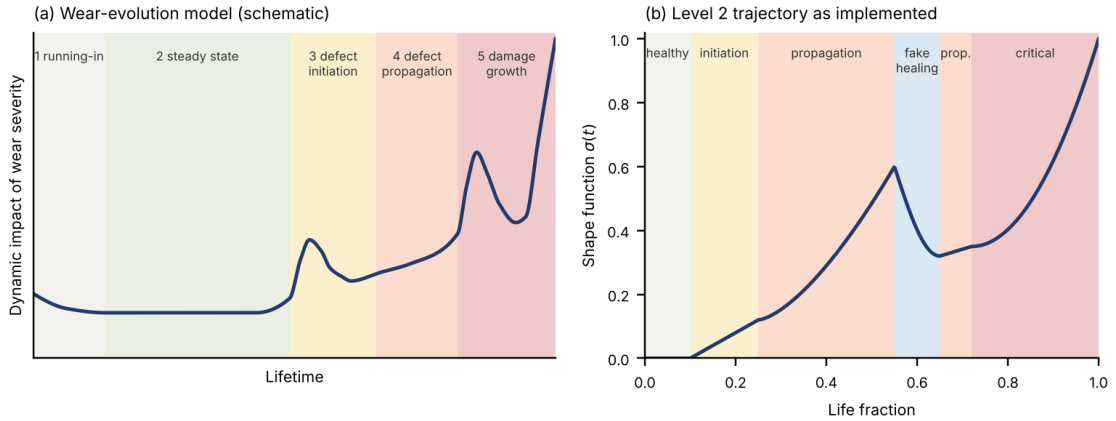


Figure 3: Stage knowledge supplied to Level 2. (a) Schematic of the five-stage wear-evolution model, redrawn and simplified from [26].

4.2 Raw-waveform injection and calibration

All levels pass through the same injection operator on the 20 kHz waveform, not on the RMS indicator. For each monitoring window a real healthy window is drawn at random from the

first 300 snapshots of Test 2, bearing 1 (about two days, over which RMS is flat). An outer-race impulse train is generated at unit severity at the Ball Pass Frequency Outer race of the Rexnord ZA-2115 geometry (about 236 Hz at 2000 RPM), each impact a decaying resonance with small timing jitter. Imbalance ($1\times$ shaft frequency) and misalignment ($2\times$) are present at 5% and 4% of the healthy RMS. The severity scale s is solved backwards under approximately energy-additive superposition,

$$s = \frac{\sqrt{\max(0, r^2 - \text{RMS}_{\text{healthy}}^2 - \text{RMS}_{\text{imb}}^2 - \text{RMS}_{\text{mis}}^2)}}{\text{RMS}_{\text{fault,unit}}}, \quad (1)$$

where r is the level’s target RMS in that window. The output is the real healthy window plus the scaled impulse train plus the two nuisance components; RMS and kurtosis are computed afterwards only as diagnostics. Injecting into a scalar indicator would discard the fault’s frequency, phase relation to shaft rotation, and harmonic structure, which are what a diagnostic model needs and what a spectral plausibility check verifies. Figure 4, from an earlier demonstration run with a synthetic carrier, shows the effect: the impulse train emerges through propagation, recedes in the fake-healing dip, and becomes large and dense at the critical stage. Only one speed (2000 RPM) enters the ablation.

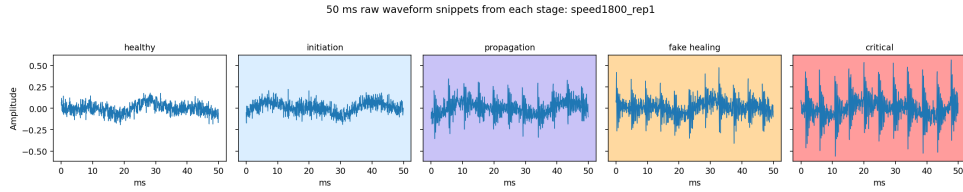


Figure 4: 50 ms raw-waveform snippets from each stage of a demonstration run (synthetic carrier), showing the outer-race impulse train emerging through propagation, receding during the fake-healing dip, and intensifying at the critical stage.

Calibration. Figure 5 compares the achieved RMS of a Level 4 run on the real healthy carrier with its target. Over five seeds the mean absolute error is 0.0035 times the healthy RMS (0.0003 g) and the maximum 0.036 times (0.003 g); the residual comes from the cross-terms that the energy-additive assumption neglects. This checks the calibration’s self-consistency and is distinct from the comparison with real degradation below.

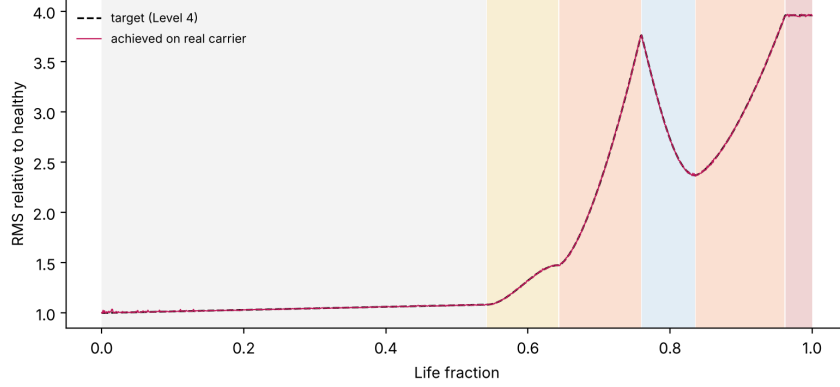


Figure 5: Achieved RMS of a Level 4 run on the real IMS Test 2 healthy carrier (solid) against its target (dashed), relative to the healthy RMS, with stage regions shaded.

5 Results

The ground truth is the windowed RMS of Test 2, bearing 1, for each of the 984 snapshots. The last two snapshots (about 0.002 g, forty times below healthy, probably sensor dropout) are excluded. All curves are expressed against life fraction on a common 1001-point grid. The real curve stays near healthy for over half its life and then rises steeply, so absolute errors are dominated by the end, and five metrics are reported: NMAE and NRMSE (mean absolute and root-mean-square error as a percentage of the real range); the log error (mean absolute log ratio, as a percentage), which weights a doubling at low and high amplitude equally; the shape error (mean absolute difference after min-max normalizing both curves); and the peak-timing error (difference in life fraction between the maxima).

Table 3: Error of each level against the real IMS Test 2, bearing 1, RMS curve, with failure level 3.96 times healthy from Test 1. Level 1a: mean $\pm$ SD over 200 realizations; others: one run.

Level	NMAE (%)	NRMSE (%)	Log error (%)	Shape (%)	Peak timing (%)
1a: random walk	15.8 ± 3.2	18.1 ± 3.3	103 ± 24	46.5 ± 7.9	45.2 ± 35.2
1b: monotone growth	10.0	11.8	58	33.9	0.1
2: stage-informed	7.9	10.1	47	28.0	0.2
3: curve snapshot	5.5	9.7	23	21.3	1.7
4: stage + snapshot	4.8	8.5	20	19.2	0.2

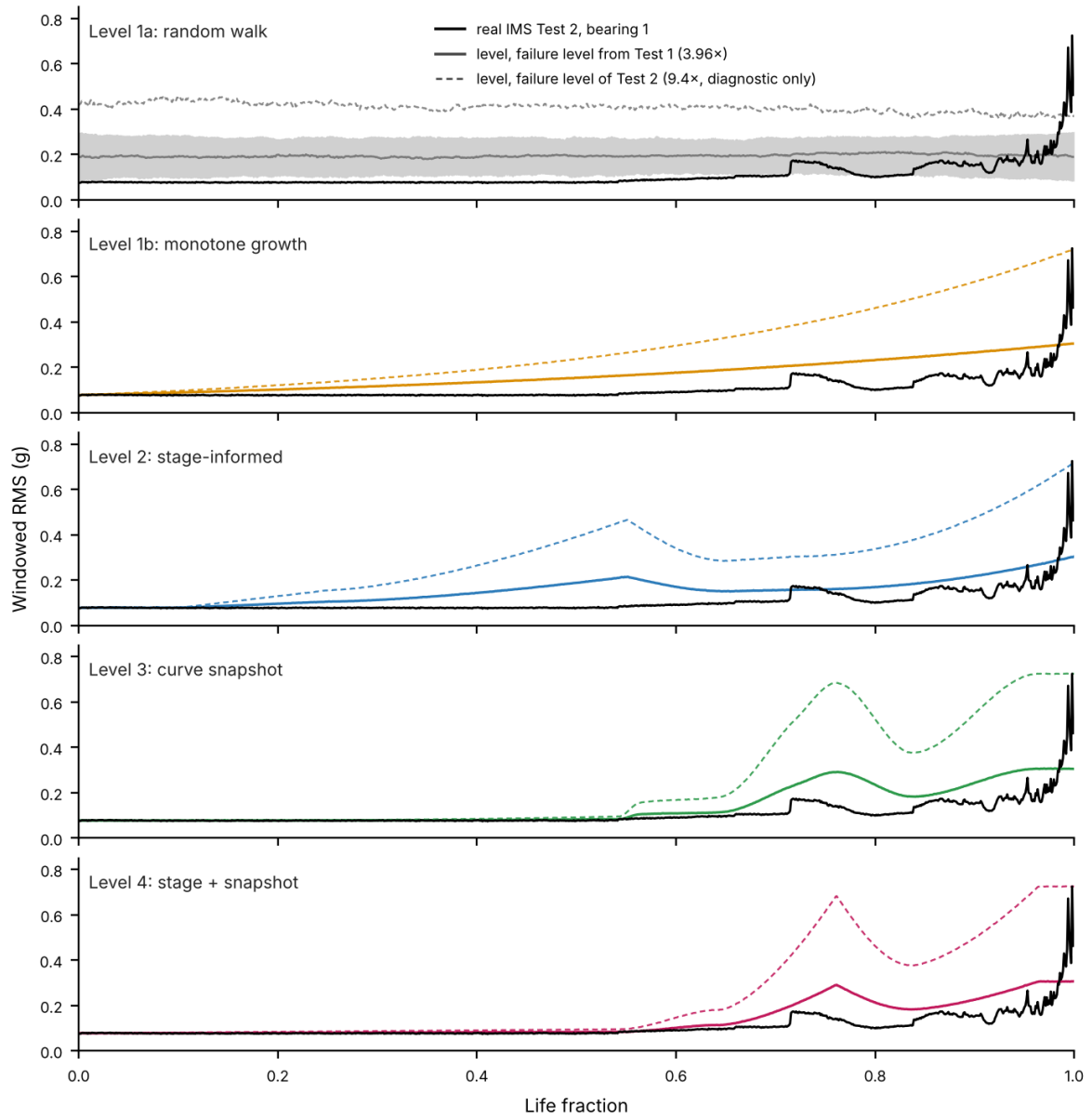


Figure 6: Windowed RMS of each level (solid, coloured) against the real IMS Test 2, bearing 1 curve (black), on a common time and amplitude scale.

With shared scale inputs, error falls at every step of added knowledge on four of five metrics (Table 3): NMAE goes from 15.8% for the random walk to 10.0%, 7.9%, 5.5%, and 4.8% for Levels 1b to 4, and the log and shape errors fall in the same order. Peak timing separates the random walk, which misses the peak by 45% of life on average with a large spread, from the other levels, which place it within 2% of the real peak at the end of life. Between those levels the metric is weak, since the real peak falls in the last 2% of life and the generic curve ends on a plateau. Level 4 improves slightly on Level 3 (4.8% against 5.5%), which is expected because the snapshot already carries the stage structure; its benefit beyond the metrics is that it produces stage labels. Levels 3 and 4 also match the onset best: the generic curve begins to rise at about

54% of life, and the real curve’s first sustained rise begins at about 55% (it exceeds 1.3 times healthy at 66%), while Levels 1b and 2 begin at 10–15%.

Three findings qualify this. First, no level reproduces the real curve’s long low plateau, stepwise rise, and spike in the last 2% of life; the generic curve lies above the real curve between about 65% and 90% of life in every case. Second, the failure level matters as much as the shape: set to the real Test 2 value of 9.4 (dashed lines in Figure 6; a diagnostic that uses the target and is not a fair test), errors rise about four-fold (Level 3 NMAE from 5.5% to 22.1%, Level 4 from 4.8% to 20.0%), because a higher end level lifts the middle of every curve, where the real curve is low. Third, the ranking depends on the scale treatment: in a first version with unequal scales the random walk had the lowest NMAE, because it stayed near the healthy level, as the real curve does for most of its life. Hence the table reports several metrics and shares the scale inputs.

6 Discussion

Figure 7 shows the framework in the form it has after the case study. We present it here because several of its features are findings of the work, not a priori design decisions. It is important to highlight the following findings:

- The original design had anchors but not the two numbers between which they are scaled, and the failure level changed the error about four-fold. The healthy and failure levels are therefore inputs of the framework, reported with their source and uncertainty.
- Each added piece of guidance (direction, stage timing, a curve) reduced the error, whereas a more elaborate injection engine was not varied. A misreading of the generic curve in an earlier version, which put its peak early, gave a peak-timing error of 29% against under 2% after the correction.
- The generic curve gets the onset time roughly right but lies far above this bearing in mid-life, and the end-of-record ratio is 3.6–4.3 for Test 1’s failed bearing 3 against 9.4 for Test 2’s bearing 1. A generic curve is a prior, not a truth.
- With unequal scales the random walk looked best on absolute error. The gate therefore needs several metrics, including log and shape errors and an onset-timing error.
- Matching a health indicator, which calibration does almost exactly, says nothing about whether the waveform is distinguishable from a real faulty one.

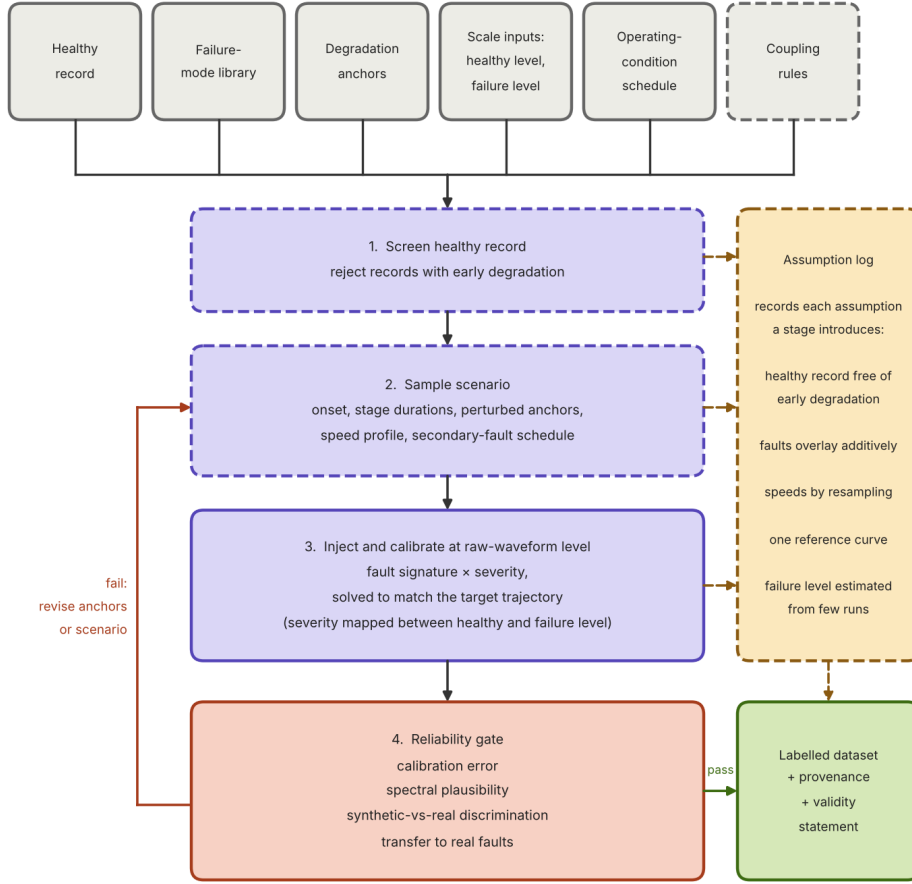


Figure 7: Automated fault-injection framework.

The results make the definition of anchors the most important open problem, and it deserves a study of its own. The questions are as follows. Where do anchors come from? Many run-to-failure histories (the two IMS tests, PRONOSTIA) can be normalized by healthy level and life fraction, and the same turning points extracted: onset, plateau level, peak, dip, final level. Anchors should be treated as distributions, for example onset between 50% and 70% of life, which would give the perturbation rule defensible bounds. Runs fail at different times and rates, so onset alignment or time warping is needed before pooling. The number of anchors follows the convergence test. The failure level can come from alarm or trip thresholds, such as vibration-severity standards, or from the spread across runs, always with its uncertainty. And success should be judged by envelope coverage, whether the real curve falls inside the 5–95% band of the synthetic family, which suits a method that generates families of curves better than a single mean error.

7 Conclusion

This paper proposed a framework for injecting faults into historical healthy data to produce labelled datasets for diagnosis and prognosis. It combines an injection model that separates fault type, severity evolution, and operating-condition dependence with design rules, a reliability gate, and a validity statement for each dataset. On IMS Test 2 the calibration reproduces its target on a real healthy carrier with a mean error of 0.0035 times the healthy RMS. A five-level

ablation against the real degradation curve, with shared healthy and failure levels, shows the error falling from 15.8% of the real range for an unguided random walk to 4.8% for stage structure combined with a real-world curve snapshot. It also shows the limits of generic knowledge: no level reproduces the real bearing's hidden-degradation plateau, and the failure level changes the error about four-fold, so it must be reported as an explicit input. Future work will define anchors from many run-to-failure histories, implement the components marked as not implemented in Table 1, run the reliability tests, and extend the study to PRONOSTIA and to other asset classes.

References

- [1] Q Nie, J Geng and C Liu 2026 A review of fault diagnosis methods: from traditional machine learning to large language model fusion paradigm *Sensors* **26**(2) 702 doi:10.3390/s26020702
- [2] Y Jin, L Hou, M Du and Y Chen 2022 A wavelet transform and self-supervised learning-based framework for bearing fault diagnosis with limited labeled data *arXiv preprint arXiv:2207.10432*
- [3] D Neupane, M R Bouadjene, R Dazeley and S Aryal 2025 Data-driven machinery fault diagnosis: a comprehensive review *Neurocomputing* **627** 129588
- [4] S Wang, Y Vidal and F Pozo 2025 An unsupervised approach to early fault detection and performance degradation assessment in bearings *Adv. Eng. Inform.* **68** 103620 doi:10.1016/j.aei.2025.103620
- [5] P E Lanigan, P Narasimhan and T E Fuhrman 2011 Supporting synthetic data-driven diagnosis through automated fault-injection *Proc. 4th Int. Conf. on Dependability (DEPEND 2011)* pp 65–71.
- [6] M Abboush, C Knieke and A Rausch 2024 Representative dataset generation framework for AI-based failure analysis during real-time validation of automotive software systems *Proc. Hawaii Int. Conf. on System Sciences (HICSS)* doi:10.24251/HICSS.2024.877
- [7] F Bagehorn, J Rios Aliaga, S Jha, R Filepp, L Shwartz, N Abe and X Yang 2022 WOLFFI: a fault injection platform for learning AIops models *Proc. IEEE/ACM Int. Conf. on Automated Software Engineering (ASE 2022), Industry Showcase* doi:10.1145/3551349.3559503
- [8] J R de Campos 2021 Advanced online failure prediction through machine learning *PhD thesis* Faculty of Sciences and Technology, University of Coimbra
- [9] Q Wang, C Taal and O Fink 2022 Integrating expert knowledge with domain adaptation for unsupervised fault diagnosis *IEEE Trans. Instrum. Meas.* (arXiv:2107.01849)
- [10] I Kim and S Lee 2023 Simulation-driven bearing fault diagnosis for condition monitoring without faulty data *Proc. PHM Society Asia-Pacific Conf.* **4**(1) 3778 doi:10.36001/phmap.2023.v4i1.3778
- [11] D-Y Ji, B Wang, H Inoue and M Kanemaru 2025 Motor fault detection with a hybrid physics-based and data-driven method *Proc. IEEE Int. Electric Machines and Drives Conf. (IEMDC)*
- [12] M Ahang, M Jalayer, A Shojaeinasab, O Ogunfowora, T Charter and H Najjaran 2022 Synthesizing rolling bearing fault samples in new conditions: a framework based on a modified CGAN *Sensors* **22**(14) 5413 doi:10.3390/s22145413

- [13] A Saxena, K Goebel, D Simon and N Eklund 2008 Damage propagation modeling for aircraft engine run-to-failure simulation *Proc. Int. Conf. on Prognostics and Health Management (PHM 2008)*
- [14] M J McGhee, V M Catterson and B Brown 2017 Prognostic modelling utilizing a high fidelity pressurized water reactor simulator *IEEE Trans. Syst. Man Cybern.: Syst.* doi:10.1109/TSMC.2017.2662478
- [15] J Xiong, J Zhou, Y Ma and O Fink 2023 Generating controlled physics-informed time-to-failure trajectories for prognostics in unseen operational conditions *Proc. 33rd European Safety and Reliability Conf. (ESREL 2023)* (Southampton, UK, 3–8 September 2023)
- [16] C Dong, J Tao and C Liu 2025 Digital twin-driven zero-shot fault diagnosis of axial piston pumps using fluid-borne noise signals *arXiv preprint* arXiv:2512.19280
- [17] C Liu, R Hu, X Fang, W Luo and C Zhu 2025 Enhancing unsupervised bearing fault diagnosis through structured prediction in latent subspace *Sci. Rep.* **15** 42146 doi:10.1038/s41598-025-26013-0
- [18] H Chen, J Wei, H Huang, Y Yuan and J Wang 2024 Review of imbalanced fault diagnosis technology based on generative adversarial networks *J. Comput. Des. Eng.* **11**(5) 99–124 doi:10.1093/jcde/qwae075
- [19] F Wei, L Tan, X Ma, H Xiao, D Patel, C G Lee and L Yang 2025 A hybrid prognostic framework: stochastic degradation process with adaptive trajectory learning to transfer historical health knowledge *Mech. Syst. Signal Process.* **224** 112171 doi:10.1016/j.ymssp.2024.112171
- [20] Performance degradation analysis and fault prognostics of solid oxide fuel cells using the data-driven method 2021 *Int. J. Hydrogen Energy*
- [21] Y Cao, M Jia, J Zhuang and X Zhao 2022 Research on sparsity measures for rotating machinery health monitoring *arXiv preprint* arXiv:2205.00178
- [22] J Lee, H Qiu, G Yu, J Lin and Rexnord Technical Services 2007 Bearing Data Set, IMS, University of Cincinnati, *NASA Prognostics Data Repository*, NASA Ames Research Center, Moffett Field, CA.
- [23] R Chang, Y Ma, W Nie, J Nie and A-A Liu 2023 Causal disentanglement hidden Markov model for fault diagnosis *arXiv preprint* arXiv:2308.03027
- [24] T von Hahn and C K Mechefske 2022 Knowledge informed machine learning using a Weibull-based loss function *arXiv preprint* arXiv:2201.01769
- [25] H Qiu, J Lee, J Lin and G Yu 2006 Wavelet filter-based weak signature detection method and its application on roller bearing prognostics *Journal of Sound and Vibration* **289** 1066–1090.
- [26] I El-Thalji and E Jantunen 2014 A descriptive model of wear evolution in rolling bearings *Engineering Failure Analysis* **45** 204–224 doi:10.1016/j.engfailanal.2014.06.004